



Lenovo ServerGuide Scripting Toolkit, Windows Edition User's Guide



Version 10.1

Note

Before using this information and the product it supports, read the information in Appendix C “Notices” on page 145.

(May 2015)

© Copyright Lenovo 2014, 2015.

Portions © Copyright IBM Corporation 2012, 2014

LIMITED AND RESTRICTED RIGHTS NOTICE: If data or software is delivered pursuant to a General Services Administration “GSA” contract, use, reproduction, or disclosure is subject to restrictions set forth in Contract No. GS-35F-05925

Contents

Tables	iii	About customizing Windows installation scenarios	24
About this publication	v	Installing Windows Server 2008 x86 editions locally from a USB key	25
Who should read this guide	v	Installing Windows Server 2008 x64 editions locally	26
Conventions and terminology	v	Installing Windows Server 2008 x64 editions over a network	27
Publications and related information	v	Installing Windows remotely	28
Web resources	v	Installing Windows remotely using Integrated Management Module	30
Chapter 1. Overview.	1	Adding an installation of the Systems Director Agent	31
Chapter 2. Downloading and extracting Lenovo ServerGuide Scripting Toolkit, Windows Edition	3	Adding the installation of UpdateXpress System Packs to your deployment	31
Prerequisites	3	Modifying the Windows PE image	32
Downloading and extracting the Toolkit files	3	Automating the deployment process	33
Configuring the source server	4	Chapter 5. Downloading and installing System Enablement Packs (SEPs)	51
Adding files to the source tree	5	Chapter 6. Supported target server hardware and software	53
Chapter 3. Quick start deployment scenarios	9	System support	53
Modes of operation	9	Operating system support	53
Deployment tasks	9	RAID controller support	53
Deploying operating systems	9	Fibre Channel HBA support	54
Configuring system settings with ASU	10	Chapter 7. ServerGuide Scripting Toolkit tools and utilities	55
Configuring a Fibre HBA boot task	12	SGTKWinPE.cmd	55
Windows installation deployments	12	Tools included with ServerGuide Scripting Toolkit	55
Deploying Windows Server 2008 Enterprise Edition x86 locally	12	Advanced Settings Utility	55
Deploying Windows Server 2008 Standard Edition x64 over a network	13	UpdateXpress System Pack Installer	56
Installing Windows applications and upgrades	13	praid.exe	57
Chapter 4. Customized deployment scenarios	15	ServerGuide Scripting Toolkit utilities	75
Customizing source server settings	15	clini.exe	75
Adding device drivers to Windows installations	16	ddcopy.exe	81
Adding Windows PE drivers	17	dscan.exe	84
Adding drivers to an existing driver set	17	hwdetect.exe	89
Adding drivers to a new device driver set	17	invraid.exe	93
Adding hotfixes to Windows PE	18	LEcho.exe	97
PRAID policy files	18	quaculi.exe	99
Adding PRAID policy files	19	savestat.cmd	101
Adding ASU files	20	tksearch.exe	103
About customizing a Fibre HBA boot configuration	20	unattend.exe	105
Configuring USB keys for deployment	23	unattend.ini	107
Using a USB key for deployment	24		

valraid.exe	109	Configuring USB keys for deployment	120
winlpcfg.exe	111	Booting from a USB key	121
Chapter 8. Incorporating Scripting Toolkit with your existing process	113	Configuring ServeRAID controllers with a 64-bit operating systems	121
Chapter 9. Troubleshooting and support	115	Installing Windows Server 2008 x64 locally with a ServeRAID-SCSI configuration	121
Known problems and limitations.	115	Installing Windows Server 2008 x64 over a network with a ServeRAID-SCSI configuration	123
Getting help and technical assistance	117	Network installation of Windows Server 2008 x64 with a ServeRAID-SCSI configuration using a USB key	125
Before you call	117	Installing Windows remotely	126
Using the documentation.	118	Appendix B. Accessibility	143
Getting help and information on the Web	118	Accessibility features for ServerGuide Scripting Toolkit	143
Software service and support	118	Appendix C. Notices.	145
Hardware service and support.	118	Trademarks	146
Appendix A. Best practices	119	Important notes	146
Building a customized installation image with a RAID adapter driver.	119	Index.	147
Using UXSPI to download updates.	119		
Considerations for installing an OS on a multi-adapter system	119		
Adding software applications for installation after first Autologin	120		

Tables

1. Sample PRAID policy files	19	27. dscan examples	86
2. Fibre HBA boot configuration variables	20	28. drvinfo.ini variables	87
3. Windows PE 2.1 editions and their INI file names.	25	29. drvinfo.ini file contents	88
4. Windows Server 2008 editions and their ISO images	26	30. hwdetect parameters	90
5. Windows 2008 editions and associated files for local installation	27	31. hwdetect examples	91
6. Windows 2008 editions and associated files for Network installation	28	32. Supported RAID adapter information.	93
7. AutoStartToolkitProcesses setting	33	33. invraid parameters	95
8. AutoContinueToolkitProcesses setting	34	34. invraid return values	95
9. AutomaticallyRestart Process setting	34	35. LEcho parameters	98
10. User definable settings for SGTWinPE.cmd	35	36. qaucli command usage examples	100
11. Toolkit commands for downloading and installing SEPs	51	37. Additional examples of qaucli usage	101
12. SGTWinPE parameters and options	55	38. savestat parameters	102
13. ASU syntax and usage	56	39. tksearch parameters	103
14. PRAID parameters common to multiple modes	58	40. tksearch examples	104
15. PRAID Deploy mode parameters	60	41. unattend utility device drivers	105
16. Capture mode parameters	61	42. unattend parameters	106
17. Policy files	64	43. unattend examples.	107
18. Policy file parameters.	64	44. valraid parameters	110
19. Supported settings for each RAID controller when using PRAID	70	45. winlpcfg syntax	111
20. Default RAID levels.	73	46. winlpcfg usage examples	112
21. clini parameters	76	47. INI file names for Windows Server 2008 x64 versions	122
22. clini examples.	78	48. ISO file names for Windows Server 2008 x64 versions	122
23. ddcopy parameters	82	49. INI file names for Windows Server 2008 x64 versions	124
24. ddcopy return values	83	50. ISO file names for Windows Server 2008 x64 versions	124
25. ddcopy examples	83	51. INI file names for Windows Server 2008 x64 versions	125
26. dscan parameters	85	52. ISO file locations for Windows Server 2008 x64 versions	125
		53. User definable settings for SGTWinPE.cmd	129

About this publication

The Lenovo ServerGuide Scripting Toolkit, Windows Edition *Users Guide* provides information about how to download and use ServerGuide Scripting Toolkit.

Who should read this guide

This guide is for system administrators or other individuals responsible for creating and deploying software and performing system configuration, maintenance, and upgrades.

Conventions and terminology

Paragraphs that start with a Note, Important, or Attention in bold have specific meanings to highlight key information:

Note: These notices provide important tips, guidance, or advice.

Important: These notices provide information or advice that might help you avoid inconvenient or difficult situations.

Attention: These notices indicate possible damage to programs, devices, or data. An attention notice appears before the instruction or situation in which damage can occur.

Publications and related information

To view a PDF file, you need Adobe Acrobat Reader, which can be downloaded for free from the Adobe website at www.adobe.com/products/acrobat/readstep.html.

Information centers and topic collections

The Lenovo ToolsCenter for System x and BladeCenter Information Center provides information for x86 products and tools.

Publications

The latest version of the Lenovo ServerGuide Scripting Toolkit, Windows Edition *Users Guide* can be downloaded from Lenovo ServerGuide for Lenovo x86 servers.

Web resources

The following websites and information center topics are resources for using Lenovo ServerGuide Scripting Toolkit, Windows Edition.

Websites

- Lenovo ServerGuide for Lenovo x86 servers at:
<http://support.lenovo.com/us/en/documents/LNVO-TOOLKIT>
Use this website to download the Lenovo ServerGuide Scripting Toolkit, Windows Edition tool and documentation.
- Lenovo ServerProven at: “<http://www-03.ibm.com/systems/info/x86servers/serverproven/compat/us/>” on page
Use this website to obtain information about the hardware compatibility of BladeCenter, and System x systems with applications and middleware.

- Lenovo Service and Support at: <http://support.lenovo.com/us/en>
Use this website to obtain service and support information for Lenovo products.

Chapter 1. Overview

Windows Scripting Toolkit is a collection of system-configuration tools and installation scripts for deploying software to your Lenovo x86 server.

provides hardware configuration utilities and operating system (OS) installation examples for Lenovo System x and BladeCenter x86-based hardware. You can use Toolkit to build custom hardware deployment solutions.

ServerGuide Scripting Toolkit, Windows Edition enables you to create a bootable Windows Preinstallation Environment (Windows PE) 2.1, 3.0, or 4.0 CD, DVD, or USB key that supports the following:

- Network and mass storage devices
- Policy based RAID configuration
- Configuration of system settings using Advanced Settings Utility (ASU)
- Configuration of Fibre Host Bus Adapters (HBAs) under WinPE
- Automated Network Operating System (NOS) installation support for:
 - Windows Server 2008, Standard, Enterprise, Datacenter, and Web Editions
 - Windows Server 2008 x64, Standard, Enterprise, Datacenter, and Web Editions
 - Windows Server 2008, Standard, Enterprise, and Datacenter Editions without Hyper-V
 - Windows Server 2008 x64, Standard, Enterprise, and Datacenter Editions without Hyper-V
 - Windows Server 2008 R2 x64, Standard, Enterprise, Datacenter, and Web Editions
 - Windows Server 2012
 - Windows Server 2012 R2
- Local self-contained DVD deployment scenarios
- Local CD/DVD and network share based deployment scenarios
- RSA II, IMM, and BladeCenter MM/AMM remote disk scenarios
- Systems Director Agent installation integrated with scripted NOS deployment
- Systems Director Agent installation integrated with scripted NOS deployment. ServerGuide Scripting Toolkit, Windows Edition supports the following versions of Systems Director Agent:
 - Systems Director Agent 5.1 or higher
 - Common Agent 6.1 or higher
 - Core Services 5.20.31 or higher

Chapter 2. Downloading and extracting Lenovo ServerGuide Scripting Toolkit, Windows Edition

The topics in this section provide information about the prerequisites, downloading and installing ServerGuide Scripting Toolkit on a source server, and configuring the source server. ServerGuide Scripting Toolkit uses a source server to create and deploy images.

The following topics will assist you in getting started with ServerGuide Scripting Toolkit:

- Verify that you have met the requirements in the “Prerequisites” on page 3 topic.
- Download and install ServerGuide Scripting Toolkit. See “Downloading and extracting the Toolkit files” on page 3.
- After the you have downloaded and installed ServerGuide Scripting Toolkit, configure the source server. See “Configuring the source server” on page 4.
- Files that will be included in deployments, must first be added to the Toolkit source tree. See “Adding files to the source tree” on page 5.

Prerequisites

To use ServerGuide Scripting Toolkit, Windows Edition (**SGTKWinPE.cmd**), complete the following prerequisites for adding the required software used by the Toolkit. These files are added to the source tree as part of setting up the source server.

- Download and install an English version of one of the following kits:
 - Windows Assessment and Deployment Kit (ADK) that includes:
 - Windows Server 2012
 - Windows Server 2012 R2

This ADK is available for download from: .

- Windows Automated Installation Kit (AIK) that includes:
 - Windows Server 2003 with Service Pack 2
 - Windows Vista SP1
 - Windows Server 2008 family
 - Windows 7 family
 - Windows Server 2008 R2 family

This AIK is available from: .

- Create WindowsPE ISO images of the source files on a CD or DVD-ROM.

Downloading and extracting the Toolkit files

This section provides instructions for downloading and extracting the ServerGuide Scripting Toolkit, Windows Edition files.

ServerGuide Scripting Toolkit, Windows Edition supports these Windows operating systems:

- Windows Server 2012
- Windows Server 2008 R2 x64 Standard, Enterprise, Datacenter, and Web Editions
- Windows Server 2008, Standard, Enterprise, Datacenter, and Web Editions

- Windows Server 2008 x64, Standard, Enterprise, Datacenter, and Web Editions
- Windows Server 2003, Standard, Enterprise, and Web Editions
- Windows Server 2003, Standard and Enterprise x64 Editions
- Windows Server 2003 R2, Standard and Enterprise Editions
- Windows Server 2003 R2, Standard and Enterprise x64 Editions
- Windows 7
- Windows Vista Business Edition
- Windows Vista Business x64 Edition
- Windows XP Professional Edition
- Windows XP Professional x64 Edition

To download and extract the Toolkit, complete these steps:

- Step 1. Download the latest version of the Toolkit.zip file from: [Lenovo ServerGuide for Lenovo x86 servers](#).
- Step 2. Create a directory on the source server to extract the Toolkit.zip files. This directory will be referred to as the Toolkit folder. Examples in this document use the value: C:\sgshare as the Toolkit directory.
- Step 3. Extract the Toolkit.zip files in the directory you created.
After extracting the file, the path should look like C:\sgshare\sgdeploy.

When you have finished extracting the Toolkit files, you must configure the source server, as described in “Configuring the source server” on page 4.

Configuring the source server

This topic provides detailed information about configuring the ServerGuide Scripting Toolkit, Windows Edition source server.

ServerGuide Scripting Toolkit, Windows Edition comes configured to use a set of defaults for access to the source server. If you do not want to use these defaults, you must change them in the .ini files used by the Toolkit, located in the sgdeploy\SGTKWinPE\Scenario\INIs directory.

Toolkit uses the following default settings:

- Server name: **sgtksrv**
- Shared directory name: **sgshare**
- User ID and password: **sgtk0clnt**

These settings are only necessary for network share deployments. For more information about changing these settings, see “Customizing source server settings” on page 15.

This procedure lists the steps for configuring the source server.

Attention: A fully automated ServerGuide Scripting Toolkit CD, DVD, or USB solution has the ability to re-partition and reformat a system without warning. For best results and to avoid accidental loss of data, remove the CD, DVD, or USB key from the system boot order on the source server as soon as it is no longer required.

- Step 1. To use the source server for network deployments, share the Toolkit folder on the network. For example: C:\sgshare. For local deployments, it is not necessary to share the Toolkit folder.
- Step 2. Verify that the C:\sgshare folder is accessible to all the systems that require access on the network.

- Step 3. Use the Toolkit Configuration Utility to add files to the source tree, by starting from a command prompt or by double-clicking it. `TKConfig.exe` is located in the `sgdeploy\tkconfig` directory. For more information, see “Adding files to the source tree” on page 5.
- Step 4. Include or update the Windows answer file. For more information, see “Using the Windows answer file for unattended installations” on page 5.
- Step 5. Create a user ID and password that has access to the Toolkit folder. Toolkit uses the default value of `sgtk0clnt` for both the ID and the password.
- Step 6. If you choose to use a different ID and password, you must change the ID and password values in the Network Settings section of the Toolkit Settings `.ini` file.
- Step 7. If deployment scenarios will be built on a different system from the source server, the required System Enablement Packs (SEPs) must be installed on the source server before you perform a deployment. For more information about SEPs, see Chapter 5 “Downloading and installing System Enablement Packs (SEPs)” on page 51.

Adding files to the source tree

This section provides information about adding files to the source tree. Files that will be included in deployments, must first be added to the Toolkit source tree.

ServerGuide Scripting Toolkit, Windows Edition uses the Toolkit Configuration Utility, which is a graphical user interface (GUI) configuration program that adds these files to the source tree:

- Systems Director Agent files
- Windows installation files
- Windows answer files

Using the Windows answer file for unattended installations

The Windows answer file allows you to perform unattended installations by providing responses to prompts encountered during installation. Four sample answer files are provided with the Toolkit, and another answer file is provided by Microsoft.

The following Toolkit sample answer files are located in the `sgdeploy\SGTKWinPE\AnswerFiles` directory:

- `win2008.xml`
- `win2008x64.xml`
- `win2011x64.xml`
- `win2012x64.xml`

Customizing the Windows Server 2008 answer file

The following procedure describes how to customize the Windows Server 2008 answer file.

- Step 1. Open the `sgdeploy\SGTKWinPE\AnswerFiles\win2008x64.xml` file.
- Step 2. Add the settings you want to customize, for example:


```
<TimeZone>%TK_TimeZone%</TimeZone>
```
- Step 3. Set the value of the setting you added to a variable that you add to the appropriate Scripting Toolkit scenario `.ini` file. Typically you will add this value to the [NOS Installation Settings] section of the `.ini` file. Scripting Toolkit environment variables are surrounded by the percent sign (%), as shown in this example:


```
[GuiUnattended]
...
<TimeZone>%TK_TimeZone%</TimeZone>
...
```

- Step 4. For each environment variable you have assigned, include a value in the corresponding scenario .ini file. Typically, you add this value to the [NOS Installation Settings] section of the .ini file.

Use the format *variable_name=value*, where *variable_name* is the name you selected in the answer file and *value* is a valid value for the variable you are using, for example:

```
TK_TimeZone=035
```

or:

```
TK_TimeZone=Pacific Standard Time
```

During deployment, the environment variables specified in the answer file are replaced with the corresponding variables from the scenario .ini file.

Customizing the Windows Server 2012 answer file

The following procedure describes how to customize the Windows Server 2012 answer file.

- Step 1. Open the sgdeploy\SGTKWinPE\AnswerFiles\win2012x64.xml file.
- Step 2. Add the settings you want to customize, for example:
<TimeZone>%TK_TimeZone%</TimeZone>
- Step 3. Set the value of the setting you added to a variable that you add to the appropriate Scripting Toolkit scenario .ini file. Typically, you add this value to the [NOS Installation Settings] section of the .ini file. Scripting Toolkit environment variables are surrounded by the percent sign (%), as shown in this example:
[GuiUnattended]
...
<TimeZone>%TK_TimeZone%</TimeZone>
...
- Step 4. For each environment variable you assign, include a value in the corresponding scenario .ini file. Typically you add this value to the [NOS Installation Settings] section of the .ini file.

Use the format *variable_name=value*, where *variable_name* is the name you selected in the answer file and *value* is a valid value for the variable you are using, for example:

```
TK_TimeZone=035
```

or:

```
TK_TimeZone=Pacific Standard Time
```

Adding Windows installation files

The following procedure describes how to add Windows installation files to the source tree.

If you intend to use the source server as an OS repository for network deployments or create an OS deployment image bundled with Windows installation files for local deployments, follow these steps to add Windows installation files to the source tree.

- Step 1. Start the Toolkit Configuration Utility.
- Step 2. Select **Add Operating System Installation Files** from the task list.
- Step 3. Follow the GUI Wizard instructions for the operating system type you want.
- Step 4. Insert the correct OS installation media into the optical drive of the source system running Windows , or browse to select the specific directory containing the OS installation media.
- Step 5. Modify the target path if necessary.
- Step 6. Copy the files from the source location to the target location.
- Step 7. Exit the Operating System Installation Files wizard.

Adding Windows device drivers

Toolkit allows you to download System Enablement Packs (SEPs), which include a driver library that contains all of the drivers necessary to complete the installation of Windows Server 2008 and Windows Server 2012.

To ensure that all devices are installed, the drivers are up to date, and no errors remain in Device Manager, you must deploy the latest UpdateXpress System Packs. By downloading the applicable UpdateXpress System Packs, you can add support for systems released after the current version of ServerGuide Scripting Toolkit, Windows Edition. For more information, see Chapter 5 “Downloading and installing System Enablement Packs (SEPs)” on page 51.

Adding Systems Director Agent files

To install Systems Director Agent during Windows OS deployment, follow these steps to add the installation files to the source tree.

Step 1. Download the Systems Director Agent files from .

Note: This download requires registration with the IBM home page.

Step 2. Extract the Systems Director Agent files to a convenient location.

Step 3. Start the Toolkit Configuration Utility.

Step 4. Select **Add Operating System Application Files** from the task list.

Step 5. Browse to the location where you extracted the Systems Director Agent files.

Step 6. Copy the files from the source location to the target location.

Step 7. Exit the Operating System Application files wizard.

Adding UpdateXpress System Packs

Follow these steps to add LenovoUpdateXpress System Packs (UXSPs) to the source tree.

Step 1. Download UpdateXpress System Pack for the desired machine type and operating system combination into the source tree in the updates\uxsp directory. When downloading multiple UXSPs for multiple machine types, place them all in this directory. When the UXSP installer runs, it automatically selects the appropriate files. If you are prompted to overwrite existing files, click **OK**.

Note: Do not change the names of any UXSP files or make any changes to the case (uppercase and lowercase) of the letters. Changing the file name or case can cause the UXSP deployment to fail.

Step 2. Set the UpdateXpress environment variable to **Yes** in the NOS Installation Settings section of the scenario .ini file you are using for this deployment, if you are using UXSP for this deployment.

See the following example.

```
[NOS Installation Settings]
```

```
TK_NOS_PerformPostOSInstallUXSPUpdates=Yes
```

UXSP will be installed after the operating system is installed. You can check the results of the UXSP installation in the Toolkit log file: C:\Lenovo_Support\Lenovo_WinPEToolkit.log.

Chapter 3. Quick start deployment scenarios

The Toolkit is a powerful and flexible tool, designed to help you create deployment scenarios that are tailored for your unique environment. The topics in this section provide information about deployment scenarios that will quickly help you begin using the Toolkit.

Topics in this section include:

- “Modes of operation” on page 9
- “Deployment tasks” on page 9
- “Windows installation deployments” on page 12

For more information about how to tailor deployments to best fit your needs, see Chapter 4 “Customized deployment scenarios” on page 15.

Modes of operation

ServerGuide Scripting Toolkit, Windows Edition uses three different modes of operation to support a variety of deployment scenarios. Each mode is designed to address a different set of deployment requirements.

These modes are:

Local deployment

Local deployments provide a simple means of deploying an image to a system that is physically accessible. This deployment mode does not require access to a network share. For local deployments, use the Toolkit to create a complete image with all of the required files and then burn that image to a CD or DVD-ROM or copy it to a USB key, which you then use to deploy the image to the system.

Network deployment

Network deployments require the target system to have access to a network share. Use the Toolkit to create an image containing the minimum required files and burn that image to a CD or DVD-ROM or copy it to a USB key. Use this media to boot the target system, and the deployment is completed by using files from the network share.

Remote deployment

Remote deployments allow the use of a Remote Supervisor Adapter II, Integrated Management Module (IMM), BladeCenter Management Module, or BladeCenter Advanced Management Module to deploy the Scripting Toolkit image remotely. Create an ISO image, CD, or DVD-ROM of the required Scripting Toolkit deployment files, and then use one of these remote deployment options to send the deployment files to the remote target system. The use of a USB key is not supported for this mode of operation.

Deployment tasks

This section provides examples of the tasks required to create and deploy an image using ServerGuide Scripting Toolkit, Windows Edition.

The examples in this section include all of the information you need to create a deployment image that can perform the most common deployment scenarios.

Deploying operating systems

This procedure describes how to complete an operating system deployment.

- Step 1. Configure RAID.
- Step 2. Install the operating system. See Chapter 1 “Overview” on page 1.
- Step 3. Perform post-OS installation tasks, such as installing UpdateXpress System Packs, the Systems Director Agent, and device driver updates.

For more detailed information about customizing your deployments, see Chapter 4 “Customized deployment scenarios” on page 15.

RAID configuration

Before installing the operating system, you must configure RAID for the target system. This section contains examples for a basic RAID configuration and a more complicated, policy file based configuration.

“Default local RAID configuration” on page 10 describes the steps to create a task to perform local RAID configuration on the target system. “RAID1 policy file configuration” on page 10 describes how to create a task to perform RAID configuration by using a PRAID policy file. For more information about policy files, see “Adding PRAID policy files” on page 19. For a list of RAID controllers supported by the Toolkit, see “RAID controller support” on page 53.

Default local RAID configuration

This example describes how to create a basic RAID configuration task by using Windows PE 2.1 x86.

- Step 1. At a command prompt, go to the following directory: `\sgshare\sgdeploy\SGTKWinPE`.
- Step 2. Create the RAID configuration ISO image by running **SGTKWinPE.cmd**, using the `Raid_Config_Only_x86.ini` file provided by the Toolkit as shown:
`SGTKWinPE.cmd ScenarioINIs\Local\Raid_Config_Only_x86.ini`
The ISO image is created in `..\WinPE_ScenarioOutput\Local_Raid_Config_Only_x86\WinPE_x86.iso`, and your media creation software is started to create bootable media from the ISO image.
- Step 3. Use the media created to boot the target system, and follow the on-screen prompts to complete the deployment.

RAID1 policy file configuration

This example describes how to use a policy file to create a RAID1 configuration task.

- Step 1. At a command prompt, go to the following directory: `\sgshare\sgdeploy\SGTKWinPE`.
- Step 2. Modify the `Raid_Config_Only_x86.ini` file provided by the Toolkit to use the `RAID1.ini` policy file. Set the variable `TK_PRAID_PolicyFile` to `RAID1.ini`.
- Step 3. Create the RAID configuration ISO image by running **SGTKWinPE.cmd**, using the `Raid_Config_Only_x86.ini` file provided by the Toolkit as shown:
`SGTKWinPE.cmd ScenarioINIs\Local\Raid_Config_Only_x86.ini`
The ISO image is created in `..\WinPE_ScenarioOutput\Local_Raid_Config_Only_x86\WinPE_x86.iso`, and your media creation software is started to create bootable media from the ISO image.
- Step 4. Boot the target system using the media created, and follow the on-screen prompts to complete the deployment.

Configuring system settings with ASU

Before installing the operating system, you can configure the system settings for the target system by using Advanced Settings Utility (ASU). This section describes the steps for a basic BIOS configuration.

The following example describes how create a task that performs a basic BIOS configuration using ASU on Windows PE 2.1 x86. This example boots the target system and then invokes ASU to perform a sample BIOS setting configuration change. You can modify the default `default_settings.ini` file supplied by the Toolkit to make other BIOS setting changes as well. This file is located in the `sgdeploy\sgtkwinpe\asufiles` directory.

- Step 1. Open a command prompt, change directory to `\sgshare\sgdeploy\SGTKWinPE`.
- Step 2. Create the ASU installation image by running **SGTKWinPE.cmd**, using the `ASU_Config_Only_x86.ini` file provided by the Toolkit as shown:
`SGTKWinPE.cmd ScenarioINIs\Local\ASU_Config_Only_x86.ini`
The ISO image is created in `..\WinPE_ScenarioOutput\Local_ASU_Config_Only_x86\WinPE_x86.iso`, and your media creation software is started to create bootable media from the image.
- Step 3. Boot the target system from the created media, and follow the on-screen prompts to complete the deployment.

Notes:

- BIOS settings can vary significantly between models. Therefore, use only ASU installation images created from identical models.
- Some systems can display a password value of `<hidden>`. You cannot replicate these systems, even if you replace the value of `<hidden>` with the valid password. To change these values, you must use the **Set** command rather than the **Replicate** command used by the script provided. Refer to the *ASU User's Guide* for more information.

Sample setting files

In addition to the `default_settings.ini` file, other sample setting files are provided in the `sgdeploy\sgtkwinpe\asufiles`:

- `bootorder_settings.ini`: Sets boot order in uEFI.
- `nodes.ini`: Shows how many nodes in a legacy multinode system or in an IMM-based system multinode partition.
- `save_config.ini`: Saves the configuration of the IMM system. You can apply the configuration to another system remotely via the **asu replicate config** command.
- `show_groups.ini`: Views all available setting groups support on an IMM based system.
- `show_values.ini`: Views values of all settings.
- `staticip_settings.ini`: Sets the static IP of IMM. You can change the IP information in this file to meet the demands of the user.
- `baud_rate.ini`: Set the baud rate of the serial port in uEFI. You can change the value of the baud rate in this file to meet demand.

For a static IP configuration, change the IP information in the `staticip_settings.ini` file, and then use the scenario ini file `Asu_Config_Only_x64_StaticIP.ini` to create the ISO as follows: `SGTKWinPE.cmd ScenarioINIs\Local\ASU_Config_Only_x64_StaticIP.ini`

For the boot order setting, change the boot order information in the `bootorder_settings.ini` file, and then use the scenario .ini file `Asu_Config_Only_x64_BootOrder.ini` to create the ISO as follows: `SGTKWinPE.cmd ScenarioINIs\Local\ASU_Config_Only_x64_BootOrder.ini`

For other ASU configurations, change the value of the item `TK_ASU_File` in the scenario .ini file so that it points to the corresponding sample setting file, and then use the modified scenario .ini file to create the ISO.

After configuring a system settings task, you can create a task to install one of the supported Windows operating systems. For more information about creating Windows installation packages, see “About customizing Windows installation scenarios” on page 24.

Configuring a Fibre HBA boot task

This procedure describes how to create a boot task to configure the boot information for Fibre Host Bus Adapters (HBAs).

Before you install the operating system, you can configure a Fibre HBA to boot from a Storage Area Network (SAN) and install an operating system to the SAN. To boot from the SAN, the Fibre HBA must be configured with a logical drive mapped to a Logical Unit Number (LUN). The HBA to be configured must have permission to access the logical drive and LUN.

After configuring a Fibre HBA boot task, you can create a task to install one of the supported Windows operating systems. For more information about creating Windows installation packages, see “About customizing Windows installation scenarios” on page 24. For a list of HBAs supported by ServerGuide Scripting Toolkit, see “Fibre Channel HBA support” on page 54.

The following example describes how create a task to perform a local Fibre configuration on the target system to allow the operating system to be installed on the SAN.

- Step 1. Open a command prompt, and change directory to `\sgshare\sgdeploy\SGTKWinPE`.
- Step 2. Create the Fibre configuration ISO image by running the command **SGTKWinPE.cmd**, using the `Fibre_Config_Only_x86.ini` file provided by Toolkit as shown:
`SGTKWinPE.cmd ScenarioNIs\Local\Fibre_Config_Only_x86.ini`
The ISO image is created in `..\WinPE_ScenarioOutput\Local_Fibre_Config_Only_x86\WinPE_x86.iso`, and your media creation software is started to create bootable media from the image.

Note: The default settings in the `Fibre_Config_Only_x86.ini` file configure the first drive mapped to LUN 0 as a boot device. To avoid errors, ensure that your SAN configuration includes at least one logical drive mapped to LUN 0.
- Step 3. Boot the target system from the created media, and follow the on-screen prompts to complete the deployment.

Windows installation deployments

The topics in this section describe the steps for deploying Windows Server 2008 Enterprise Edition x86 locally and Windows Server 2008 Standard Edition x64 over a network.

After creating a RAID configuration task, you can create a task to install one of the supported Windows operating systems. For more information about creating Windows installation packages, see “About customizing Windows installation scenarios” on page 24.

Notes: By default, Windows generates the Windows Computer Name during the deployment process. You can specify the Computer Name by replacing the * character in the answer file that corresponds to your deployment scenario with a name that complies with the restrictions:

- Has a maximum length of 15 characters
- Has at least one letter
- Has any of the following characters: `~ ! @ # $ ^ & * () = + [] { } \ | ; : ' " , < > / ?`

Deploying Windows Server 2008 Enterprise Edition x86 locally

This procedure provides an example of how to create a local deployment of Windows Server 2008 Enterprise Edition x86.

Note: If the target system contains a RAID controller, RAID will be configured as part of the installation.

- Step 1. In a command prompt window, enter `CD` to change to the `\sgshare\sgdeploy\SGTKWinPE` directory.

- Step 2. Create the Windows installation image by running **SGTKWinPE.cmd**, using the Win2008_x86_EE.ini file provided by the Toolkit as shown in:SGTKWinPE.cmd ScenarioINIs\Local\Win2008_x86_EE.ini
The following ISO image is created:
sgdeploy\WinPE_ScenarioOutputs\Local_Win2008_x86_EE\WinPE_x86.iso. The media creation software is started and creates bootable media from the image.
- Step 3. Boot the target system from the bootable media, and follow the on-screen prompts to complete the deployment.

Deploying Windows Server 2008 Standard Edition x64 over a network

This procedure provides an example of how to create a network deployment of Windows Server 2008 Standard Edition x64.

If the target system contains a RAID controller, RAID will be configured as part of the installation.

- Step 1. In a command prompt window, enter CD to change to the : \sgshare\sgdeploy\SGTKWinPE directory.
- Step 2. Verify the credentials used to access the shared Toolkit folder used in Win2008_x64_SE.ini.
- Step 3. Create the Windows installation image by running **SGTKWinPE.cmd**, using the Win2008_x64_SE.ini file provided by the Toolkit as shown in:
SGTKWinPE.cmd ScenarioINIs\Network\Win2008_x64_SE.ini
The following ISO image is created:
..\WinPE_ScenarioOutput\Network_Win2008_x64_SE\WinPE_x64.iso. The media creation software is started and creates bootable media from the image.
- Step 4. Boot the target system from the created media, and follow the on-screen prompts to complete the deployment.

Installing Windows applications and upgrades

You can use the Toolkit to create a complete deployment package that configures RAID, installs the Windows operating system, and installs selected applications and updates.

Complete the following prerequisite steps:

- Add the Systems Director Agent files to the source tree as described in “Adding Systems Director Agent files” on page 7
- Add the UXSP files to the source tree as described in “Adding UpdateXpress System Packs” on page 7.

This procedure provides an example of how to create a RAID1 configuration image that includes:

- A policy file
- Local installation of Windows Server 2008 Enterprise Edition x86
- An installation of Systems Director Agent and UpdateXpress System Pack updates.

- Step 1. At a command prompt, change the directory to \sgshare\sgdeploy\SGTKWinPE.
- Step 2. Modify the ScenarioINIs\Local\Win2008_x86_EE.ini file as follows:
- a. To configure RAID1 by using a policy file, set TK_PRAID_PolicyFile=RAID1.ini.
 - b. To install Systems Director Agent set TK_NOS_PerformDirectorAgentInstallation=Yes.
 - c. To install the UpdateXpress System Pack updates, set
TK_NOS_PerformPostOSInstallUXSPUpdates=Yes.
- Step 3. Create the ISO image for deployment by running the command **SGTKWinPE.cmd**, using the file you modified:

```
SGTKWinPE.cmd ScenarioINIs\Local\Win2008_x86_EE.ini
```

The ISO file for deployment is created in:

```
..\WinPE_ScenarioOutput\Local_Win2008_x86_EE\WinPE_x86.iso, and your media creation software is started for creating bootable media from the image.
```

- Step 4. Boot the target system from the bootable media, and follow the on-screen prompts to complete the deployment.

Chapter 4. Customized deployment scenarios

This section provides information about how to customize deployment scenarios.

You can customize your deployment scenarios in the following ways:

- Customize source server settings
- Add PRAID policy files
- Add ASU files
- Customize a Fibre HBA boot configuration
- Customize your Windows installation
- Add the installation of Systems Director Agent to your deployment
- Add installation of UpdateXpress System Packs to your deployment
- Modify the Windows PE image
- Add custom scripts to the SGTkWinPE process
- Add files to the Windows PE image
- Automate the deployment process

Customizing source server settings

This topic describes the variables used by ServerGuide Scripting Toolkit, Windows Edition and explains how to customize them for your server.

ServerGuide Scripting Toolkit, Windows Edition uses variables to set the following source server attributes:

- Server name
- Shared folder name
- User ID and password
- Network share drive assignment

These values are located in the [Network Share Settings] section of the .ini files used by Toolkit.

Even if you accept the other defaults as part of the installation, it might be necessary to change the server name variable to match your server.

The Toolkit also uses variables to set the following TCP/IP configuration attributes:

- DHCP
- Client IP address and subnet mask
- Gateway address
- Windows PE firewall

These values are located in the [Network TCP/IP Settings] section of the .ini files used by Toolkit.

The default value for DHCP is Yes, indicating that Toolkit will use DHCP to obtain IP addresses. If you use the default value for DHCP, it is not necessary to modify any of the other IP address variables. If you choose to use static addresses, you must check the other IP address variables to ensure that they are appropriate for your network configuration.

For more information about these variables, see the appropriate sections in Table 10 “User definable settings for SGTkWinPE.cmd” on page 35.

Adding device drivers to Windows installations

This topic explains the process for adding device drivers to a Windows installation task.

- To add device drivers to your Windows installations, you must install System Enablement Packs (SEPs). Any uninstalled SEPs are automatically installed when you run the **SGTKWinPE.cmd** script. You can also install SEPs manually by using the **InstallSEP s.cmd** script. For more information about SEPs, see Chapter 5 “Downloading and installing System Enablement Packs (SEPs)” on page 51.
- After an SEP has been installed, the device drivers used by the Toolkit are saved in the `sgdeploy\drvs` directory. During installation, the Toolkit reads the Supported Systems and Windows Version fields in the `DrvSet.ini` file and uses that information to locate the correct device driver set for the deployment. The Toolkit then matches this information to the `DrvInfo.ini` file in each device driver subdirectory and adds the drivers to the deployment, if the platform ID of the system being deployed is in the Supported Systems field of the file.

Note: The platform ID of a system is the first two characters of its BIOS or uEFI build ID. For example, a System x3250 M3 can have a uEFI build ID of `GYE135A`, making the platform ID for that system `GY`. Identifying and installing device drivers by a system's machine type is still supported, but only if that system's platform ID is not present in either the `DrvSet.ini` or the `DrvInfo.ini` file.

