IBM ServerGuide Scripting Toolkit

User's Reference Version 1.3.01

Version 1.3.01

Seventh Edition (March 2006)

Contents

About this document

This book provides information about installing, configuring, and using the IBM® ServerGuide™ Scripting Toolkit. It also provides reference information for the utilities that come with the ServerGuide Scripting Toolkit.

How this book is organized

"Chapter 1. Introducing ServerGuide Scripting Toolkit" on page 3 contains a brief overview of the ServerGuide Scripting Toolkit features and prerequisites for use.

"Chapter 2. Installing files and configuring the source tree" on page 7 contains information about unpacking the ServerGuide Scripting Toolkit files and working with the source tree.

"Chapter 3. Customizing Toolkit scenarios" on page 17 contains information about customizing ServerGuide Scripting Toolkit scenarios.

"Chapter 4. Configuring RAID and Fibre Channel" on page 29 contains information about how to use the ServerGuide Scripting Toolkit to configure IBM ServeRAID™ controllers and Fibre Channel host bus adapters during deployment scenarios.

"Chapter 5. Deploying Microsoft Windows operating systems" on page 35 contains information about deploying Windows® operating systems.

"Chapter 6. Deploying Linux operating systems" on page 41 contains information about deploying Linux® operating systems.

"Chapter 7. Performing secure data disposal" on page 49 contains information about how to use the ServerGuide Scripting Toolkit for secure data disposal.

"Chapter 8. Altiris Deployment Solution support" on page 51 contains information about how to integrate the ServerGuide Scripting Toolkit with existing Altiris Deployment Solution deployment scenarios.

"Chapter 9. Microsoft Automated Deployment Services support" on page 65 contains information about how to integrate the ServerGuide Scripting Toolkit with existing Microsoft® Automated Deployment Services deployment scenarios.

"Chapter 10. ServerGuide Scripting Toolkit utilities" on page 73 contains information about the utilities used for the supported deployment scenarios.

"Appendix A. Working with Windows device drivers" on page 125 contains information about how to add, update, and remove Windows device drivers in the ServerGuide Scripting Toolkit device-driver set.

"Appendix B. Enhancing deployment scenarios" on page 129 contains information about incorporating BIOS code and firmware code updates during the deployment process, dynamically updating the Windows answer file, and using RemoteUX to perform post-install updates.

"Appendix C. Getting help and technical assistance" on page 135 contains information about how to obtain help and service for IBM hardware and software products.

"Appendix D. Notices" on page 137 contains important notice and trademark information.

Notices that are used in this book

This book contains the following notices designed 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 is placed just before the instruction or situation in which damage could occur.

IBM deployment resources on the World Wide Web

The following Web pages provide resources for understanding, using, and troubleshooting deployment and systems-management tools.

IBM ServerGuide page

> http://www.ibm.com/servers/eserver/xseries/systems_management/serverguide.html

> This Web page provides an overview of the IBM ServerGuide™ program.

IBM ServerGuide Scripting Toolkit page

> http://www.ibm.com/servers/eserver/xseries/systems_management/sgstk.html

> This Web page provides an overview of IBM ServerGuide Scripting Toolkit and links to the latest version of the toolkit for download.

IBM Update*Xpress* page

> http://www.ibm.com/servers/eserver/xseries/systems_management/ibm_director/extensions/xpress.html

> This Web page provides an overview of IBM Update*Xpress*. It also contains links to pages from which you can download Update*Xpress*, order an Update*Xpress* CD, or purchase a subscription to Update*Xpress* updates.

IBM ServerProven® page

> http://www.ibm.com/servers/eserver/serverproven/compat/us/

> The ServerProven® Web page provides information about xSeries®, BladeCenter™, and IntelliStation® hardware compatibility with IBM systems-management tools.

IBM Support page

> http://www.ibm.com/pc/support/

> This is the IBM Support Web site for IBM hardware and systems-management software.

IBM Systems Management Software: Download/Registration page

> http://www.ibm.com/pc/us/eserver/xseries/systems_management/dwnl.html

> Use this Web page to download IBM systems-management software, including IBM Director.