To add a device driver to an existing driver set, follow these steps:

- Step 1. Ensure that either the platform ID or machine type, but not both, of the system supported by the driver is present in the Supported Systems field of the `DrvSet.ini` file in the root directory of the driver set to which you are adding it.
- Step 2. Create a new subdirectory with no spaces or special characters in the root directory of the driver set.
- Step 3. Copy your driver files into the new directory. Do not create any subdirectories under this new directory. Here the term "driver files" refers to raw data files (typically `*.inf`, `*.sys`, `*.dll`, and `*.cat`).
- Step 4. If the device driver that you have is an executable program, install it to a system, and then fetch the raw data file from the installation directory.
- Step 5. Create a new `DrvInfo.ini` file in the new directory. The easiest way to do this is to make a copy of a `DrvInfo.ini` file from one of the other driver folders already present in the driver set.
- Step 6. Update the following fields in the `DrvInfo.ini` file as follows:

Installation Mode

Set to: **PnP** for all drivers for Windows Server 2008 and Windows Server 2012.

Supported Systems

The 2-digit platform ID or 4-digit machine type (but not both) of the system to which you are installing.

Path

If the **Installation Mode** parameter is set to **Textmode**, this parameter must point to the `txtsetup.oem` file. Your driver directory must contain this file.

The following example shows the required `DrvInfo.ini` fields for a network driver installation on a LenovoBladeCenter HS22 with platform ID `P9`, machine type `7978`:

```
Installation Mode = PnP
Supported Systems = P9
Path =
```


The following example shows the required `DrvInfo.ini` fields for a Windows Server 2008 storage driver installation using the machine type for a System x3850 M2, machine type 7141, platform ID A3:

```
Installation Mode = Textmode
Supported Systems = 7141
Path = txtsetup.oem
```

Adding Windows PE drivers

You can add device drivers to an existing Windows PE device driver set or you can create a new driver set.

To add device drivers to your Windows installations, System Enablement Packs (SEPs) must be installed. For more information about SEPs, see Chapter 5 “Downloading and installing System Enablement Packs (SEPs)” on page 51.

- After an SEP is installed, the Windows PE device drivers are located in subdirectories of the `sgdeploy\SGTKWinPE\Drivers` directory. The subdirectories are named `WinPE_architecturetype_TK_YYYY-MM-DD`, where *architecturetype* is the processor architecture, x86 or x64, and *YYYY-MM-DD* is the date that the drivers were created.
- Each driver set has a `DrvSet.ini` file in the root of the directory. The Toolkit Windows PE build process reads this file and uses the fields **Windows Version**, **Creation Date**, and **Creation Time** to determine which drivers are the most recent, and therefore will be installed. By modifying the directory structure and creating a new `DrvSet.ini` file, you can either add drivers to existing driver sets, or you can create a new driver set.

Adding drivers to an existing driver set

This topic describes the process for adding drivers to a supplied Windows PE device driver set.

- Step 1. Create a new subdirectory with no spaces or special characters in the root directory of the latest WinPE driver set of the desired architecture.
- Step 2. Copy the new driver files into the new directory. Do not create any subdirectories under this new directory, or the drivers will not be installed correctly. In this step, driver files refer to raw data files (typically `*.inf`, `*.sys`, `*.dll`, and `*.cat`).
- Step 3. If the device driver that you have is an executable program, install it to a system, and then fetch the raw data file from installation directory.
- Step 4. Run the build process to install each driver in the set, including your addition.

Adding drivers to a new device driver set

This topic describes the process for creating a new device driver set.

If there are no conflicting file names, you can put multiple drivers in a single directory and all of them will be installed.

- Step 1. Create a subdirectory for the new driver set in: `sgdeploy\SGTKWinPE\Drivers`. The subdirectory name cannot have any spaces or special characters.
- Step 2. Create subdirectories in this new folder and copy your driver files into the subdirectories.
- Step 3. Create a file called `DrvSet.ini` in the root directory of your new driver set. The easiest way to do this is to make a copy of a `DrvSet.ini` file from one of the WinPE driver sets that came with the Toolkit.
- Step 4. Update the fields in the `DrvSet.ini` file by using the following format:
[Main]
Supported Systems = all
Windows Version = WinPE21*architecture_type*

Scripting Toolkit Version = 2.1
Creation Date = *DD Month YYYY*
Creation Time = *HH:MM:SS*
BuildID = 2008-06-01
Where:

- **Windows Version** indicates the architecture type of your driver set, such as: *WinPE21x86*, *WinPE21x64*, *WinPE40x86*, or *WinPE40x64*.
 - The WinPE21 version drivers are also installed with Windows PE 3.0. You can use the same indicators for deployment scenarios built with WAIK 1.1 or 2.0.
 - The WinPE40 version drivers are installed with Windows PE 4.0 for deployment scenarios built with ADK.
- **Creation Date** is the day you created the driver set. Use the format: *DD Month YYYY*.
- **Creation Time** is the time you created the driver set. Use the format *HH:MM:SS*.

Step 5. Run the build process to install all of the drivers in the set.

Adding hotfixes to Windows PE

This topic describes the process for adding hotfixes to Windows PE.

These steps describe how to install the hotfixes released by Microsoft for Remote NDIS support in Windows PE. The process can be used to install other hotfixes to Windows PE provided by Microsoft, but the Toolkit has only been tested to support the fixes specifically mentioned in this section.

To use the WinPE 2.1 hotfix with ServerGuide Scripting Toolkit, Windows Edition, follow these steps:

1. Download the hotfix packages here:
<http://download.microsoft.com/download/2/8/1/2814E049-BF80-44A2-86BC-FDD02EDFC16C/MSKB981138.zip>.
2. Copy the hotfix files to `sgdeploy\updates\winpe2\architecture`, where *architecture* is either x86 or x64, depending on the architecture for which you are building.
3. Build your deployment scenario by using the **SGTKWinPE.cmd** script; the fixes will be applied to the boot image.

To use the WinPE 3.0 hotfix with ServerGuide Scripting Toolkit, Windows Edition, follow these steps:

1. Download the hotfix packages according to the instructions here: .
2. Run the executable hotfix packages, and enter the passwords provided with the download.
3. Extract the packages to: `sgdeploy\updates\winpe3`.
4. Build your deployment scenario by using the **SGTKWinPE.cmd** script; the fixes will be applied to the boot image.

To use the WinPE 4.0 hotfix with ServerGuide Scripting Toolkit, Windows Edition, follow these steps:

1. Acquire the hotfix packages.
2. Run the executable hotfix packages and enter the passwords provided with the download.
3. Extract the packages to: `sgdeploy\updates\winpe4`.
4. Build your deployment scenario by using the **SGTKWinPE.cmd** script; the fixes will be applied to the boot image.

PRAID policy files

Add PRAID policy files to a deployable image to customize your RAID configuration with PRAID.

By default, RAID devices are configured with default settings. To override these default settings, you can use a policy file, which can be either one that is provided or a custom policy file.

- For more information about PRAID, see “praid.exe” on page 57.

Note: PRAID.exe can capture an existing RAID configuration that you can then use for RAID replication. For more information, see “praid.exe” on page 57.

- For more information about SGTkWinPE user settings, see “SGTKWinPE user settings” on page 35.
- Table 1 “Sample PRAID policy files” on page 19 lists the sample policy files provided in the `\sgdeploy\SGTKWinPE\PolicyFiles` directory and the policies they implement.

This table provides a list of PRAID policy file names and a description of the policies each implements.

Table 1. Sample PRAID policy files

Policy file name	Policy implemented
RAID0.ini	Creates a RAID 0 array using all drives
RAID1-1.ini	Creates a RAID 1 array with the first two drives and a RAID 1 array with the 3rd and 4th drives
RAID1-5.ini	Creates a RAID 1 array with the first two drives and a RAID 5 array with the remaining drives
RAID1.ini	Creates a RAID 1 array with the first two drives
RAID1HSP.ini	Creates a RAID 1 array with the first two drives and designates the 3rd drive as a hot-spare drive
RAID5.ini	Creates a RAID 5 array using all drives
RAID5HSP.ini	Creates a RAID 5 array using all drives except for one drive designated as a hot-spare drive
RAID6.ini	Creates a RAID 6 array using all drives.
RAID6HSP.ini	Creates a RAID 6 array using all drives except for one drive designated as a hot-spare drive.
RAID10.ini	Creates a RAID 10 array using the first four drives
RAID50.ini	Creates a RAID 50 array using all drives
RAID60.ini	Creates a RAID 60 array using all drives

Adding PRAID policy files

Customize your RAID configuration with PRAID by adding PRAID policy files to your deployable image.

By default, RAID devices are configured with default settings. To override these default settings, you can use a policy file, which can be either one that is provided or a custom policy file.

To use a policy file, complete these steps:

- Step 1. Create a custom policy file and copy it to the `\sgdeploy\SGTKWinPE\PolicyFiles` directory.
- Step 2. Specify the name of the desired sample or custom policy file by using the `TK_PRAID_PolicyFile` variable in the [PRAID Settings] section of the SGTkWinPE user settings .ini file.

Any policy files that exist in the `\sgdeploy\SGTKWinPE\PolicyFiles` directory will be copied to the Windows PE image, but only one can be activated at a time for automatic execution.

Adding ASU files

By adding customized configuration files to the Windows PE image, you enable ASU to modify deployments, such as changing the boot order.

Any ASU files that exist in the `\sgdeploy\SGTKWinPE\ASUFiles` directory will be copied to the Windows PE image, but only the one specified by `TK_ASU_File` will be activated for automatic execution.

The `TK_ASU_File` variable is used to customize the ASU files in the image.

This variable specifies the ASU configuration file to use. This value cannot contain any path information. The specified ASU configuration file must exist in the `SGTKWinPE\ASUFiles` directory.

The default value of this variable is `default_settings.ini`.

To add ASU files to the image, complete the following steps.

- Step 1. Create or obtain the customized file and copy it to `\sgdeploy\SGTKWinPE\ASUFiles`.
- Step 2. Specify the name of the new file by using the `TK_ASU_File` variable in the [ASU Settings] section of the `STGKWinPE` user settings `.ini` file.

About customizing a Fibre HBA boot configuration

You can use Toolkit variables to customize the configuration of Fibre HBAs on a target system, allowing them to boot from SAN targets.

By default, ServerGuide Scripting Toolkit configures the first QLogic HBA on the system to boot from the first available SAN target for QLogic Fibre HBAs only. Emulex Fibre HBAs are not supported. For more information, see “Known problems and limitations” on page 115.

The BIOS configures the first disk drive that it finds as a boot device, that is also a `LUN 0`. ServerGuide Scripting Toolkit uses the variables in the following table to configure Fibre HBAs.

Note: In some examples that follow, single lines are broken into multiple lines for formatting reasons. When using these settings, you must present all of the information for each variable on a single line.

Table 2. Fibre HBA boot configuration variables

Variable	Description
<code>TK_FIBRE_COUNT</code>	<p>Specifies the number of HBA ports to configure.</p> <p>Valid values are $1n$, where n is the number of HBA ports available.</p> <p>This variable affects the use of the following variables:</p> <ul style="list-style-type: none">• <code>TK_FIBRE_N_HBA_ID</code>• <code>TK_FIBRE_N_BOOT_DISABLE</code>• <code>TK_FIBRE_N_BOOT_PRIM</code>• <code>TK_FIBRE_N_BOOT_ALT1</code>• <code>TK_FIBRE_N_BOOT_ALT2</code>• <code>TK_FIBRE_N_BOOT_ALT3</code> <p>Where N is the HBA number to be configured.</p>

Table 2. Fibre HBA boot configuration variables (continued)

Variable	Description
	<p>Note: You must complete one of each of these variables for every HBA port you configure. So if TK_FIBRE_COUNT=2, you must complete one set of these variables for the first port and one for the second.</p>
TK_FIBRE_N_HBA_ID	<p>Identifies the Qlogic/Emulex HBA to be configured, where <i>N</i> is the HBA number to be configured.</p> <p>Valid values are:</p> <p><i>hba_instance</i> The instance number of an HBA port. Valid values are integers from 0 to <i>n</i>-1, where <i>n</i> is the number of HBAs in the system. For example, to configure HBA instance 0: TK_FIBRE_1_HBA_ID=0.</p> <p><i>hba_wwpn</i> The World Wide Port Name of an HBA port, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. For example, to configure HBA: 90-87-AA-BB-65-34-BB-E0: TK_FIBRE_1_HBA_ID= 90-87-AA-BB-65-34-BB-E0 Default: 0</p> <p>Identifies the Brocade HBA to be configured, where <i>N</i> is the HBA number to be configured.</p> <p>Valid values are:</p> <p><i>hba_instance</i> The instance number of an HBA port. A valid format is <i>N/P</i>, where <i>N</i> is the adapter number from 1 to <i>N</i>, and <i>P</i> is the port number from 0 to <i>p</i>-1. For example: TK_FIBRE_1_HBA_ID=1/0.</p> <p><i>hba_wwpn</i> The World Wide Port Name of an HBA port, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. For example, to configure HBA: 90-87-AA-BB-65-34-BB-E0: TK_FIBRE_1_HBA_ID= 90-87-AA-BB-65-34-BB-E0 Default: 0</p>
TK_FIBRE_N_BOOT_DISABLE	<p>Disable the selected current boot device settings on the specified HBA port, where <i>N</i> is the HBA number to be configured.</p> <p>Valid values are:</p> <p>No Does not clear or disable any boot settings.</p> <p>All Disables the primary and all alternate boot settings:</p> <ul style="list-style-type: none"> • Prim • Alt1 • Alt2 • Alt3 <p>Prim Disables only the primary boot setting.</p>

Table 2. Fibre HBA boot configuration variables (continued)

Variable	Description
	<p>Alt1 Disables the Alternative 1 boot setting.</p> <p>Alt2 Disables the Alternative 2 boot setting.</p> <p>Alt3 Disables the Alternative 3 boot setting.</p> <p>Default: No</p>
<p>TK_FIBRE_ <i>N</i> _BOOT_PRIM = <i>target_wwn</i><i>target_wwpn</i><i>lun_id</i></p>	<p>Defines the primary boot target settings, where <i>N</i> is the HBA number to be configured, and:</p> <ul style="list-style-type: none"> • <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>lun_id</i> is the Logical Unit Number of a device. <p>Default: 0 0 0</p> <p>Example: TK_FIBRE_1_BOOT_PRIM= BB-CC-AA-BB-65-34-BB-F1 BB-CC-AA-BB-FF-34-BB-F1 9</p>
<p>TK_FIBRE_ <i>N</i> _BOOT_ALT1 = <i>target_wwn</i><i>target_wwpn</i><i>lun_id</i></p>	<p>Configures the operating system to use the indicated target as the first alternate boot device, where <i>N</i> is the HBA number to be configured, and</p> <ul style="list-style-type: none"> • <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>lun_id</i> is the Logical Unit Number of a device. <p>Default: blank</p> <p>Example: TK_FIBRE_1_BOOT_ALT1= BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5</p>
<p>TK_FIBRE_ <i>N</i> _BOOT_ALT2 = <i>target_wwn</i><i>target_wwpn</i><i>lun_id</i></p>	<p>Configures the operating system to use the indicated target as the second alternate boot device, where <i>N</i> is the HBA number to be configured, and</p> <ul style="list-style-type: none"> • <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>lun_id</i> is the Logical Unit Number of a device. <p>Default: blank</p> <p>Example: TK_FIBRE_1_BOOT_ALT2=</p>

Table 2. Fibre HBA boot configuration variables (continued)

Variable	Description
	BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5
TK_FIBRE_N_BOOT_ALT3 = <i>target_wwn</i> <i>target_wwpn</i> <i>lun_id</i>	<p>Configures the operating system to use the indicated target as the third alternate boot device, where <i>N</i> is the HBA number to be configured, and</p> <ul style="list-style-type: none"> <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. <i>lun_id</i> is the Logical Unit Number of a device. <p>Default: blank</p> <p>Example: TK_FIBRE_1_BOOT_ALT3= BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5</p>

Configuring USB keys for deployment

This procedure describes how to create a bootable Windows PE USB key for any ServerGuide Scripting Toolkit, Windows Edition deployment.

Configuring a USB key for a Toolkit deployment requires the following:

- A system running one of the following Windows sessions:
 - Windows Vista
 - Windows Server 2008
 - Windows 7
 - Windows Server 2008 R2
 - Windows 2.1 PE
 - Windows 3.0 PE
 - Windows 4.0 PE
- A USB key with a storage capacity of at least 64 MB larger than your Windows PE image.

To configure your USB key for deployment, follow this procedure:

Step 1. Insert your USB key in the system.

Step 2. Open a command prompt.

Step 3. Use the **diskpart.exe** command to format the USB key as a single FAT32 partition spanning the entire key, and set the partition to active, as shown in the following example:

```
diskpart
select disk 1
clean
create partition primary
select partition 1
active
format fs=fat32
assign
exit
```

When the USB key has been formatted, you can copy the Toolkit image files to the USB key for deployment. After you format a USB key for Toolkit deployments, you can replace the deployment images on the key and use it for subsequent deployments without reformatting it.

Using a USB key for deployment

This procedure describes how to copy a ServerGuide Scripting Toolkit, Windows Edition deployment image to a USB key for deployment.

Before you can use a USB key for deployment, format it as described in “Configuring USB keys for deployment” on page 23.

Complete the following steps to perform a Toolkit deployment from a USB key.

- Step 1. Create a deployment image. See "Chapter 3. Quick start deployment scenarios."
- Step 2. Copy the contents from the resulting ... \sgdeploy\WinPE_ScenarioOutput directory onto the USB key.
- Step 3. Ensure that you have copied the following directories:
 - boot
 - efi
 - sgdeploy
 - sources
 - bootmgr
- Step 4. Boot the target system from the USB key.
The deployment will execute automatically.

About customizing Windows installation scenarios

This section describes the process and variables used for all of the supported Windows installation scenarios. It describes the parts of a deployment that can be customized and the variables required to customize it.

The Toolkit supports the following Windows installation scenarios:

Windows Edition	Scenario file
Windows Server 2008 R2 DataCenter Edition	Win2008_R2_x64_DE.ini
Windows Server 2008 R2 Enterprise Edition	Win2008_R2_x64_EE.ini
Windows Server 2008 R2 HPC Edition	Win2008_R2_x64_HPC.ini
Windows Server 2008 R2 Standard Edition	Win2008_R2_x64_SE.ini
Windows Server 2008 R2 Web Edition	Win2008_R2_x64_WE.ini
Windows Server 2008 DataCenter Edition	Win2008_x64_DE.ini, Win2008_x86_DE.ini
Windows Server 2008 without Hyper-V DataCenter Edition	Win2008_x64_DE_no_hv.ini, Win2008_x86_DE_no_hv.ini
Windows Server 2008 Enterprise Edition	Win2008_x64_EE.ini, Win2008_x86_EE.ini
Windows Server 2008 without Hyper-V Enterprise Edition	Win2008_x64_EE_no_hv.ini, Win2008_x86_EE_no_hv.ini
Windows Server 2008 Standard Edition	Win2008_x64_SE.ini, Win2008_x86_SE.ini
Windows Server 2008 without Hyper-V Standard Edition	Win2008_x64_SE_no_hv.ini, Win2008_x86_SE_no_hv.ini
Windows Server 2008 Web Edition	Win2008_x64_WE.ini, Win2008_x86_WE.ini

Windows Edition	Scenario file
Windows Server 2012 DataCenter Edition	Win2012_x64_DE.ini
Windows Server 2012 Standard Edition	Win2012_x64_SE.ini
Windows Server 2012 Foundation Edition	Win2012_x64_FE.ini
Windows Server 2012 Storage Edition	Win2012_x64_SS.ini
Windows Server 2012 R2 Standard Edition	Win2012_r2_x64_SE.ini
Windows Server 2012 R2 DataCenter Edition	Win2012_r2_x64_DE.ini

Installing Windows Server 2008 x86 editions locally from a USB key

This procedure describes how to install Windows Server 2008 x86 editions locally from a USB key.

To complete this process you will need a bootable USB key, created as described in “Configuring USB keys for deployment” on page 23.

Create and deploy a Windows Server 2008 x86 installation image by completing these steps:

- Step 1. Insert the USB key into the source server.
- Step 2. Set the Windows Product Key by using the variable *TK_NOS_ProductKey* in the .ini file provided for the Windows Server 2008 edition that you are deploying. You can also set the product key by customizing the answer file located at `sgshare\sgdeploy\SGTKWinPE\AnswerFiles\win2008_x64_EE.xml`.
- Step 3. Open a command prompt and change directory to: `sgshare\sgdeploy\SGTKWinPE`.
- Step 4. Run the **SGTKWinPE.cmd** command by using the provided .ini file that corresponds to the version of the operating system you are deploying. This table provides a list of Windows PE 2.1 editions and their INI file names.

Table 3. Windows PE 2.1 editions and their INI file names

Windows PE 2.1 edition	INI file name
Windows Server 2008 DataCenter Edition	ScenarioINIs\Local\Win2008_x86_DE.ini
Windows Server 2008 Enterprise Edition	ScenarioINIs\Local\Win2008_x86_EE.ini
Windows Server 2008 Standard Edition	ScenarioINIs\Local\Win2008_x86_SE.ini
Windows Server 2008 Web Edition	ScenarioINIs\Local\Win2008_x86_WE.ini
Windows Server 2008 without Hyper-V Enterprise Edition	ScenarioINIs\Local\Win2008_x86_EE_no_hv.ini
Windows Server 2008 without Hyper-V DataCenter Edition	ScenarioINIs\Local\Win2008_x86_DE_no_hv.ini
Windows Server 2008 without Hyper-V Standard Edition	ScenarioINIs\Local\Win2008_x86_SE_no_hv.ini

The Windows PE ISO image is created in the output directory that corresponds to your Windows version. For example, `SGTKWinPE.CMD ScenarioINIs\Local\Win2008_x86_version.ini`.

This table provides a list of Windows Server 2008 editions and their ISO images.

Table 4. Windows Server 2008 editions and their ISO images

Windows edition	ISO image
Windows Server 2008 DataCenter Edition	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x86_DE\WinPE_x86.iso
Windows Server 2008 Enterprise Edition	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x86_EE\WinPE_x86.iso
Windows Server 2008 Standard Edition	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x86_SE\WinPE_x86.iso
Windows Server 2008 Web Edition	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x86_WE\WinPE_x86.iso
Windows Server 2008 without Hyper-V Datacenter Edition	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x86_DE_no_hv\WinPE_x86.iso
Windows Server 2008 without Hyper-V Enterprise Edition	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x86_EE_no_hv\WinPE_x86.iso
Windows Server 2008 without Hyper-V Standard Edition	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x86_SE_no_hv\WinPE_x86.iso

- Step 5. Copy the contents from the resulting sgdeploy\WinPE_ScenarioOutput\ directory to your USB key.
 Step 6. Boot the target system from the USB key, and follow the prompts to complete the deployment.

Installing Windows Server 2008 x64 editions locally

This section describes how to perform a local installation of Windows Server 2008 x64 editions from a DVD.

To complete this process you will need:

- A blank DVD
- A properly-configured source server with a DVD R/W drive
- DVD creation software

Note: When working with a ServeRAID-SCSI controller, separate media are required for RAID configuration and x64 operating system installation. See “No x64 support for SCSI RAID controller configuration” on page 117 for more information.

Create and deploy a Windows Server 2008 x64 installation image by completing these steps:

- Step 1. Label the blank DVD and insert it into the DVD R/W drive of the source server.
- Step 2. Set the Windows Product Key by using the variable TK_N0S_ProductKey in the provided .ini file for the Windows Server 2008 edition that you are deploying. You can also set the product key by customizing the answer file: sgshare\sgdeploy\SGTKWinPE\AnswerFiles\win2008_x64_EE.xml.
- Step 3. Open a command prompt and change directory to sgshare\sgdeploy\SGTKWinPE.
- Step 4. Run the **SGTKWinPE.cmd** command, using the provided .ini file that corresponds to the version of the operating system you are deploying. The Windows PE ISO image is created in the output directory that corresponds to your Windows version. This table lists the Windows 2008 editions and the associated INI and ISO image file names used for local installation.

Table 5. Windows 2008 editions and associated files for local installation

Windows edition	INI file	ISO image
Windows Server 2008 DataCenter Edition	ScenarioINIs\Local\Win2008_x64_DE.ini	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_DE\WinPE_x64.iso
Windows Server 2008 Enterprise Edition	ScenarioINIs\Local\Win2008_x64_EE.ini	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_EE\WinPE_x64.iso
Windows Server 2008 Standard Edition	ScenarioINIs\Local\Win2008_x64_SE.ini	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_SE\WinPE_x64.iso
Windows Server 2008 Web Edition	ScenarioINIs\Local\Win2008_x64_WE.ini	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_WE\WinPE_x64.iso
Windows Server 2008 without Hyper-V Enterprise Edition	ScenarioINIs\Local\Win2008_x64_EE_no_hv.ini	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_EE_no_hv\WinPE_x64.iso
Windows Server 2008 without Hyper-V DataCenter Edition	ScenarioINIs\Local\Win2008_x64_DE_no_hv.ini	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_DE_no_hv\WinPE_x64.iso
Windows Server 2008 without Hyper-V Standard Edition	ScenarioINIs\Local\Win2008_x64_SE_no_hv.ini	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_SE_no_hv\WinPE_x64.iso
Windows Server 2008 R2 DataCenter Edition	ScenarioINIs\Local\Win2008_R2_x64_DE.ini	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_R2_x64_DE\WinPE_x64.iso
Windows Server 2008 R2 Enterprise Edition	ScenarioINIs\Local\Win2008_R2_x64_EE.ini	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_R2_x64_EE\WinPE_x64.iso
Windows Server 2008 R2 Standard Edition	ScenarioINIs\Local\Win2008_R2_x64_SE.ini	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_R2_x64_SE\WinPE_x64.iso
Windows Server 2008 R2 Web Edition	ScenarioINIs\Local\Win2008_R2_x64_WE.ini	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_R2_x64_WE\WinPE_x64.iso

Your DVD creation software starts and is ready to create a bootable DVD using the ISO image.

Step 5. Follow the instructions in your DVD creation software to complete the DVD.

Step 6. Boot the target system by using the newly-created DVD, and follow the on-screen prompts to complete the deployment.

Installing Windows Server 2008 x64 editions over a network

This procedure describes how to install Windows Server 2008 x64 editions over a network share from CD or DVD.

To complete this process you will need the following resources:

- A blank CD or DVD
- A properly-configured source server with a DVD R/W or CD R/W drive. For more information, see “Configuring the source server” on page 4.
- DVD or CD creation software

Note: When working with a ServeRAID-SCSI controller, separate media are required for RAID configuration and x64 operating system installation. See “No x64 support for SCSI RAID controller configuration” on page 117 for more information.

Create and deploy a Windows Server 2008 x64 installation image by completing these steps:

- Step 1. Label the blank disc and insert it into the appropriate R/W drive of the source server.
- Step 2. Set the Windows product key by using the variable TK_NOS_ProductKey in the .ini file for the Windows Server 2008 edition you are deploying. You can also set the product key by customizing the answer file for the appropriate edition, for example:
sgshare\sgdeploy\SGTKWinPE\AnswerFiles\win2008_x64_EE.xml
- Step 3. Open a command prompt and change directory to sgshare\sgdeploy\SGTKWinPE.
- Step 4. Run the SGTKWinPE.cmd command by using the provided .ini file that corresponds to the version of the operating system you are deploying. The Windows PE ISO image is created in the output directory that corresponds to your Windows version. This table lists the Windows 2008 editions and their associated INI and ISO image file names used for network installation.

Table 6. Windows 2008 editions and associated files for Network installation

Windows edition	INI file	ISO image
Windows Server 2008 DataCenter Edition	ScenarioINIs\Network\Win2008_x64_DE.ini	sgdeploy\WinPE_ScenarioOutput\Network_Win2008_x64_DE\WinPE_x64.iso
Windows Server 2008 Enterprise Edition	ScenarioINIs\Network\Win2008_x64_EE.ini	sgdeploy\WinPE_ScenarioOutput\Network_Win2008_x64_EE\WinPE_x64.iso
Windows Server 2008 Standard Edition	ScenarioINIs\Network\Win2008_x64_SE.ini	sgdeploy\WinPE_ScenarioOutput\Network_Win2008_x64_SE\WinPE_x64.iso
Windows Server 2008 Web Edition	ScenarioINIs\Network\Win2008_x64_WE.ini	sgdeploy\WinPE_ScenarioOutput\Network_Win2008_x64_WE\WinPE_x64.iso
Windows Server 2008 without Hyper-V DataCenter Edition	ScenarioINIs\Network\Win2008_x64_DE_no_hv.ini	sgdeploy\WinPE_ScenarioOutput\Network_Win2008_x64_DE_no_hv\WinPE_x64.iso
Windows Server 2008 without Hyper-V Enterprise Edition	ScenarioINIs\Network\Win2008_x64_EE_no_hv.ini	sgdeploy\WinPE_ScenarioOutput\Network_Win2008_x64_EE_no_hv\WinPE_x64.iso
Windows Server 2008 without Hyper-V Standard Edition	ScenarioINIs\Network\Win2008_x64_SE_no_hv.ini	sgdeploy\WinPE_ScenarioOutput\Network_Win2008_x64_SE_no_hv\WinPE_x64.iso
Windows Server 2008 R2 DataCenter Edition	ScenarioINIs\Network\Win2008_R2_x64_DE.ini	sgdeploy\WinPE_ScenarioOutput\Network_Win2008_R2_x64_DE\WinPE_x64.iso
Windows Server 2008 R2 Enterprise Edition	ScenarioINIs\Network\Win2008_R2_x64_EE.ini	sgdeploy\WinPE_ScenarioOutput\Network_Win2008_R2_x64_EE\WinPE_x64.iso
Windows Server 2008 R2 Standard Edition	ScenarioINIs\Network\Win2008_R2_x64_SE.ini	sgdeploy\WinPE_ScenarioOutput\Network_Win2008_R2_x64_SE\WinPE_x64.iso
Windows Server 2008 R2 Web Edition	ScenarioINIs\Network\Win2008_R2_x64_WE.ini	sgdeploy\WinPE_ScenarioOutput\Network_Win2008_R2_x64_WE\WinPE_x64.iso

Your DVD creation software starts and is ready to create a bootable disc using the ISO image.

- Step 5. Follow the instructions in your media creation software to complete the disc.
- Step 6. Boot the target system by using the created disc, and follow the on-screen prompts to complete the deployment.

Installing Windows remotely

This procedure describes how to perform an unattended remote installation of Windows using an RSA II adapter, Integrated Management Module (IMM), BladeCenter Management Module, or BladeCenter Advanced Management Module.

To perform this task you need:

- An ISO image or a physical CD or DVD created according to the instructions in the previous sections for local and network unattended installation scenarios for Windows .
- Network connectivity to the Remote Supervisor Adapter II, Integrated Management Module, BladeCenter Management Module, or BladeCenter Advanced Management Module from the source system.
- A properly configured browser on the source system:
 - Ensure that the Java plug-in is installed. The control panel will reflect a Java Control Panel icon if a proper plug-in is installed. In addition, the Java plug-in may require that browser caching be disabled for the remote interface to work properly.
 - Disable pop-up blocking in the web browser on the source server.
- For network deployment scenarios, you must have:
 - A source system that contains a properly set up source tree. (For more information, see “Configuring the source server” on page 4.)
 - Network connectivity from the target server to the source system shared directory.
- For Windows Server 2008 installations, ISO images are not supported. You must use a physical CD or DVD.
- Network deployment scenarios take significantly less time than local deployments.
- Ensure that there is no disc is in the CD or DVD-ROM drive of your target server.
- If you encounter problems, try updating the RSA II, IMM, MM, or AMM firmware to the latest level.

To deploy your installation image through RSA II, MM, or AMM, complete these steps:

- Step 1. Open a web browser on the source system.
- Step 2. Log in to the adapter with your user ID and password, and select a timeout value for your session. Unless you select **no timeout**, your remote session might time out before your deployment is complete.
- Step 3. Click **Continue** to proceed to the task menu.
- Step 4. From the Task menu, select **Remote Control**.
- Step 5. Click the appropriate link or button to start a remote control session. Your remote control session opens in a new window. If the new window does not open, make sure all popup blockers are disabled, and retry. If the Remote Disk and Remote Console sections do not appear in the window within a few minutes, try clicking in the window. If they still fail to load, check to make sure the Java plug-in is installed.
- Step 6. For the MM and AMM, select the desired target blade in both the **Media Tray** and **KVM Owner** lists.
- Step 7. Select your deployment media in the **Available Resources** box.
 - For a CD or DVD, select the drive containing your disc, then add it the list of resources by clicking the >> box.
 - For an image, click **Select Image**, click the >> box, then browse to your image file. Select the image file, and click **Open**.
- Step 8. Select the deployment media again in the **Selected Resources** box.
- Step 9. Select **Write Protect**.
- Step 10. Click the **Mount** button to mount the disc or image.
- Step 11. Reboot the target system.
- Step 12. When the target system reboots it will load the deployment media. Use the interactive Remote Console video window to respond to any prompts. If you are performing the RAID configuration

only portion of a Windows Server 2008 x64 deployment, the target server will restart automatically. When it does, wait for Windows PE to exit completely and then unmount the drive to "eject" the deployment media. Then select your media again as above and proceed with the installation.

Step 13. The Network Operating System Installation instructions prompt will be displayed and explain how to continue to the unattended portion of the operating system installation.

Step 14. Press a key to reboot the system.

Step 15. After Windows PE has exited completely and the target server is booting, unmount the drive to eject the deployment media.
The target server boots from the hard disk to continue the operating system deployment. You can monitor progress through the Remote Console.

Installing Windows remotely using Integrated Management Module

This procedure describes how to perform an unattended installation of Windows by using an Integrated Management Module (IMM).

To complete this process you will need:

- An ISO image or a physical CD or DVD created according to the instructions in the previous sections for local and network Windows unattended installation scenarios.
- Network connectivity to the Integrated Management Module from the source system.
- A properly configured browser on the source system:
 - Ensure that the Java plug-in is installed. The control panel should display a Java Control Panel icon if a proper plug-in is installed. In addition, the Java plug-in may require that browser caching be disabled for the remote interface to work properly.
 - Disable popup blocking in the web browser on the source server.
- For network deployment scenarios:
 - A source system that contains a properly set up source tree.
 - Network connectivity to the source system shared directory from the target server.
- For Windows Server 2008 installations, ISO images are not supported. Physical CDs or DVDs are required.
- Network deployment take significantly less time than local deployments.
- Make sure that no disc is present in the CD or DVD-ROM drive of your target server.
- If you encounter problems, try updating the IMM firmware to the latest level.

To deploy your installation image through the IMM, complete these steps:

Step 1. Open a web browser on the source system.

Step 2. Enter the host name or IP address of the IMM in address field.

Step 3. Log in to the adapter with your user ID and password, and select a timeout value for your session. Unless you select **no timeout**, your remote session might time out before your deployment is complete.

Step 4. Click **Continue** to proceed to the task menu.

Step 5. From the task menu, select **Remote Control**.

Step 6. Click the appropriate **Start Remote Control** button. Your remote control session opens in a new window. If the new window does not open, make sure all popup blockers are disabled, and then retry the procedure.

If the Remote Disk and Remote Console sections do not appear in the window within a few minutes, try clicking in the window. If they still fail to load, check to make sure the Java plug-in is installed.

Step 7. Select your deployment media in the **Client View** list of the Virtual Media Session window.

For a CD or DVD, ensure that the media is in the drive, select the drive, and check the **Mapped** box for the drive to map it.

For an image, click the **Add image...** button, select the image file, and click **Open**.

Step 8. Check the **Read Only** box.

Step 9. Reboot the target system.

Step 10. When the target system reboots it will load the deployment media. Use the interactive Remote Console video window to respond to any prompts. If you are working with ServeRAID-SCSI adapters and performing only the RAID configuration portion of a Windows x64 deployment, the target server will restart automatically.

Step 11. When the target server restarts, wait for Windows PE to exit completely and then deselect the **Mapped** box to "eject" the deployment media. Then select your media again, and proceed with the installation.

Step 12. The Network Operating System Installation instructions prompt will be displayed and explain how to continue to the unattended portion of the operating system installation.

Step 13. Press a key to reboot the system.

Step 14. After Windows PE has exited completely and the target server is booting, deselect the **Mapped** box to "eject" the deployment media.

The target server boots from the hard disk to continue the operating system deployment. You can monitor progress via the Remote Console.

Adding an installation of the Systems Director Agent

This procedure describes how to add the installation of the Systems Director Agent to your deployment.

To complete this process, you need:

- The Systems Director Agent installation files, obtained from either the Systems Director Agent installation CD or the IBM website: .
- A properly configured source server. (See "Configuring the source server" on page 4.)

To add the installation of the Systems Director Agent to your deployment:

Step 1. Add the Systems Director Agent installation files to the source tree. For more information, see "Adding Systems Director Agent files" on page 7.

Step 2. Set the Systems Director Agent installation variable, *TK_NOS_PerformDirectorAgentInstallation*, in the scenario .ini files you are using to Yes. The variable is located in the [NOS Installation Settings] section of the SGTKWinPE user settings. For more information, see "SGTKWinPE user settings" on page 35.

The Systems Director Agent is installed after the Windows operating system installation is complete.

Adding the installation of UpdateXpress System Packs to your deployment

This topic describes how to install UpdateXpress System Pack updates as part of your deployment.

To complete this process, you will need to either obtain the UpdateXpress System Pack from this website: , or use the UpdateXpress System Pack Installer (UXSPi) provided with ServerGuide Scripting Toolkit to acquire updates using: `sgdeploy\updates\uxsp\ibm_utl_uxspi_x.xx_winsrvr_32-64.exe`. For more information about how to obtain updates using UXSPi, see “Using UXSPi to download updates” on page 119.

To add the installation of UXSP updates to your deployment:

- Step 1. Download UpdateXpress System Pack for the desired machine type and operating system combination into the source tree in the `updates\uxsp` directory. Always choose the same destination directory for each UpdateXpress System Pack download. If you are prompted, overwrite any existing files that are duplicated in the new download.
- Step 2. Set the UpdateXpress environment variable, `TK_NOS_PerformPostOSInstallUXSPUpdates` under the Applications Settings section in the scenario .ini file, to Yes. This variable is located in the [NOS Installation Settings] section of the SGTKWinPE user settings. For more information, see “SGTKWinPE user settings” on page 35.

UXSP updates are installed after the Windows operating system installation is complete.

Verify that the updates were installed by checking the log file `C:\Lenovo_Support\Lenovo_WinPEToolkit.log`.

Modifying the Windows PE image

You can manually modify the Windows PE build image by creating directories, adding files, modifying files and scripts, or making any other changes required by your deployment. The following sections describe the methods for modifying the Windows PE build image.

If you modify the image manually, you must recreate the ISO deployment file. When you have completed your changes, run the **SGTKWinPE.cmd** script by using the .ini file that you used previously, adding the `/ISO` switch to regenerate the ISO file. Then run the **SGTKWinPE.cmd** script again, using the .ini file that was previously used, and add the `/StartISO` switch to recreate the CD or DVD, as described in the following example procedure.

The example uses the `ScenarioINIs\Network\Windows_x86_EE.ini` settings file.

- Step 1. Change directory to `\sgshare\sgdeploy\sgtkwinpe`.
- Step 2. Run the following command to generate the initial image without creating an ISO image or starting the media burning software: `SGTKWinPE.cmd ScenarioINIs\Network\Windows_x86_EE.ini /Image`
- Step 3. Manually modify the generated Windows PE image in the `\sgshare\sgdeploy\WinPE_ScenarioOutput\Network_Win2008_x86_EE\ISO` directory.
- Step 4. Create the ISO file with the modified image by using the **SGTKWinPE** command and the `/ISO` option, as shown: `SGTKWinPE.cmd ScenarioINIs\Network\Windows_x86_EE.ini /ISO`
- Step 5. Start the CD burning software by using the **SGTKWinPE** command and the `/StartISO` option as shown: `SGTKWinPE.cmd ScenarioINIs\Network\Windows_x86_EE.ini /StartISO`

Requirements for adding custom scripts to the Windows PE build process

This topic describes the requirements for adding custom scripts to the creation of the Windows PE image.

LenovoScripting Toolkit uses three variables to control the custom scripts and the time they are run during the SGTKWinPE process:

- `TK_PATH_BeforeWIMMounted_Custom_Script`
- `TK_PATH_WhileWIMMounted_Custom_Script`
- `TK_PATH_AfterWIMMounted_Custom_Script`

To run a custom script during the SGTkWinPE process, set the appropriate variable to the path of the custom script that you want to run. For more information, see “SGTKWinPE user settings” on page 35.

A template script, `Template.cmd`, is provided in the `BuildScripts` directory to use as a base for creating your own custom scripts. At a minimum, any custom script must set the following environment variables on completion:

RC=number

The return code that informs the `SGTKWinPE.cmd` script file that an error occurred when the custom script file was run. The number 0 indicates success, and any other number indicates an error.

RMSG=message

If `RC=number` is set to a non-zero value, set the `RMSG` environment variable to a message that indicates the nature of the error. This message is written to the log file and displayed on the console to indicate the error.

Automating the deployment process

This section describes the steps for automating the deployment process for installations. During deployment, you might be prompted to provide input or take action to continue an installation. The Toolkit provides SGTkWinPE user settings that automate responses to these prompts and create installations that can be run unattended.

You can use these settings to:

- Start the deployment process
- Continue the deployment process
- Restart the deployment
- Prevent stoppages during Windows installation

Automatic start of the Toolkit processes

This topic lists the settings required to automatically start the Toolkit processes.

After the Windows PE image boots, to start the specified Toolkit processes, you are required to press a key. To have the Toolkit processes start automatically, you can use the following variables, which are available in the [Toolkit Process Settings] section of the SGTkWinPE settings .ini files. These variables include the ability to activate the automatic start of the Toolkit processes by setting a timer:

Table 7. *AutoStartToolkitProcesses* setting

Setting	Description
<code>TK_Process_AutoStartToolkitProcesses</code>	<ul style="list-style-type: none">• Enables the automatic start of the Toolkit processes. Setting this value to <code>No</code> causes the user to be prompted to start the Toolkit processes when booting from the Windows PE CD or DVD.• Valid values are: <code>Yes</code>, <code>No</code>• Default: <code>No</code>
<code>TK_Process_AutoStartTimer</code>	<ul style="list-style-type: none">• Automatic start countdown timer in seconds.• Default: <code>60</code>

Automatic continuation of the Toolkit processes

Some Toolkit processes require that a server is restarted before the processes can continue, such as a RAID configuration. By default, the remaining toolkit processes automatically continue after a restart. This

feature can be disabled, requiring you to press a key to continue the remaining Toolkit processes after the Windows PE image boots.

To activate the automatic continuation of the Toolkit processes, the variables listed in this table are available in the [Toolkit Process Settings] section of the SGTkWinPE settings .ini files. This option also uses the timer.

Table 8. *AutoContinueToolkitProcesses* setting

Setting name	Description
TK_Process_AutoContinueToolkitProcesses	<ul style="list-style-type: none"> Enables automatic continuation of Toolkit processes during required reboots and is useful for debugging. Valid setting values: <ul style="list-style-type: none"> – Yes – No – Default: Yes Selecting No turns this setting off and results in prompts to continue the Toolkit processes after Windows PE CD or DVD reboots.
TK_Process_AutoStartTimer	<ul style="list-style-type: none"> Automatic start countdown timer in seconds. Default: 60

Automatic system restart during deployment

The automatic restart setting restarts a system after the Toolkit processes are complete.

The *TK_Process_AutomaticallyRestart* variable in the [Toolkit Process Settings] section of the SGTkWinPE settings .ini files activates the automatic restart function.

Table 9. *AutomaticallyRestart Process* setting

Setting	Description
TK_Process_AutomaticallyRestart	<ul style="list-style-type: none"> Determines whether to restart a system after the selected tasks are complete. Valid setting values: <ul style="list-style-type: none"> – Yes – No – Default: No

The *TK_Process_AutomaticallyRestart* option is used for one of the following situations:

- For new systems that have no file system on a mass storage device.
 - To fully automate the restart of the server at the end of the Toolkit processes:
 - Set the *ForceWinPEBoot* option to No to enable the Microsoft *bootfix.bin* boot procedure, and then set the *TK_Process_AutomaticallyRestart* option to Yes.
 - To full automate the NOS installation procedure.
- Deploying an image using the RSA II adapter.

Unattended mode for Windows installations

ServerGuide Scripting Toolkit provides a sample *unattend.txt* file for each supported version of Microsoft Windows .

The answer files are located in the `SGTKWinPE\AnswerFiles` directory. By default, the `TK_NOS_AnswerFile` variable in the `SGTKWinPE` settings `.ini` files is set to point to one of these sample files.

If you use one of the sample default files during the deployment step, first customize the correct file before starting a deployment. If the Windows version being deployed requires a product ID key, this key must be provided in the scenario `.ini` file or the deployment stops and requests for this information.

SGTKWinPE user settings

Use the information in this topic to customize an image produced by the `SGTKWinPE.cmd` command.

The `SGTKWinPE` settings `.ini` file provides input to the `SGTKWinPE.cmd` file and controls whether options and processes of ServerGuide Scripting Toolkit are enabled or disabled. This allows you to create a wide array of Windows PE solutions. For example, a default image from Microsoft that simply boots to a command prompt, a DVD for local deployment, or a CD for network deployment.

Toolkit provides example `SGTKWinPE` settings `.ini` files for deploying each supported operating system for both local and network deployments in the `sgdeploy\SGTKWinPE\ScenarioINIs` directory.

Before modifying these files or any other Toolkit files, create backup copies of the files you are modifying.

Table 10 “User definable settings for `SGTKWinPE.cmd`” on page 35 describes the user definable parameters in the `.ini` files used by `SGTKWinPE.cmd`.

Note: Note that while some examples are broken across multiple lines for formatting reasons, when using these settings, you must include all the information for each variable on a single line.

Table 10. User definable settings for `SGTKWinPE.cmd`

Setting	Description
[SGTKWinPE Build Settings]	
<code>TK_Build_ScenarioOutputDirectory</code>	Specifies the name of the directory to build the scenario in. Multiple scenarios of the same type can be maintained by changing this value at the start of each build, for example by appending a number. Default: blank
<code>TK_Build_WinPE_Arch_Type</code>	The architecture of the Windows PE image to create. Valid values are: <code>x86</code> , <code>x64</code> Default: <ul style="list-style-type: none"> • <code>x86</code> for Windows Server 2008 x86 deployments • <code>x64</code> for Windows Server 2008 x64 deployments
<code>TK_Build_WinPEPackages</code>	Lists additional packages to install to the Windows PEimage through the <code>peimg /install=pckg</code> command. The Toolkit will automatically install the following packages: <code>*Scripting*</code> , <code>*MDAC*</code> , <code>*WMI*</code> . Valid values are described in the Advanced Installation Kit documentation. Default: blank

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_Build_PeImgPrep	<p>Reduces the size of the Windows PE image by removing packages that are not installed from the image to reduce the size of the image.</p> <p>Valid values are: Yes, No</p> <p>Default: No</p> <p>Note: This setting applies only to the Automated Installation Kit (AIK) for Windows Vista SP1 and Windows Server 2008. If you are using the AIK for Windows 7 and Windows Server 2008 R2, this setting is ignored.</p>
TK_Build_ForceWinPEBoot	<p>Forces the Windows PE CD or DVD to boot without prompting when a file system is present.</p> <p>Ordinarily, when a file system is present when Windows PE boots, it prompts for user input and allows 8 seconds for the user to press a key. The ForceWinPEBoot option disables this feature and forces the Windows PE CD or DVD to always boot.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_Build_InstallWinPEDrivers	<p>Installs the Lenovo Windows PE driver set as part of the Windows PE image.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_Build_EnableSGToolkitEnvironment	<p>Enables the ServerGuide Scripting Toolkit environment in the Windows PE image. If the environment is disabled, then no Toolkit files or environment are copied to the Windows PE image.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_Build_CreateISO	<p>Automatically create the ISO image.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_Build_StartISO	<p>Starts the software associated with ISO files. Typically this will start the CD or DVD burning process for the created ISO or RamdiskISO file. The RamdiskISO file takes precedence if it exists.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_Build_SuppressPrompts	<p>Suppresses interactive prompts during SGTkWinPE script execution.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_Build_DisplayBuildSettings	<p>Display the Toolkit settings from the input SGTkWinPE settings .ini file during SGTkWinPE script execution.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
[Toolkit Environment Settings]	
TK_Environment_ToolkitFilesLocation	<p>Defines whether the Scripting Toolkit binary and script files are to be copied to the local bootable media or run live over the network from the Toolkit source server.</p> <p>Valid values are: Local, Network</p> <p>Default:</p> <ul style="list-style-type: none"> • Local for local deployment scenarios • Network for network deployment scenarios
TK_Environment_PerformToolkitProcesses	<p>Automatically perform Toolkit processes after booting the Windows PE CD or DVD (depending on the values of the individual process settings).</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Yes</p> <ul style="list-style-type: none"> • If set to No, the Toolkit processes may be run manually. • If set to Disabled, the Toolkit processes can not be run manually for local scenarios since the required files are not copied to the DVD.
TK_Environment_DisplayToolkitSettings	<p>Display the relevant Toolkit settings from the SGTkWinPE settings .ini file during deployment scenario execution.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
[Toolkit Process Settings]	
TK_Process_DataFilesLocation	<p>Specifies whether the NOS, driver files and PRAID policy files should be copied to the local bootable media or obtained over the network from the Toolkit source server (local deployment versus network deployment).</p> <p>Valid values are: Local, Network</p> <p>Default:</p> <ul style="list-style-type: none"> • Local for local deployment scenarios • Network for network deployment scenarios

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_Process_PerformRAIDConfiguration	<p>Automatically perform the RAID configuration task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Yes</p> <p>If set to Disabled, the files required for local RAID configuration scenarios are not copied to the DVD.</p>
TK_Process_PerformFibreConfiguration	<p>Automatically perform the Fibre configuration task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Disabled</p>
TK_Process_PerformASUConfiguration	<p>Automatically perform the ASU configuration task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Disabled</p> <p>If this variable is set to Disabled, the files required for local ASU configuration scenarios are not copied to the DVD.</p>
TK_Process_PerformPartitioning	<p>Automatically perform the partitioning task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Yes</p> <p>If set to Disabled, the files required for local partitioning scenarios are not copied to the DVD.</p>
TK_Process_PerformNOSInstallation	<p>Automatically perform the NOS Installation task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Yes</p> <p>If set to Disabled, the files required for local NOS installation scenarios are not copied to the DVD.</p>
TK_Process_AutomaticallyRestart	<p>Specifies that the system be restarted after the selected tasks are complete.</p> <p>Valid values are: Yes, No</p> <p>Default: No</p>

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_Process_AutoStartToolkitProcesses	<p>Enables automatic start of the Toolkit processes. Set this value to Yes to have the Toolkit processes start automatically when the Windows PE CD is booted without prompting the user to press a key.</p> <p>Attention: If the RAID or partitioning processes are set to Yes, this will re-partition or reformat the hard drive automatically without warning when the CD or DVD is booted.</p> <p>Valid values are: Yes, No</p> <p>Default: No</p>
TK_Process_AutoStartTimer	<p>Automatically starts countdown timer in seconds.</p> <p>Default: 60</p>
TK_Process_AutoContinueToolkitProcesses	<p>Enables automatic continuation of Toolkit processes during required reboots. Turning this off causes the user to be prompted to continue the Toolkit processes when Windows PE is rebooted from CD or DVD. Useful for debugging.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
[PRAID Settings]	
TK_PRAID_PolicyFile	<p>Specifies the PRAID policy file to use for the configuration. This value should not contain any path information. The specified PRAID policy file must exist in the SGTkWinPE\PolicyFiles directory.</p> <p>Default: RAIDAuto.ini</p>
TK_PRAID_Flags	<p>Modifies the behavior of RAID configuration or replication.</p> <p>Valid values are: blank, /r:n, /e1, /e2, /e3, /v:n where <i>n</i> is 1, 3, or 5. See “praid.exe” on page 57 for more information.</p> <p>Default: /r</p>
[Fibre Settings]	
TK_FIBRE_COUNT	<p>Specifies the number of HBA ports to configure.</p> <p>Valid values are 1<i>n</i>, where <i>n</i> is the number of HBA ports available.</p> <p>This variable affects the use of the following variables:</p> <ul style="list-style-type: none"> • TK_FIBRE_N_HBA_ID • TK_FIBRE_N_BOOT_DISABLE • TK_FIBRE_N_BOOT_PRIM • TK_FIBRE_N_BOOT_ALT1 • TK_FIBRE_N_BOOT_ALT2 • TK_FIBRE_N_BOOT_ALT3

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
	<p>Where <i>N</i> is the HBA number to be configured.</p> <p>Note: You must complete one of each of these variables for every HBA port you configure. So if TK_FIBRE_COUNT=2, you must complete one set of these variables for the first port and one for the second.</p>
TK_FIBRE_N_HBA_ID	<p>Identifies the Qlogic/Emulex HBA to be configured, where <i>N</i> is the HBA number to be configured.</p> <p>Valid values are:</p> <p><i>hba_instance</i></p> <p>the instance number of an HBA port. Valid values are integers from 0 to <i>n</i>-1, where <i>n</i> is the number of HBAs in the system.</p> <p>For example, to configure HBA instance 0: TK_FIBRE_1_HBA_ID=0.</p> <p><i>hba_wwpn</i></p> <p>The World Wide Port Name of an HBA port, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>.</p> <p>For example, to configure HBA: 90-87-AA-BB-65-34-BB-E0: TK_FIBRE_1_HBA_ID= 90-87-AA-BB-65-34-BB-E0</p> <p>Default: 0</p> <hr/> <p>Identifies the Brocade HBA to be configured, where <i>N</i> is the HBA number to be configured.</p> <p>Valid values are:</p> <p><i>hba_instance</i></p> <p>the instance number of an HBA port. The valid format is <i>N/P</i>, where <i>N</i> is the adapter number from 1 to <i>N</i>, and <i>P</i> is the port number from 0 to <i>p</i>-1.</p> <p>For example: TK_FIBRE_1_HBA_ID=1/0.</p> <p><i>hba_wwpn</i></p> <p>the World Wide Port Name of an HBA port, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>.</p> <p>For example, to configure HBA: 90-87-AA-BB-65-34-BB-E0: TK_FIBRE_1_HBA_ID= 90-87-AA-BB-65-34-BB-E0</p> <p>Default: 0</p>

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_FIBRE_N_BOOT_DISABLE	<p>Disables the selected current boot device settings on the specified HBA port, where <i>N</i> is the HBA number to be configured.</p> <p>Valid values are</p> <p>No Does not clear or disable any boot settings.</p> <p>All Disables the primary and all alternate boot settings: Prim, Alt1, Alt2, and Alt3.</p> <p>Prim Disables only the primary boot setting.</p> <p>Alt1 Disables the Alternative 1 boot setting.</p> <p>Alt2 Disables the Alternative 2 boot setting.</p> <p>Alt3 Disables the Alternative 3 boot setting.</p> <p>Default: No</p>
TK_FIBRE_N_BOOT_PRIM = <i>target_wwn</i> <i>target_wwpn</i> <i>lun_id</i>	<p>Defines the primary boot target settings, where <i>N</i> is the HBA number to be configured, and:</p> <ul style="list-style-type: none"> • <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>lun_id</i> is the Logical Unit Number of a device. <p>Default: 0 0 0</p> <p>Example: TK_FIBRE_1_BOOT_PRIM= BB-CC-AA-BB-65-34-BB-F1 BB-CC-AA-BB-FF-34-BB-F1 9</p>
TK_FIBRE_N_BOOT_ALT1 = <i>target_wwn</i> <i>target_wwpn</i> <i>lun_id</i>	<p>Configures the operating system to use the indicated target as the first alternate boot device, where <i>N</i> is the HBA number to be configured, and</p> <ul style="list-style-type: none"> • <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>lun_id</i> is the Logical Unit Number of a device. <p>Default: blank</p> <p>Example: TK_FIBRE_1_BOOT_ALT1=</p>

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
	BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5
TK_FIBRE_N_BOOT_ALT2 = <i>target_wwn</i> <i>target_wwpn</i> <i>lun_id</i>	Configures the operating system to use the indicated target as the second alternate boot device, where <i>N</i> is the HBA number to be configured, and <ul style="list-style-type: none"> • <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>lun_id</i> is the Logical Unit Number of a device. Default: blank Example: TK_FIBRE_1_BOOT_ALT2= BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5
TK_FIBRE_N_BOOT_ALT3 = <i>target_wwn</i> <i>target_wwpn</i> <i>lun_id</i>	Configures the operating system to use the indicated target as the third alternate boot device, where <i>N</i> is the HBA number to be configured, and <ul style="list-style-type: none"> • <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>lun_id</i> is the Logical Unit Number of a device. Default: blank Example: TK_FIBRE_1_BOOT_ALT3= BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5
[ASU Settings]	
TK_ASU_File	Specifies the ASU configuration file to use. This value cannot contain any path information. The specified ASU configuration file must exist in the SGTkWinPE\ASUFiles directory. Default: default_settings.ini
[Partition Settings]	
TK_Partition_DiskNum	Specifies the disk number on which to create new partition. Valid values are the disk numbers found by diskpart.exe. Default: AUTO Note: The AUTO setting is the first disk on the system.

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_Partition_Size	<p>Specifies the partition size in MB.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • Max • <i>number</i> <p>Max indicates to use all available disk space.</p> <p>Default: Max</p>
TK_Partition_FileSystem	<p>Specifies the file system type to use when formatting the drive.</p> <p>Valid values are: NTFS, FAT32</p> <p>Default: NTFS</p> <p>Note: NTFS is the only valid value for Windows Server 2008 installations.</p>
TK_Partition_SR_Size	<p>Specifies the partition size, in MB, for a System Reserved Partition.</p> <p>The System Reserved Partition is a primary active partition created during the partitioning step. The BitLocker Drive Encryption function requires an active partition and formatted NTFS.</p> <p>Valid values are integers greater than 108. The default is 108 MB.</p> <p>k2</p> <p>Note: This setting is supported only for Windows Server 2008 R2. This setting is ignored when booting WinPE in native uEFI mode.</p> <p>For more information about BitLocker Drive Encryption, see: http://technet.microsoft.com/en-us/library/cc731549%28WS.10%29.aspx</p>
[NOS Installation Settings]	
TK_NOS_NetworkOperatingSystem	<p>Specifies the NOS to use for the deployment. This must be a valid directory name within the sgdeploy\os directory in the Toolkit source server.</p> <p>Default: Varies per scenario based on the NOS being deployed. For example, for Windows Server 2008 Enterprise Edition x86 deployments, the .ini file name is Win2008_x86_EE.ini. The corresponding OS directory in the .ini file is Win2008_x64.</p>

Table 10. User definable settings for SGTKWinPE.cmd (continued)

Setting	Description
TK_NOS_AnswerFile	<p>Specifies the answer file to use for the deployment. This must be a valid file name within the SGTKWinPE\AnswerFiles directory in the Toolkit source server.</p> <p>Default:</p> <ul style="list-style-type: none"> • win2008.xml for Windows Server 2008 x86 • win2008x64.xml for Windows Server 2008 x64 • win2012x64.xml for Windows Server 2012 x64
TK_NOS_DeploymentDriverLibrary	<p>Specifies the deployment driver library to use for the deployment.</p> <p>Valid values are: Auto, <i>dirname</i></p> <p>If set to Auto, Toolkit searches all directories within the sgdeploy\drvs directory and selects the newest DDL present that supports the machine and OS being deployed. If <i>dirname</i> is specified, this must be a valid directory name within the sgdeploy\drvs directory in the Toolkit source server. For example, w23_drv would be specified for sgdeploy\drvs\w23_drv.</p> <p>Default: Auto</p>
TK_NOS_PerformDirectorAgentInstallation	<p>Automatically performs the Director Agent application installation task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Disabled</p> <p>If set to Disabled, the files required to install the Director Agent are not copied to the DVD for local deployment scenarios.</p>
TK_NOS_PerformPostOSInstallUXSPUpdates	<p>Automatically performs the post-OS install UXSP updates task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Yes</p> <p>If set to Disabled, the files required to install UpdateXpress System Packs are not copied to the DVD for local deployment scenarios.</p>
TK_NOS_ProductKey	<p>Specifies the product key to be used when using the answer files provided with the Toolkit.</p> <p>Default: blank</p>

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_NOS_Win2008_ImageName	<p>Sets the Image Name within the Windows 2008 installation image. This variable selects the Windows 2008 Edition.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • Windows Longhorn SERVERSTANDARD • Windows Longhorn SERVERSTANDARDV • Windows Longhorn SERVERDATACENTER • Windows Longhorn SERVERDATACENTERV • Windows Longhorn SERVERENTERPRISE • Windows Longhorn SERVERENTERPRISEV • Windows Longhorn SERVERWEB • Windows Server 2008 R2 SERVERSTANDARD • Windows Server 2008 R2 SERVERDATACENTER • Windows Server 2008 R2 SERVERENTERPRISE 7 • Windows Server 2008 R2 SERVERWEB <p>This value is set by the scenario .ini file.</p> <p>Note: To install the core version of an edition, append CORE to the image name. For example, to install the core version of Windows Server 2008 Standard Edition, use the value Windows Longhorn SERVERSTANDARDCORE.</p>
TK_NOS_Win2008_UILanguage	<p>Specifies the language to install from the Windows 2008 installation image.</p> <p>The default for this value is en-us, which allows the unattended installation of English copies of Windows Server 2008. To perform an unattended installation of Windows Server 2008 images in other languages, change this setting to the value for the language you want to install, for example ja-JP for Japanese.</p> <p>See the lang.ini file from the Windows 2008 media or the Windows Automated Installation Kit documentation for valid values for this variable.</p> <p>Default: en-us</p>
[Director Agent Settings]	
TK_DirAgent_DirectorAgent	<p>Specifies the path to the Systems Director Agent application files within the sgdeploy\apps directory.</p> <p>Default: dwin</p>
[UXSP Settings]	

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_UXSP_FilesLocation	<p>Specifies whether the UpdateXpress System Packs should be copied to the local bootable media or obtained over the network from the Toolkit source server or from the specified remote server.</p> <p>Valid values are: Local, Network</p> <p>Default:</p> <ul style="list-style-type: none"> • Local for local deployment scenarios • Network for network deployment scenarios
TK_UXSP_MakeLocal_Copy	<p>Specifies whether or not the UXSP files should be copied to the target server before they are installed. If TK_UXSP_FilesLocation = Network, the files will be copied to the C:\LenovoSGTK\luxsp directory. If TK_UXSP_FilesLocation = Local, the UXSP files are always copied to the target server before they are installed.</p> <p>Valid values are: Yes, No</p> <p>Default: No</p>
TK_UXSP_RemoteServer	<p>If TK_UXSP_FilesLocation = Network, specifies the network path to the shared directory containing the UXSP files. If blank (no value is specified), then the UXSPs are obtained from the Toolkit source server.</p> <p>Example value: \\MyServerName\UXSPsDirectory</p> <p>Default: blank</p>
TK_UXSP_RemoteServer_UserName	<p>Specifies the user name and optionally the domain name with which to access the server specified with TK_UXSP_RemoteServer.</p> <p>Valid syntax:</p> <ul style="list-style-type: none"> • <i>UserName</i> • <i>UserName@DomainName</i> • <i>DomainName\UserName</i> <p>Default: blank</p>
TK_UXSP_RemoteServer_Password	<p>Specifies the password that corresponds to the user name specified with TK_UXSP_RemoteServer_UserName.</p> <p>Default: blank</p>

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_UXSP_ApplyLatest	<p>Specifies whether UXSPi should apply the latest updates to the target system if no UXSPfiles are found for the target system. Set this variable to Yes to force the UpdateXpress System Pack Installer (UXSPi) to apply the latest updates to the target system if no UXSP files are found for that system.</p> <p>Valid values: Yes, No</p> <p>Default: No</p> <p>Note: In order for individual updates to be installed properly, any XML files included with the update must have lowercase file extensions (for example, xml not XML).</p>
TK_UXSP_UXSPIUpdateFlags	<p>Specifies user-provided command line arguments for processing by the UpdateXpress System Pack Installer (UXSPi) in Update mode. To provide command line arguments to be processed by UXSPi, set this variable to the command line arguments.</p> <p>See "UpdateXpress System Pack Installer" on page 56 for a list of command line arguments to use with UXSPi in Update mode.</p> <p>Default: blank</p>
[Network TCP/IP Settings]	
TK_NetTCPIP_UseDHCP	<p>Specifies whether the Windows PE networking settings should use DHCP. If this option is turned on, then the static specifications for network settings below are disabled automatically by Windows PE.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_NetTCPIP_ClientIPAddress	<p>Specifies the client IP address to use for the networking Windows PE environment.</p> <p>Default: 192.168.0.131</p>
TK_NetTCPIP_Gateway	<p>Specifies the Gateway IP address to use for the networking Windows PE environment.</p> <p>Default: 192.168.0.2</p>
TK_NetTCPIP_WinPEFirewall	<p>Specifies whether the Windows PE firewall should be activated.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_NetTCPIP_Identifier	<p>Specifies the network adapter to configure with static IP settings</p> <p>Default: Local Area Connection</p>
[Network Share Settings]	

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_NetShare_ServerName	<p>Specifies the Toolkit source server computer name.</p> <p>Valid values are: either the host name or IP address of the Toolkit source server.</p> <p>Default: sgtksrv</p>
TK_NetShare_ShareName	<p>Specifies the Toolkit source server shared directory name.</p> <p>Default: sgshare</p>
TK_NetShare_UserName	<p>Specifies the user name, and optionally the domain name, to use when establishing a network connection to the Toolkit source server.</p> <p>Valid syntax:</p> <ul style="list-style-type: none"> • <i>UserName</i> • <i>UserName@DomainNameDomainName\UserName</i> <p>Default: sgtk0clnt</p>
TK_NetShare_Password	<p>Specifies the password to use when establishing a network connection to the Toolkit source server.</p> <p>Default: sgtk0clnt</p>
TK_NetShare_ShareDriveAssignment	<p>Specifies the Share Drive Assignment to use when establishing a network connection to the Toolkit Source Server.</p> <p>Default the default network share drive is Y:</p>
[SGTKWinPE Paths]	
TK_Path_WinPE_Output	<p>Specifies the path for the Windows PE Build Output directory.</p> <p>Default: %TK_Path_SGDeploy%\WinPE_ScenarioOutput</p>
TK_Path_WAIK_Source	<p>This variable has been deprecated. The AIK information is found automatically in the registry. You can uncomment this variable and use it to bypass the registry check, but it is not required.</p> <p>Default (if used): %ProgramFiles%\Windows AIK</p>
[SGTKWinPE Filenames]	
TK_File_WinPE_ISO_Filename	<p>The Windows PE ISO file name</p> <p>Default: WinPE_%TK_Build_WinPE_Arch_Type%.iso</p>
TK_File_WinPE_Log_Filename	<p>The Windows PE build process log file name. Default: WinPE_%TK_Build_WinPE_Arch_Type%.log</p>
[SGTKWinPE Custom Scripts]	
TK_Path_BeforeWIMMounted_Custom_Script	<p>Specifies a custom script to run during the SGTkWinPE build process before the WinPE WIM image is mounted for editing. Use the Template.cmd script as a starting point for writing custom scripts, and then put the custom script in the correct location.</p>

Table 10. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_Path_WhileWIMMounted_Custom_Script	Specifies a custom script to run during the SGTkWinPE build process while the WinPE WIM image is mounted for editing. Use the Template.cmd script as a starting point for writing custom scripts, and then put the custom script in the correct location.
TK_Path_AfterWIMMounted_Custom_Script	Specifies a custom script to run during the SGTkWinPE build process after the WinPE WIM image is mounted for editing. Use the Template.cmd script as a starting point for writing custom scripts, and then put the custom script in the correct location.

Chapter 5. Downloading and installing System Enablement Packs (SEPs)

System Enablement Packs (SEPs) are collections of system-specific utilities, drivers, and scripts required to support a specific set of machine types. You can use SEPs to add support for systems that were released after the most current version of the Toolkit.

For ServerGuide Scripting Toolkit to support a specific machine type, you must download and install the corresponding SEP. After you have downloaded the SEP, you must install it on the Toolkit source server to be able to create deployments for the supported machine types.

Scripting Toolkit provides the **DownloadSEP s.cmd** command to download System Enablement Packs (SEPs) for use in creating deployments. The command file is located in the `sgdeploy\SGTKWinPE` directory.

Step 1. Run the **DownloadSEP s.cmd** command to download the SEP for the specified system. For example:

```
DownloadSEPs.cmd 7327,7328
```

Step 2. Install the System Enablement Packs (SEPs) after they have been downloaded using one of the following methods:

Table 11. Toolkit commands for downloading and installing SEPs

Command	Action
DownloadSEP s.cmd	Installs the SEPs that it has downloaded.
InstallSEP s.cmd	Installs the SEPs that are stored either in the default location, <code>sgshare\sgdeploy\updates\uxsp</code> , or in another location specified in the command line.
SGTKWinPE.cmd	Creates a deployment scenario that automatically prompts you to install any required SEPs.

Chapter 6. Supported target server hardware and software

ServerGuide Scripting Toolkit supports the deployment of Windows operating systems on LenovoServer and LenovoSeries servers. In general, ServerGuide Scripting Toolkit provides support for LenovoServerProven or third-party adapters in the following categories:

- Ethernet
- Fibre Channel
- IDE and IDE RAID
- SAS and SAS RAID
- SATA and SATA RAID
- SCSI and SCSI RAID (includes Ultra-SCSI)

This section contains the following information about specific hardware and software support for deployment scenarios:

- “System support” on page 53
- “Operating system support” on page 53
- “RAID controller support” on page 53
- “Fibre Channel HBA support” on page 54

Additional information about these topics is contained in the `readme.htm` file.

You can download the latest version of the `readme.htm` file from the ServerGuide Scripting Toolkit Web page. See Lenovo deployment resources on the Web for information.

System support

This section details what systems are supported by ServerGuide Scripting Toolkit.

You can use the Scripting Toolkit to deploy supported operating systems to any Lenovo System x, BladeCenter, iDataPlex server, or Flex system that is in the “” on page .

Operating system support

The Lenovo website provides information about the operating system and server combinations that are supported by ServerGuide Scripting Toolkit.

You can use the Scripting Toolkit to deploy supported operating systems. To determine what operating system and server combinations are supported, see Lenovo ServerProven.

RAID controller support

Lenovo provides detailed online information about what RAID controller and server combinations are supported by ServerGuide Scripting Toolkit.

To determine which RAID controller and server combinations are supported, refer to the following web pages:

- Storage Controllers
- Lenovo ServerProven

Fibre Channel HBA support

The Lenovo website provides detailed information what FC HBA and server combinations are supported by the ServerGuide Scripting Toolkit.

To determine what FC HBA and server combinations are supported, see Shared Storage Adapters and Lenovo ServerProven.

Chapter 7. ServerGuide Scripting Toolkit tools and utilities

This section describes the tools and utilities shipped with ServerGuide Scripting Toolkit.

- Each tool has a brief description and instructions for using it with ServerGuide Scripting Toolkit, and pointers to additional information on the tool and its use.
- Each utility includes a description and examples of the syntax and parameters.

SGTKWinPE.cmd

Use the **SGTKWinPE** script to build a Windows PE image and to run the scenario scripts provided by Scripting Toolkit.

The syntax for the **SGTKWinPE** script is:

```
SGTKWinpe filename [/Image | /ISO | /StartISO | /?]
```

SGTKWinPE uses these parameters and options.

Table 12. SGTKWinPE parameters and options

Parameter	Description
filename	The fully qualified path and file name for the SGTKWinPE settings file. A file name is required.
/Image	Creates the Windows PE image without creating the ISO files.
/ISO	Creates an ISO image of the Windows PE image.
/StartISO	Starts the registered software associated with the ISO files.
/?	Displays Help.

Tools included with ServerGuide Scripting Toolkit

ServerGuide Scripting Toolkit includes the following tools that make Toolkit more efficient:

- “Advanced Settings Utility” on page 55
- “UpdateXpress System Pack Installer” on page 56

Advanced Settings Utility

ServerGuide Scripting Toolkit includes Advanced Settings Utility (ASU), and uses a subset of the ASU functions to capture and deploy firmware settings as part of the scripted deployments.

You can use ASU to modify firmware settings from the command line on multiple operating-system platforms.

The following examples illustrate ASU syntax and usage.

Table 13. **ASU** syntax and usage

Command	Description
<code>asu.exe show bios</code>	<ul style="list-style-type: none"> Displays and captures CMOS settings. Use redirection to store this output in a file as shown here: <code>asu.exe show bios > bios_settings.ini</code>
<code>asu.exe replicate filename</code>	<ul style="list-style-type: none"> Applies CMOS settings from a file. ASU finds and reads the contents of the specified <i>filename</i>. If the contents are valid CMOS settings, they are applied, one line at a time, to the server. The following example applies the <code>bios_settings.ini</code> settings: <code>asu.exe replicate bios_settings.ini</code> <p>Note: Due to a difference in BIOS settings and valid values between models, only settings captured from an identical model can be replicated.</p>

Updating Advanced Settings Utility

This procedure describes how to update Advanced Settings Utility for ServerGuide Scripting Toolkit, Windows Edition.

You need the following files and resources to update the ASU executable file:

- The `lnvgy_utl_asu_asutversion_windows_i686.exe` file, where *version* is the updated version of ASU.
- The `lnvgy_utl_asu_asutversion_windows_x86-64.exe` file, where *version* is the updated version of ASU.
- A scripting toolkit source server.

Use the following procedure to update ASU when the ServerGuide Scripting Toolkit, Windows Edition requires a new version.

- Step 1. On the source server, navigate to `..\sgdeploy\SGTKWinPE\Bin\win32`.
- Step 2. Copy the `lnvgy_utl_asu_asutversion_windows_i686.exe` file to that directory.
- Step 3. Execute the file.
- Step 4. When prompted, enter **A** to overwrite all of the old files.
- Step 5. When the update is complete, navigate to `..\sgdeploy\SGTKWinPE\Bin\winx64`.
- Step 6. Copy the `lnvgy_utl_asu_asutversion_windows_x86-64.exe` file to that directory.
- Step 7. Delete the `asu.exe` file.
- Step 8. Execute `lnvgy_utl_asu_asutversion_windows_x86-64.exe`.
- Step 9. When prompted, enter **A** to overwrite all of the old files.
- Step 10. Rename `asu64.exe` to `asu.exe` in this directory.