IBM xSeries® Systems Management page

> http://www.ibm.com/servers/eserver/xseries/systems_management/

> This Web page presents an overview of IBM systems management and IBM Director. Click **IBM Director** for the latest information and documentation.

Chapter 1. Introducing ServerGuide Scripting Toolkit

This chapter contains the following sections:
- "Feature overview"
- "Prerequisites"
- "Supported deployment scenarios" on page 4
- "Supported target server hardware and software" on page 4 for IBM eServer™ and IBM eServer™ xSeries servers

Feature overview

The ServerGuide Scripting Toolkit is a collection of system-configuration tools and sample scripts that you can use to deploy operating systems to IBM eServer or IBM eServer™ xSeries servers in a repeatable, predictable manner. The ServerGuide Scripting Toolkit includes IBM PC DOS 7.1 and can use the device-driver set from IBM ServerGuide.

This task list uses the following terminology:
- The system from which you deploy the operating system is the *source system*.
- The server to which you deploy the operating system is the *target server*.

The ServerGuide Scripting Toolkit can perform the following tasks.
- Detect machine type and model, serial number, and other information about the target server hardware
- Save persistent-state information across server restarts to monitor the deployment process
- Report errors encountered during the deployment process
- Delete existing partitions and data from the hard disks of the target server
- Erase all data on the hard disks of the target server
- Update BIOS code on the target server
- Update IBM ServeRAID BIOS and firmware code on adapters and controllers in the target server
- Configure supported RAID adapters and controllers in the target server
- Configure supported Fibre Channel host bus adapters (Windows operating systems only) in the target server
- Create a primary operating-system installation partition on the first hard disk of the target server, and format the new partition (support for FAT32 partitions for Windows only)
- Install an operating system on the target server (Microsoft Windows or Linux)
- Install device drivers on the target server (for Windows only)
- Install IBM Director Agent for Windows
- Integrate with third-party products like Altiris Deployment Solution, and Microsoft Automated Deployment Solution.

The ServerGuide Scripting Toolkit does not currently include installation script files for Windows Preinstallation Environment 2005 or Linux boot environments.

Prerequisites

The ServerGuide Scripting Toolkit utilities were tested with IBM PC DOS 7.1. If you run the utilities with another version of DOS, replace the fdisk32.com and format32.com commands with commands applicable to the version of DOS you are using.

Use the InstallShield wizard to install the ServerGuide Scripting Toolkit on a 32-bit source platform running one of the following operating systems:

- Red Hat Enterprise Linux (RHEL) AS, Version 3
- Microsoft Windows 2000 Server
- Microsoft Windows 2000 Advanced Server
- Microsoft Windows 2003 Server Standard Edition
- Microsoft Windows 2003 Server Enterprise Edition

> **Requirement:** Microsoft Automated Deployment Services (ADS) requires Windows 2003 Server Enterprise Edition with SP1.

To install the Scripting Toolkit onto the source server, you need a user account and password on the source system with sufficient authority to install the files. You should also obtain a static IP address for the source system to use the source system to deploy code to target systems.

The ServerGuide Scripting Toolkit might require some of the following items that are not part of the Web-downloadable package:

- Licensed copies of all operating systems to be deployed
- The syslinux.com executable for Linux installations.
- The latest version of the *IBM ServerGuide Setup and Installation* CD-ROM that supports the target server
- The latest version of the *IBM Director Setup and Installation* CD
- The latest version of the *IBM UpdateXpress* CD

Supported deployment scenarios

The ServerGuide Scripting Toolkit supports the following standalone deployment scenarios:

- DOS-startable CD-ROM or standalone DOS-startable diskette with data CD-ROM
- DOS-startable diskette with network share
- Remote Supervisor Adapter II or BladeCenter® Management Module and network share

You can use any of these deployment scenarios to deploy a supported Windows operating system. You must use the DOS-startable diskette with standalone network share deployment scenario to deploy a Linux operating system.

The ServerGuide Scripting Toolkit also supports Windows and Linux deployment through the Altiris Deployment Solution for Servers. See "Chapter 8. Altiris Deployment Solution support" on page 51 for more information.