UpdateXpress System Pack Installer

The Windows edition of ServerGuide Scripting Toolkit includes UpdateXpress System Pack Installer (UXSPi) to acquire updates for your deployment scenarios.

UXSPi can perform the following functions:

- Acquire firmware and driver updates for supported machine type and operating system combinations from a remote location, such as the IBM support website: .
- Inventory a system to be updated and compare the inventory to the list of available updates, and then recommend and deploy a set of updates for the system.
- Create bootable media on CD-ROM, DVD, or USB key to use in applying firmware to supported systems.

For more information about running UpdateXpress System Pack Installer (UXSPi), refer to the UXSPi user's guide in the `sgdeploy\SGTKWinPE\Docs\uxspi` directory.

praid.exe

`praid.exe` is a scriptable executable program that offers a single user interface for both configuring and replicating all RAID controllers supported by ServerGuide Scripting Toolkit.

PRAID works with both the 32- and 64-bit versions of the Windows Preinstallation Environment.

PRAID has three modes of operation:

- **Deploy mode** for scripted configuration of RAID controllers.
- **Capture mode** for replicating RAID controller settings.
- **Restore-defaults mode** for resetting RAID controllers to factory-default settings only.

Deploy mode

Used in Deploy mode, PRAID offers the following features:

- Configures all RAID controllers in a server with a single call to the program.
- Automatically resets all RAID controllers to factory-default settings before configuring.
- Uses customizable logic to decide which configuration (policy) is applied to a server based on system hardware. The logic can involve:
 - Machine type of the server
 - Serial number of the server
 - Number of drives connected to the RAID controller
 - RAID controller type
 - Controller number (order) of the RAID controller
- Can be highly customized for specific RAID configurations, or highly generalized to handle many different RAID configurations.
- Provides a default or AUTO mode for automatically creating arrays and logical drives by using default settings. This mode requires no knowledge of the number, size, or location of the drives connected to the RAID controllers.
- Automatically applies default values for any RAID configuration parameters that you do not supply. You supply only the parameters that you want to change.
- Default values for each configuration parameter are equivalent to the default settings of the ServeRAID Manager express configuration method, where applicable.
- Allows up to 50 policies for configuring RAID controllers to be specified in a single policies file.

Note: When you use PRAID in Deploy mode, you must use the `/r` parameter.

To delete RAID configuration on all controllers, specify `/r`. To delete RAID configuration on a specific controller, specify `/r#` where `#` is the controller number.

For example, `praid /f:policy.ini /r /y`.

Capture mode

Used in Capture mode, PRAID offers the following features:

- Captures the RAID configurations of all supported controllers to the policies file, a text file, with a common format.
- Captured RAID configurations can be immediately used with PRAID in Deploy mode to easily replicate the RAID configuration to many servers.
- Allows customizable logic when saving the captured parameters to determine when each captured configuration should be deployed.
- Saves useful information about each captured configuration, including the system machine type, date, and time when the configuration was captured.
- Allows you to edit any RAID configurations that you capture before deploying them to other systems.

Restore-defaults mode

Used in Restore-defaults mode, PRAID offers the following features:

- Deletes all arrays and logical drives on all RAID controllers.
- Sets other RAID controller settings back to factory defaults.

Environment requirements

RAID adapter information supported by PRAID is provided.

PRAID works by parsing the output of other RAID-configuration utilities. To accomplish this, the utilities must be in the system search path. Refer to Table 32 “Supported RAID adapter information” on page 93 for supported RAID adapter information.

PRAID usage

Each of the modes supported by PRAID requires a specific syntax, but they all share some common parameters.

The following table, Table 14 “PRAID parameters common to multiple modes” on page 58, lists these shared parameters.

Table 14. PRAID parameters common to multiple modes

Parameter	Description	Usage
<code>/r:n</code> Restore-defaults mode	<p>Restores the RAID controller with the controller number specified by <i>n</i> to factory-default settings and then returns immediately. No RAID configuration is performed if you use this parameter.</p> <p>If no value is specified for the controller number, all RAID controllers are reset to factory-default settings.</p> <p>Used alone, the parameter provides the Restore-defaults mode. You must use this parameter in conjunction with the Deploy mode parameters to reset controllers to the factory default settings before deploying a new configuration.</p>	<p><code>praid /r</code> Restores all controllers to factory-default settings.</p> <p><code>praid /r:3</code> Restores controller three to factory-default settings. No other controllers are affected.</p> <p><code>PRAID /f:policies.ini /r /v:5 /e1</code> Configures the RAID controllers in the system by using the policies file <code>policies.ini</code>, sets the verbose mode to maximum, and returns an error code if there were no matching policies for any controllers.</p>
<code>/f:policies_file</code>	The policy file name. This parameter is required for the Capture mode and	<p><code>praid /f:myfile.ini</code> Uses the policies file, <code>myfile.ini</code>, to configure all RAID controllers.</p>

Table 14. PRAID parameters common to multiple modes (continued)

Parameter	Description	Usage
Specifies the policy file	<p>Deploy mode unless the /d parameter is used.</p> <p>In Deploy mode, this parameter points to the policies that you want PRAID to use when configuring the RAID controllers. You cannot use this parameter with the /d parameter.</p> <p>In Capture mode, this parameter points to the file where you want the captured configurations to be written. If the file does not exist, PRAID creates it. If the file does exist, PRAID appends to the end of it.</p> <p>The /f parameter is valid in both Deploy and Capture modes.</p>	<p>praid /c /f:myfile.ini</p> <p>Captures the RAID configuration of all controllers to the policy file, myfile.ini.</p>
/y Suppresses prompting	<p>Suppresses the confirmation prompt. This parameter is optional.</p> <p>If you select the /y parameter, PRAID does not prompt you before resetting controllers to factory-default settings. PRAID always resets all controllers to factory-default settings before configuring them.</p> <p>If you do not supply this parameter, PRAID pauses to warn you before resetting the RAID controllers to factory-default settings.</p> <p>The /y parameter is valid in the Deploy and Restore-defaults modes.</p> <p>This parameter is optional.</p>	<p>praid /f:myfile.ini /y</p> <p>Uses the policies in myfile.ini to configure the RAID controllers and does not prompt before resetting all controllers to factory-default settings.</p>
/e2 Error code 2 if no supported controllers found	<p>Returns an error code of 2 if there were no supported RAID controllers found in the system.</p> <p>By default, PRAID does not return an error if no controllers are found in the system.</p> <p>This parameter is valid in all modes.</p> <p>This parameter is optional.</p>	<p>praid /c /f:c:\myfile.ini /e2</p> <p>Captures the RAID configuration of all RAID controllers to c:\myfile.ini, and returns an error if no controllers are found in the system.</p>

Table 14. PRAID parameters common to multiple modes (continued)

Parameter	Description	Usage
/e3 Error code 3 if no supported drives found	Returns an error code of 3 if at least one controller was found with no drives attached. By default, PRAID does not return an error if no drives are attached to a RAID controller. This parameter is valid in any mode. This parameter is optional.	praid /d /e3 Configures all RAID controllers with default settings and returns an error if one or more controllers has no drives attached.
/v:n Verbose level	Sets the verbosity level, where n is: <ul style="list-style-type: none"> • 0 - quiet • 3 - default • 5 - maximum This parameter is valid in any mode. This parameter is optional.to max'.	praid /d /v:5 Configures all RAID controllers with default settings and sets the verbose level

Deploy mode

The syntax for Deploy mode is:

PRAID.EXE /f:polices /r /d /p:path /e1 /e2 /e3
/v:n /y /b

The parameters unique to Deploy mode are described below.

Table 15. PRAID Deploy mode parameters

Parameter	Description	Usage
/d	Configure with defaults Configures all controllers in the system by using default settings instead of a policies file. The default settings used are the same as the default settings for the policies file. You cannot use this parameter with the /f parameter. See Table 20 “Default RAID levels” on page 73 for the default values that are assigned for each RAID controller based on the number of drives attached to the controller. This parameter is required unless the /f parameter is specified.	praid /d /r Configures all RAID controllers in the system by using default settings.
/e1	Error if no policy found Returns an error code of 1 if one or more controllers are not configured due to the fact that there was no policy found to configure them This parameter is optional.	praid /f:policy.ini /r /e1 Configures all RAID controllers using the policies file, policy.ini, and returns an error if no matching policy was found.

Capture mode

The syntax for Capture mode is:

```
PRAID.EXE /c[:p] /f:policies /e2 /e3 /v:n
```

The parameters unique to Capture mode are described below.

Table 16. Capture mode parameters

Parameter	Description	Usage
/c[:p]	<p>Capture mode Indicates capture mode. The :p portion is optional. If you do not include the optional portion, then :p assumes the default value: t,d.</p> <p>You can use :p to provide a list of parameters describing the AppliesTo that should be created when capturing the parameters to a policy. See “AppliesTo.n parameter” on page 66.</p> <p>:p is a list containing any of the following:</p> <ul style="list-style-type: none">• t Use the type of the RAID controller in the AppliesTo.1 entry for the policy.• c Use the controller number (scan order relative to all other RAID controllers in the system) in the AppliesTo.1 entry for the policy.• d Use the number of drives connected to the RAID controller in the AppliesTo.1 entry for the policy. <p>Note: You must specify the name of the policies file by using the /f parameter when using the /c parameter.</p> <p>If the file exists, the policy or policies created are appended to the end of the file. If the file does not exist, a new file is created. If there are multiple RAID controllers in the system, their configurations are placed in the file in scan order.</p>	<pre>praid /c:m,t /f:myfile.ini</pre> <p>Captures the configuration of all RAID controllers to the myfile.ini file by using the machine type of the server and the RAID controller type as the AppliesTo.1 entry.</p>

Restore-defaults mode

The syntax for Restore-defaults mode is:

```
PRAID /r:n /e2 /v:n /y
```

PRAID usage examples

This topic shows deploy mode, capture mode, and restore-defaults mode examples.

Deploy mode examples

```
PRAID /r /d /y
```

- Uses default settings to configure all RAID controllers in the system
- Does not prompt before setting controllers to factory-default settings
- When required, performs drive synchronization without prompting

The following example is useful for unattended scripted installations.

```
PRAID /f:policies.ini /r /v:5 /e1
```

- Uses the policies file, policies.ini, to configure the RAID controllers in the system
- Sets the verbose mode to maximum
- Returns an error code if there were no matching policies for one or more controllers

Capture mode examples

```
PRAID /c /f:c:\mydata\policies.ini
```

Captures the configuration of all RAID controllers into the file: C:\mydata\policies.ini.

```
PRAID /c:m,t /f:policies.ini
```

- Captures the configuration of all RAID controllers into the file policies.ini
- Uses the system machine type and RAID controller type as the AppliesTo.1 entry in the policies file for each captured configuration

Restore-defaults mode examples

```
PRAID /r /v:0 /y
```

- Restores all RAID controllers to factory-default settings.
- Operates in silent mode; no messages are printed to the screen.
- Does not prompt the user before restoring factory-default settings.

PRAID return codes

The following list provides the PRAID return codes.

- 0 - Success.
- 1 - Execution was successful, but the /e1 parameter was supplied, and at least one controller was not configured because there was no matching policy.
- 2 - Execution was successful, but the /e2 parameter was supplied, and no controllers were found in the system.
- 3 - Execution was successful, but the /e3 parameter was supplied, and at least one controller was not configured because no drives were attached.
- 4 - A syntax error occurred on the command line.
- 5 - Either a syntax error occurred in the policies file or the policies file could not be opened.
- 6 - Reserved.
- 7 - An error occurred while resetting a controller to the default settings.
- 8 - An error occurred while gathering information about a controller.
- 9 - An error occurred in the policy file.
- 10 - An error occurred during processing.
- 11 - An error occurred during deployment.

Policies file

When used in Configure mode, the policies file directs how PRAID configures the RAID controllers in a system by using keywords and values that can be customized by the user. In Capture mode, PRAID creates or appends to the end of a policies file the parameters that can configure other RAID controllers identically to the ones in the current system.

To create a policies file, you can use any of the following methods:

- Run PRAID in Capture mode to create a policies file from a RAID controller that is already configured.
- Use one of the example policies files provided with ServerGuide Scripting Toolkit, and customize it to configure your RAID controllers.
- Use an ASCII text editor to create a new policies file.

The policies file is an ASCII text file that is organized in an INI file format. Each section in the policies file represents a single policy for configuring RAID controllers. Each section name in the INI file indicates the start of a new policy. You can have up to 50 policies in a single policies file.

The policies file must contain one or more uniquely-named sections that use the format [Policy.name], where name is a unique user-assigned name that identifies the policy. Name can be any combination of letters, numbers, underscores, periods, or dashes.

Some examples of valid section names are: [Policy.1], [Policy.mypolicy], and [Policy.My-RAID5-config].

How PRAID selects a policy

Each section in the policies file represents a single policy for configuring the RAID controllers. In Configure mode, each RAID controller is configured by using a single policy, but a single policy can be used to configure multiple controllers.

Each policy in a policies file contains one or more AppliesTo.n entries, where n is the number of the **AppliesTo** parameter within the policy. This entry is required in each section, so every section must contain at least an AppliesTo.1 entry. See “Policies file parameters” on page 63 for a full description of the AppliesTo.n entry.

These entries are followed by a list of hardware parameters, including machine type, number of drives connected to the RAID controller, and scan order, which are evaluated against the current system hardware. If all of the hardware parameters of an AppliesTo.n entry match the hardware being evaluated, this policy is used to configure the hardware.

For each policy in the policies file, the AppliesTo.n entries for that policy are evaluated in order starting with AppliesTo.1. If none of the AppliesTo.n entries match the current hardware, then the policy is not applied and the AppliesTo.n entries in the next policy are evaluated. This continues until either a match is found or no more policies exist in the file. If the end of the file is reached without a match, then the controller is not configured. Because the policies are evaluated in order, you should place more specific policies at the beginning of the policies file.

Policies file parameters

This section describes the parameters used in the policies file.

The Policy.name header and AppliesTo.1 entry are the only parameters required. None of the values are case-sensitive.

If you do not specify a value for any of the other parameters, they will be assigned a default value when applicable. If a parameter is not valid for a RAID controller, it will be ignored.

In addition to the examples in this reference guide, ServerGuide Scripting Toolkit also provides two example policies files that you can modify for your own use.

Table 17. Policy files

File name	Description
RAID1-5.ini	<ul style="list-style-type: none"> • Creates a RAID-1 array using the first two drives, and a RAID-5 array using the remaining drives. • Valid for: <ul style="list-style-type: none"> – ServeRAID-4H – 4Mx – 4Lx – 5i – 6i, – 6M, – 6i+, – 7k – 8i
RAID5HSP.ini	<ul style="list-style-type: none"> • Creates a single RAID-5 array with a single hot-spare drive by using all available drives. • Valid for: <ul style="list-style-type: none"> – ServeRAID-4H – 4Mx – 4Lx – 5i – 6i – 6i+ – 6M – 7k – 7t – 8i
template.ini	Provides a policies file template that contains all parameters with details about each one.
syntax.txt	Provides a syntax specification for the policies file.

The following tables lists the policy file parameters.

Table 18. Policy file parameters

Keyword	Required?	Default	Description
Policy.name	Yes	None	This header designates the start of a new policy. See “Policy.name parameter” on page 66 for additional information.
AppliesTo.n	Yes	None	Use this parameter to describe when the current policy should be chosen to configure the RAID controllers. See “AppliesTo.n parameter” on page 66 for additional information.

Table 18. Policy file parameters (continued)

Keyword	Required?	Default	Description
ReadAhead	No	<ul style="list-style-type: none"> ADAPTIVE (for ServeRAID 4H, 4MX, 4Lx, 5i, 6i, 6i+, 6M, and 7k) 0N (for ServeRAID-7t and 8i, 8k, and 8k-l) 	Specifies the read ahead setting that should be applied to the RAID controller. See “ReadAhead parameter” on page 66 for additional information.
RebuildRate	No	HIGH	Specifies the rebuild rate that should be applied to the RAID controller. See “RebuildRate parameter” on page 67 for additional information.
StripeSize	No	<ul style="list-style-type: none"> 8 (for ServeRAID 4H, 4Mx, 4Lx, 5i, 6i, 6i+, 6M, and 7k) 64 (for ServeRAID-7t, 8i, 8k, 8k-l, 7e-SCSI, 7e-SATA, 8e-SATA, 8e-SAS, and LIS-IDEal-RAID) 	Specifies the stripe-unit size in KB that the controller should use for its arrays. See “StripeSize parameter” on page 67 for additional information.
Array_Mode	No	AUTO	Defines the array-creation policy to use when selecting physical disk drives to include in an array. See “Array_Mode parameter” on page 67 for additional information.
Array_Defaults	No	<ul style="list-style-type: none"> 0%:1 for ServeRAID-8e-SATA and 8e-SAS, LSI-SCSI-RAID when at least 3 drives are available 0%:1 for ServeRAID-4H, 4Mx, 4Lx, 5i, 6i, 6i+, 6M, and 7k, when one or more arrays has 4 or more physical drives 0%:0 for all other cases 	Defines the default values to use for the variance and number of hot-spare drives when AUTO is specified for Array_Mode. See “Array_Defaults parameter” on page 68 for additional information.
Array.letter	No	None	Lets you specify how many arrays are created and the physical drives that you would like in each array. See “Array.letter parameter” on page 68 for additional information.
Hotspares	No	None	Defines a list of specific physical drives to designate as hot-spare drives. See “Hotspares parameter” on page 69 for additional information.
Logical_Mode	No	AUTO	Defines the logical-drive creation policy to use when creating logical drives. See “Logical_Mode parameter” on page 69 for additional information.

Table 18. Policy file parameters (continued)

Keyword	Required?	Default	Description
Logical_Defaults	No	FILL:AUTO:AUTO	Defines the default logical drive settings that should be used when creating logical drives. See “Logical_Defaults parameter” on page 69 for additional information.
Logical.num	No	None	Lets you specify how many logical drives are created and the specific parameters for each logical drive. See “Logical.num parameter” on page 70 for additional information.

***Policy.name* parameter**

The *policy.name* header designates the start of a new policy.

Description

You can specify *name* by using any combination of letters, numbers, underscores, periods, or dashes. There is no maximum length for *name*, but the maximum length for a single line in the policies file is 256 characters. You can have up to 50 policies in a single policies file.

[Policy.RAID-5-Hotspare]

***AppliesTo.n* parameter**

This parameter is used to describe when the current policy should be chosen to configure the RAID controllers.

Description

You can define up to 20 *AppliesTo.n* entries per policy. You must have an *AppliesTo.1* entry for each policy, and *AppliesTo.n* is the only required parameter of a policy.

AppliesTo.n includes a comma delimited list that contains one or more of the following parameters:

- *m:mtype*, where *mtype* is the four digit machine type of a LenovoServer or xSeries server.
- *s:serial*, where *serial* is the serial number of a LenovoServer or xSeries server.
- *c:contn*, where *contn* is the controller number (scan order) of the RAID controller with respect to all other RAID controllers in the system. The number assigned to a particular controller is dependent on the controller's physical PCI slot and the order in which your system scans its PCI slots.
- *t:ctype*, where *ctype* is the type of the controller. The type is not case-sensitive and must be one of the controller types listed in the table of RAID adapters supported by PRAID.
- *d:drives*, where *drives* is an integer value specifying the number of drives connected to the controller. Only drives in a Ready state after the controller is reset to factory-default settings are counted.
- **ALL** Indicates that the policy should be used for all RAID controllers. This parameter is useful if you declare a default policy that is not covered by any of the other policies.

Example using the *m,s,c,t*, and *d* parameters:

AppliesTo.1 = *m:8865,t:ServeRAID-7k*

AppliesTo.2 = *c:1,d:15,s:87R478U*

Example using the **ALL** parameter:

AppliesTo.1 = **ALL**

***ReadAhead* parameter**

The **ReadAhead** parameter specifies the read ahead setting that should be applied to the RAID controller.

Description

If this parameter is not applicable for a RAID controller, it is ignored. See “Supported settings for RAID controllers” on page 70 for the list of ReadAhead settings supported by PRAID for each RAID controller. Possible settings are:

- Adaptive
- On
- Off

ReadAhead = On

RebuildRate parameter

The **RebuildRate** parameter specifies the rebuild rate that should be applied to the RAID controller.

Description

If this parameter is not applicable for a RAID controller, then it is ignored. See “Supported settings for RAID controllers” on page 70 for the list of **RebuildRate** parameter settings supported by PRAID for each RAID controller.

- High
- Medium
- Low

RebuildRate = High

StripeSize parameter

The **StripeSize** parameter specifies the stripe-unit size in KB that the controller should use for its arrays.

Description

If this parameter is not applicable for a RAID controller, then it is ignored. See “Supported settings for RAID controllers” on page 70 for the list of **StripeSize** parameter settings supported by PRAID for each RAID controller. Possible values are any stripe size supported by the controller.

StripeSize = 32

Array_Mode parameter

The **Array_Mode** parameter defines the array-creation policy to use when selecting physical disk drives to include in an array.

Description

Possible values for the **Array_Mode** parameter are as follows:

Auto

Creates arrays using drives that have the same size in MB. This is the default. Each set of drives with same size will be combined into a single array. The maximum number of drives allowed per array is determined by the limits of the RAID controller. Only drives in a Ready state after resetting the controller to factory-default settings are used in arrays. Hot-spare drives are created based on the rules supplied with the **Array_Defaults** parameter.

The **Array_Defaults** parameter allows you to modify the default behavior of the AUTO mode for arrays.

Custom

Allows you to specify the exact physical disk drives to use in the array. If you specify this value, you must also specify the **Array.letter** parameter with a list of drives for each array that you want to create. If you want hot-spare drives to be created, you must use the **Hotspares** parameter to list the hot-spare drives.

```
Array_mode = CUSTOM
```

Array_Defaults parameter

The **Array_Defaults** parameter defines the default values to use for the variance and number of hot-spare drives when AUTO is specified for Array_Mode.

Description

The **Array_Defaults** parameter is not valid if Array_Mode is set to CUSTOM.

The value of **Array_Defaults** is expressed in the format: variance:hotspares, where:

variance specifies the percentage variance to use when selecting drives to add to the array. This parameter is useful when you are using drives that may vary slightly in size. Variance is based on a percentage of the drive's size in MB. Valid values are:

- 0% Only drives with an equal size in MB are combined into a single array.
- 5% All drives within 5 percent of the same size in MB are combined into a single array.
- 10% All drives within 10 percent of the same size in MB are combined into a single array.
- 100% All drives, regardless of size in MB, are combined into a single array.

and

hotspares is an integer that specifies the total number of hot-spare drives to create. The largest drives are chosen as hot-spare drives first. If not enough drives are available to create hot-spare drives, then PRAID does not create any hot-spare drives.

```
Array_Defaults = 5%:1
```

Array.letter parameter

The *Array.letter* parameter specifies how many arrays are created and the exact physical drives that you want in each array.

Description

You can use any of the following methods to specify the physical drives:

- The channel number and SCSI ID (for SCSI) or bus number and target ID (for SATA/SAS) of each drive. The channel number or bus number is always 1-based. The SCSI ID or target ID is always 0-based.
- A list of integer values indicating that the nth drive should be included in the array.
- The keyword ALL to indicate that all remaining drives attached to the controller that are not specified in previous arrays should be included in the current array.

The first array must be labeled Array.A. Additional arrays are labeled sequentially, for example, Array.B, Array.C, and so on. The maximum number of arrays allowed per controller is determined by the limits of the specific RAID controller.

Example using channel number and SCSI ID:

```
Array.A = 1:1,1:2
```

```
Array.B = 1:3,1:4,1:5,2:1,2:2,2:3,2:4,2:5,2:6
```

```
Array.C = ALL
```

Example using integer values:

Array.A = 1,2,3
Array.B = ALL

Hotspares parameter

The **Hotspares** parameter defines a list of specific physical drives to designate as hot-spare drives.

Description

You can use the following methods to specify the physical drives:

- The channel number and SCSI ID (for SCSI) or bus number and target ID (for SATA/SAS) of each drive. The channel number or bus number is always 1-based. The SCSI ID or target ID is always 0-based.
- A list of integer values indicating that the nth drive should be included in the array.
- The keyword ALL to indicate that all remaining drives attached to the controller that are not specified in previous arrays should be included in the current array.

Example using channel number and SCSI ID:

```
Hotspares = 1:12,2:14
```

Example using integer value:

```
Hotspares = 12, 13
```

Logical_Mode parameter

The **Logical_Mode** parameter defines the logical-drive creation policy to use when creating logical drives.

Description

Possible values for this parameter are:

AUTO

Indicates that defaults should be used for all parameters. Default parameters are:

- One logical drive is created on each array, using all available space.
- The RAID level is set by using the AUTO (default) scheme.
- Write-cache mode is set by using the default value for the controller.

You can adjust these default values by using the `Logical_Defaults` parameter.

CUSTOM

Indicates that you want to specify all of the parameters for each logical drive that is created. If you specify CUSTOM, then you must specify the parameters for each logical drive by using the `Logical.num` parameter.

```
Logical_Mode = CUSTOM
```

Logical_Defaults parameter

The **Logical_Defaults** parameter defines the default logical drive settings that should be used when creating logical drives.

Description

This parameter is only valid when AUTO is specified for `Logical_Mode`. Values for this parameter are expressed in the format: `size:raidlevel:writemode`, where:

Size specifies the size of each logical drive. One logical drive of the specified size is created on each array. Size can be in any of the following formats:

- A positive integer specifies the size in MB.

- A percentage specifies that a percentage of the total space should be used.
- FILL indicates that all available space on the array should be used.

Raidlevel specifies the RAID level for the logical drive. See “Supported settings for RAID controllers” on page 70 for the list of RAID level settings supported by PRAID for each controller.

Writecmode is an optional parameter that specifies the write-cache mode for each logical drive. If the write-cache mode cannot be set for a specific configuration, then this parameter is ignored. See “Supported settings for RAID controllers” on page 70 for the list of write_cache mode settings supported by PRAID for each RAID controller.

Valid values are:

- ON
- OFF
- AUTO uses the default write-cache mode for the controller. (Recommended for most users.) AUTO is the default value if writecmode is not specified.

Logical_Defaults = 50%:5EE:AUTO

Logical.num parameter

The *Logical.num* parameter specifies the number of logical drives to create and the parameters for each logical drive.

Description

You can set the array letter where the logical drive is located, logical drive size, RAID level, and write-caching mode for each logical drive. The first logical drive must be labeled *Logical.1*. Additional logical drives are numbered *Logical.2*, *Logical.3*, and so on. You must specify at least one logical drive for each array. The maximum number of drives allowed per array and the maximum total number of logical drives allowed is determined by the specific RAID controller.

Values for this parameter are expressed in the format: *array:size:raidlevel:writecmode* where *array* specifies the array letter, and *size*, *raidlevel*, and *writecmode* are as described in “*Logical_Defaults* parameter” on page 69.

```
Logical.1 = A:50%:0
Logical.2 = A:50%:5EE
Logical.3 = B:FILL:1:ON
Logical.4 = C:4096:AUTO:AUTO
```

Supported settings for RAID controllers

Supported settings for each RAID controller when using PRAID are listed with details about the read policy, write policy, RAID levels, and stripe size.

Table 19 “Supported settings for each RAID controller when using PRAID” on page 70 lists the supported settings for each RAID controller when using PRAID.

In some cases, when using PRAID, the list of supported settings might differ from the supported settings of the RAID controller. These known cases are indicated in the following table.

Table 19. Supported settings for each RAID controller when using PRAID

Bold settings are defaults.

Table 19. Supported settings for each RAID controller when using PRAID (continued)

RAID adapters	Read policy	Write policy	RAID Levels ¹	Stripe Size (KB)
ServeRAID-B5015	<ul style="list-style-type: none"> • ON • OFF 	[n/a]	R1, R5	4, 8, 16, 32, 64, 128 , 256, 512, 1024
LSI-IDEal-RAID	[n/a]	[n/a]	R0, R1	32, 64 , 128, 256, 512, 1024, 2048, 4096
LSI-MegaRAID-8480	[n/a]	[n/a]	R0, R1, R10, R5, R50	4, 8, 16, 32, 64 , 128
LSI-SAS-1078-IR	[n/a]	[n/a]	R0, R1	[n/a]
LSI-SAS-RAID	[n/a]	[n/a]	R0, R1, R1E	[n/a]
LSI-SCSI-RAID	[n/a]	[n/a]	R1	[n/a]
ServeRAID-7t	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	RVOLUME, R0, R1, R10, R5	16, 32, 64
ServeRAID-8i	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	RVOLUME, R0, R1, R10, R1E, R5, R50, R5EE, R6, R60	16, 32, 64, 128, 256 , 512, 1024
ServeRAID-8k	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	RVOLUME, R0, R1, R10, R1E, R5, R6	16, 32, 64, 128, 256 , 512, 1024
ServeRAID-8k-l	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	RVOLUME, R0, R1, R10	16, 32, 64, 128, 256 , 512, 1024
ServeRAID-8s	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	RVOLUME, R0, R1, R10, R1E, R5, R50, R6	16, 32, 64, 128, 256 , 512, 1024
ServeRAID-BR10ie	[n/a]	[n/a]	R0, R1, R1E	[n/a]
ServeRAID-BR10il	[n/a]	[n/a]	R0, R1, R1E	[n/a]
ServeRAID-M1015	[n/a]	[n/a]	R0, R1, R10	8, 16, 32, 64
ServeRAID-M1015R5	[n/a]	[n/a]	R0, R1, R10, R5, R50	8, 16, 32, 64
ServeRAID-M1xxx	[n/a]	[n/a]	R0, R1, R10	8, 16, 32, 64
ServeRAID-M1xxx_R5	[n/a]	[n/a]	R0, R1, R10, R5, R50	8, 16, 32, 64
ServeRAID-M5014	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50	8, 16, 32, 64 , 128
ServeRAID-M5014-R6-R60	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64 , 128
ServeRAID-M5015	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50	8, 16, 32, 64 , 128

Table 19. Supported settings for each RAID controller when using PRAID (continued)

RAID adapters	Read policy	Write policy	RAID Levels ¹	Stripe Size (KB)
ServeRAID-M5015-R6-R60	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64 , 128
ServeRAID-M5025	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50	8, 16, 32, 64 , 128, 256, 512, 1024
ServeRAID-M5025-R6-R60	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64 , 128, 256, 512, 1024
ServeRAID-M5xxx	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64 , 128, 256, 512, 1024
ServeRAID-M51xx	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10	8, 16, 32, 64 , 128, 256, 512, 1024
ServeRAID-M51xx_R5	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50	8, 16, 32, 64 , 128, 256, 512, 1024
ServeRAID-M51xx_R6	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R6, R60	8, 16, 32, 64 , 128, 256, 512, 1024
ServeRAID-M51xx_R5_R6	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64 , 128, 256, 512, 1024
ServeRAID-M5210	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0,R1,R10	8,16,32, 64
ServeRAID-M5210-R5	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50, R6, R60	8,16,32, 64
ServeRAID-M5215	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50	64, 128, 256, 512, 1024
ServeRAID-M1215	<ul style="list-style-type: none"> • ON • OFF • AUTO • Write 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10	64
ServeRAID-M1215-R5	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50	64

Table 19. Supported settings for each RAID controller when using PRAID (continued)

RAID adapters	Read policy	Write policy	RAID Levels ¹	Stripe Size (KB)
ServeRAID-M1210e	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10	64
ServeRAID-MR10i	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64 , 128
ServeRAID-MR10ii	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64 , 128
ServeRAID-MR10is	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64 , 128, 256, 512, 1024
ServeRAID-MR10k	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50, R6, R60	16, 32, 64 , 128, 256, 512, 1024
ServeRAID-MR10M	<ul style="list-style-type: none"> • ON • OFF • AUTO 	<ul style="list-style-type: none"> • ON • OFF 	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64 , 128
ServeRAID-C100	[n/a]	[n/a]	R0, R1, R10	64
ServeRAID-C100-R5	[n/a]	[n/a]	R0, R1, R10, R5	64
ServeRAID-C105	[n/a]	[n/a]	R0, R1, R10	64
SAS2004	[n/a]	[n/a]	R0, R1, R10, R1E	[n/a]

- RAID levels 5E and 5EE support only one logical drive per array.
- Default RAID levels are described in “Default RAID levels” on page 73.

Default RAID levels

The default RAID level that is applied to a logical drive depends on the number of drives in the array and the controller type. These default values are designed to match the default values of the express configuration method in ServeRAID Manager where applicable.

The following table shows the default RAID values that PRAID uses when AUTO is specified for raidlevel.

Table 20. Default RAID levels

Controller	Drives in array				
	1	2	3	4	5 or more
ServeRAID-B5015	[n/a]	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare
LSI-IDEal-RAID	[n/a]	RAID 1	[n/a]	[n/a]	[n/a]
LSI-MegaRAID-8480	RAID 0	RAID 0	RAID 0	RAID 0+Hotspare	RAID 0+Hotspare
LSI-SAS-1078-IR	[n/a]	RAID 1	RAID 1+Hotspare	RAID 1+Hotspare	RAID 1+Hotspare

Table 20. Default RAID levels (continued)

Controller	Drives in array				
	1	2	3	4	5 or more
LSI-SAS-RAID	[n/a]	RAID 1	RAID 1E+Hotspare	RAID 1E+Hotspare	RAID 1E+Hotspare
LSI-SCSI-RAID	[n/a]	RAID 1	RAID 1+Hotspare	RAID 1+Hotspare	RAID 1+Hotspare
ServeRAID-7t	RAID 0	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare
ServeRAID-8i	VOLUME	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare
ServeRAID-8k	VOLUME	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare
ServeRAID-8k-l	VOLUME	RAID 1	RAID 1+Hotspare	RAID 10	RAID 10+Hotspare
ServeRAID-8s	VOLUME	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare
ServeRAID-BR10ie	[n/a]	RAID 1	RAID 1E	RAID 1E + Hotspare	RAID 1E + Hotspare
ServeRAID-BR10il	[n/a]	RAID 1	RAID 1E	RAID 1E + Hotspare	RAID 1E + Hotspare
ServeRAID-M1015	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1015R5	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1xxx	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1xxx_R5	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5014	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5014-R6-R60	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5015	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5015-R6-R60	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5025	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5025-R6-R60	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5xxx	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M51xx	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M51xx_R5	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M51xx_R6	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M51xx_R5_R6	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5210	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5210-R5	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5215	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1215	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1215-R5	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1210e	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-MR10i	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-MR10il	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-MR10is	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-MR10k	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-MR10M	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare

Table 20. Default RAID levels (continued)

Controller	Drives in array				
	1	2	3	4	5 or more
ServeRAID-C100	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-C100-R5	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-C105	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
SAS2004	[n/a]	RAID 0	RAID 0	RAID 10	RAID 10

ServerGuide Scripting Toolkit utilities

This section describes utilities that come with ServerGuide Scripting Toolkit.

Conventions

The command-line syntax examples in this documentation use the following conventions:

- Variables are shown in *italics*.
- Required parameters are shown within angle <> brackets.
- Optional parameters are shown within square [] brackets.
- Required or optional parameters from which you must make a unique choice are separated by a vertical bar (|).

Note:

Although the examples in this documentation are sometimes shown on multiple lines due to formatting constraints, you must enter all parameters for a utility on a single command line.

clini.exe

The **clini** utility is a command line INI utility that writes information to and reads information from an INI file, and merges information from one INI file with another.

The **clini** utility performs the following functions:

- Writes information to an INI file:
 - Adds new sections, items, or values
 - Removes sections, items, or values
 - Changes existing sections, items, or values
 - Changes or appends to values of existing sections or items
 - Comments or uncomments sections, items, or values
- Reads information from an INI file:
 - Reads items and store all or part of the value as an environment variable
 - Reads items and check all or part of the value for strings, substrings, or tokens
- Merges information from one INI to another.

Two versions of the utility come with ServerGuide Scripting Toolkit:

- A 32-bit version for Windows 32-bit operating systems and for the Windows Preinstallation Environment (Windows PE) 2.1/3.0 (32-bit)
- A 64-bit version for Windows x64 operating systems and for Windows PE 2.1/3.0 (x64).

Storing a value as an environment variable is done by creating a batch file that contains a command to set the environment variable. You must then call the batch file to set the environment variable. By default, the batch file is named `cliniset.bat`. If the batch file already exists, it is deleted and recreated with the new information.

Note: Because the media is read-only, this feature cannot be used on a bootable Windows PE CD or DVD.

In addition to setting values, the **clini** utility can append values to existing items in an INI file. By default, no delimiter is used to append values. A delimiter can be specified, if required. Appending values provides the ability to build values in the INI file by issuing multiple commands. When reading values from an INI file to set an environment variable, the values can be tokenized to specify a particular token.

The **clini** utility checks the number of characters on the command line and displays a message if the characters exceed the limit. The **/O** parameter overrides character-limit checking.

The **clini** utility has the following command-line syntax:

```
clini <filename> <[filename2 [/ES] [/A|U|P]]> <[/S:section]
[/I:item] [/V:value|/A:value|/U:value|/E:variable]
[/=:string|/C:string|/CT:string]> [/B:file_name]
[/D:delimiter] [/T:n] [/R] <[/CMT|/UCMT] [/AI]
[/CC:character]> [/NS] [/N] [/O]
```

The **clini** utility uses these parameters:

Table 21. *clini* parameters

Parameter	Description
<i>filename</i>	Defines the fully qualified path to the INI file to process.
<i>filename2</i>	<ul style="list-style-type: none"> Defines the fully qualified path to an INI file to merge information into from <i>filename</i>. All values in <i>filename</i> are copied into <i>filename2</i>, replacing the value of any preexisting items in <i>filename2</i>.
/ES	Specifies to merge only the items or values in the empty section.
/A	<ul style="list-style-type: none"> Specifies to append values from items in <i>filename</i> to the items in <i>filename2</i> instead of replacing them. An optional delimiter can be specified using the <i>/D:delimiter</i> parameter.
/U	<ul style="list-style-type: none"> Specifies to uniquely append values from items in <i>filename</i> to the items in <i>filename2</i> instead of replacing them; only if the value doesn't already exist. An optional delimiter can be specified using the <i>/D:delimiter</i> parameter.
/P	<ul style="list-style-type: none"> Specifies that the data in <i>filename2</i> is persistent. If duplicate items are found, they are not replaced.
/S:section	Specifies the name of the section within the INI file to write or to read.
/I:item	Specifies the name of the item within the INI file to write or to read.
/V:value	Specifies the value to write to the INI file.
/A:value	<ul style="list-style-type: none"> Specifies the value to append to the existing item in the INI file. The /I parameter is required to use the <i>/A:value</i> parameter.
/U:value	<ul style="list-style-type: none"> Specifies a unique value to append to the existing item in the INI file, only if this value does not already exist for the item. The /I parameter is required to use the <i>/U:value</i> parameter.

Table 21. *clini* parameters (continued)

Parameter	Description
<i>/E</i>	<ul style="list-style-type: none"> Convert multiple Items to Environment Variables. The Item name is used for the environment variable name. Use the /NS parameter to replace any spaces in the item names with underscore characters when creating the Environment Variables, if spaces are not desired.
<i>/E:variable</i>	<ul style="list-style-type: none"> Specifies the environment variable used to store the value of the item from the INI file. The /I parameter is required to use the <i>/E:variable</i> parameter. If the item specified by the /I parameter does not exist, or the section specified by the /S parameter does not exist, the environment variable has no value in the batch file created by the clini utility. If the environment variable exists on the system, it is deleted when the batch file runs.
<i>/=:string</i>	Verifies that the value of the item is equal to <i>string</i> , returning a value of 0 if true and 100 if false.
<i>/C:string</i>	Verifies that value of the item has <i>string</i> as a substring, returning a value of 0 if true and 100 if false.
<i>/CT:string</i>	<ul style="list-style-type: none"> Verifies that the value of the item has <i>string</i> as one of the tokens, returning a value of 0 if true and 100 if false. The default delimiter is a comma unless the /D:delimiter option is specified.
<i>/B:filename</i>	<ul style="list-style-type: none"> Defines the fully qualified path and file name of the batch file to create for setting the environment variable. The default is <code>CLIniSet.bat</code> if no file name is specified for this parameter. This parameter is only valid when the /E parameter is used.
<i>/D:delimiter</i>	<ul style="list-style-type: none"> Specifies a delimiter to use when appending values to an item in an INI file or reading tokens from an INI file. This parameter is not valid if the /V parameter is used. The /D parameter is valid only with the /A, /U, or /E parameters. Using the /D parameter without one of these three parameters results in a syntax error.
<i>/T:n</i>	<ul style="list-style-type: none"> Specifies the token in a delimited value to set as the specified environment variable, where <i>n</i> is a positive integer. The default delimiter is a comma unless otherwise specified with the /D parameter. This parameter is only valid with the /E parameter.
<i>/R</i>	Removes the specified section, item, or value from the INI file. Removing the last item in a section also removes the section.
<i>/CMT</i>	<ul style="list-style-type: none"> Specifies to comment out the line indicated by the Section, Item, or Value parameter, if it exists in the INI file. It also allows use of the /AI parameter.
<i>/UCMT</i>	<ul style="list-style-type: none"> Specifies to Uncomment the line indicated by the Section, Item, or Value parameter, if it exists in the INI file. It also allows use of the /AI parameter.
<i>/CC:character</i>	<ul style="list-style-type: none"> Specifies the comment character to use when commenting or uncommenting lines. If omitted, the default comment character is the semicolon. This parameter is only valid with the /CMT or /UCMT parameters.

Table 21. *clini* parameters (continued)

Parameter	Description
/AI	<ul style="list-style-type: none"> Specifies to explicitly treat the /V parameter as the value to all items when commenting or uncommenting. This parameter is only valid when using the /CMT or /UCMT parameters.
/N	Deletes an existing INI file and creates a new INI file. This parameter is not valid with the /E parameter.
/NS	Omits spaces around "=" when writing items into INI files. By default, the clini utility concatenates spaces around "=" when writing items.
/O	<ul style="list-style-type: none"> Overrides the command-line character count. The number of characters on the command line is automatically determined by this utility. An error message is displayed when the character limit is reached, unless you override this feature. The Windows command line is limited to 8189 characters.

The **clini** utility returns the following values to indicate status:

- 0 - Success or true.
- 1 - Syntax error.
- 2 - Program error.
- 3 - Destination is read-only.
- 4 - Current working directory is read-only.
- 5 - File not found.
- 100 - False.

The following examples illustrate **clini** utility usage.

Table 22. *clini* examples

Example	Description
<code>clini info.ini /S:Hardware /I:Machine Type /V:8549 /N</code>	Deletes any existing info.ini file and creates a new INI file named info.ini with a section called Hardware that contains one item, Machine Type, which has a value of "8549."
<code>clini info.ini /S:Hardware /I:Machine Name /V:Server1</code>	Adds the item Machine Name with a value of Server1 to the existing Hardware section of the info.ini file
<code>clini info.ini /S:Hardware /I:Machine Type /E:MachineType</code> <code>call CLIniSet.bat</code>	Reads the Machine Type value from the info.ini file, and stores it as an environment variable called <i>MachineType</i> .
<code>clini info.ini /S:Hardware /I:Machine Type2 /V:%MachineType%</code>	Writes the value of the environment variable <i>MachineType</i> to the INI file named info.ini, using section Hardware and item Machine Type2.
<code>clini info.ini /S:Hardware /I:Machine Type2 /E:MachineType2 /B:d:\EnvSet1.bat</code> <code>call d:\EnvSet1.bat</code>	Reads the machine type value from the info.ini file and stores it as an environment variable called <i>MachineType2</i> using a custom path and name for the batch file created to set the environment variable.

Table 22. *clini* examples (continued)

Example	Description
<pre>clini info.ini /S:MySection /E Call cliniset.bat</pre>	<p>This example creates environment variables for all the items found in section MySection.</p>
<pre>clini info.ini /AI /E /B:setthem.bat Call setthem.bat</pre>	<p>This example creates environment variables for all the items found in any section of the info.ini file and uses an alternate name for the CLIniSet.bat file.</p>
<pre>clini info.ini /S:MySection /I:MyItem /E Call cliniset.bat</pre>	<p>This example creates an environment variable called <i>My_Item</i> if it exists in the info.ini file.</p>
<pre>clini info.ini /S:MySection /I:My Item /E /NS Call cliniset.bat</pre>	<p>This example creates an environment variable called <i>My_Item</i> (converts the space to an underscore for the environment variable name) if the item exists in the info.ini file.</p>
<p>After running the first five examples above, in sequence, the info.ini file contains the following information:</p> <pre>[Hardware] Machine Type = 8549 Machine Type2 = 8549 Machine Name = Server1</pre> <p>Also, two new environment variables are created, as shown below:</p> <pre>MachineType = 8549 MachineType2 = 8549</pre>	
<pre>clini info.ini /S:User /I:Name /V:Toolkit /N clini info.ini /S:User /I:Name /A: User or clini info.ini /S:User /I:Name /V:Toolkit /N clini info.ini /S:User /I:Name /A:User /D:" "</pre>	<p>Creates a new file named info.ini with a section called User and one item called Name, which is set equal to "Toolkit User". The resulting info.ini file contains:</p> <pre>[User] Name = Toolkit User</pre>
<pre>clini info.ini /S:Section /I:Item /A:Value1 /D:, /N clini info.ini /S:Section /I:Item /A:Value2 /D:, clini info.ini /S:Section /I:Item /A:Value3 /D:, clini info.ini /S:Section /I:Item /A:Value2 /D:,</pre>	<p>Creates a new file named info.ini with a comma delimited list of values. The resulting info.ini file contains:</p> <pre>[Section] Item = Value1,Value2,Value3,Value2</pre>
<pre>clini info.ini /S:Section /I:Item /U:Value1 /D:, /N clini info.ini /S:Section /I:Item /U:Value2 /D:, clini info.ini /S:Section /I:Item /U:Value3 /D:, clini info.ini /S:Section /I:Item /U:Value2 /D:,</pre>	<p>Creates a new file named info.ini with a comma delimited list of unique values. The resulting info.ini file contains:</p> <pre>[Section] Item = Value1,Value2,Value3</pre>
<pre>clini info.ini /S:Section /I:Item /E:MyEVariable /T:2 or clini info.ini /S:Section /I:Item /E:MyEVariable /T:2 /D:,</pre>	<p>Reads information from the info.ini file created in the previous example, and sets the second value of the item to the <i>MyEVariable</i> environment variable. The resulting CLIniSet.bat file contains:</p> <pre>Set MyEVariable=Value2</pre>

Table 22. *clini* examples (continued)

Example	Description
<pre>Content of doit.bat: @Echo off clini info.ini /S:Section /I:Item /V:Value1 /N clini info.ini /S:Section /I:Item /=:Value1 if errorlevel 100 goto itsfalse if errorlevel 1 goto error if errorlevel 0 goto itstrue :error Echo Error occurred Goto end :itsfalse Echo It's false Goto end :itstrue Echo It's true :end</pre>	<p>This example creates a file called info.ini with the following content:</p> <pre>[Section] Item = Value1</pre> <p>Then it checks to see if the value of Item in [Section] is equal to Value1 and displays a message.</p> <p>After running doit.bat, the follow message is displayed: It's true</p>
<pre>Content of doit.bat: @Echo off clini info.ini /S:Section /I:Item /V:Value1 /N clini info.ini /S:Section /I:Item /C:alu if errorlevel 100 goto itsfalse if errorlevel 1 goto error if errorlevel 0 goto itstrue :error Echo Error occurred Goto end :itsfalse Echo It's false Goto end :itstrue Echo It's true :end</pre>	<p>This example creates a file called info.ini with the following content:</p> <pre>[Section] Item = Value1</pre> <p>Then it checks to see if the value of Item in [Section] contains substring alu and displays a message.</p> <p>After running doit.bat, the follow message is displayed: It's true</p>
<pre>Content of doit.bat: @Echo off clini info.ini /S:Section /I:Item /V:V1,V2,V3 /N clini info.ini /S:Section /I:Item /CT:V2 if errorlevel 100 goto itsfalse if errorlevel 1 goto error if errorlevel 0 goto itstrue :error Echo Error occurred Goto end :itsfalse Echo It's false Goto end :itstrue Echo It's true :end</pre>	<p>This example creates a file called info.ini with the following contents:</p> <pre>[Section] Item = V1,V2,V3</pre> <p>Then it checks to see if the value of Item in [Section] contains token V2 in a comma delimited list and displays a message.</p> <p>After running doit.bat, the follow message is displayed: It's true</p>

Table 22. *clini* examples (continued)

Example	Description
<code>Clini info1.ini info2.ini</code>	This example copies all the sections, items, and values from <code>info1.ini</code> into <code>info2.ini</code> . Any existing values for items in <code>info2.ini</code> are replaced.
<code>Clini info1.ini info2.ini /P</code>	This example copies all the sections, items, and values from <code>info1.ini</code> into <code>info2.ini</code> . Any values for existing items in <code>info2.ini</code> are kept. Only new items and values are copied over from <code>info1.ini</code> .
<code>Clini info1.ini info2.ini /S:MySection</code>	This example copies all the items and values from the section called <code>MySection</code> in <code>info1.ini</code> into the section called <code>MySection</code> in <code>info2.ini</code> replacing any values that may already exist in the section called <code>MySection</code> in <code>info2.ini</code> .
<code>Clini info1.ini info2.ini /S:MySection /I:MyItem</code>	This example copies the value from the section called <code>MySection</code> , for the item called <code>MyItem</code> in <code>info1.ini</code> into the same section and item in <code>info2.ini</code> replacing the existing value in <code>info2.ini</code> if it already exists.
<code>Clini info1.ini info2.ini /ES</code>	This example copies all the items and values from the empty section (items and values that are not in a section) in <code>info1.ini</code> into <code>info2.ini</code> replacing any existing items in the empty section in <code>info2.ini</code> .
<code>Clini info1.ini info2.ini /A</code>	This example appends all the values from the sections and items from <code>info1.ini</code> to <code>info2.ini</code> .
<code>Clini info1.ini info2.ini /U</code>	This example uniquely appends all the values from the sections and items from <code>info1.ini</code> to <code>info2.ini</code> if the value does not already exist in <code>info2.ini</code> .
<code>Clini info1.ini info2.ini /U /D:</code>	This example uniquely appends all the values from the sections and items from <code>info1.ini</code> to <code>info2.ini</code> using a comma as the delimiter if the value does not already exist in <code>info2.ini</code> .
<code>Clini info.ini /V:My Ini Line /CMT</code>	This example comments out the line <code>My Ini Line</code> in the empty section in the <code>info.ini</code> file with a semicolon if the line exists.
<code>Clini info.ini /S:MySection /V:My Ini Line /UCMT</code>	This example uncomments the line <code>My Ini Line</code> in the <code>MySection</code> section of the <code>info.ini</code> if the line exists.
<code>Clini info.ini /I:MyItem /CMT</code>	This example comments out the line indicated by the item <code>MyItem</code> in the empty section of the <code>info.ini</code> file if the item exists.
<code>Clini info.ini /S:MySection /I:MyItem /CMT /CC:#</code>	This example comments out the line indicated by the item <code>MyItem</code> in the section <code>MySection</code> in the <code>info.ini</code> file with a <code>#</code> sign if the item exists.
<code>Clini info.ini /s:MySection /AI /V:My Value /CMT</code>	This example comments out the lines indicated by any item that has a value of <i>My Value</i> of all the items in the section <code>MySection</code> in the <code>info.ini</code> file if the item exists.
<code>Clini info.ini /s:MySection /CMT</code>	This example comments out the section header indicated by <code>MySection</code> in the <code>info.ini</code> file if the section exists.

ddcopy.exe

The **ddcopy** utility, is a Device Driver Copy utility that copies only those drivers in the driver set that support a specific machine.

Two versions of the utility come with ServerGuide Scripting Toolkit:

- A 32-bit version for Windows 32-bit operating systems and for the Windows Preinstallation Environment (Windows PE) 2.1/3.0 (32-bit)
- A 64-bit version for Windows x64 operating systems and for Windows PE 2.1/3.0 (x64).

Microsoft uses the term *device-driver directory* to refer to the directory that contains drivers for an individual device. All device-driver files are located in individual device-driver directories that contain the files for an individual device driver. The following directories are device-driver directories:

- \adaptec
- \asm
- \hal
- \ideraid
- \srvraid

However, when specifying the source path for the **ddcopy** command, always specify the directory that contains the drvset.ini file. In this case, the source directory is the C:\w03_drv\%oem%\\$1\drv directory.

When you issue a **ddcopy** command against a directory that contains device-driver directories, Ddcopy copies all of the device drivers that are specific to the specified machine types and any other files located in the drv directory to the new location.

The SupportedSystems keyword in the drvset.ini file is modified to reflect the new machine list.

The ddcopy.exe utility has the following command-line syntax:

```
ddcopy <source_path> <destination_path> [/M:machine_types or platform_ids]
[/C:category] | [/V:n] [/?]
```

The **ddcopy** utility uses these parameters.

Table 23. ddcopy parameters

Parameter	Description
<i>source_path</i>	Defines the fully qualified path to the directory that contains the device-driver directories and the drvset.ini file.
<i>destination_path</i>	Specifies the fully qualified path of the target directory for copying the device drivers.
<i>/M:machine_types or platform_ids</i>	Specifies machine types or platform IDs to limit the number of drivers that are copied. Multiple machine types or platform IDs are allowed when delimited by commas.
<i>/C:category</i>	Specifies the driver categories to limit the device drivers to be copied. Multiple categories can be specified using a comma as the delimiter. If omitted, then all the device driver categories are copied. Valid values are: <ul style="list-style-type: none"> • Network • Video • Management • Chipset • Mass Storage • Application • Tape • Hotfix

Table 23. *ddcopy* parameters (continued)

Parameter	Description
	This is only valid with driver sets from ServerGuide 7.4.12 or greater.
/V:n	Specifies the verbose level used to report status during the deployment process. Valid values for n are: <ul style="list-style-type: none"> • 0 - quiet mode • 3 - default • 5 - maximum information

The **ddcopy** utility returns the following values to indicate status:

Table 24. *ddcopy* return values

Value	Description
0	Success
1	Syntax error
2	Program error
3	Failed to copy
4	Machine type or platform ID not found
5	Destination is read-only
6	File not found

The following examples illustrate **ddcopy** utility usage.

Table 25. *ddcopy* examples

Example	Description
<code>ddcopy d:\drivers\%oem%\\$1\drv c:\wininst\%oem%\\$1\drv</code>	Copies all the drivers from the d:\drivers\%oem%\\$1\drv directory to the c:\wininst\%oem%\\$1\drv directory
<code>ddcopy d:\drivers\%oem%\\$1\drv c:\wininst\%oem%\\$1\drv /M:8832</code>	Copies the drivers that are specifically for machine type 8832 from the d:\drivers\%oem%\\$1\drv directory to the c:\wininst\%oem%\\$1\drv directory and updates the supported systems field in the drivers' DrvInfo.ini file and the c:\wininst\%oem%\\$1\drv\drvset.ini file.

Table 25. *ddcopy* examples (continued)

Example	Description
<pre>ddcopy d:\drivers\%oem%\\$1\drv c:\wininst\%oem%\\$1\drv /M:8832,8865</pre>	<p>Copies the drivers for machine type 8832 and machine type 8865 from the d:\drivers\%oem%\\$1\drv directory to the c:\wininst\%oem%\\$1\drv directory and updates the supported systems field in the drivers' DrvInfo.ini file and the c:\wininst\%oem%\\$1\drv\drvset.ini file.</p>
<pre>ddcopy d:\drivers\%oem%\\$1\drv c:\mydrvs /m:8832 /c:network,video</pre>	<p>Copies the Network and Video drivers specific to the machine type 8832 from the d:\drivers\%oem%\\$1\drv directory to the c:\mydrvs directory and updates the supported systems field in the drivers' DrvInfo.ini file as well as the c:\mydrvs\drvset.ini file.</p>

dscan.exe

The **dscan** utility is a driver scan utility that performs the following functions:

- Scans a device driver or set of device drivers to determine the installation mode (text mode, Plug and Play, or executable) and write this information to the drvinfo.ini file that is located in each device-driver directory. The drvinfo.ini file is used by the **unattend** utility during the installation of Windows operating systems.
- Creates a text mode directory, copy all text mode device drivers into that directory, then dynamically create a master txtsetup.oem file that contains all of the unique information that is in the individual txtsetup.oem files. Known unattended installation defects are automatically addressed.

Two versions of the utility come with ServerGuide Scripting Toolkit:

- A 32-bit version for Windows 32-bit operating systems and for the Windows Preinstallation Environment (Windows PE) 2.1/3.0 (32-bit)
- A 64-bit version for Windows x64 operating systems and for Windows PE 2.1/3.0 (x64).

Microsoft uses the term *device-driver directory* to refer to the directory that contains drivers for an individual device. All device-driver files are located in individual device-driver directories that contain the files for an individual device driver. The following directories are device-driver directories:

- \adaptec
- \asm
- \hal
- \ideraid
- \srvraid

However, when specifying the source path for the **ddcopy** command, always specify the directory that contains the drvset.ini file. In this case, the source directory is the C:\w03_drv\%oem%\\$1\drv directory.

When you issue a **dscan** utility against a directory that contains device-driver directories, **dscan** performs its tasks against all of the subdirectories that the directory contains, with the exception of the drvutils directory. The drvutils directory contains two utilities, Holdit.exe and Reboot.exe, which are used by the **unattend** utility.

The Driver Scan utility stores information in an INI file named `drvinfo.ini` in the device-driver directory, for use by the **unattend** utility. If the `drvinfo.ini` file already exists for the device driver, it is left unchanged. For more information, see

- “`driverinfo.ini`” on page 87
- “`unattend.exe`” on page 105

The Driver Scan utility can also merge text mode device drivers into a single directory. This merges the device-driver files and the `txtsetup.oem` files for use in unattended installations. If the destination directory for text mode drivers already exists, it is automatically deleted and recreated.

The **dscan** utility automatically assumes that the device driver being scanned is applicable to all target servers. To make a device driver server-specific, you must modify the `drvinfo.ini` file to reflect the servers that the device driver supports.

The **dscan** utility command-line syntax:

```
dscan <driver_path> [/S]/SS[/T[:path]] [/M:machine_type/platform_ID] [/H:filename  
[/OW] [/V:n] [/W:n] [/O:file_name] [/?]
```

The **dscan** utility uses these parameters.

Table 26. *dscan* parameters

Parameter	Description
<i>driver_path</i>	<ul style="list-style-type: none"> • Defines the fully qualified path to the directory to scan for device drivers. Each driver is assumed to be in a separate subdirectory within this path. • If <i>driver_path</i> has <code>\$oem\$</code> in the path, the Driver Scan utility creates the <code>\$oem\$\textmode</code> directory and merges the text mode device drivers. • If the /SS parameter is used, the path is assumed to be the path to a single device driver.
/S	<ul style="list-style-type: none"> • Specifies to scan device drivers and create <code>drvinfo.ini</code> files only, if necessary. • Text mode device drivers are not merged when this parameter is used.
/SS	<ul style="list-style-type: none"> • Specifies to scan a single device driver and create the <code>drvinfo.ini</code> file only, if necessary. • Text mode device drivers are not merged when this parameter is used.
/T[:path]	<ul style="list-style-type: none"> • Specifies to build the text mode device drivers only. • Other device drivers are not scanned, and <code>drvinfo.ini</code> files are not created when this parameter is used. • If <i>path</i> is specified, the text mode device drivers are merged to the specified path. Otherwise, the <i>driver_path</i> parameter must have <code>\$oem\$</code> in the path so that the text mode device drivers are merged into the <code>\$oem\$\textmode</code> directory.
/M:machine_type/platform_ID	<ul style="list-style-type: none"> • Specifies a machine type, where <i>machine_type</i> is the machine type of the target server or platform ID, where <i>platform_ID</i> is the platform ID of the target server, that is used to limit merging of the text-mode device drivers. • If this parameter is not specified, all text-mode device drivers are merged. The /T parameter is required to use this parameter.
/H:filename	<ul style="list-style-type: none"> • Specifies a fully-qualified path and file name for the <code>hwdetect.ini</code> file that was created by the hwdetect utility. • Limits the merging of the text mode device drivers to only those drivers detected in the system.

Table 26. dscan parameters (continued)

Parameter	Description
/OW	<ul style="list-style-type: none"> Overwrites the text-mode drivers without deleting and recreating the text-mode directory. This parameter cannot be used with the /S parameter or the /SS parameter.
/V:n	Specifies the verbose level used to report status during the deployment process. Valid values for <i>n</i> are: <ul style="list-style-type: none"> 0 - quiet mode 3 - default 5 - maximum information
/W:n	Specifies the version of Microsoft Windows for the device drivers: <ul style="list-style-type: none"> 0 for Windows 2000 1 for Windows Server 2003 2 for Windows 2000 Professional 3 for Windows XP 4 for Windows Server 2003 x64
/O:file_name	Combines the information in the drvinfo.ini files into a single file specified by the file_name value.
/?	Displays usage information.

The dscan.exe utility returns the following values to indicate status:

- 0 - Success.
- 1 - Syntax error.
- 2 - Program error.
- 3 - Destination is read-only.

The following examples illustrate Driver Scan utility usage.

Table 27. dscan examples

Example	Description
dscan c:\insttemp\%oem%\\$1\drv	Scans a device-driver set in c:\insttemp\%oem%\\$1\drv, creates the drvinfo.ini files for each device driver, and builds the text-mode directory.
dscan c:\drv /S	Scans a device-driver set in c:\drv and creates drvinfo.ini files for each device driver, but does not build the text mode device drivers.
dscan c:\drv\mydriver /SS	Scans a single device driver in d:\drv\mydriver and creates the drvinfo.ini file for that device driver, but does not build the text mode device driver.
dscan c:\w2\%oem%\\$1\drv /T	Builds the text mode directory in c:\w2\%oem%\textmode using device drivers found in c:\w2\%oem%\\$1\drv, but does not create any drvinfo.ini files.
dscan c:\drivers /T:c:\other\textmode	Builds the text mode directory in c:\other\textmode using device drivers found in c:\drivers, but does not create drvinfo.ini files.

driverinfo.ini

The drvinfo.ini file contains information specific to each device driver and is used by the **unattend** utility to add device-driver information to the answer file for Windows deployment scenarios.

You can either create this file or have the Driver Scan utility create it automatically.

The drvinfo.ini file contains one section, called [Driver Information], and can contain the following variables:

Table 28. drvinfo.ini variables

Variable name	Description
Automatically Reboots	<ul style="list-style-type: none">• Specifies whether the executable device driver automatically restarts (reboots) the target server after the device-driver installation has completed. This variable is only valid when Installation Mode is set to "Executable". Valid values are True or False. The default value is False.• If an executable-device-driver installation program restarts the server and this variable is set to False, then any remaining installation procedures are not completed.• To use this variable, the drvutils directory must contain the Holdit.exe and Reboot.exe utilities.• This variable is only supported for Windows 2000.
Installation Mode	<ul style="list-style-type: none">• Specifies the installation method for the device driver. Valid values are: Executable, Manual, PnP, or Textmode.• If set to <i>Manual</i>, the unattend utility does not install the device driver.
Order Before	<ul style="list-style-type: none">• Specifies that the device driver is added to the answer file before another specified device driver.• This variable is valid only when Installation Mode is set to "PnP" or "Executable".• Valid values are a comma-delimited list of the names of the device-driver directories, or the special keyword "All".• If more than one device driver has a value of "All", the device drivers are installed in alphabetical order before those that do not have the specification.
Order After	<ul style="list-style-type: none">• Specifies that the device driver is added to the answer file after all other device drivers.• This variable is only valid when Installation Mode is set to "PnP" or "Executable".• The only valid value is the special keyword "All".• If more than one device driver has this value set to "All", the device drivers are installed in alphabetical order after those that do not have the specification.
Parameters	<ul style="list-style-type: none">• Defines any required command-line parameters required by the executable device driver.• This variable is only valid when Installation Mode is set to <i>Executable</i>.
Path	<p>Specifies the path to the installation file. This variable has a different function, depending on the setting of the Installation Mode variable, as indicated below:</p> <ul style="list-style-type: none">• Executable - Path specifies the path to the executable installation file• Manual - Path variable is ignored• PnP - Path specifies the path to the INF installation files• Textmode - Path specifies the path to the txtsetup.oem file

Table 28. *drvinfo.ini* variables (continued)

Variable name	Description
PCIVenDevID	<ul style="list-style-type: none"> Specifies the PCI Vendor ID or Device ID information used to limit the installation of executable device drivers to only when the specified device is in the target server. This variable is only valid when Installation Mode is set to <i>Executable</i>. Entries must be in the same format as those in the <code>txtsetup.oem</code> file, with multiple entries delimited by commas. For example: PCIVenDevID = PCI\VEN_1002&DEV_5159&SUBSYS_029A1014
Reboot Required	<ul style="list-style-type: none"> Specifies whether the executable device driver requires the target server to restart (reboot) after the installation of the device driver is completed. This variable is only valid when Installation Mode is set to <i>Executable</i>. Valid values are True or False. To use this variable, the <code>drvutils</code> directory must contain the <code>Holdit.exe</code> and <code>Reboot.exe</code> utilities. This variable is only supported for Windows 2000.
Supported Locales	<ul style="list-style-type: none"> Specifies the locales supported by this device driver. This value can be All, or a comma-delimited list of locales, as specified by the Localization variable in the <code>ProdSpec.ini</code> file from the <code>i386</code> directory of the applicable operating system installation directory. You cannot use both Supported Systems and Unsupported Systems in the same <code>drvinfo.ini</code> file.
Supported Systems	<ul style="list-style-type: none"> Specifies the servers supported by this device driver. This value can be All, None, or a comma-delimited list of server machine types or platform IDs. Cannot use both Supported Systems and Unsupported Systems in the same <code>drvinfo.ini</code> file.
Unsupported Locales	<ul style="list-style-type: none"> Specifies locales that are not supported by this device driver. This value must be a comma-delimited list of locales, as specified by the Localization variable in the <code>ProdSpec.ini</code> file from the <code>i386</code> directory of the applicable operating system installation directory. Cannot use both Supported Locales and Unsupported Locales in the same <code>drvinfo.ini</code> file.
Unsupported Systems	<ul style="list-style-type: none"> Specifies servers that are not supported by this device driver. This value must be a comma-delimited list of server machine types or platform IDs. Cannot use both Supported Locales and Unsupported Locales in the same <code>drvinfo.ini</code> file.

The following examples illustrate `drvinfo.ini` file contents.

Table 29. *drvinfo.ini* file contents

Example	Description
<pre>[Driver Information] Installation Mode=PnP Path= Parameters= Automatically Reboots= Reboot Required= PCIVenDevID= Order Before= Supported Systems=All</pre>	<p>Supports a plug-and-play device driver with INF files in the root of the device driver directory, and supports all target servers.</p>

Table 29. *drvinfo.ini* file contents (continued)

Example	Description
<pre> ;Unsupported Systems= Supported Locales= ;Unsupported Systems= </pre>	
<pre> [Driver Information] Installation Mode=PnP Path=win2000 Parameters= Automatically Reboots= Reboot Required= PCIvenDevID= Order Before= Supported Systems=8673,8679,8687 ;Unsupported Systems= Supported Locales= ;Unsupported Systems= </pre>	<p>Supports a plug-and-play device driver with INF files in the win2000 directory in the root of the device driver directory, and only supports target servers with machine types 8673, 8679, and 8687.</p>
<pre> [Driver Information] Installation Mode=PnP Path=win2000 Parameters= Automatically Reboots= Reboot Required= PCIvenDevID= Order Before=All Supported Systems=8673,8679,8687 ;Unsupported Systems= Supported Locales= ;Unsupported Systems= </pre>	<p>Supports a plug-and-play device driver that must be installed before any other plug-and-play device driver, with INF files in the win2000 directory in the root of the device driver directory, and supports only target servers with machine types 8673, 8679, and 8687.</p>
<pre> [Driver Information] Installation Mode=Executable Path=win2000\setup.exe Parameters=-Q Automatically Reboots=False Reboot Required=True PCIvenDevID=PCI\VEN_1002&DEV_5159&SUBSYS_1014029A Order Before=All ;Supported Systems= Unsupported Systems=8687 Supported Locales= ;Unsupported Systems= </pre>	<p>Supports an executable device-driver installation using <code>setup.exe</code>, with a <code>-Q</code> parameter, in the win2000 directory in the root of the device driver directory, and supports all target servers (except those of machine type 8687) that have a device installed matching the specified Vendor/Device ID.</p>

hwdetect.exe

The **hwdetect** (hardware detect) utility performs basic hardware detection functions that are typically obtained by using SMBIOS and a PCI scan.

The **hwdetect** utility has options that can be used to dump all of the hardware information to an output file, or it can be used to query hardware information and return values that set the *errorlevel* environment variable or the return code in Windows PE.

hwdetect performs basic hardware scan functions and more complex PCI device detection options. The basic hardware scan functions can only be used singularly. The PCI device detection functions may be used in combination with each other to produce a query based on multiple restrictions.

You can only use the **hwdetect** utility basic hardware scan functions one at a time. The PCI-device detection functions can be combined or used more than once on the same command line.

Two versions of the utility come with ServerGuide Scripting Toolkit:

- A 32-bit version for Windows 32-bit operating systems and for the Windows Preinstallation Environment (Windows PE) 2.1/3.0 (32-bit)
- A 64-bit version for Windows x64 operating systems and for Windows PE 2.1/3.0 (x64).

The **hwdetect** utility command-line syntax:

```
hwdetect [/s|/p|/i|/m:type] [/vid:vendor_id] [/did:device_id] [/svid:sub_vendor_id]
[/sdid:sub_device_id] [/bn:bus_number] [/dn:device_number] [/add:num]
```

Table 30. *hwdetect* parameters

Parameter	Description
Basic hardware scan functions	
<i>/s</i>	<p>Determines if the target server is a LenovoeServer or LenovoeServerxSeries server. The return values are:</p> <ul style="list-style-type: none"> • 0 for a LenovoeServer or LenovoeServerxSeries server • 1 for a non-LenovoServer or LenovoeServerxSeries server
<i>/p</i>	<ul style="list-style-type: none"> • Displays all hardware information for the target server in a variable=value format. • The return value is 0 when successful. • You can use the > output-redirect option to save the output to an output file. For example, <code>hwdetect /p > filename</code>.
<i>/i</i>	<ul style="list-style-type: none"> • Displays all hardware information for the target server in an INI-file format. • The return value is 0 when successful. • You can use the > output-redirect option to save the output to an output file.
<i>/m:type</i>	<p>Compares the machine type of the target xSeries server to the specified machine type, <i>type</i>. The return values are:</p> <ul style="list-style-type: none"> • 0 when the machine types are different or no basis for comparison exists • 1 when the machine types match

Parameter	Description
PCI-device detection functions	
<i>/vid:vendor_id</i>	<ul style="list-style-type: none"> • Determines whether there is a PCI adapter in the target server that matches the specified vendor ID, where <i>vendor_id</i> is a hexadecimal value. • The return value is the number of matching adapters in the target server, or 0 if there are no matches.
<i>/did:device_id</i>	<ul style="list-style-type: none"> • Determines whether there is a PCI adapter in the target server that matches the specified device ID, where <i>device_id</i> is a hexadecimal value. • The return value is the number of matching adapters in the target server, or 0 if there are no matches.
<i>/svid:sub_vendor_id</i>	<ul style="list-style-type: none"> • Determines whether there is a PCI adapter in the target server that matches the specified sub-vendor ID, where <i>subvendor_id</i> is a hexadecimal value. • The return value is the number of matching adapters in the target server, or 0 if there are no matches.
<i>/sdid:sub_device_id</i>	<ul style="list-style-type: none"> • Determines whether there is a PCI adapter in the target server that matches the specified sub-device ID, where <i>subdevice_id</i> is a hexadecimal value. • The return value is the number of matching adapters in the target server, or 0 if there are no matches.

Parameter	Description
<code>/bn:bus_number</code>	<ul style="list-style-type: none"> Causes the PCI scan to begin at the specified bus number, instead of starting at bus 0, by default. This parameter is only valid when more than one <code>/vid</code>, <code>/did</code>, <code>/svid</code>, or <code>/sdiid</code> parameter is specified on the command line.
<code>/dn:device_number</code>	<ul style="list-style-type: none"> Causes the PCI scan to begin at the specified device number, instead of starting at device number 0, by default. This parameter is only valid when the <code>/bn</code> parameter is specified on the command line.
<code>/add:num</code>	<ul style="list-style-type: none"> Adds an integer number, <i>num</i>, to the return value before exiting. This is useful to obtain a sum of different PCI adapters, with different PCI IDs, in a target server. The return value is the resultant sum of all other return values plus <i>num</i>.

The following examples illustrate **hwdetect** utility usage.

Table 31. *hwdetect* examples

Example	Description
<pre>hwdetect /s if errorlevel 1 goto NONLenovo if errorlevel 0 goto Lenovo :NONLenovo rem Perform non-Lenovo equipment specific steps here goto FINISH :Lenovo rem Perform Lenovo equipment specific steps here :FINISH</pre>	Determines if the target server is a Lenovo server or not, and branches accordingly to perform equipment-specific steps
<pre>hwdetect /m:8676 if errorlevel 1 goto 8676 hwdetect /m:8669 if errorlevel 1 goto 8669 echo System not supported! goto done :8676 call 8676.bat goto done :8669 call 8669.bat goto done :done</pre>	Determines if the target server is either a machine type 8676 or machine type 8669 server, and branches accordingly to call a system-specific batch file or displays a message of not supporting other machine types
<pre>hwdetect /i>hwdetect.out clini hwdetect.out /S:CI /I:Vendor_ID.0 /E:Vendor CLIniSet.bat</pre>	Creates an output file that lists the hardware configuration for the target server, so that the clini utility can search for a specific PCI adapter from a vendor and set an environment variable accordingly

Table 31. *hwdetect* examples (continued)

Example	Description
<pre>hwdetect /vid:0x9005 /did:0x0250 if errorlevel 1 call 6Mstuff.bat</pre>	<p>Determines if there is at least one IBM ServeRAID 6i/6i+/6M PCI adapter in the target server, and calls a batch file to process adapter-specific tasks</p>
<pre>hwdetect /vid:0x9005 /did:0x0250 if errorlevel 0 set TOTAL=0 if errorlevel 1 set TOTAL=1 if errorlevel 2 set TOTAL=2 if errorlevel 3 set TOTAL=3 hwdetect /add:%TOTAL% /vid:0x1014 /did:0x01BD if errorlevel 0 set TOTAL=0 if errorlevel 1 set TOTAL=1 if errorlevel 2 set TOTAL=2 if errorlevel 3 set TOTAL=3 if errorlevel 4 set TOTAL=4 if errorlevel 5 set TOTAL=5 if errorlevel 6 set TOTAL=6 echo There are %TOTAL% Lenovo ServeRAID adapters in this system</pre>	<p>Determines the total number of Lenovo ServeRAID adapters in the target server, assuming there are no more than three of each type: Lenovo ServeRAID 4, Lenovo ServeRAID 5, and Lenovo ServeRAID 6/6i/6M</p>
<pre>hwdetect /i>hwdetect.out</pre>	<ul style="list-style-type: none"> • Displays hardware configuration information about the target server. • The >hwdetect.out parameter is a Windows output-redirect option that causes the output from the hwdetect utility to be saved in the specified file.