Supported target server hardware and software

The ServerGuide Scripting Toolkit supports deployment of Windows operating systems and Linux operating systems on IBM eServer and IBM eServer™ xSeries servers. In general, the ServerGuide Scripting Toolkit provides support for ServerProven IBM 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)

The README file contains the following information about specific hardware and software support for deployment scenarios:

- A list of supported IBM eServer and IBM eServer™ xSeries servers
- A list of supported RAID adapters and controllers, and Fibre Channel host bus adapters
- A list of deployable operating systems and supported languages for those operating systems
- Any limitations of support for otherwise supported IBM eServer and IBM eServer™ xSeries servers

You can download the latest version of the readme.htm file from the ServerGuide Scripting Toolkit Web page. See "IBM deployment resources on the World Wide Web" on page 2 for information.

Obtaining the ServerGuide Scripting Toolkit

The ServerGuide Scripting Toolkit is available from the Web as an InstallShield executable file and a ZIP file. Install using the InstallShield wizard on a Windows platform. Only InstallShield installations are supported by IBM. The ZIP file is supported only for extracting specific items, such as the Hwdetect utility.

The Scripting Toolkit can be obtained directly from the download site at http://www.ibm.com/pc/support/site.wss/document.do?lndocid=MIGR-53564.

You can also obtain the Scripting Toolkit from the main IBM Web site at http://www.ibm.com :

1. Select **Products**.
2. Select **Intel® processor-based** under **Servers**.
3. Select **Systems Management** on the left or **Systems Management software** under **Related products**.
4. Select **ServerGuide Scripting Toolkit** under **Deploy** to go to the IBM ServerGuide Scripting Toolkit page at http://www.ibm.com/servers/eserver/xseries/systems_management/sgstk.html.
5. Select **ServerGuide Scripting Toolkit Downloads** under **Downloads and Documents** on the right side of the page.

See "IBM deployment resources on the World Wide Web" on page 2 for more information about downloading IBM deployment tools.

Chapter 2. Installing files and configuring the source tree

This chapter contains information about setting up the source tree. After downloading the code and before you can use the ServerGuide Scripting Toolkit, you must install the code and set up the source tree. The major steps are:

1. Install the ServerGuide Scripting Toolkit files.
2. Set up the operating system files.
3. Set up the device driver files.
4. Set up the operating system utilities files.
5. Set up the application files.

The directories in the source tree of the ServerGuide Scripting Toolkit are:

Directory	Contents
sgshare	Network shared directory
sgdeploy	ServerGuide Scripting Toolkit assembly directory
sgtk	ServerGuide Scripting Toolkit base directory (populated by the installation program)
os	Operating systems base directory
osutils	Operating system utilities directory
drvs	Operating systems device drivers base directory
updates	BIOS and firmware files directory
apps	Application installation files directory

Installing the files

The ServerGuide Scripting Toolkit is available for downloading from the IBM Web site as an InstallShield wizard executable and as a ZIP file. Download and use the InstallShield wizard on a Windows system to install the ServerGuide Scripting Toolkit.

If the ServerGuide Scripting Toolkit is not already installed, the wizard presents a series of installation panels to collect parameters for the installation. Otherwise, upgrade or uninstall the existing installation before installing the new ServerGuide Scripting Toolkit.

Installing when a previous installation exists

When the installed version is prior to Version 1.3

When the installed ServerGuide Scripting Toolkit is prior to Version 1.3, manually remove the `sgtk` directory before installing the current version of the ServerGuide Scripting Toolkit.

When the installed version is Version 1.3 or later

When the installed ServerGuide Scripting Toolkit is Version 1.3 or later, perform the following procedure:

1. Save any source tree data and other data that you might have collected.
2. Use the Windows Add/Remove Programs utility to remove the previous installation.
3. Install the current version.

If you do not use the Windows Add/Remove Programs utility to remove a previously installed Version 1.3 or later ServerGuide Scripting Toolkit, the current version automatically performs a silent uninstall of the previously installed version, but does not automatically save any data from the previous installation.