Below is an example of the *hwdetect.out* file using the last parameter in the table above:

```
[System]
Machine_Type=8674
Model_Number=42X
Serial_Number=78Z9506
Product_Name=eserver xSeries 330
BIOS_version=1.04
BIOS_Build_Level=EME112A
BIOS_DATE=06/28/2002
BIOS_Manufacturer=Lenovo
BIOS_Language=US
Number_Of_Enclosures=1
Enclosure_Type.0=23
Processor_Slots=2
Active_Processors=1
Processor_Family.0=17
Processor_Speed_MHz.0=1400
Processor_X64 = TRUE
Total_Enabled_Memory_Mb=256
ROM_Diagnostics_Build_Level=EME112A
ISMP_Build_Level=BR8T30A
RSA_Build_Level=GEE834A
System_UUID = 8030E01060F010B010605090D0A020F0
Blade_Chassis_UUID = 0F020A0D0900F00F020A0D0900F00F02
Blade_Slot = 02

[PCI]
Total_Number_Devices=10
Bus_Number.0=0
Device_Number.0=1
Function_Number.0=0
Class_Code.0=0000
Revision.0=0
```

```

Header_Type.0=0
Vendor_ID.0=5333
Device_ID.0=8A22
Subvendor_ID.0=1014
Subdevice_ID.0=01C5
Bus_Number.1=0
Device_Number.1=2
Function_Number.1=0
Class_Code.1=0000
Revision.1=0
Header_Type.1=0
Vendor_ID.1=8086
Device_ID.1=1229
Subvendor_ID.1=1014
Subdevice_ID.1=105C

```

Running **hwdetect /p** produces the same output with the exception that the section names are tacked onto the beginning of each keyword:

```

System_Machine_Type = 8674
System_Model_Number = 42X
System_Serial_Number = 78Z9506

```

```

PCI_Bus_Number.0 = 0
PCI_Device_Number.0 = 1
...

```

1. The BIOS_DATE value is listed in mm/dd/yyyy format.
2. The Enclosure_Type.0=23 is based on SMBIOS 2.3 spec. 23 = Main chassis.
3. There is an entry for Processor_Family and Processor_Speed_MHz for each microprocessor in the server.
4. The ROM_Diagnostics_Build_Level is empty for servers that do not support ROM diagnostics.
5. PCI devices are listed in the order they are scanned.
6. PCI devices are listed in the *Value.n* format, where *Value* is the variable name and *n* is the nth PCI device scanned.
7. The header_type field is not available for versions of hwdetect running on Windows 32 or 64-bit operating systems.
8. The vendor, device, subvendor, and subdevice values are in hexadecimal notation.

invraid.exe

The **invraid** utility is used to obtain RAID settings.

The table below provides the supported RAID adapter information by PRAID. PRAID works by parsing the output of other RAID configuration utilities. To accomplish this, the utilities must be in the system search path.

Table 32. Supported RAID adapter information

Adapter	Controller type	Utility
ServeRAID 7t	ServeRAID-7t	arcconf
ServeRAID 8i	ServeRAID-8i	
ServeRAID 8k	ServeRAID-8k	
ServeRAID 8k l	ServeRAID-8k-l	
ServeRAID 8s	ServeRAID-8s	
ServeRAID B5015	ServeRAID-B5015	brcli

Table 32. Supported RAID adapter information (continued)

Adapter	Controller type	Utility
LSI SAS 1078 IR	LSI-SAS-1078-IR	cfggen
LSI SAS (1064/1064E/1068/1078)	LSI-SAS-RAID	
LSI SCSI (1020/1030)	LSI-SCSI-RAID	
ServeRAID BR10i	ServeRAID-BR10i	
ServeRAID BR10il	ServeRAID-BR10il	
ServeRAID 7e SATA	ServeRAID-7e-SATA	hrconf
ServeRAID 7e SCSI	ServeRAID-7e-SCSI	
ServeRAID 8e SAS	ServeRAID-8e-SAS	
ServeRAID 8e SATA	ServeRAID-8e-SATA	
ServeRAID 6M	ServeRAID-6M	ipssend
LSI MegaRAID 8480	LSI-MegaRAID-8480	storcli
ServeRAID C105	ServeRAID-C105	
ServeRAID C100	ServeRAID-M100	
ServeRAID C100 R5	ServeRAID-M100-R5	
ServeRAID M1xxx Series	ServeRAID-M1xxx	
ServeRAID M1xxx Series R5	ServeRAID-M1xxx_R5	
ServeRAID M5014	ServeRAID-M5014	
ServeRAID M5014 R6/R60	ServeRAID-M5014-R6-R60	
ServeRAID M5015	ServeRAID-M5015	
ServeRAID M5015 R6/R60	ServeRAID-M5015-R6-R60	
ServeRAID M5025	ServeRAID-M5025	
ServeRAID-M5025-R6-R60	ServeRAID M5025 R6/R60	
ServeRAID M51xx Series	ServeRAID-M51xx	
ServeRAID M51xx Series R5	ServeRAID-M51xx_R5	
ServeRAID M51xx Series R5/R6	ServeRAID-M51xx_R5_R6	
ServeRAID M51xx Series R6	ServeRAID-M51xx_R6	
ServeRAID MR10i	ServeRAID-MR10i	
ServeRAID MR10ie	ServeRAID-MR10ie	
ServeRAID MR10il	ServeRAID-MR10il	
ServeRAID MR10is	ServeRAID-MR10is	
ServeRAID MR10k	ServeRAID-MR10k	
ServeRAID MR10M	ServeRAID-MR10M	
ServeRAID M5210	ServeRAID-M5210	
ServeRAID M5210 R5	ServeRAID-M5210_R5	
ServeRAID M1215	ServeRAID-M1215	
ServeRAID M1215 R5	ServeRAID-M1215-R5	
ServeRAID M1210e	ServeRAID-M1210e	

Table 32. Supported RAID adapter information (continued)

Adapter	Controller type	Utility
ServeRAID M5215	ServeRAID-M5215	
ServeRAID H1110/H1135	SAS2004	sas2ircu

Usage

The **invraid** utility has the following command-line syntax:

```
invraid [/I | /P] /L /F
```

Table 33. invraid parameters

Parameter	Description
/I	Displays information about the all host adapters in the system in an INI file format.
/P	Dumps information about all host adapters in a system in a keyword=value format.
/L	Specifies the light version of the RAID utility.
/F	Dumps information about all host adapters in the system to a file.

Return values

Table 34. invraid return values

Return Value	Description
0	Success
1	Syntax Error
2	Program Error

Examples

To dump the information about all RAID controllers in a system to a file in INI file format with the name myraid.ini, use the **/I** parameter as shown here:

```
invraid.exe /i /f:myraid.ini
```

Returns:

```
[System]
```

```
Machine_Type = 7977
```

```
Serial_Number = KOKN689
```

```
Total_Number_Of_Controllers = 2
```

```
[RAIDController.1]
```

```
Model = ServeRAID-8k-l
```

```
BIOSVersion = 5.2-0 (15412)
```

```
FirmwareVersion = 5.2-0 (15412)
```

```
DriverVersion = 5.2-0 (15317)
```

```
RebuildRate = HIGH
```

```
StripeSize = 256
```

```
ReadAhead = ADAPTIVE
```

```
PCI = 9005:0286:FFFF:FFFF
```

```
[RAIDController.1.Array]
```

```
Total_Number_Of_Arrays = 1
```

```
ID.1 = A
```

```
Members.1 = 1,2,3,4
```

```
[RAIDController.1.Hotspares]
Total_Number_Of_Hotspares = 0
```

```
[RAIDController.1.Logical]
Total_Number_Of_Logicals = 1
Array.1 = A
Size.1 = 279800
Raid_Level.1 = 10
WriteCache.1 = AUTO
```

```
[RAIDController.1.Physical]
Total_Number_Of_Physicals = 4
Channel.1 = 1
ID.1 = 0
Size.1 = 140013
Type.1 = SAS
Serial_Number.1 = JDX2JN8K
Channel.2 = 1
ID.2 = 1
Size.2 = 140013
Type.2 = SAS
Serial_Number.2 = Q5902T4N
Channel.3 = 1
ID.3 = 2
Size.3 = 140013
Type.3 = SAS
Serial_Number.3 = Q5902TPA
Channel.4 = 1
ID.4 = 3
Size.4 = 140013
Type.4 = SAS
Serial_Number.4 = Q5902TS8
```

```
[RAIDController.2]
Model = ServeRAID-8s
BIOSVersion = 5.2-0 (15411)
FirmwareVersion = 5.2-0 (15411)
DriverVersion = 5.2-0 (15317)
PCI = 9005:0285:1014:034D
```

```
[RAIDController.2.Array]
Total_Number_Of_Arrays = 0
```

```
[RAIDController.2.Hotspares]
Total_Number_Of_Hotspares = 0
```

```
[RAIDController.2.Logical]
Total_Number_Of_Logicals = 0
```

```
[RAIDController.2.Physical]
Total_Number_Of_Physicals = 0
```

Use the **/P** parameter to return the same information, but with the section title from the properties file shown for each value:

```
invraid /p > myfile.ini
```

Returns:

```
System_Machine_Type = 7977
System_Serial_Number = KOKN689
```

```
RAIDController.1.Model = ServeRAID-8k-l
```



```

RAIDController.1.BIOSVersion = 5.2-0 (15412)
RAIDController.1.FirmwareVersion = 5.2-0 (15412)
RAIDController.1.DriverVersion = 5.2-0 (15317)
RAIDController.1.RebuildRate = HIGH
RAIDController.1.StripeSize = 256
RAIDController.1.ReadAhead = ADAPTIVE
RAIDController.1.PCI = 9005:0286:FFFF:FFFF
RAIDController.1.Array.ID.1 = A
RAIDController.1.Array.Members.1 = 1,2,3,4
RAIDController.1.Logical.Array.1 = A
RAIDController.1.Logical.Size.1 = 279800
RAIDController.1.Logical.Raid_Level.1 = 10
RAIDController.1.Logical.WriteCache.1 = AUTO
RAIDController.1.Physical.Channel.1 = 1
RAIDController.1.Physical.ID.1 = 0
RAIDController.1.Physical.Size.1 = 140013
RAIDController.1.Physical.Type.1 = SAS
RAIDController.1.Physical.Serial_Number.1 = JDX2JN8K
RAIDController.1.Physical.Channel.2 = 1
RAIDController.1.Physical.ID.2 = 1
RAIDController.1.Physical.Size.2 = 140013
RAIDController.1.Physical.Type.2 = SAS
RAIDController.1.Physical.Serial_Number.2 = Q5902T4N
RAIDController.1.Physical.Channel.3 = 1
RAIDController.1.Physical.ID.3 = 2
RAIDController.1.Physical.Size.3 = 140013
RAIDController.1.Physical.Type.3 = SAS
RAIDController.1.Physical.Serial_Number.3 = Q5902TPA
RAIDController.1.Physical.Channel.4 = 1
RAIDController.1.Physical.ID.4 = 3
RAIDController.1.Physical.Size.4 = 140013
RAIDController.1.Physical.Type.4 = SAS
RAIDController.1.Physical.Serial_Number.4 = Q5902TS8
RAIDController.2.Model = ServeRAID-8s
RAIDController.2.BIOSVersion = 5.2-0 (15411)
RAIDController.2.FirmwareVersion = 5.2-0 (15411)
RAIDController.2.DriverVersion = 5.2-0 (15317)
RAIDController.2.PCI = 9005:0285:1014:034D

```

LEcho.exe

The **LEcho** utility displays and writes messages to log files and sets the system error level.

The **LEcho** utility performs the following functions:

- Writes a message to the display
- Writes a message to a log file
- Sets the system error level with a specific code
- Displays a message to the screen while pausing or running a timer for a discreet amount of time

ServerGuide Scripting Toolkit provides 32- and 64-bit versions of **LEcho**.

For **LEcho.exe** to write a message to a log file, you must set the environment variable **LECHO_LOG** to a fully qualified path and file name by using a command similar to the following:

```
set LECHO_LOG=C:\SGTKWinPE\Lecholog.txt
```

LEcho checks the number of characters on the command line against the current command line limits of 8000 characters for the Windows environment. A message is displayed if the characters exceed the limit.

Usage

The **LEcho** utility has the following command-line syntax:

```
LEcho [message] [/F] [/R:n]
[/E:n] [/P] [/P:n] [/T:n]
[/SC] [/SN] [/N] [/LO] [/DO] [/?]
```

Table 35. LEcho parameters

Parameter	Description
<i>message</i>	The message to display to the screen or log file.
/F	<p>Formats the message using the following variables:</p> <ul style="list-style-type: none"> • %d or %nd formats the system date. The format is indicated by <i>n</i>: <ul style="list-style-type: none"> – 0 = Sun 12/31/2006 (Default) – 1 = Sunday, December 31, 2006 – 2 = Sun, Dec 31, 2006 – 3 = Dec 31, 2006 – 4 = 12-31-2006 – 5 = 12/31/2006 – 6 = 2006-12-31 – 7 = 2006-Dec-31 – 8 = 2006-December-31 – 9 = 20061231 • %t or %nt formats the system time. The format is indicated by <i>n</i>. <ul style="list-style-type: none"> – 0 = 16:12:13 (Default) – 1 = 04:12:13 PM • \a sets an alert (bell) • \b - backspace • \f - form feed • \n - newline • \r - carriage return • \t - horizontal tab
/R:n	Repeats the message <i>n</i> times.
/E:n	Displays the error message and sets the system error level to <i>n</i> .
/P	Pauses until a key is pressed.
/P:n	Pauses for <i>n</i> seconds or until a key is pressed
/T:n	Initiates a timer for <i>n</i> seconds. This timer cannot be ended prematurely.
/SC	Suppresses the output of the countdown timer.
/SN	Suppresses the newline character.
/N	Creates a new, blank log file. If the log file already exists, it is overwritten.
/LO	Writes the message to the log file only.
/DO	Writes the message only to the display.
/?	Displays a help message containing the application syntax.

LEcho return codes

The **LEcho** utility returns the following values to indicate status:

- 0 - Success or true.
- 1 - Syntax error.
- 5 - Cannot write to log file.
- 100 - False.
- 255 - Program error.

LEcho examples

The following examples illustrate how to use the **LEcho** utility.

Example	Description
LEcho	Sends a blank line to the display and the log file if LECHO_LOG is set.
LEcho My Message	Sends the text My Message to the display and to the log file if LECHO_LOG is set.
LEcho /T	Displays a message indicating the current system time to the display and to the log file if LECHO_LOG is set.
LEcho My Message /T	Sends the text 13:55:24 My Message to the display and the log file if LECHO_LOG is set. 13:55:24 indicates the current system time.
LEcho My Message /T /DO	Sends the text 13:55:24 My Message to the display only. 13:55:24 indicates the current system time.
LEcho My Message /E:200	Sends the text My Message to the display and the log file if LECHO_LOG is set and then sets the system error level to 200.
LEcho /E:155	Sets the system error level to 155. No text is displayed or logged.
LEcho New Log File /N /LO	<ul style="list-style-type: none">• Creates a new log file from LECHO_LOG.• If a log file already exists, it is deleted and a new one is created.• Sends the text New Log File to the new log file only; no text is displayed to the screen.
LEcho %d\t%t My Message /F	Sends the text Sun 12-31-2006 16:12:13 My Message to the screen and log file if LECHO_LOG is set.
LEcho Pausing for 30 seconds. /P:30	Sends the text Pausing for 30 seconds and starts a countdown timer at 30. You can bypass the countdown by pressing any key.
LEcho Running a 30 second timer. /T:30 /SC /SO	Sends the text Running a 30 second timer. to the screen only and returns control to the environment after 30 seconds. No timer is displayed.

qacli.exe

The **qacli** utility configures Fibre Host Bus Adapters (HBAs).

Two versions of the utility come with ServerGuide Scripting Toolkit:

- A 32-bit version for Windows 32-bit operating systems and for the Windows Preinstallation Environment (Windows PE) 2.1/3.0 (32-bit)
- A 64-bit version for Windows x64 operating systems and for Windows PE 2.1/3.0 (x64).

You can also download this utility from QLogic at . For additional information about the **qaucli** utility, refer to the documentation in the `sgdeploy\SGTKWinPE\Docs\qaucli` directory.

The **qaucli** utility uses these commands.

Table 36. *qaucli* command usage examples

Command	Description
<code>qaucli.exe -pr fc e (view ?)</code>	Shows the current boot device information on all HBAs.
<code>qaucli.exe -pr fc -e (hba_instance hba_wwpn target_wwnn target_wwpn lun_id [prim alt1 alt2 alt3])</code>	Configures the operating system to boot from a particular target. hba_instance The HBA instance number of an HBA port. hba_wwpn The World Wide Port Name of an HBA port. target_wwnn The World Wide Node Name of a target device, in the format <i>nn-nn-nn-nn-nn-nn-nn-nn</i> or <i>nnnnnnnnnnnnnnnnnnnn</i> . target_wwpn The World Wide Port Name of a target device, in the format <i>nn-nn-nn-nn-nn-nn-nn-nn</i> or <i>nnnnnnnnnnnnnnnnnnnn</i> . lun_id The Logical Unit Number of a LUN. prim The primary boot port name. alt<i>n</i> The name of the alternate boot port. You can specify up to three alternate boot ports.
<code>qaucli.exe -pr fc -e (hba_instance hba_wwpn) (view ?)</code>	Shows the current boot device information for the specified HBA port.
<code>qaucli.exe -pr fc -e (hba_instance hba_wwpn) (enable 0 0 0)</code>	Configures the operating system to boot from the first target found by the BIOS. The default LUN is 0.
<code>qaucli.exe -pr fc -e (hba_instance hba_wwpn) disable [prim alt1 alt2 alt3]</code>	Clears the selected boot device settings on the indicated HBA port.
<code>qaucli.exe -pr fc -l (hba_instance hba_wwpn)</code>	Displays information about the LUNs attached to the specified HBA port.

Examples

The following examples provide additional examples of **qaucli** usage.

Note: While some of these examples are broken across multiple lines, when using, you must enter all of the parameters on a single line.

Table 37. Additional examples of qaucli usage

Example	Description
qaucli.exe -pr fc -e view	Displays the current boot device information on all HBAs.
qaucli.exe -pr fc e E0-FF-EE-DE-CD-34-56-30 E0-00-ED-DE-CD-34-56-30 E0-10-ED-DE-CD-34-56-30 1 prim	Configures HBA E0-FF-EE-DE-CD-34-56-30 E0-00-ED-DE-CD-34-56-30 E0-10-ED-DE-CD-34-56-30 to boot from the primary target.
qaucli.exe -pr fc e E0-FF-EE-DE-CD-34-56-30 view	Displays the current boot setting information for HBA port E0-FF-EE-DE-CD-34-56-30.
qaucli.exe -pr fc e E0-FF-EE-DE-CD-34-56-30 disable prim	Clears the selected boot device setting on HBA port E0-FF-EE-DE-CD-34-56-30.
qaucli.exe -pr fc l E0-FF-EE-DE-CD-34-56-30	Displays information about the LUNs attached to HBA port E0-FF-EE-DE-CD-34-56-30.

savestat.cmd

The **savestat.cmd** utility allows you to store and retrieve up to 21 values to persistent storage.

You can use the **savestat** utility to return to your place in an installation, even when a reboot is required. This utility is designed to return values that set the *errorlevel* environment variable so that you can branch in a script or batch file based on the result of the utility's execution.

The utility runs in 32- and 64bit versions of Windows Preinstallation Environment 2.1/3.0, Windows Server 2003, and Windows Server 2008.

Because **savestat.cmd** uses the persistent storage capability of **ASU**, the following files must be available for the script to work:

- asu.exe
- device.cat
- ibm_mdiss_server_os.inf
- savestat.vbs script
- savestat.def

savestat usage

The **savestat** utility command-line syntax:

```
SAVESTAT [/q] /reset
SAVESTAT [/q] /set1=value [.../set2=value ... /set21=value]
SAVESTAT [/q] /getn
SAVESTAT [/q] /validate
SAVESTAT [/q] /signature
```

The **savestat** utility uses these parameters.

Table 38. *savestat* parameters

Parameter	Description	Usage
<code>/setn=value</code>	Saves an integer value, <i>value</i> , to the <i>n</i> th location in persistent-storage memory, where <i>n</i> is an integer from 1-21. Return codes: <ul style="list-style-type: none"> • 0 if successful • 1 if not successful 	<code>savestat /setn=value</code> Where: <ul style="list-style-type: none"> • <i>n</i> is an integer from 1-21 • <i>value</i> is an integer from 0-254
<code>/getn</code>	Retrieves the value currently set in the <i>n</i> th location in persistent-storage memory, where <i>n</i> is an integer from 1-21. Return codes: <ul style="list-style-type: none"> • The value stored at the location specified by <i>n</i>, if successful. • 255 if not successful. 	<code>savestat /getn</code> Where <i>n</i> is an integer from 1-21
<code>/reset</code>	Resets all persistent-storage memory to values of zero. Return codes: <ul style="list-style-type: none"> • 0 if successful • 1 if not successful 	<code>savestat /reset</code>
<code>/signature</code>	Verifies that the persistent storage contains the savestat signature. Return codes: <ul style="list-style-type: none"> • 0 if storage contains the signature • 1 if storage does not contain the signature 	<code>savestat /signature</code>
<code>/validate</code>	Verifies that the system is supported by savestat . Return codes: <ul style="list-style-type: none"> • 0 if the system is supported • 1 if the system is not supported 	<code>savestat /validate</code>
<code>/q</code>	Invokes the quiet mode. This parameter is optional and can be used with any other savestat parameter.	<code>savestat /q /set1=100</code>

savestat examples

The following examples illustrate **savestat** utility usage.

Example	Description
savestat /set2=100	Stores the value 100 in the second persistent-storage memory location
<pre> savestat /get2 if errorlevel 100 goto end if errorlevel 1 goto level1 :level1 call level1.bat :end </pre>	Retrieves the value of the second persistent-storage memory location and branches in the batch file according to the value returned

tksearch.exe

The **tksearch** utility performs a Toolkit search for the following information:

- The latest ServerGuide driver sets starting from a given path.
- The latest ServerGuide driver set or sets that supports one or more specified machine types.
- The latest ServerGuide driver set or sets that supports a specified Network Operating System.
- The architecture and type of a Network Operating System.

Two versions of the utility come with ServerGuide Scripting Toolkit:

- A 32-bit version for Windows 32-bit operating systems and for the Windows Preinstallation Environment (Windows PE) 2.1/3.0 (32-bit)
- A 64-bit version for Windows x64 operating systems and for Windows PE 2.1/3.0 (x64).

tksearch usage

The **tksearch** utility syntax:

```
tksearch driverpath [/W:n | /WP:nospath | WP:prodspec.ini] [/M:machtype/platform_ID] [/B:filename] [/?]
```

The **tksearch** utility uses these parameters.

Table 39. *tksearch* parameters

Parameter	Description
<i>driverpath</i>	The fully qualified directory path to start searching for driver sets. For example: F:\sgdeploy\drvs.
<i>/W:n</i>	The preferred version of Windows to device drivers to search for: <ul style="list-style-type: none"> • 0 = Windows Server 2000 • 1 = Windows Server 2003 (Default) • 2 = Windows 2000 Professional • 3 = Windows XP • 4 = Windows Server 2003 x64
<i>/WP:nospath</i>	The fully qualified path to directory to start searching for the Windows prodspec.ini file. For example, F:\sgdeploy\os\w23_std
<i>/WP:prodspec.ini</i>	The fully qualified path to the Windows prodspec.ini file. For example, F:\sgdeploy\os\w23_std\i386\prodspec.ini

Table 39. *tksearch* parameters (continued)

Parameter	Description
<i>/I:machinetype/platform_ID</i>	<ul style="list-style-type: none"> Limits the search to the specified machine types or platform IDs. To specify multiple machine types or platform IDs use a comma as the delimiter, for example, <i>/I:8853,8854</i>.
<i>/B:filename</i>	<ul style="list-style-type: none"> The name of the batch file in which to place the resulted environment variables. The default is <i>.\DrvSet.bat</i>. Environment variables are: <ul style="list-style-type: none"> <i>TK_NOS_Type</i> - (Win2000, Win2003) <i>TK_NOS_Arch</i> - (I386, AMD64) <i>TK_NOS_Arch_Type</i> - (x86, x64) <i>TK_NOS_DDL_Type</i> - (Win2000 Server, Win2003 Server, Win2003 Server x64) <i>TK_NOS_DDL_Path_#</i> - (number of drivers sets found) <i>TK_NOS_DDL_Path_1</i> - (corresponding drivers set) <i>TK_NOS_DDL_Path_2</i> - (corresponding drivers set) <i>TK_NOS_DDL_Path_3</i> (corresponding drivers set)

Return codes

The **tksearch** utility returns the following values to indicate status:

- 0 - Success: One or more driver libraries were found.
- 1 - Success: No driver libraries were found.
- 2 - Error: Command-line syntax error.
- 3 - Error: Writing output batch file.
- 4 - Error: Driver path not found.
- 5 - Error: NOS not found in specified path.
- 6 - Error: General application error.

tksearch examples

The following examples illustrate **tksearch** utility usage.

Table 40. *tksearch* examples

Example	Description
<code>tksearch f:\sgdeploy\drvs</code>	Search all ServerGuide drivers sets regardless of machine type or NOS type.
<code>tksearch f:\sgdeploy\drvs /W:4</code>	Search all ServerGuide drivers sets for Windows Server 2003 x64 regardless of machine type.
<code>tksearch f:\sgdeploy\drvs /WP:f:\w2000\i386\prodspec.ini</code>	Search all ServerGuide drivers sets for Windows Server 2000 regardless of machine type.
<code>tksearch f:\sgdeploy\drvs /M:8853,7978</code>	Search all ServerGuide drivers sets for systems with machine type 8853 and 7978 regardless the NOS type.
<code>tksearch f:\sgdeploy\drvs /M:8853 /W:f:\sgdeploy\os\w23_std</code>	Search all ServerGuide drivers sets for systems with machine type 8853 and Windows 2003 Standard.

Table 40. *tksearch examples (continued)*

Example	Description
<code>tksearch f:\sgdeploy\drvs /M:8853,7978 /B:</code>	Search all ServerGuide drivers sets for systems with machine type 8853 and 7978 regardless the NOS type. The search results are saved as environment variables in Batch file <code>drvset.bat</code> .
<code>tksearch f:\sgdeploy\drvs /M:8853 /WP:F:\sgdeploy\os\w23_ee /B:drivers.bat</code>	Search all ServerGuide drivers sets for systems with machine type 8853 and Windows 2003 The search results are saved as environment variables in the batch file <code>drivers.bat</code> .

unattend.exe

The **unattend** utility dynamically adds server-specific device-driver information to the Microsoft Windows answer file for an unattended installation deployment.

The device drivers on the *Lenovo ServerGuide Setup and Installation CD* are already configured for use with the **unattend** utility.

Two versions of the utility come with ServerGuide Scripting Toolkit:

- A 32-bit version for Windows 32-bit operating systems and for the Windows Preinstallation Environment (Windows PE) 2.1/3.0 (32-bit)
- A 64-bit version for Windows x64 operating systems and for Windows PE 2.1/3.0 (x64).

The **unattend** utility processes three types of device drivers and adds the device-driver information to the answer file in one or more locations, depending on the type of device driver.

Table 41. *unattend utility device drivers*

Device Driver	Notes
Text mode	<ul style="list-style-type: none"> • Text mode device-driver information is added to the [MassStorageDevices] and [OemBootFiles] sections. • Entries are not duplicated, and existing entries are not changed. • The unattend utility uses the information in the <code>hwdetect.ini</code> file and the <code>txtsetup.oem</code> file (located in the <code>\\$oem\$\textmode</code> directory of the target server) to add the text mode device driver information to the answer file. • This utility also adds the Microsoft retail text mode device drivers using information specified in the <code>txtsetup.sif</code> file from the <code>i386</code> directory of the Windows operating-system installation source files.
Hardware abstraction layer (HAL)	<ul style="list-style-type: none"> • HAL device-driver information is assigned to the <code>ComputerType</code> keyword in the [Unattended] section. • Any value previously assigned to this keyword is overwritten.
Plug and Play	<ul style="list-style-type: none"> • The <code>OemPnPDriversPath</code> keyword in the [Unattended] section is set to the path to the PnP device driver directory. • Any value previously assigned to this keyword is overwritten. • The unattend utility uses information from the <code>hwdetect.ini</code> file (the output of the <code>hwdetect.exe</code> utility) and the <code>drvinfo.ini</code> file in each device-driver directory in the target server to determine the device drivers to add to the answer file. • Device drivers that are not supported on the target server are deleted from the device drivers directory.

unattend utility usage

The **unattend** utility has the following command-line syntax:

```
unattend [/?] <file_name|/U:file /D:path /H:file /I:path> [/U:file] [/D:path]
[/H:file_name] [/I:path] [/S:drive] [/C] [/T] [/P] [/E] [/V:n]
```

Table 42. unattend parameters

Parameter	Description
/?	Displays all parameters.
file_name	Specifies a fully qualified path and file name for the unattend.ini file that contains command-line parameters for the unattend.exe utility. You can put parameters in this file instead of typing them all on the command line, which is useful for long command lines that exceed the limit of 127 characters. <ol style="list-style-type: none">Any settings for the /U, /D, /H, or /I parameters you place on the command line will override settings in the unattend.ini file.If you do not specify a setting for <i>filename</i>, you must specify the /U, /D, /H, and /I parameters on the command line.
/U:file	Specifies a fully qualified path and file name for the answer file.
/D:path	Specifies a fully qualified path to the device drivers directory in the target server. If <i>path</i> does not include \$oem\$, you must use the /T parameter to process text mode device drivers.
/H:file	Specifies a fully qualified path and file name for the hwdetect.ini file that was created by the hwdetect.exe utility.
/I:path	Specifies a fully qualified path to the i386 directory in the target server.
/S:drive	Specifies the drive letter on the target server to which the operating system is being installed.
/C	Creates a default unattend.ini file
/T	Causes the unattend.exe utility to add only the text mode device-driver information to the answer file.
/P	Causes the unattend.exe utility to add only the plug-and-play device driver information to the answer file.
/E	Causes the unattend.exe utility to add only the executable device driver information to the answer file.
/V:n	Specifies the verbose level used to report status during the deployment process. Valid values for <i>n</i> are: <ul style="list-style-type: none">0 - quiet mode3 - default5 - maximum information

unattend utility return codes

The **unattend** utility returns the following values to indicate status:

- 0 - Success.
- 1 - Syntax error.
- 2 - Program error.
- 3 - Destination is read-only.
- 4 - No device-driver information files found.

unattend utility examples

The following examples illustrate how to use the **unattend** utility.

Table 43. unattend examples

Example	Description
unattend /U:c:\unattend.txt /D:c:\w2\%oem%\\$1\drv /H:c:\hwdetect.ini /I:c:\i386	Adds plug-and-play and executable device drivers from c:\w2\%oem%\\$1\drv and text mode device drivers from c:\w2\%oem%\textmode to the answer file and deletes device drivers not specific to the target server.
unattend /U:C:\unattend.txt /D:c:\w2\%oem%\textmode /H:c:\hwdetect.ini /I:c:\i386 /T	Adds only text mode device drivers from the c:\w2\%oem%\textmode directory to the answer file.

unattend.ini

The unattend.ini file contains all required parameters for the unattend.exe utility in a single file. Parameters specified on the command line will override settings in this file.

The unattend.ini file contains two sections, called [Unattend] and [GUIRunOnce]. The [Unattend] section contains variables that you can set instead of providing command-line parameters. The [GUIRunOnce] section enables you to specify a set of commands to run on the target server after the operating system is installed. These commands can run either before the executable device drivers are installed or after they have run.

None of the commands can cause the server to restart (reboot).

Run commands that require user interaction after the executable device drivers are installed.

The unattend.ini file can contain the following variables:

Variable name	Description
[Unattend] section	
Drivers Path	Specifies a fully qualified path to the device-drivers directory in the target server. If %oem% is not in the path, you must use the /T parameter to process text mode device drivers.
Executable	Causes the unattend.exe utility to add only the executable device driver information to the answer file. Valid values are True and False.
HWDetectIni	Specifies a fully qualified path and file name for the hwdetect.ini file that was created by the hwdetect.exe utility.
I386 Path	Specifies a fully qualified path to the i386 directory in the target server.
PnP	Causes the unattend.exe utility to add only the plug-and-play device driver information to the answer file. Valid values are True and False.
System Drive	Specifies the drive letter on the target server to which the operating system is being installed.
Textmode	Causes the unattend.exe utility to add only the text mode device driver information to the answer file. Valid values are True and False.
UnattendTxt	Specifies a fully qualified path and file name for the answer file.
Verbose Level	Specifies the verbose level used to report status during the deployment process. Valid values are: <ul style="list-style-type: none"> • 0 - quiet mode • 3 - default • 5 - maximum information
[GUIRunOnce] section	

Variable name	Description
<i>name_Command</i>	Specifies the <i>name</i> of the command to run.
<i>name_Supported_Systems</i>	The <i>name</i> value specifies the servers on which to run the command. This value can be either All, None, or a comma-delimited list of server machine types or platform IDs. You cannot use both <i>name_Supported_Systems</i> and <i>name_Unsupported_Systems</i> in the same unattend.ini file.
<i>name_Unsupported_Systems</i>	Specifies the servers on which not to run the command, <i>name</i> . This value must be a comma-delimited list of server machine types or platform IDs. You cannot use both <i>name_Supported_Systems</i> and <i>name_Unsupported_Systems</i> in the same unattend.ini file.
After Drivers	Specifies a comma-delimited list of commands to run after the executable device drivers are installed. Each command must have a <i>name_Command</i> variable and either a <i>name_Supported_Systems</i> or <i>name_Unsupported_Systems</i> variable defined.
Before Drivers	Specifies a comma-delimited list of commands to run before the executable device drivers are installed. Each command must have a <i>name_Command</i> variable and either a <i>name_Supported_Systems</i> or <i>name_Unsupported_Systems</i> variable defined.

The following are examples of contents in the unattend.ini file:

Example	Description
<pre>[Unattend] UnattendTxt=c:\unattend.txt Drivers Path=c:\w2\%oem%\\$1\drv HWDetectIni=c:\hwdetect.ini I386 Path=c:\i386 System Drive= Textmode= PnP= Executable= Verbose Level=</pre>	<p>Adds plug-and-play device drivers from c:\w2\%oem%\\$1\drv and text mode device drivers from c:\w2\%oem%\textmode to the answer file and deletes device drivers not specific to the target server.</p>
<pre>[Unattend] UnattendTxt=c:\unattend.txt Drivers Path=c:\w2\%oem%\textmode HWDetectIni=c:\hwdetect.ini I386 Path=c:\i386 System Drive= Textmode=True PnP= Executable= Verbose Level=</pre>	<p>Adds only the text mode device drivers from c:\w2\%oem%\textmode directory to the answer file.</p>
<pre>[Unattend] UnattendTxt=c:\unattend.txt Drivers Path=c:\w2\%oem%\\$1\drv HWDetectIni=c:\hwdetect.ini I386 Path=c:\i386 System Drive= Textmode= PnP= Executable= Verbose Level= [GUIRunOnce] Before Drivers=LaunchIt,MoveIt After Drivers>DeleteIt,FinishIt</pre>	<p>Adds the plug-and-play device drivers and executable device drivers from c:\w2\%oem%\\$1\drv and the text mode device drivers from c:\w2\%oem%\textmode to the answer file, deletes device drivers not specific to the target server, and runs some specific commands both before and after executable device drivers are installed on specific servers</p>

Example	Description
<pre>LaunchIt_Command="CMD.EXE /C c:\RunMe.exe" LaunchIt_Supported_Systems=All MoveIt_Command="CMD.EXE /C Move c:\WinInst\Readme.htm c:\\" MoveIt_Supported_Systems=8676,8870 DeleteIt_Command="CMD.EXE /C RMDIR c:\WinInst /q" DeleteIt_Unsupported_Systems=8870 FinishIt_Command="CMD.EXE /C c:\ShowMsg.exe" FinishIt_Supported_Systems=All</pre>	

valraid.exe

The **valraid** utility validates policy files against inventory files generated by the **invraid** utility.

valraid has two modes of operation:

- **Simulation mode** - simulates the effect a policy file would have on a controller.
- **Check mode** - determines whether the policy file matches the configuration represented in the inventory file.

Simulation mode

Used in simulation mode, **valraid** simulates the effect that a policy file would have on a RAID configuration if it were applied using the **praid** utility, which makes it useful for creating PRAID policy files. The policy files can be tested without running **praid** on the target system.