Installing a new version

The InstallShield wizard installs the ServerGuide Scripting Toolkit according to the mode of use. Use this information to identify how you plan to install the ServerGuide Scripting Toolkit:

- Standalone

  Select standalone to use the ServerGuide Scripting Toolkit without the Altiris Deployment Solution or Microsoft Automated Deployment Services (ADS). You can use the ServerGuide Scripting Toolkit from several standlaone environments, which include a bootable compact disc (CD) or digital versatile disc (DVD), a bootable diskette with a data disc, or a network file system.

- Integrated with Altiris Deployment Solution, Windows version

  Requirement: Install the Altiris Deployment Solution, Windows version before installing the ServerGuide Scripting Toolkit.

- Integrated with Microsoft Automated Deployment Services (ADS)

  Requirement: Install ADS before installing the ServerGuide Scripting Toolkit.

The mode of use affects where the wizard installs the ServerGuide Scripting Toolkit and other processing that the wizard performs. The following topics describe installing the ServerGuide Scripting Toolkit for each mode of use.

Standalone Windows server installation instructions

Complete the following steps to download and install the latest version of the ServerGuide Scripting Toolkit files.

1. Download the latest version of the ServerGuide Scripting Toolkit executable file, as described in "Obtaining the ServerGuide Scripting Toolkit" on page 5.
2. From a computer running a Windows operating system, run the executable file that you downloaded in the previous step.
3. The Welcome banner window opens. Click **Next**.
4. When the License Agreement window opens, read the information; then, select **I accept the terms in the license agreement** and click **Next**.
5. Select **Standalone** from the menu.
6. Adjust the destination path, if necessary. For example, if you plan to perform Web-based HTTP installations, specify a location for the top-level directory, sgdeploy, that is within the directory for the installed Web server.
7. When the InstallShield Wizard Complete window opens, click **Finish**.
8. **Naming rules for shared directory:** Share the directory where the Scripting Toolkit files are installed. By default, this is C:\sgshare. The share name that you use for this Scripting Toolkit directory must be DOS 8.3-compliant, without a space in the name of the share.
9. Run the Toolkit Configuration Utility to configure the source tree, as described in "Toolkit Configuration Utility" on page 11.

Standalone Linux server installation instructions

This procedure uses a Windows system as a staging server to install the Scripting Toolkit on a Samba shared drive that the Linux system can access. The Windows system also is a staging server for configuring the source tree on the Samba drive.

Complete the following steps to download and install the latest version of the ServerGuide Scripting Toolkit files to a drive that the Linux system can access.

Requirement: You must install the files from a Windows system that is capable of running a Windows 32-bit executable file.

1. Download the latest version of the ServerGuide Scripting Toolkit executable file, as described in "Obtaining the ServerGuide Scripting Toolkit" on page 5.

2. Save the downloaded executable file to a Samba share drive, accessible to both the Windows machine that you plan to use to install the files and the Linux system that is to be the source system.

   More information: See the manpage for Samba on the Linux system to learn more.

3. From the Windows computer, run the executable file that you downloaded.

4. The Welcome banner window opens. Click **Next**.

5. When the License Agreement window opens, read the information; then, select **I accept the terms in the license agreement** and click **Next**.

6. Select **Standalone** from the menu.

7. Adjust the destination path in the Destination Folder panel, if necessary, to point to the Samba drive, then click **Next**.

8. Click **Install** on the Ready to Install panel.

9. When the InstallShield Wizard Complete window opens, click **Finish**.

10. Share the directory on the Samba drive where the Scripting Toolkit files are now installed. By default, this is C:\sgshare.

    Naming rules: If you use a different share name, the name of the share must be DOS 8.3-compliant, without a space in the name of the share.

11. Run the Toolkit Configuration Utility to configure the source tree, as described in "Toolkit Configuration Utility" on page 11.

12. To use the same Windows computer to install files on another Linux server:

    a. Disconnect from the Samba share where you installed the files.

    b. Click **Start > Control Panel > Add or Remove Programs > IBM ServerGuide Scripting Toolkit > Remove** to uninstall the ServerGuide Scripting Toolkit and remove the registry entries made on the Windows computer during the installation of the file on the Samba drive.