Check mode

Used in check mode, **valraid** determines whether the policy file specified matches the RAID configuration represented in the inventory file. This capability can be used in OS deployment scripts to skip the RAID configuration step if the controller is already configured with the required RAID configuration and thus avoiding an extra reboot before installing the OS. **valraid** will set the return code to 20 to indicate that the policy file does not match the configuration represented by the inventory file.

valraid usage

The two modes of operation share most parameters, but the syntax is mode-specific.

The simulation mode syntax is:

```
valraid /ini:input_inventory_file /inp:input_policy_file
/outi:output_inventory_file /outp:output_policy_file
/raid:linifiles
```

The check mode syntax is:

```
valraid /c /ini:input_inventory_file /inp:input_policy_file /raid:linifiles
```

The following examples illustrates **valraid** parameters and usage.

Table 44. valraid parameters

Parameter	Description	Example
<i>/ini:input_inventory_file</i>	<ul style="list-style-type: none"> Specifies the input inventory file. To generate the inventory file, run invraid against a target system. 	<pre>valraid /ini:myfile.inv /inp:policy.ini /outi:newfile.inv /outp:newpolicy.ini /raid:/inifiles</pre>
<i>/inp:input_policy_file</i>	Specifies the input policy file.	<pre>valraid /ini:myfile.inv /inp:policy.ini /outi:newfile.inv /outp:newpolicy.ini /raid:/inifiles</pre>
<i>/outi:output_inventory_file</i>	<ul style="list-style-type: none"> This option is valid only for simulation mode. Specifies the file name for the output inventory file. The inventory file represents the RAID configuration that results from using the praid utility to apply <i>input_policy_file</i> to the system described in <i>input_inventory_file</i>. 	<pre>valraid /ini:myfile.inv /inp:policy.ini /outi:newfile.inv /outp:newpolicy.ini /raid:/inifiles</pre>
<i>/outp:output_policy_file</i>	<ul style="list-style-type: none"> This option is valid only for simulation mode. Specifies the file name for the output policy file. The output policy file can be applied to a target system by using the praid utility. 	<pre>valraid /ini:myfile.inv /inp:policy.ini /outi:newfile.inv /outp:newpolicy.ini /raid:/inifiles</pre>
<i>/raid:inifiles</i>	<ul style="list-style-type: none"> Specifies the directory that contains the RAID configuration .ini files. The default is <code>/opt/lvgy/sgtk/sgdeploy/sgtklinux/.data/valraid</code> 	<pre>valraid /ini:myfile.inv /inp:policy.ini /outi:newfile.inv /outp:newpolicy.ini /raid:/inifiles</pre>
<i>/c</i>	<ul style="list-style-type: none"> Specifies check mode to compare the configuration from <i>input_inventory_file</i> to the configuration represented in <i>input_policy_file</i>. The default is simulation mode. 	<pre>valraid /c /ini:myfile.inv /inp:policy.ini /raid:inifiles</pre>

Return codes

The **valraid** utility returns the following values to indicate status:

- 0 - Success.
- 1 - An error occurred while parsing the input policy file.
- 2 - An error occurred while parsing the input inventory file.
- 3 - The controller is not supported.
- 4 - The RAID level is not supported.
- 5 - The Stripe Size is not supported.

- 6 - The number of arrays is not supported.
- 7 - The number of drives in an array is not supported.
- 8 - The number of logical volumes in an array is not supported.
- 9 - There are not enough drives to create hotspare.
- 10 - There are not enough drives of the same size.
- 11 - An error occurred while opening the input policy file.
- 12 - An error occurred while opening the input inventory file.
- 13 - An error occurred while opening the output inventory file.
- 14 - An error occurred while writing to the output inventory file.
- 15 - An error occurred while opening the output policy file.
- 16 - An error occurred while writing to the output policy file.
- 17 - Partial drive sizing is not supported.
- 18 - A command line syntax error occurred.
- 19 - There is no policy match.
- 20 - The controller is not configured, and it does not match the policy file.

winlpcfg.exe

The **winlpcfg** utility configures Fibre Host Bus Adapters (HBAs).

Two versions of the utility come with ServerGuide Scripting Toolkit:

- A 32-bit version for Windows 32-bit operating systems and for the Windows Preinstallation Environment (Windows PE) 2.1/3.0 (32-bit)
- A 64-bit version for Windows x64 operating systems and for Windows PE 2.1/3.0 (x64).

You can download **winlpcfg** from Emulex at: <http://www.emulex.com>. For additional information about the **winlpcfg** utility, refer to the documentation in the `sgdeploy\SGTKWinPE\Docs\winlpcfg` directory.

The **winlpcfg** utility uses the following syntax:

Table 45. *winlpcfg* syntax

Syntax	Description
<pre>winlpcfg.exe help winlpcfg.exe ? winlpcfg.exe help <i>command</i> winlpcfg.exe ? <i>command</i></pre>	<ul style="list-style-type: none"> • Displays winlpcfg Help. • To view a list of all available commands, enter: winlpcfg.e xe help or winlpcfg.e xe ?. • To view Help for a specific command, add the command name. For example: <pre>winlpcfg.exe help download</pre> or <pre>winlpcfg.exe ? download</pre>
<pre>winlpcfg.exe listwwn</pre>	<p>Lists all of the adapters installed in the system and shows the factory-assigned WWN, the nonvolatile WWPN, and the WWNN used to identify the adapter in the SAN.</p>

Table 45. winlpcfg syntax (continued)

Syntax	Description
winlpcfg.exe listhba	Lists the following information for all installed adapters in the system: <ul style="list-style-type: none"> • Adapter number • IEEE address assigned by the manufacturer • Firmware version • Adapter type • Possible mailbox errors
winlpcfg.exe readbootdevice n=adapter_number	Displays the WWN, LUN, and the topology in use for the indicated boot device.
winlpcfg.exe enableboot n=adapter_number i=index	Enables or disables the BootBIOS specified by the index number on the specified adapter.
winlpcfg.exe setbootdevice n=adapter_number w0= wwpn_word_0 w1=wwpn_word_2 l=lun t= topology	<ul style="list-style-type: none"> • Sets the boot device to the indicated adapter, WWPN, and topology. • Select: <ul style="list-style-type: none"> – 0 for Arbitrated Loop – 1 for Point to Point
winlpcfg.exe readaltboot n=adapter_number	<ul style="list-style-type: none"> • Displays the WWN and LUN of all possible alternate boot devices. • Up to seven alternate boot devices are supported.
winlpcfg.exe setaltboot i=index w0=wwpn_word_0 w1=wwpn_word_2 l=lun	<ul style="list-style-type: none"> • Specifies an alternate boot device. • You can set up to seven boot devices by specifying indexes from 1 to 7.

Examples

The following examples illustrate **winlpcfg** utility usage.

Note: While some of these examples are broken across multiple lines, when using **winlpcfg**, you must enter all of the parameters on a single line.

Table 46. winlpcfg usage examples

Example	Description
winlpcfg.exe help	Displays all available commands.
winlpcfg listwwn	Displays the WWNs of all adapters in the system
winlpcfg listhba	Lists all adapters in the system.
winlpcfg readbootdevice n=1	Displays the WWN, LUN, and topology for adapter number one.
winlpcfg enableboot n=6 i=1	Enables BootBios on adapter number 6.
winlpcfg setbootdevice n=1 w0=a1b2c3d4 w1=b946a4e8 l=46 t=0	Sets the boot device to adapter number one, LUN 46, with an Arbitrated Loop topology.
winlpcfg readaltboot n=1	Displays the WWN and LUN number of all possible alternate boot devices.
winlpcfg setaltboot n=1 i=1 w0=12345678 w1=a842b6 l=3	Sets the alternate boot device on adapter 1, LUN 3.

Chapter 8. Incorporating Scripting Toolkit with your existing process

Before using Scripting Toolkit procedures to perform deployments, use the **hwdetect** utility to determine if the combined process is being executed on hardware supported by the Scripting Toolkit.

You can then add appropriate branches in the batch files to use either the existing process or the Scripting Toolkit process.

For example, you might use the /s option of **hwdetect** to determine if the current system is an LenovoeServer, xSeries, or BladeCenter server:

```
hwdetect.exe /s
if errorlevel 1 goto NONLenovo
if errorlevel 0 goto Lenovo

:NONLenovo
rem Perform non-Lenovo equipment specific processing here.

:Lenovo
rem Perform Lenovo eServer or xSeries equipment specific processing here.
```

Chapter 9. Troubleshooting and support

Use this section to troubleshoot and resolve problems with ServerGuide Scripting Toolkit, Windows Edition.

Known problems and limitations

This section provides information and alternative solutions for known problems and limitations of ServerGuide Scripting Toolkit, Windows Edition.

Toolkit does not work on a network share drive or any drive that is not formatted using NTFS

Toolkit uses the Microsoft WindowsImagex.exe utility during the generation of the Windows PE ISO. The Imagex.exe utility fails during the creation of a deployment scenario if Toolkit is installed on a mapped network drive or a USB device that is formatted with anything other than the NTFS file system.

Unknown HID SYS device in Windows Device Manager

When a Lenovo Remote Supervisor Adapter-II is installed in a server, an unknown HID SYS device may appear in the Windows Device Manager list if the adapter OS Type is set to *Linux*. You can remove the unknown HID SYS device by completing the following steps after the OS installation:

1. Boot the machine and press **F1** to enter System Setup.
2. Navigate to **Advanced Setup**.
3. Navigate to **RSA II Setting**
4. Set the OS Type of the Remote Supervisor Adapter-II to **Other**.
5. Save and exit from System Setup.
6. Reboot the machine and start Windows.
7. Navigate to Device Manager.
8. Right-click the unknown HID SYS device in the Windows Device Manager list.
9. Select **Update Driver** from the menu.
10. Select **No, not at this time** to connect to Windows Update.
11. Click **Next**.
12. Click **Next**.
13. Click **Finish**.

Partitioning disks appears to hang at 0 percent complete

When using Scripting Toolkit to partition disks of 300 GB or larger, the process might appear to hang at 0 percent complete. The process is not hung. It could take 15-20 minutes for the partitioning process to complete.

Windows ComputerName must be alphanumeric

The *ComputerName* variable used for Windows installations must be alphanumeric, and must contain at least one letter. Valid values of *ComputerName* must be 15 characters or less.

If *ComputerName* does not meet these criteria, you will receive an error during unattended Windows 2008 installations saying:

```
Windows could not parse or process the unattended answer file for pass
(specialize).
```

```
The settings specified in the answer file cannot be applied.
```

```
The error was detected while processing settings for component
```

[Microsoft-Windows-Shell-Setup].

Limitations for RSA-II installations

You might encounter errors when using RSA-II to install Windows Server 2008 using an ISO image. These errors vary depending on the type of system getting the installation. In order to avoid these errors, when using RSA-II to install Windows Server 2008, use a mounted physical CD or DVD instead of an ISO image.

Slow network installations on System x3850 and x3950

Network installations using onboard Ethernet communications on the System x3850 and x3950 are very slow. You can avoid this problem by using an external Ethernet adapter.

Error for non existent PS/2 mouse in Device Manager

Installations of Windows Server 2008 can result in an error message being displayed in Device Manager for a non existent PS/2 mouse.

Because the error refers to a device that is not present, it can be safely ignored.

To fix this error on a System x 3250 server update the BIOS to level 1.42a or higher.

Savestat.cmd will not save to location 9 on xSeries 226 with BIOS PME170CUS

On the xSeries 226 with BIOS Level PME170CUS, **savestat.cmd** cannot save a value to byte nine in persistent storage.

BladeCenter HS22 unable to access bootable deployment media

When performing deployments to a BladeCenter HS22, type 1936 or 7870, the system might be unable to access the bootable media being used for the deployment. This error is caused by the system's inability to find the mounted media tray.

When this problem occurs, it is possible for all blades in the chassis to lose access to the media tray. To correct the problem, restart the BladeCenter Advanced Management Module (AMM) to restore access to the media tray.

To correct the problem, update the firmware for the system and the BladeCenter chassis to the latest level available.

If it isn't feasible to update the firmware, or if the problem persists, you can avoid it by using the Remote Control feature from the AMM to perform your deployment. For more information on this deployment method, please refer to "Installing Windows remotely" on page 28.

Yellow exclamation point for Microsoft ISATAP adapter

A yellow exclamation point icon might be displayed by Device Manager for the Microsoft ISATAP adapter after installing Windows Server 2008. In most cases, the adapter is functioning properly and you can continue to use the device normally.

For more information on this situation, see "" on page .

Bootting from SAN using Brocade Fibre Channel Adapters not supported

Bootting from SAN using the Brocade Fibre Channel Adapters listed in this document is not supported.

Default Fibre configuration not supported on Emulex HBAs

The Fibre HBA Toolkit variables: *Target WWNN*, *Target WWPN*, and *LUN* number must be set to configure the Primary, Alternate 1, Alternate 2 and Alternate 3 boot device settings. The default settings will NOT work on Emulex Fibre HBA adapters.

No x64 support for SCSI RAID controller configuration

Lenovo Scripting Toolkit supports only 32-bit RAID configuration of ServeRAID SCSI and LSI SCSI RAID controllers. If 64-bit RAID configuration is attempted, you might receive a "No controller found" message. To avoid this issue, use the x86 RAID configuration for the SCSI controllers.

The following controllers are affected by this issue:

- ServeRAID 4H
- ServeRAID 4Lx
- ServeRAID 4Mx
- ServeRAID 5i
- ServeRAID 6i/6i+
- ServeRAID 6M
- ServeRAID 7k
- LSI SCSI (1020/1030)

ServeRAID BR10i adapter not supported on iDataPlex dx360 M2 with 12 Bay Storage Chassis (Machine type 7321)

IBM Systems Director 6.11 Platform Agent installation fails on Windows 2008

When installing the platform agent on Windows Server 2008, it might fail with return code 400. This indicates that the target system does not include Windows Installer 4.5.

To resolve this issue, follow the instructions in Microsoft Knowledge Base article 942288: .

Booting from SAN is not supported for the QLogic 10Gb Dual Port CNA for IBM System x (42C1800)

When installing Windows using ServerGuide, disks show special ownership

When you install Windows 2008 using ServerGuide, ownership of the disk is assigned to **TrustedInstaller**. This is different from the default assignment of **Administrators** in a native Windows installation. To use the default assignment, edit the security property of the installation file to assign ownership to **Administrators**.

Getting help and technical assistance

If you need help, service, or technical assistance or just want more information about Lenovo products, you will find a wide variety of sources available from Lenovo to assist you. The topics in this section contain information about where to go for additional information about Lenovo products, what to do if you experience a problem with your xSeries or IBMIntelliStation system, and whom to call for service, if it is necessary.

Before you call

Before you call for assistance concerning a product, complete these steps to try to solve the problem yourself:

- Check all cables to make sure that they are connected.
- Check the power switches to make sure that the system is turned on.
- Use the troubleshooting information in your system documentation, and use the diagnostic tools that come with your system. Information about diagnostic tools is in the *Hardware Maintenance Manual and*

Troubleshooting Guide on the *LenovoSeries Documentation CD* or in the *IBMIntelliStation Hardware Maintenance Manual*.

- Go to the [Lenovo Support website](#) at to check for technical information, hints, tips, and new device drivers or to submit a request for information.

You can solve many problems without outside assistance by following the troubleshooting procedures that Lenovo provides in the online help or in the publications that are provided with your system and software. The information that comes with your system also describes the diagnostic tests that you can perform. Most xSeries and IntelliStation systems, operating systems, and programs come with information that contains troubleshooting procedures and explanations of error messages and error codes. If you suspect a software problem, see the information for the operating system or program.

Using the documentation

Information about your LenovoSeries or IBMIntelliStation system and preinstalled software, if any, is available in the documentation that comes with your system in a variety of formats: books, online books, readme files, and help files.

See the troubleshooting information in your system documentation for instructions for using the diagnostic programs. The troubleshooting information or the diagnostic programs might tell you that you need additional or updated device drivers or other software. Lenovo maintains pages on the Web where you can get the latest technical information and download device drivers and updates. To access these pages, go to and follow the instructions. Also, you can order publications through the IBM Publications Ordering System at .

Getting help and information on the Web

Support information for your Lenovo products, including supported options, is available at .

Support information for IBM IntelliStation is available at: .

Software service and support

Through [Lenovo Support Line](#), you can get telephone assistance, for a fee, with usage, configuration, and software problems with xSeries servers and appliances. For information about products that are supported by Support Line in your country or region, go to .

For more information about Support Line and other Lenovo services, go to , or go to for support telephone numbers.

Hardware service and support

You can receive hardware service through [Lenovo Services](#) or through your Lenovo reseller, if your reseller is authorized by Lenovo to provide warranty service. Refer to for support telephone numbers.

Appendix A. Best practices

This section provides best practices information for using ServerGuide Scripting Toolkit.

Building a customized installation image with a RAID adapter driver

This procedure describes the steps necessary for building a customized installation image with an additional RAID adapter driver.

- Step 1. Set up the source server, making sure to add files to the source tree. See “Adding files to the source tree” on page 5.
- Step 2. Add the RAID adapter driver to Windows PE. See “Adding Windows PE drivers” on page 17.
- Step 3. Add the RAID adapter driver to Windows installation. See “Adding Windows device drivers” on page 7.
- Step 4. Customize installation scenarios and create image. See “About customizing Windows installation scenarios” on page 24.

Using UXSPi to download updates

This procedure describes how to acquire firmware and driver updates for your Lenovo servers by using the UpdateXpress System Pack Installer (UXSPi) graphical user interface (GUI).

For more information about using UXSPi, refer to the *LenovoUpdateXpress System Pack Installer User's Guide* (lnvgy_utl_uxspi_x.xx_anyos_noarch), located in the `sgdeploy\SGTKWinPE\docs\uxspi\` folder where you installed Lenovo ServerGuide Scripting Toolkit, Windows Edition.

To acquire driver and firmware updates for your system with UXSPi in GUI mode, follow these instructions:

- Step 1. Navigate to the `sgdeploy\updates\uxsp` directory.
- Step 2. Double-click `lnvgy_utl_uxspi_x.xx_winsrvr_32-64.exe` to start UXSPi. The main UXSPi window opens.
- Step 3. Click **Next** to proceed to the Select Command window.
- Step 4. Select **Create a repository of updates**, and click **Next**.
- Step 5. Follow the instructions in the wizard to download the latest UXSPs or firmware updates.

Considerations for installing an OS on a multi-adapter system

This section describes the special considerations for installing an operating system on a multi-adapter system.

When you perform a Windows installation, ServerGuide Scripting Toolkit, Windows Edition attempts to install Windows to the first disk on the system presented by the **diskpart** command line utility.

Due to limitations of the system, when multiple storage adapters are present on the system, the first disk presented by the **diskpart** utility is not always the first disk on the system.

You can control which disk the operating system installs by using the `TK_Partition_DiskNum` variable in the Partitioning section of the `SGTKWinPE.ini` settings file. To determine the value for the disk you want to install to, complete RAID configuration and then run the utility **GetDiskData.cmd** to determine the available disks.

When a system contains a Fibre HBA that has been configured with a logical drive mapped to Logical Unit Number (LUN) 0, Toolkit uses this drive as the boot device. Therefore, Toolkit fails if a different drive is selected for the Windows installation. If you want to install to a different drive, disable the BIOS for the HBA before beginning your deployment.

Adding software applications for installation after first Autologin

This procedure describes how to add software applications to ServerGuide Scripting Toolkit as part of the post-OS installation stage.

You can add software applications by editing and customizing the post-OS installation script.

Note: This procedure is not suitable for use with interactive applications that require additional user input. Attempting to install applications that require user interaction can cause the post-OS installation stage to hang while waiting for user input.

Step 1. Add the software application files to `sgdeploy\SGTKWinPE\Scripts\Custom`. All of the files in this directory are copied to the installation image.

Step 2. Edit the installation script file `sgdeploy\SGTKWinPE\Scripts\Custom\CustomPostInstall.cmd` to include the commands that invoke the added software applications.

Configuring USB keys for deployment

This procedure describes how to create a bootable Windows PE USB key for any ServerGuide Scripting Toolkit, Windows Edition deployment.

Configuring a USB key for a Toolkit deployment requires the following:

- A system running one of the following Windows sessions:
 - Windows Vista
 - Windows Server 2008
 - Windows 7
 - Windows Server 2008 R2
 - Windows 2.1 PE
 - Windows 3.0 PE
 - Windows 4.0 PE
- A USB key with a storage capacity of at least 64 MB larger than your Windows PE image.

To configure your USB key for deployment, follow this procedure:

Step 1. Insert your USB key in the system.

Step 2. Open a command prompt.

Step 3. Use the **diskpart.exe** command to format the USB key as a single FAT32 partition spanning the entire key, and set the partition to active, as shown in the following example:

```
diskpart
select disk 1
clean
create partition primary
select partition 1
active
format fs=fat32
assign
```


exit

When the USB key has been formatted, you can copy the Toolkit image files to the USB key for deployment. After you format a USB key for Toolkit deployments, you can replace the deployment images on the key and use it for subsequent deployments without reformatting it.

Booting from a USB key

This procedure describes how to boot from a USB key.

To boot from a USB key, the key must be configured for Toolkit deployment. For more information about configuring a USB key for deployments, see: “Configuring USB keys for deployment” on page 120.

To boot from a USB key, use the Boot Menu program as follows:

- Step 1. Reboot the server.
- Step 2. When prompted, press F12 to enter the setup menu.
- Step 3. Select the USB device. For more information about booting from a USB key, consult the documentation for your system.

Configuring ServeRAID controllers with a 64-bit operating systems

The topics in this section describe how to configure ServeRAID-SCSI controllers in conjunction with the installation of a supported 64bit operating system.

ServeRAID-SCSI controllers use the **ipssend** utility to perform RAID configuration. This utility is not supported for 64-bit versions of Windows Preinstallation Environment. Therefore, when you combine RAID configuration of affected controllers with the installation of a supported 64bit operating system, you must follow the steps described in the following topics to create separate pieces of media for RAID configuration and operating system installation.

The following RAID controllers use the **ipssend** utility and are therefore affected by this limitation:

- ServeRAID 4H
- ServeRAID 4Lx
- ServeRAID 4Mx
- ServeRAID 5i
- ServeRAID 6i/6i+
- ServeRAID 6M
- ServeRAID 7k

Installing Windows Server 2008 x64 locally with a ServeRAID-SCSI configuration

This procedure describes how to install Windows Server 2008 x64 locally with a ServeRAID-SCSI configuration.

To perform this task, you need:

- Either two blank DVDs, or one blank CD and one blank DVD
- A source server with a properly configured source tree
- Media creation software and a DVD R/W drive

- Step 1. Because the **ipssend** utility is not supported under Windows PE, you must create two separate pieces of configuration media:
- RAID configuration media, which can be either a DVD or a CD, created using Windows PE x86.
 - Operating system installation media, which must be a DVD created using Windows PE x64.

Step 2. Create the RAID configuration disc.

- Label the disc and insert it into the appropriate R/W drive on the source server.
- Open a command window and change directory to `sgshare\sgdeploy\SGTKWinPE`.
- Enter the following command:
`SGTKWinPE.cmd ScenarioINIs\Local\Raid_Config_Only_x86.ini`

The ISO image of the RAID configuration disc is created at `sgdeploy\WinPE_ScenarioOutput\Local_RAID_Config_Only_x86\WinPE_x86.iso`, and the media creation software is started. Follow the instructions in your media creation software to create a disc using the ISO image.

Step 3. Create the Windows 2008 x64 deployment disc.

- Customize the unattend file: `sgshare\sgdeploy\SGTKWinPE\AnswerFiles\win2008_x64_EE.xml`
- Label the DVD and insert it in the drive.
- From the `sgshare\sgdeploy\SGTKWinPE` directory, issue the following command:
`SGTKWinPE.cmd ScenarioINIs\Local\Windows_version.ini`
 Where *Windows_version* is the .ini file that corresponds to the version of Windows Server 2008 that you are installing, as shown below:

Table 47. INI file names for **Windows Server 2008 x64** versions

Version	INI file
Windows Server 2008 x64 DataCenter Edition	Win2008_x64_DE.ini
Windows Server 2008 x64 Enterprise Edition	Win2008_x64_EE.ini
Windows Server 2008 x64 Standard Edition	Win2008_x64_EE.ini
Windows Server 2008 x64 Web Edition	Win2008_x64_WE.ini
Windows Server 2008 x64 without Hyper-V DataCenter Edition	Win2008_x64_DE_no_hv.ini
Windows Server 2008 x64 without Hyper-V Enterprise Edition	Win2008_x64_EE_no_hv.ini
Windows Server 2008 x64 without Hyper-V Standard Edition	Win2008_x64_SE_no_hv.ini

The ISO image is created in the location that corresponds to the version of Windows Server 2008 you are installing, as shown below.

Table 48. ISO file names for **Windows Server 2008 x64** versions

Version	INI file
Windows Server 2008 x64 DataCenter Edition	<code>sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_DE\WinPE_x64.iso</code>
Windows Server 2008 x64 Enterprise Edition	<code>sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_EE\WinPE_x64.iso</code>
Windows Server 2008 x64 Standard Edition	<code>sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_SE\WinPE_x64.iso</code>
Windows Server 2008 x64 Web Edition	<code>sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_WE\WinPE_x64.iso</code>

Table 48. ISO file names for Windows Server 2008 x64 versions (continued)

Version	INI file
Windows Server 2008 x64 without Hyper-V DataCenter Edition	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_DE_no_hv\WinPE_x64.iso
Windows Server 2008 x64 without Hyper-V Enterprise Edition	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_EE_no_hv\WinPE_x64.iso
Windows Server 2008 x64 without Hyper-V Standard Edition	sgdeploy\WinPE_ScenarioOutput\Local_Win2008_x64_SE_no_hv\WinPE_x64.iso

The media creation software is started. Follow the instructions in your media creation software to create a disc using the ISO image.

Step 4. Complete the deployment.

- a. Perform RAID configuration by booting the target server from the RAID configuration disc you created. When the RAID configuration is complete, the target server restarts.
- b. After the restart, remove the RAID configuration disc and boot the target system from the operating system installation disc you created to complete the deployment.

Installing Windows Server 2008 x64 over a network with a ServeRAID-SCSI configuration

This topic describes the prerequisites and steps for installing Windows Server 2008 x64 over a network with a ServeRAID-SCSI configuration.

To perform this task, you need:

- Two blank DVDs or CDs
- A source server with:
 - a properly configured source tree
 - media creation software
 - a drive capable of writing CDs or DVDs
- Network connectivity between the source and target servers

Step 1. Because the **ipssend** utility is not supported under Windows PE x64, you must create two separate pieces of configuration media:

- a. A RAID configuration disc, which can be either a DVD or a CD, created using Windows PE x86.
- b. An operating system installation disc, which can be either a DVD or a CD, created using Windows PE x64.

Step 2. Create the RAID configuration disc.

- a. Label the disc and insert it into the appropriate R/W drive on the source server.
- b. Open a command window and change directory to `sgshare\sgdeploy\SGTKWinPE`.
- c. Enter the following command:
`SGTKWinPE.cmd ScenarioINIs\Network\Raid_Config_Only_x86.ini`

The ISO image of the RAID configuration disc is created at:

`sgdeploy\WinPE_ScenarioOutput\Network_RAID_Config_Only_x86\WinPE_x86.iso`, and the media creation software is started. Follow the instructions in your media creation software to create a disc using the ISO image.

Step 3. Create the Windows Server 2008 x64 deployment disc.

- a. Customize the unattend file: `sgshare\sgdeploy\SGTKWinPE\AnswerFiles\win2008_x64_EE.xml`.

- b. Label the DVD and insert it into the drive.
- c. From the `sgshare\sgdeploy\SGTKWinPE` directory, issue the following command:

```
SGTKWinPE.cmd ScenarioINIs\Network\Windows_version.ini
```

Where *Windows_version* is the .ini file corresponding to the version of Windows Server 2008 you are installing, as shown below:

Table 49. INI file names for Windows Server 2008 x64 versions

Version	INI file
Windows Server 2008 x64 DataCenter Edition	Win2008_x64_DE.ini
Windows Server 2008 x64 Enterprise Edition	Win2008_x64_EE.ini
Windows Server 2008 x64 Standard Edition	Win2008_x64_SE.ini
Windows Server 2008 x64 Web Edition	Win2008_x64_WE.ini
Windows Server 2008 x64 without Hyper-V DataCenter Edition	Win2008_x64_DE_no_hv.ini
Windows Server 2008 x64 without Hyper-V Enterprise Edition	Win2008_x64_EE_no_hv.ini
Windows Server 2008 x64 without Hyper-V Standard Edition	Win2008_x64_SE_no_hv.ini

The ISO image is created in the location that corresponds to the version of Windows Server 2008 that you are installing, as shown below.

Table 50. ISO file names for Windows Server 2008 x64 versions

Version	INI file
Windows Server 2008 x64 DataCenter Edition	sgdeploy\WinPE_ScenarioOutput\ Network_Win2008_x64_DE\WinPE_x64.iso
Windows Server 2008 x64 Enterprise Edition	sgdeploy\WinPE_ScenarioOutput\ Network_Win2008_x64_EE\WinPE_x64.iso
Windows Server 2008 x64 Standard Edition	sgdeploy\WinPE_ScenarioOutput\ Network_Win2008_x64_SE\WinPE_x64.iso
Windows Server 2008 x64 Web Edition	sgdeploy\WinPE_ScenarioOutput\ Network_Win2008_x64_WE\WinPE_x64.iso
Windows Server 2008 x64 without Hyper-V DataCenter Edition	sgdeploy\WinPE_ScenarioOutput\ Network_Win2008_x64_DE_no_hv\WinPE_x64.iso
Windows Server 2008 x64 without Hyper-V Enterprise Edition	sgdeploy\WinPE_ScenarioOutput\ Network_Win2008_x64_EE_no_hv\WinPE_x64.iso
Windows Server 2008 x64 without Hyper-V Standard Edition	sgdeploy\WinPE_ScenarioOutput\ Network_Win2008_x64_SE_no_hv\WinPE_x64.iso

The media creation software is started. Follow the instructions in your media creation software to create a disc using the ISO image.

Step 4. Complete the deployment

- a. Perform RAID configuration by booting the target server from the RAID configuration disc you created. When the RAID configuration is complete, the target server will restart.
- b. After the restart, remove the RAID configuration disc and boot the target system from the operating system installation disc you created to complete the deployment.

Network installation of Windows Server 2008 x64 with a ServeRAID-SCSI configuration using a USB key

This topic describes the prerequisites and steps for installing Windows Server 2008 x64 with a ServeRAID-SCSI configuration using a USB key.

To perform this task, you need:

- A USB key with a bootable WinPE image
- A source server with a properly configured source tree

Step 1. Because the **ipssend** utility is not supported under Windows PE x64, you must create two separate sets of deployment files:

- a. RAID configuration files, created using Windows PE x86
- b. Operating system installation files, created using Windows PE x64

Step 2. Create the RAID configuration files.

- a. Open a command window and change directory to `sgshare\sgdeploy\SGTKWinPE`.
- b. Enter the following command:
`SGTKWinPE.cmd ScenarioINIs\Network\Raid_Config_Only_x86.ini`

The RAID configuration files are created in:
`sgdeploy\WinPE_ScenarioOutput\Network_RAID_Config_Only_x86\WinPE_x86.iso`

Step 3. Create the Windows Server 2008 x64 deployment disc.

- a. Customize the unattend file: `sgshare\sgdeploy\SGTKWinPE\AnswerFiles\win2008x64.txt`.
- b. Label the DVD and insert it into the drive.
- c. From the `sgshare\sgdeploy\SGTKWinPE` directory, issue the following command:

`SGTKWinPE.cmd ScenarioINIs\Network\Windows_version.ini`

Where *Windows_version* is the .ini file corresponding to the version of Windows Server 2008 you are installing, as shown below:

Table 51. INI file names for **Windows Server 2008 x64** versions

Version	INI file
Windows Server 2008 x64 Enterprise Edition	Win2008_x64_EE.ini
Windows Server 2008 x64 Standard Edition	Win2008_x64_SE.ini

The ISO image is created in the location corresponding to the version of Windows Server 2008 that you are installing, as shown below.

Table 52. ISO file locations for **Windows Server 2008 x64** versions

Version	ISO file location
Windows Server 2008 x64 Enterprise Edition	<code>sgdeploy\WinPE_ScenarioOutput\Network_Win2008_x64_EE\WinPE_x64.iso</code>
Windows Server 2008 x64 Standard Edition	<code>sgdeploy\WinPE_ScenarioOutput\Network_Win2008_x64_SE\WinPE_x64.iso</code>

The media creation software is started. Follow the instructions in your media creation software to create a disc using the ISO image.

Step 4. Complete the deployment

- a. Copy the RAID configuration and operating system deployment files from the locations above.

- b. Boot the target server by using the USB key and follow the on-screen prompts to complete the deployment.

Installing Windows remotely

This procedure describes how to perform an unattended remote installation of Windows using an RSA II adapter, Integrated Management Module (IMM), BladeCenter Management Module, or BladeCenter Advanced Management Module.

To perform this task you need:

- An ISO image or a physical CD or DVD created according to the instructions in the previous sections for local and network unattended installation scenarios for Windows .
- Network connectivity to the Remote Supervisor Adapter II, Integrated Management Module, BladeCenter Management Module, or BladeCenter Advanced Management Module from the source system.
- A properly configured browser on the source system:
 - Ensure that the Java plug-in is installed. The control panel will reflect a Java Control Panel icon if a proper plug-in is installed. In addition, the Java plug-in may require that browser caching be disabled for the remote interface to work properly.
 - Disable pop-up blocking in the web browser on the source server.
- For network deployment scenarios, you must have:
 - A source system that contains a properly set up source tree. (For more information, see “Configuring the source server” on page 4.)
 - Network connectivity from the target server to the source system shared directory.
- For Windows Server 2008 installations, ISO images are not supported. You must use a physical CD or DVD.
- Network deployment scenarios take significantly less time than local deployments.
- Ensure that there is no disc is in the CD or DVD-ROM drive of your target server.
- If you encounter problems, try updating the RSA II, IMM, MM, or AMM firmware to the latest level.

To deploy your installation image through RSA II, MM, or AMM, complete these steps:

- Step 1. Open a web browser on the source system.
- Step 2. Log in to the adapter with your user ID and password, and select a timeout value for your session. Unless you select **no timeout**, your remote session might time out before your deployment is complete.
- Step 3. Click **Continue** to proceed to the task menu.
- Step 4. From the Task menu, select **Remote Control**.
- Step 5. Click the appropriate link or button to start a remote control session. Your remote control session opens in a new window. If the new window does not open, make sure all popup blockers are disabled, and retry. If the Remote Disk and Remote Console sections do not appear in the window within a few minutes, try clicking in the window. If they still fail to load, check to make sure the Java plug-in is installed.
- Step 6. For the MM and AMM, select the desired target blade in both the **Media Tray** and **KVM Owner** lists.
- Step 7. Select your deployment media in the **Available Resources** box.
 - For a CD or DVD, select the drive containing your disc, then add it the list of resources by clicking the **>>** box.

- For an image, click **Select Image**, click the >> box, then browse to your image file. Select the image file, and click **Open**.

Step 8. Select the deployment media again in the **Selected Resources** box.

Step 9. Select **Write Protect**.

Step 10. Click the **Mount** button to mount the disc or image.

Step 11. Reboot the target system.

Step 12. When the target system reboots it will load the deployment media. Use the interactive Remote Console video window to respond to any prompts. If you are performing the RAID configuration only portion of a Windows Server 2008 x64 deployment, the target server will restart automatically. When it does, wait for Windows PE to exit completely and then unmount the drive to "eject" the deployment media. Then select your media again as above and proceed with the installation.

Step 13. The Network Operating System Installation instructions prompt will be displayed and explain how to continue to the unattended portion of the operating system installation.

Step 14. Press a key to reboot the system.

Step 15. After Windows PE has exited completely and the target server is booting, unmount the drive to eject the deployment media.

The target server boots from the hard disk to continue the operating system deployment. You can monitor progress through the Remote Console.

Configuring the source server

This topic provides detailed information about configuring the ServerGuide Scripting Toolkit, Windows Edition source server.

ServerGuide Scripting Toolkit, Windows Edition comes configured to use a set of defaults for access to the source server. If you do not want to use these defaults, you must change them in the .ini files used by the Toolkit, located in the sgdeploy\SGTKWinPE\ScenarioINIs directory.

Toolkit uses the following default settings:

- Server name: **sgtksrv**
- Shared directory name: **sgshare**
- User ID and password: **sgtk0clnt**

These settings are only necessary for network share deployments. For more information about changing these settings, see "Customizing source server settings" on page 15.

This procedure lists the steps for configuring the source server.

Attention: A fully automated ServerGuide Scripting Toolkit CD, DVD, or USB solution has the ability to re-partition and reformat a system without warning. For best results and to avoid accidental loss of data, remove the CD, DVD, or USB key from the system boot order on the source server as soon as it is no longer required.

Step 1. To use the source server for network deployments, share the Toolkit folder on the network. For example: C:\sgshare. For local deployments, it is not necessary to share the Toolkit folder.

Step 2. Verify that the C:\sgshare folder is accessible to all the systems that require access on the network.