Microsoft Automated Deployment Services (ADS) installation instructions

Requirements:

- Microsoft Windows 2003 Server Enterprise Edition with SP1 is required for the ServerGuide Scripting Toolkit to run with ADS. No other operating system supports this scenario.

- ADS must already be installed on the source system before you install the ServerGuide Scripting Toolkit files.

The InstallShield wizard that installs the Scripting Toolkit checks for the installation of ADS when you select the installation option. The wizard also checks the operating system to verify that it is Windows 2003 Server Enterprise Edition with SP1. If either condition is not met, the wizard grays out the Next button so that you cannot continue the installation. Cancel the installation and verify that the prerequisites are installed before reinstalling.

Complete the following steps to download and install the latest version of the ServerGuide Scripting Toolkit files.

1. Download the latest version of the ServerGuide Scripting Toolkit executable file, as described in "Obtaining the ServerGuide Scripting Toolkit" on page 5.

2. From the computer running the Windows 2003 Server Enterprise Edition with SPI operating system, run the executable file that you downloaded.

3. The Welcome banner window opens. Click **Next**.

4. When the License Agreement window opens, read the information; then, select **I accept the terms in the license agreement** and click **Next**.

5. Select **Integrate with Microsoft Automated Deployment Services (ADS)** from the menu.

6. Adjust the destination path, if necessary. For example, if you plan to perform Web-based HTTP installations, specify a location for the top-level directory, sgdeploy, that is within the directory for the installed Web server.

7. When the InstallShield Wizard Complete window opens, click **Finish**.

8. **Naming rules for shared directory:** Share the directory where the Scripting Toolkit files are installed. By default, this is C:\sgshare. The share name that you use for this Scripting Toolkit directory must be DOS 8.3-compliant, without a space in the name of the share.

9. Run the Toolkit Configuration Utility to configure the source tree, as described in "Toolkit Configuration Utility" on page 11.

Altiris Deployment Solution, Windows version, installation instructions

Requirements: The Altiris Deployment Solution must already be installed on the source system before you install the ServerGuide Scripting Toolkit files.

The InstallShield wizard that installs the Scripting Toolkit checks for the installation of the Altiris Deployment Solution when you select the installation option. If Altiris is not installed, the wizard grays out the Next button so that you cannot continue the installation. Cancel the installation and install Altiris before reinstalling the ServerGuide Scripting Toolkit.

This procedure loads the Scripting Toolkit supplied binary file into the Altiris console. The newly created set of jobs is contained in a new folder labeled with the Scripting Toolkit version number and build date. In addition, a custom IBM Tools menu is created within the Altiris console. The IBM Tools menu is under the Altiris Tools menu option.

Complete the following steps to download and install the latest version of the ServerGuide Scripting Toolkit files for use with the Altiris Deployment Solution for Windows:

1. Download the latest version of the ServerGuide Scripting Toolkit executable file, as described in "Obtaining the ServerGuide Scripting Toolkit" on page 5.

2. From a computer running a Windows operating system, run the executable file that you just downloaded.

3. The Welcome banner window opens. Click **Next**.

4. When the License Agreement window opens, read the information; then, select **I accept the terms in the license agreement** and click **Next**.

5. Select **Integrate with Altiris Deployment Solution, Windows version** from the menu.

6. When the InstallShield Wizard Complete window opens, click **Finish**.

7. Run the Toolkit Configuration Utility to configure the source tree, as described in "Toolkit Configuration Utility" on page 11.

Configuring the source tree

The source tree is a set of directories that contains the files used by the ServerGuide Scripting Toolkit to configure hardware and deploy operating systems. Before using the ServerGuide Scripting Toolkit, you must copy operating-system files, device-driver files, and application files to the source tree.

Note: To perform an HTTP-based installation of a Windows or Red Hat Linux system, the source tree must be in a directory beneath the root directory of a Web server.

Adding files to the source tree

Before you can use the ServerGuide Scripting Toolkit to configure hardware or deploy operating systems to a target server, you must add applicable files to the source tree. Depending on the hardware configuration and operating systems you intend to deploy, all of the following might not be required:

- IBM Director Agent application files
- Operating system installation files for one of the supported operating systems
- Windows device-driver files (for Windows 2000 Server and Windows Server 2003)
- The syslinux.com operating system utility for Linux installs
- System BIOS code
- ServeRAID BIOS and firmware update files

Toolkit Configuration Utility

The Toolkit Configuration Utility, TKConfig.exe, is a wizard for Windows that replaces the manual steps for configuring the source tree of the Scripting Toolkit source server. The Toolkit Configuration Utility works with the Scripting Toolkit to help you set up the various device driver and operating system files for the Scripting Toolkit source server. The TKConfig program reads the Scripting Toolkit installation location from the Windows registry.

The installation wizard for the Scripting Toolkit installs the TKConfig program within the \sgdeploy\sgtk directory. Several methods exist for starting the Toolkit Configuration Utility:

- Launching the Toolkit Configuration Utility from the final panel of the wizard
- Using the shortcut in the Start menu (**Start > Programs > IBM ServerGuide Scripting Toolkit > Toolkit Configuration Utility**)
- Running the `...\sgdeploy\sgtk\TkConfig.exe` command file
- When integrated with the Altiris Deployment Solution, using the link in the **Altiris Tools > IBM Tools** menu

Clicking **Cancel** during the Toolkit Configuration Utility process of copying files immediately stops the copy operation and returns you to the Main Menu. The files already copied are not deleted, but the directories already created are not valid. To recover from a cancelled copy operation, run the Toolkit Configuration Utility again.

The Toolkit Configuration Utility uses the following general procedure for each update to the source tree:

1. Prompts for media (compact disc or hard drive location)
2. Validates compact disc media
3. Removes previously installed files from the source tree
4. Makes necessary directories
5. Copies files and subdirectories into created directories
6. Displays completion message and returns to the main menu

Adding application (IBM Director Agent for Windows) files: To install IBM Director Agent using the ServerGuide Scripting Toolkit, use the Toolkit Configuration Utility to copy IBM Director Agent installation files into the sgdeploy\apps\dawin directory of the source tree.

```
sgdeploy
ServerGuide Scripting Toolkit,
support files directory
```

```
apps
Applications directory
```
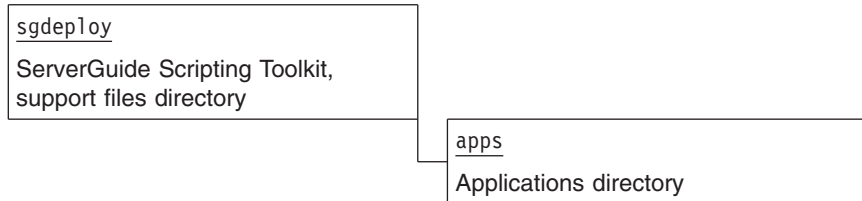
Adding IBM Director Agent installation files to the source tree is supported only for Director Agent 5.10 or higher. Two sources of the files exist: the CD and a ZIP file that you can download.

Using the *IBM Director Setup and Installation* CD: Complete the following procedure to add the IBM Director Agent installation files to the source tree from the CD:

1. Obtain the IBM Director Agent files.

   Locate the *IBM Director Setup and Installation* CD. If you do not have the disk, use the following procedure to download the ZIP file.

2. Start the Toolkit Configuration Utility, as described previously.

3. Click **Add application files**, then click **Next**.

4. Select **Add IBM Director agent installation files**, then click **Next**.

5. Select **IBM Director Agent for Windows (32-bit)** from the list.

6. Select the CD source media.

7. Click **Next**.

8. Insert the *IBM Director Setup and Installation* CD into the CD drive of the source system.

9. Click **OK** at the completion of the file copy to return to the Main Menu.

10. Click **Exit** to end the Toolkit Configuration Utility, or use the Configuration Utility to install other files into the source tree.