Step 3. Use the Toolkit Configuration Utility to add files to the source tree, by starting from a command prompt or by double-clicking it. TKConfig.exe is located in the sgdeploy\tkconfig directory. For more information, see "Adding files to the source tree" on page 5.

Step 4. Include or update the Windows answer file. For more information, see "Using the Windows answer file for unattended installations" on page 5.

- Step 5. Create a user ID and password that has access to the Toolkit folder. Toolkit uses the default value of `sgtk0clnt` for both the ID and the password.
- Step 6. If you choose to use a different ID and password, you must change the ID and password values in the Network Settings section of the Toolkit Settings `.ini` file.
- Step 7. If deployment scenarios will be built on a different system from the source server, the required System Enablement Packs (SEPs) must be installed on the source server before you perform a deployment. For more information about SEPs, see Chapter 5 “Downloading and installing System Enablement Packs (SEPs)” on page 51.

Customizing source server settings

This topic describes the variables used by ServerGuide Scripting Toolkit, Windows Edition and explains how to customize them for your server.

ServerGuide Scripting Toolkit, Windows Edition uses variables to set the following source server attributes:

- Server name
- Shared folder name
- User ID and password
- Network share drive assignment

These values are located in the [Network Share Settings] section of the `.ini` files used by Toolkit.

Even if you accept the other defaults as part of the installation, it might be necessary to change the server name variable to match your server.

The Toolkit also uses variables to set the following TCP/IP configuration attributes:

- DHCP
- Client IP address and subnet mask
- Gateway address
- Windows PE firewall

These values are located in the [Network TCP/IP Settings] section of the `.ini` files used by Toolkit.

The default value for DHCP is `Yes`, indicating that Toolkit will use DHCP to obtain IP addresses. If you use the default value for DHCP, it is not necessary to modify any of the other IP address variables. If you choose to use static addresses, you must check the other IP address variables to ensure that they are appropriate for your network configuration.

For more information about these variables, see the appropriate sections in Table 10 “User definable settings for `SGTKWinPE.cmd`” on page 35.

SGTKWinPE user settings

Use the information in this topic to customize an image produced by the **SGTKWinPE.cmd** command.

The `SGTKWinPE settings .ini` file provides input to the `SGTKWinPE.cmd` file and controls whether options and processes of ServerGuide Scripting Toolkit are enabled or disabled. This allows you to create a wide array of Windows PE solutions. For example, a default image from Microsoft that simply boots to a command prompt, a DVD for local deployment, or a CD for network deployment.

Toolkit provides example `SGTKWinPE settings .ini` files for deploying each supported operating system for both local and network deployments in the `sgdeploy\SGTKWinPE\ScenarioINIs` directory.

Before modifying these files or any other Toolkit files, create backup copies of the files you are modifying.

Table 53 “User definable settings for SGTkWinPE.cmd” on page 129 describes the user definable parameters in the .ini files used by **SGTKWinPE.cmd**.

Note: Note that while some examples are broken across multiple lines for formatting reasons, when using these settings, you must include all the information for each variable on a single line.

Table 53. User definable settings for SGTkWinPE.cmd

Setting	Description
[SGTKWinPE Build Settings]	
TK_Build_ScenarioOutputDirectory	<p>Specifies the name of the directory to build the scenario in. Multiple scenarios of the same type can be maintained by changing this value at the start of each build, for example by appending a number.</p> <p>Default: blank</p>
TK_Build_WinPE_Arch_Type	<p>The architecture of the Windows PE image to create.</p> <p>Valid values are: x86, x64</p> <p>Default:</p> <ul style="list-style-type: none"> • x86 for Windows Server 2008 x86 deployments • x64 for Windows Server 2008 x64 deployments
TK_Build_WinPEPackages	<p>Lists additional packages to install to the Windows PEImage through the peimg /install=pckg command. The Toolkit will automatically install the following packages: *Scripting*, *MDAC*, *WMI*.</p> <p>Valid values are described in the Advanced Installation Kit documentation.</p> <p>Default: blank</p>
TK_Build_PEImgPrep	<p>Reduces the size of the Windows PE image by removing packages that are not installed from the image to reduce the size of the image.</p> <p>Valid values are: Yes, No</p> <p>Default: No</p> <p>Note: This setting applies only to the Automated Installation Kit (AIK) for Windows Vista SP1 and Windows Server 2008. If you are using the AIK for Windows 7 and Windows Server 2008 R2, this setting is ignored.</p>
TK_Build_ForceWinPEBoot	<p>Forces the Windows PE CD or DVD to boot without prompting when a file system is present.</p> <p>Ordinarily, when a file system is present when Windows PE boots, it prompts for user input and allows 8 seconds for the user to press a key. The ForceWinPEBoot option disables this feature and forces the Windows PE CD or DVD to always boot.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_Build_InstallWinPEDrivers	<p>Installs the LenovoWindows PE driver set as part of the Windows PE image.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_Build_EnableSGToolkitEnvironment	<p>Enables the ServerGuide Scripting Toolkit environment in the Windows PE image. If the environment is disabled, then no Toolkit files or environment are copied to the Windows PE image.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_Build_CreateISO	<p>Automatically create the ISO image.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_Build_StartISO	<p>Starts the software associated with ISO files. Typically this will start the CD or DVD burning process for the created ISO or RamdiskISO file. The RamdiskISO file takes precedence if it exists.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_Build_SuppressPrompts	<p>Suppresses interactive prompts during SGTkWinPE script execution.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_Build_DisplayBuildSettings	<p>Display the Toolkit settings from the input SGTkWinPE settings .ini file during SGTkWinPE script execution.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
[Toolkit Environment Settings]	
TK_Environment_ToolkitFilesLocation	<p>Defines whether the Scripting Toolkit binary and script files are to be copied to the local bootable media or run live over the network from the Toolkit source server.</p> <p>Valid values are: Local, Network</p> <p>Default:</p> <ul style="list-style-type: none"> • Local for local deployment scenarios • Network for network deployment scenarios

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_Environment_PerformToolkitProcesses	<p>Automatically perform Toolkit processes after booting the Windows PE CD or DVD (depending on the values of the individual process settings).</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Yes</p> <ul style="list-style-type: none"> • If set to No, the Toolkit processes may be run manually. • If set to Disabled, the Toolkit processes can not be run manually for local scenarios since the required files are not copied to the DVD.
TK_Environment_DisplayToolkitSettings	<p>Display the relevant Toolkit settings from the SGTkWinPE settings .ini file during deployment scenario execution.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
[Toolkit Process Settings]	
TK_Process_DataFilesLocation	<p>Specifies whether the NOS, driver files and PRAID policy files should be copied to the local bootable media or obtained over the network from the Toolkit source server (local deployment versus network deployment).</p> <p>Valid values are: Local, Network</p> <p>Default:</p> <ul style="list-style-type: none"> • Local for local deployment scenarios • Network for network deployment scenarios
TK_Process_PerformRAIDConfiguration	<p>Automatically perform the RAID configuration task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Yes</p> <p>If set to Disabled, the files required for local RAID configuration scenarios are not copied to the DVD.</p>
TK_Process_PerformFibreConfiguration	<p>Automatically perform the Fibre configuration task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Disabled</p>
TK_Process_PerformASUConfiguration	<p>Automatically perform the ASU configuration task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Disabled</p> <p>If this variable is set to Disabled, the files required for local ASU configuration scenarios are not copied to the DVD.</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_Process_PerformPartitioning	<p>Automatically perform the partitioning task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Yes</p> <p>If set to Disabled, the files required for local partitioning scenarios are not copied to the DVD.</p>
TK_Process_PerformNOSInstallation	<p>Automatically perform the NOS Installation task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Yes</p> <p>If set to Disabled, the files required for local NOS installation scenarios are not copied to the DVD.</p>
TK_Process_AutomaticallyRestart	<p>Specifies that the system be restarted after the selected tasks are complete.</p> <p>Valid values are: Yes, No</p> <p>Default: No</p>
TK_Process_AutoStartToolkitProcesses	<p>Enables automatic start of the Toolkit processes. Set this value to Yes to have the Toolkit processes start automatically when the Windows PE CD is booted without prompting the user to press a key.</p> <p>Attention: If the RAID or partitioning processes are set to Yes, this will re-partition or reformat the hard drive automatically without warning when the CD or DVD is booted.</p> <p>Valid values are: Yes, No</p> <p>Default: No</p>
TK_Process_AutoStartTimer	<p>Automatically starts countdown timer in seconds.</p> <p>Default: 60</p>
TK_Process_AutoContinueToolkitProcesses	<p>Enables automatic continuation of Toolkit processes during required reboots. Turning this off causes the user to be prompted to continue the Toolkit processes when Windows PE is rebooted from CD or DVD. Useful for debugging.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
[PRAID Settings]	
TK_PRAID_PolicyFile	<p>Specifies the PRAID policy file to use for the configuration. This value should not contain any path information. The specified PRAID policy file must exist in the SGTkWinPE\PolicyFiles directory.</p> <p>Default: RAIDAUTO.INI</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_PRAID_Flags	<p>Modifies the behavior of RAID configuration or replication.</p> <p>Valid values are: blank, /r:n, /e1, /e2, /e3, /v:n where <i>n</i> is 1, 3, or 5. See “praid.exe” on page 57 for more information.</p> <p>Default: /r</p>
[Fibre Settings]	
TK_FIBRE_COUNT	<p>Specifies the number of HBA ports to configure.</p> <p>Valid values are 1<i>n</i>, where <i>n</i> is the number of HBA ports available.</p> <p>This variable affects the use of the following variables:</p> <ul style="list-style-type: none"> • TK_FIBRE_N_HBA_ID • TK_FIBRE_N_BOOT_DISABLE • TK_FIBRE_N_BOOT_PRIM • TK_FIBRE_N_BOOT_ALT1 • TK_FIBRE_N_BOOT_ALT2 • TK_FIBRE_N_BOOT_ALT3 <p>Where <i>N</i> is the HBA number to be configured.</p> <p>Note: You must complete one of each of these variables for every HBA port you configure. So if TK_FIBRE_COUNT=2, you must complete one set of these variables for the first port and one for the second.</p>
TK_FIBRE_N_HBA_ID	<p>Identifies the QLogic/Emulex HBA to be configured, where <i>N</i> is the HBA number to be configured.</p> <p>Valid values are:</p> <p><i>hba_instance</i></p> <p>the instance number of an HBA port. Valid values are integers from 0 to <i>n</i>-1, where <i>n</i> is the number of HBAs in the system.</p> <p>For example, to configure HBA instance 0: TK_FIBRE_1_HBA_ID=0.</p> <p><i>hba_wwpn</i></p> <p>The World Wide Port Name of an HBA port, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>.</p> <p>For example, to configure HBA: 90-87-AA-BB-65-34-BB-E0: TK_FIBRE_1_HBA_ID= 90-87-AA-BB-65-34-BB-E0</p> <p>Default: 0</p> <p>Identifies the Brocade HBA to be configured, where <i>N</i> is the HBA number to be configured.</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
	<p>Valid values are:</p> <p><i>hba_instance</i></p> <p>the instance number of an HBA port. The valid format is <i>N/P</i>, where <i>N</i> is the adapter number from 1 to <i>N</i>, and <i>P</i> is the port number from 0 to p-1.</p> <p>For example: TK_FIBRE_1_HBA_ID=1/0.</p> <p><i>hba_wwpn</i></p> <p>the World Wide Port Name of an HBA port, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>.</p> <p>For example, to configure HBA: 90-87-AA-BB-65-34-BB-E0: TK_FIBRE_1_HBA_ID= 90-87-AA-BB-65-34-BB-E0</p> <p>Default: 0</p>
TK_FIBRE_N_BOOT_DISABLE	<p>Disables the selected current boot device settings on the specified HBA port, where <i>N</i> is the HBA number to be configured.</p> <p>Valid values are</p> <p>No Does not clear or disable any boot settings.</p> <p>All Disables the primary and all alternate boot settings: Prim, Alt1, Alt2, and Alt3.</p> <p>Prim Disables only the primary boot setting.</p> <p>Alt1 Disables the Alternative 1 boot setting.</p> <p>Alt2 Disables the Alternative 2 boot setting.</p> <p>Alt3 Disables the Alternative 3 boot setting.</p> <p>Default: No</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
<p>TK_FIBRE_N_BOOT_PRIM = <i>target_wwn</i><i>target_wwpn</i><i>lun_id</i></p>	<p>Defines the primary boot target settings, where <i>N</i> is the HBA number to be configured, and:</p> <ul style="list-style-type: none"> • <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>lun_id</i> is the Logical Unit Number of a device. <p>Default: 0 0 0</p> <p>Example: TK_FIBRE_1_BOOT_PRIM= BB-CC-AA-BB-65-34-BB-F1 BB-CC-AA-BB-FF-34-BB-F1 9</p>
<p>TK_FIBRE_N_BOOT_ALT1 = <i>target_wwn</i><i>target_wwpn</i><i>lun_id</i></p>	<p>Configures the operating system to use the indicated target as the first alternate boot device, where <i>N</i> is the HBA number to be configured, and</p> <ul style="list-style-type: none"> • <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>lun_id</i> is the Logical Unit Number of a device. <p>Default: blank</p> <p>Example: TK_FIBRE_1_BOOT_ALT1= BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5</p>
<p>TK_FIBRE_N_BOOT_ALT2 = <i>target_wwn</i><i>target_wwpn</i><i>lun_id</i></p>	<p>Configures the operating system to use the indicated target as the second alternate boot device, where <i>N</i> is the HBA number to be configured, and</p> <ul style="list-style-type: none"> • <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>lun_id</i> is the Logical Unit Number of a device. <p>Default: blank</p> <p>Example: TK_FIBRE_1_BOOT_ALT2= BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_FIBRE_N_BOOT_ALT3 = <i>target_wwn</i> <i>target_wwpn</i> <i>lun_id</i>	<p>Configures the operating system to use the indicated target as the third alternate boot device, where <i>N</i> is the HBA number to be configured, and</p> <ul style="list-style-type: none"> • <i>target_wwn</i> is the World Wide Node Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>target_wwpn</i> is the World Wide Port Name of a device, in the format <i>xx-xx-xx-xx-xx-xx-xx-xx</i> or <i>xxxxxxxxxxxxxxxx</i>. • <i>lun_id</i> is the Logical Unit Number of a device. <p>Default: blank</p> <p>Example: TK_FIBRE_1_BOOT_ALT3= BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5</p>
[ASU Settings]	
TK_ASU_File	<p>Specifies the ASU configuration file to use. This value cannot contain any path information. The specified ASU configuration file must exist in the SGTkWinPE\ASUFiles directory.</p> <p>Default: default_settings.ini</p>
[Partition Settings]	
TK_Partition_DiskNum	<p>Specifies the disk number on which to create new partition.</p> <p>Valid values are the disk numbers found by diskpart.exe.</p> <p>Default: AUTO</p> <p>Note: The AUTO setting is the first disk on the system.</p>
TK_Partition_Size	<p>Specifies the partition size in MB.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • Max • <i>number</i> <p>Max indicates to use all available disk space.</p> <p>Default: Max</p>
TK_Partition_FileSystem	<p>Specifies the file system type to use when formatting the drive.</p> <p>Valid values are: NTFS, FAT32</p> <p>Default: NTFS</p> <p>Note: NTFS is the only valid value for Windows Server 2008 installations.</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_Partition_SR_Size	<p>Specifies the partition size, in MB, for a System Reserved Partition.</p> <p>The System Reserved Partition is a primary active partition created during the partitioning step. The BitLocker Drive Encryption function requires an active partition and formatted NTFS.</p> <p>Valid values are integers greater than 108. The default is 108 MB.</p> <p>k2 Note: This setting is supported only for Windows Server 2008 R2. This setting is ignored when booting WinPE in native uEFI mode. For more information about BitLocker Drive Encryption, see: http://technet.microsoft.com/en-us/library/cc731549%28WS.10%29.aspx</p>
[NOS Installation Settings]	
TK_NOS_NetworkOperatingSystem	<p>Specifies the NOS to use for the deployment. This must be a valid directory name within the sgdeploy\os directory in the Toolkit source server.</p> <p>Default: Varies per scenario based on the NOS being deployed. For example, for Windows Server 2008 Enterprise Edition x86 deployments, the .ini file name is Win2008_x86_EE.ini. The corresponding OS directory in the .ini file is Win2008_x64.</p>
TK_NOS_AnswerFile	<p>Specifies the answer file to use for the deployment. This must be a valid file name within the SGTkWinPE\AnswerFiles directory in the Toolkit source server.</p> <p>Default:</p> <ul style="list-style-type: none"> • win2008.xml for Windows Server 2008 x86 • win2008x64.xml for Windows Server 2008 x64 • win2012x64.xml for Windows Server 2012 x64
TK_NOS_DeploymentDriverLibrary	<p>Specifies the deployment driver library to use for the deployment.</p> <p>Valid values are: Auto, <i>dirname</i></p> <p>If set to Auto, Toolkit searches all directories within the sgdeploy\drvs directory and selects the newest DDL present that supports the machine and OS being deployed. If <i>dirname</i> is specified, this must be a valid directory name within the sgdeploy\drvs directory in the Toolkit source server. For example, w23_drv would be specified for sgdeploy\drvs\w23_drv.</p> <p>Default: Auto</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_NOS_PerformDirectorAgentInstallation	<p>Automatically performs the Director Agent application installation task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Disabled</p> <p>If set to Disabled, the files required to install the Director Agent are not copied to the DVD for local deployment scenarios.</p>
TK_NOS_PerformPostOSInstallUXSPUpdates	<p>Automatically performs the post-OS install UXSP updates task.</p> <p>Valid values are: Yes, No, Disabled</p> <p>Default: Yes</p> <p>If set to Disabled, the files required to install UpdateXpress System Packs are not copied to the DVD for local deployment scenarios.</p>
TK_NOS_ProductKey	<p>Specifies the product key to be used when using the answer files provided with the Toolkit.</p> <p>Default: blank</p>
TK_NOS_Win2008_ImageName	<p>Sets the Image Name within the Windows 2008 installation image. This variable selects the Windows 2008 Edition.</p> <p>Valid values are:</p> <ul style="list-style-type: none"> • Windows Longhorn SERVERSTANDARD • Windows Longhorn SERVERSTANDARDV • Windows Longhorn SERVERDATACENTER • Windows Longhorn SERVERDATACENTERV • Windows Longhorn SERVERENTERPRISE • Windows Longhorn SERVERENTERPRISEV • Windows Longhorn SERVERWEB • Windows Server 2008 R2 SERVERSTANDARD • Windows Server 2008 R2 SERVERDATACENTER • Windows Server 2008 R2 SERVERENTERPRISE 7 • Windows Server 2008 R2 SERVERWEB <p>This value is set by the scenario .ini file.</p> <p>Note: To install the core version of an edition, append CORE to the image name. For example, to install the core version of Windows Server 2008 Standard Edition, use the value Windows Longhorn SERVERSTANDARDCORE.</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_NOS_Win2008_UILanguage	<p>Specifies the language to install from the Windows 2008 installation image.</p> <p>The default for this value is en-us, which allows the unattended installation of English copies of Windows Server 2008. To perform an unattended installation of Windows Server 2008 images in other languages, change this setting to the value for the language you want to install, for example ja-JP for Japanese.</p> <p>See the lang.ini file from the Windows 2008 media or the Windows Automated Installation Kit documentation for valid values for this variable.</p> <p>Default: en-us</p>
[Director Agent Settings]	
TK_DirAgent_DirectorAgent	<p>Specifies the path to the Systems Director Agent application files within the sgdeploy\apps directory.</p> <p>Default: dwin</p>
[UXSP Settings]	
TK_UXSP_FilesLocation	<p>Specifies whether the UpdateXpress System Packs should be copied to the local bootable media or obtained over the network from the Toolkit source server or from the specified remote server.</p> <p>Valid values are: Local, Network</p> <p>Default:</p> <ul style="list-style-type: none"> • Local for local deployment scenarios • Network for network deployment scenarios
TK_UXSP_MakeLocal_Copy	<p>Specifies whether or not the UXSP files should be copied to the target server before they are installed. If TK_UXSP_FilesLocation = Network, the files will be copied to the C:\LenovoSGTK\uxsp directory. If TK_UXSP_FilesLocation = Local, the UXSP files are always copied to the target server before they are installed.</p> <p>Valid values are: Yes, No</p> <p>Default: No</p>
TK_UXSP_RemoteServer	<p>If TK_UXSP_FilesLocation = Network, specifies the network path to the shared directory containing the UXSP files. If blank (no value is specified), then the UXSPs are obtained from the Toolkit source server.</p> <p>Example value: \\MyServerName\UXSPsDirectory</p> <p>Default: blank</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_UXSP_RemoteServer_UserName	<p>Specifies the user name and optionally the domain name with which to access the server specified with TK_UXSP_RemoteServer.</p> <p>Valid syntax:</p> <ul style="list-style-type: none"> • <i>UserName</i> • <i>UserName@DomainName</i> • <i>DomainName\UserName</i> <p>Default: blank</p>
TK_UXSP_RemoteServer_Password	<p>Specifies the password that corresponds to the user name specified with TK_UXSP_RemoteServer_UserName.</p> <p>Default: blank</p>
TK_UXSP_ApplyLatest	<p>Specifies whether UXSPi should apply the latest updates to the target system if no UXSPfiles are found for the target system. Set this variable to Yes to force the UpdateXpress System Pack Installer (UXSPi) to apply the latest updates to the target system if no UXSP files are found for that system.</p> <p>Valid values: Yes, No</p> <p>Default: No</p> <p>Note: In order for individual updates to be installed properly, any XML files included with the update must have lowercase file extensions (for example, xml not XML).</p>
TK_UXSP_UXSPIUpdateFlags	<p>Specifies user-provided command line arguments for processing by the UpdateXpress System Pack Installer (UXSPi) in Update mode. To provide command line arguments to be processed by UXSPi, set this variable to the command line arguments.</p> <p>See “UpdateXpress System Pack Installer” on page 56 for a list of command line arguments to use with UXSPi in Update mode.</p> <p>Default: blank</p>
[Network TCP/IP Settings]	
TK_NetTCPIP_UseDHCP	<p>Specifies whether the Windows PE networking settings should use DHCP. If this option is turned on, then the static specifications for network settings below are disabled automatically by Windows PE.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_NetTCPIP_ClientIPAddress	<p>Specifies the client IP address to use for the networking Windows PE environment.</p> <p>Default: 192.168.0.131</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_NetTCPIP_Gateway	<p>Specifies the Gateway IP address to use for the networking Windows PE environment.</p> <p>Default: 192.168.0.2</p>
TK_NetTCPIP_WinPEFirewall	<p>Specifies whether the Windows PE firewall should be activated.</p> <p>Valid values are: Yes, No</p> <p>Default: Yes</p>
TK_NetTCPIP_Identifier	<p>Specifies the network adapter to configure with static IP settings</p> <p>Default: Local Area Connection</p>
[Network Share Settings]	
TK_NetShare_ServerName	<p>Specifies the Toolkit source server computer name.</p> <p>Valid values are: either the host name or IP address of the Toolkit source server.</p> <p>Default: sgtksrv</p>
TK_NetShare_ShareName	<p>Specifies the Toolkit source server shared directory name.</p> <p>Default: sgshare</p>
TK_NetShare_UserName	<p>Specifies the user name, and optionally the domain name, to use when establishing a network connection to the Toolkit source server.</p> <p>Valid syntax:</p> <ul style="list-style-type: none"> • <i>UserName</i> • <i>UserName@DomainNameDomainName\UserName</i> <p>Default: sgtk0clnt</p>
TK_NetShare_Password	<p>Specifies the password to use when establishing a network connection to the Toolkit source server.</p> <p>Default: sgtk0clnt</p>
TK_NetShare_ShareDriveAssignment	<p>Specifies the Share Drive Assignment to use when establishing a network connection to the Toolkit Source Server.</p> <p>Default the default network share drive is Y:</p>
[SGTKWinPE Paths]	
TK_Path_WinPE_Output	<p>Specifies the path for the Windows PE Build Output directory.</p> <p>Default: %TK_Path_SGDeploy%\WinPE_ScenarioOutput</p>

Table 53. User definable settings for SGTkWinPE.cmd (continued)

Setting	Description
TK_Path_WAIK_Source	<p>This variable has been deprecated. The AIK information is found automatically in the registry. You can uncomment this variable and use it to bypass the registry check, but it is not required.</p> <p>Default (if used): %ProgramFiles%\Windows AIK</p>
[SGTKWinPE Filenames]	
TK_File_WinPE_ISO_Filename	<p>The Windows PE ISO file name</p> <p>Default: WinPE_%TK_Build_WinPE_Arch_Type%.iso</p>
TK_File_WinPE_Log_Filename	<p>The Windows PE build process log file name. Default: WinPE_%TK_Build_WinPE_Arch_Type%.log</p>
[SGTKWinPE Custom Scripts]	
TK_Path_BeforeWIMMounted_Custom_Script	<p>Specifies a custom script to run during the SGTkWinPE build process before the WinPE WIM image is mounted for editing. Use the Template.cmd script as a starting point for writing custom scripts, and then put the custom script in the correct location.</p>
TK_Path_WhileWIMMounted_Custom_Script	<p>Specifies a custom script to run during the SGTkWinPE build process while the WinPE WIM image is mounted for editing. Use the Template.cmd script as a starting point for writing custom scripts, and then put the custom script in the correct location.</p>
TK_Path_AfterWIMMounted_Custom_Script	<p>Specifies a custom script to run during the SGTkWinPE build process after the WinPE WIM image is mounted for editing. Use the Template.cmd script as a starting point for writing custom scripts, and then put the custom script in the correct location.</p>

Appendix B. Accessibility

Lenovo strives to provide products with usable access for everyone, regardless of age or ability.

For detailed information, see “Accessibility features for ServerGuide Scripting Toolkit” on page 143.

Accessibility features for ServerGuide Scripting Toolkit

Accessibility features help users who have a disability, such as restricted mobility or limited vision, to use information technology products successfully.

Lenovo and accessibility

See the Lenovo Accessibility website at <http://www.lenovo.com/lenovo/us/en/accessibility.html> for more information about the commitment that Lenovo has to accessibility.

Accessibility

The following list includes the major accessibility features in Lenovo ServerGuide Scripting Toolkit, Windows Edition:

- Can be operated using only the keyboard
- Communicates all information independent of color
- Supports the attachment of alternate output devices
- Provides online documentation in an accessible format

Keyboard navigation

This product uses standard Microsoft Windows navigation keys.

The command line interface (CLI) is controlled by the keyboard.

You can use the following keyboard shortcuts from the graphical user interface:

Shortcut (Linux)	Shortcut (Windows)	Action
Alt+C	Alt+C	Close the graphical user interface.
Alt+N	Alt+N	Go to the next page.
Alt+P	Alt+P	Go to the previous page.
Tab	Tab	Go to the next control.
Shift+Tab	Shift+Tab	Move to the previous control.
Left arrow	Left arrow	Move back one character.
Right arrow	Right arrow	Move forward one character.
Backspace	Backspace	Delete the character to the left of the cursor.
Delete	Delete	Delete the character under the cursor.
Up arrow	Up arrow	Move focus and selection upwards through the radio buttons.
Down arrow	Down arrow	Move focus and selection downwards through the radio buttons.
Space	Space	Select or clear an option.

Appendix C. Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area.

Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

*Lenovo (United States), Inc.
1009 Think Place - Building One
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing*

LENOVO PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. Lenovo may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Trademarks

Lenovo, the Lenovo logo, Flex System, System x, and NeXtScale System are trademarks of Lenovo in the United States, other countries, or both.

Intel and Intel Xeon are trademarks of Intel Corporation in the United States, other countries, or both.

Internet Explorer, Microsoft, and Windows are trademarks of the Microsoft group of companies.

Linux is a registered trademark of Linus Torvalds.

Other company, product, or service names may be trademarks or service marks of others.

Important notes

Processor speed indicates the internal clock speed of the microprocessor; other factors also affect application performance.

When referring to processor storage, real and virtual storage, or channel volume, KB stands for 1 024 bytes, MB stands for 1 048 576 bytes, and GB stands for 1 073 741 824 bytes.

When referring to hard disk drive capacity or communications volume, MB stands for 1 000 000 bytes, and GB stands for 1 000 000 000 bytes. Total user-accessible capacity can vary depending on operating environments.

Lenovo makes no representations or warranties with respect to non-Lenovo products. Support (if any) for the non-Lenovo products is provided by the third party, not Lenovo.

Some software might differ from its retail version (if available) and might not include user manuals or all program functionality.

Index

A

- accessibility
 - features 143
 - keyboard 143
 - shortcut keys 143
- add device drivers 7, 16
- add UXSP updates to deployment 32
- adding custom scripts to Windows PE image 32
- adding device drivers 17
- adding IBM Director Agent to deployment 31
- adding, software components 120
- Advanced Settings Utility 10, 55
 - ASU executable file 56
- answer file 5
- AppliesTo.n parameter 66
- Array_Defaults 68
- Array_Defaults parameter 68
- Array_letter 68
- Array_Mode 67
- ASU 55
 - configuring system settings 10
- ASU executable file 56
- ASU files
 - add 20
- automated deployment 33
- automatic start 33
- automatically continue Toolkit processes 34
- automatically restart deployment 34
- automatically start countdown timer 34
- automatically starting Toolkit 33
- automating deployment process 33

B

- best practices 119
- boot from 121
- boot from SAN target 20
- boot task 12
- build installation image 119

C

- Capture mode 57
- capture, firmware settings 55
- clini.exe 75
- configuration variables 20
- configure
 - Fibre HBA 12
 - Fibre HBA boot configuration 20
 - Fibre Host Bus Adapters (HBAs) 111
 - RAID 10
 - system settings
 - using ASU 10
 - USB key 23, 120

- Configure
 - HBAs 99
- configure RAID 13
- configuring RAID
 - ServerRAID-SCSI controllers
 - 64-bit 121
- configuring RAID controllers 57, 63
- countdown timer 34
- create
 - boot task 12
 - policies file 63
 - RAID configuration disc 123
 - Windows 2008 x64 deployment disc 123
- create arrays 68
- create logical drive 69
- create OS installation image 119
- custom script environment variables 32
- customize
 - configuration of Fibre HBAs 20
- customize server settings 15, 128
- customized installation 24
- customizing 35, 128
- customizing deployment scenarios 15

D

- ddcopy utility 82
- ddcopy.exe 82
- deploy
 - Windows 2008 Enterprise Edition x86
 - locally 12
- deploy from USB key 24
- Deploy mode 57
- deploy, firmware settings 55
- deployment 113
 - configure USB key 23, 120
 - tasks 9
 - use USB key 24
- deployment image
 - copy to USB key 24
- deployment modes
 - deployment methods 9
 - network deployment 9
 - remote deployment 9
- deployment scenarios
 - customizing 15
 - PRAID policy files 19
 - SGTKWinPE settings 35, 128
 - quick start deployment scenarios 9
- deployment tasks 9
- device drivers
 - add 7
- disability 143
- Documentation 118
- download

- driver updates 119
- firmware updates 119
- Systems Director Agent 7
 - Toolkit files 3
- driverinfo.ini 87
- drivers
 - updating 119
- drvutils directory 84
- dscan.exe 84
- DVD installation 26

E

- Emulex website 111
- environment requirements 58
- examples 61
- extract
 - Toolkit files 3

F

- features, accessibility 143
- File
 - unattend.ini 107
- Finding help 118
- firmware 119
 - updating drivers 119
- format
 - USB key 23, 120

H

- hardware
 - supported 53
- hardware detection 89
- help 117–118
- hints and tips 119
- Holdit.exe 84
- hotfix 18
- Hotspares 69
- hwdetect utility 89, 113
- hwdetect.exe 89

I

- IBM Director Agent 13
- IBM Director Agent installation 31
- important notices 146
- install
 - Director Agent 13
 - updates 56
 - UXSP updates 13
- install applications 13
- install UpdateXpress System Pack updates 32
- install UXSP updates during deployment 32
- installation 12
 - network
 - Windows 2008 Server x64 Edition 27
 - Windows Server 2008 x64 125

- remote 29, 126
- ServerGuide Scripting Toolkit 3
- unattended 5, 30, 105
- Windows 2008 26
- Windows answer file 5
- Installation
 - Local
 - Windows Server 2008 x64 121
- installation scenarios 24
- installation, multi-adapter system 119
- installing hotfixes 18
- installing IBM Director Agent as part of deployment 31
- Integrated Management Module (IMM) 30
- introduction 1
- invraid.exe 93
- ipssend 121

K

- keyboard 143
- Known problems 115
- known problems and limitations 119

L

- LEcho utility 97
- Lenovo Support 118
- Lenovo website 118
- Limitations 115
- local 12
- local deployment 9, 12
- local installation 25–26
 - Windows Server 2008 x64
 - with ServeRAID-SCSI 121
- Logical_Defaults 69
- Logical_Mode 69
- Logical.num 70

M

- manually modify Windows PE 32
- modes
 - capture 61
 - Deploy 61
 - PRAID
 - supported modes 58
 - Restore-defaults 61
 - supported by PRAID 58
- modes of operation 9

N

- network 12
- network deployment 12
- network installation 123, 125
 - Windows Server 2008 x64 123
- notes, important 146
- notices 145

O

- operating systems
 - supported by the Scripting Toolkit 53
- Order publications 118

P

- parameter
 - policies file 70
 - policy file
 - StripeSize 67
 - ReadAhead 67
 - RebuildRate 67
- parameter, Array_Mode 67
- parameter, policy file 68–69
- parameters
 - policies file 63
- parameters, PRAID 58
- PE drivers 17
- persistent storage 101
- policies file 63, 66
- policies file parameters 63
- policy file 10, 67, 69–70
 - configuration
 - RAID1 10
- policy file parameter 66–67
 - AppliesTo.n 66
 - Array_Defaults 68
 - Array.letter 68
 - Array_Mode 67
 - Hotspares 69
 - Logical_Defaults 69
 - Logical_Mode 69
 - Logical.num 70
 - policy.name 66
 - ReadAhead 67
 - RebuildRate 67
 - StripeSize 67
- policy.name header 66
- post installation 120
- PRAID 58
 - modes of operation 57
- PRAID policy files
 - customizing 19
- praid.exe 57
- Prerequisites 3

Q

- qaucli utility 99

R

- RAID adapter 93
- RAID configuration 10
- RAID controller configuration 63
- RAID controller support 53
- RAID controllers
 - supported settings 70
- RAID levels 73

- RAID1 10
- ReadAhead 67
- Reboot.exe 84
- Rebuild Rate 67
- remote installation 29, 126
 - Windows
 - Integrated Management Module 30
- Required software 3
- requirements
 - environment 58
- restart deployment 34
- Restore-defaults mode 57
- return codes 62
- running custom scripts 32

S

- savestat utility 101
- scenarios
 - deployment 9
- Script
 - SGTKWinPE.CMD 55
- Scripting Toolkit
 - supported operating systems 53
 - supported systems 53
 - tools 55
 - utilities 55, 75
- SEPs 51
- service, hardware 118
- set up
 - source server 4, 127
- settings for
 - StripeSize 67
- setup 6
- SGTKWinPE process 32
- SGTKWinPE user settings
 - customizing 35, 128
- SGTKWinPE.cmd 55
- shortcut keys 143
- software
 - supported 53
- source server
 - configuration 4–5, 127
 - default settings 4, 127
- source tree 6
 - adding files 5
 - UXSPs 7
 - Director Agent files
 - adding 7
 - setup
 - Director Agent installation 7
- start Toolkit processes 33
- StripeSize 67
- support 117
 - add 51
 - RAID controllers 53
 - system 53
- Support 118
 - Telephone numbers 118
- Support Line 118

- support, hardware 118
- support, HBA and server 54
- support, Lenovo 118
- supported
 - hardware 53
 - software 53
- supported settings for
 - RAID controllers 70
- supported systems 53
- syntax
 - tksearch.exe utility 103
- System Enablement Packs 51
 - download 51
 - install 51

T

- tksearch.exe
 - syntax 103
- Toolkit
 - modes of operation 9
 - Prerequisites 3
- Toolkit Search 103
- trademarks 146
- troubleshooting 117
- Troubleshooting 115
- Troubleshooting help 118

U

- unattend 105
- unattend.ini file 107
- unattended installation 30, 33, 105
 - Windows 35
- UpdateXpress System Pack Installer 56
- UpdateXpress System Packs
 - adding to source tree 7
- updating
 - ASU executable file 56
 - drivers 119
- usage
 - Capture mode 61
 - Deploy mode 61
 - Restore-defaults mode 61
- USB key 121
- USB key installation 25, 125
- user settings 35, 128
- utilities
 - Scripting Toolkit 75
- utility
 - ipssend 121
 - savestat 101
 - tksearch.exe 103
 - Toolkit Search 103
 - unattend.exe 105
 - WINLPCFG 111
- Utility
 - qaucli 99
- utility, ddcopy 82
- UXSP updates 13

- UXSPi 56
- UXSPI 119

W

- web resources v
- Windows
 - remote installation 29, 126
- Windows answer file 5
- Windows installation 16, 24
- windows installation files 6
- Windows PE 32
 - image
 - modify 32
- Windows PE hotfixes 18
- Windows Server 2008 x64
 - Local installation 121
 - network installation 27, 125
- Windows Server 2008 x86 25
- WINLPCFG utility
 - download 111
- WinPE image 55
- Workarounds 115
- writing messages to file 97

Lenovo[™]