Downloading and using the ZIP file: Complete the following procedure to add the IBM Director Agent installation files to the source tree from a downloaded ZIP file:

1. Obtain the IBM Director Agent files.

   Download the Director Agent installation files from the IBM Director Downloads page at http://www-03.ibm.com/servers/eserver/xseries/systems_management/xseries_sm/dwnl.html:

   a. Locate and download the latest installation ZIP file (dir5.*x*_agent_windows.zip) for IBM Director Agent from the Web site.

   b. Unpack the IBM Director Agent ZIP file into a temporary directory, such as the C:\director_agent directory:

2. Start the Toolkit Configuration Utility.

3. Click **Add application files**, then click **Next**.

4. Select **Add IBM Director agent installation files**, then click **Next**.

5. Select **IBM Director Agent for Windows (32-bit)** from the list.

6. Select the appropriate source media.

7. Click **Next**.

8. Provide the path to the diragent.rsp file. For example, type, paste, or browse to select the `C:\director_agent\FILES\diragent.rsp` file.
9. Click **OK** at the completion of the file copy to return to the Main Menu.
10. Click **Exit** to end the Toolkit Configuration Utility, or use the Configuration Utility to install other files into the source tree.

The target folder name is reserved to prevent problems when using the Altiris Deployment Solution and its Post-OS installation job.

Adding device-driver files for Windows operating systems: Windows operating systems require some updated device drivers that are included as part of the ServerGuide Scripting Toolkit.

```
sgdeploy
```
ServerGuide Scripting Toolkit,
support files directory

```
drvs
```
ServerGuide device-driver directory
(empty by default)

Perform the following procedure to add the device drivers to the source tree:
1. Start the Toolkit Configuration Utility.
2. Click **Add device driver files**, then click **Next**.
3. Select the set of device driver files for either Windows 2000 or Windows 2003.
4. When prompted, insert the ServerGuide Setup and Installation disc into the disc drive of the source system.
5. Click **OK** at the completion of the file copy to return to the main menu.
6. Click **Exit** to end the Toolkit Configuration Utility, or use the Utility to install other files into the source tree.

This function is supported only for ServerGuide 7.3 or higher.

Setting up multiple ServerGuide device driver sets: Depending on the types of xSeries servers you intend to support, it might be necessary to set up more than one version of the ServerGuide set of Windows device drivers. For example, you might want to set up multiple device driver sets:
- When trying to support older xSeries servers that are no longer supported by the current version of ServerGuide.
- When trying to support a newly released xSeries server, but the ServerGuide released to support this new server is server-specific.

To set up multiple versions of the ServerGuide set of Windows device drivers:
1. Start the Toolkit Configuration Utility.
2. Click **Add device driver files**, then click **Next**.
3. Select the appropriate operating system version, then click **Next**.
4. Change the value of the **Target Folder** field to a unique identifier not longer than 8 characters. For example:

- For the Windows 2000 device driver set from the ServerGuide 7.3 release, create a folder named w2k_73.
- For the Windows 2003 device driver set from the ServerGuide 7.3.05 release, create a folder named w23_7305.

5. Click **OK** at the completion of the file copy to return to the main menu.

6. Click **Exit** to end the Toolkit Configuration Utility, or use the Utility to install other files into the source tree.

7. To use the additional device driver set, update the Target Folder value in the USRVARS.BAT file.

Adding operating-system installation files: In order to deploy an operating system using the ServerGuide Scripting Toolkit, you must copy operating-system installation files into the source tree.

```
sgdeploy
ServerGuide Scripting Toolkit,
support files directory
```
```
os
Operating-systems directory (empty
by default)
```
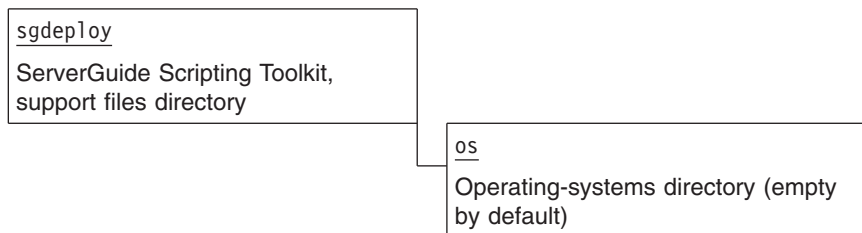
To add the operating system installation files to the source tree, perform the following steps:

1. Start the Toolkit Configuration Utility.

2. Click **Add operating system installation files**, then click **Next**.

3. Select the operating system that you are installing, then click **Next**.

4. Select the edition of the operating system, then click **Next**.

5. Provide the path to the CD drive, where you will load each operating system disc as prompted.

6. **Windows only:** Edit the value in the source path edit field, if necessary, or use **Browse** to provide the source path to the operating system files on the CD. The default \i386\prodspec.ini value should work without change if the CD is a standard Windows operating system CD.

7. Optionally change the **Target Folder** value. If you do change the value, edit the USRVARS.BAT file to change the corresponding value.

8. Click **Next**.

9. When prompted, insert the correct OS installation CD, based on your previous selection, into the CD drive of the source system running Windows.

10. Click **OK** at the completion of the file copy to return to the main menu.

11. Click **Exit** to end the Toolkit Configuration Utility, or use the Utility to install other files into the source tree.
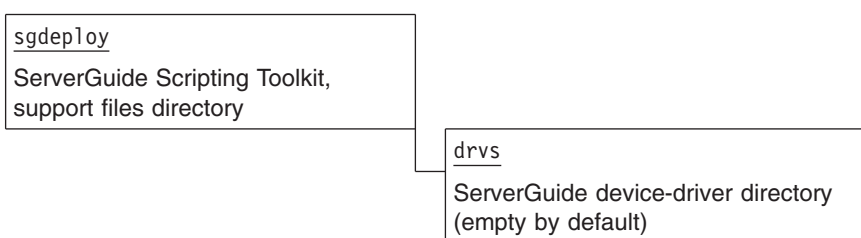
The Toolkit Configuration Utility creates the necessary operating system directory structure.

Default directories for supported operating systems: The Toolkit Configuration Utility, TKConfig.exe, has a default directory for each supported operating system under the sgdeploy\os directory.

The following table lists the default directories and the number of operating system CDs that are required to populate the operating system directories in the source tree:

Table 1. Default source tree directories for operating system files

Supported Operating System	Source tree directory under sgdeploy\os	Number of CDs
Windows 2000 Server	w2k_srv	1
Windows 2000 Advanced Server	w2k_adv	1
Windows Server 2003 Standard Edition	w23_std	1
Windows Server 2003 Enterprise Edition	w23_ent	1
Windows Server 2003 Web Edition	w23_web	1
Red Hat Enterprise Linux (RHEL) 3 AS 32-bit systems	rhas3	4
Red Hat Enterprise Linux 3 ES 32-bit systems	rhes3	4
Red Hat Enterprise Linux 4 AS 32-bit systems	rhas4	4
Red Hat Enterprise Linux 4 ES 32-bit systems	rhes4	4
Red Hat Enterprise Linux 3 AS 64-bit systems	rhas3x64	4
Red Hat Enterprise Linux 3 ES 64-bit systems	rhes3x64	4
Red Hat Enterprise Linux 4 AS 64-bit systems	rhas4x64	5
Red Hat Enterprise Linux 4 ES 64-bit systems	rhes4x64	5
SUSE Linux Enterprise Server (SLES) 9 32-bit systems	sles9	6 plus 3 Service Pack discs
SUSE Linux Enterprise Server (SLES) 9 64-bit systems	sles9x64	6 plus 3 Service Pack discs

SLES 9 source tree considerations: Always use the Toolkit Configuration Utility, TKConfig.exe, to populate the SUSE Linux Enterprise Server (SLES) 9 source tree as opposed to manually copying the files to the source tree.

The SLES 9 source tree configuration that the Toolkit Configuration Utility creates is verified. Subtle directory name requirements for the source tree for the SLES 9 CD components are not intuitive. Copying the files manually might create a non-valid source tree.

Adding operating system utility files: The syslinux.com utility is required for configuring the Linux system and starting the Linux kernel.

```
sgdeploy
ServerGuide Scripting Toolkit,
support files directory
```
```
osutils
Operating system utility files directory,
which is empty by default
```
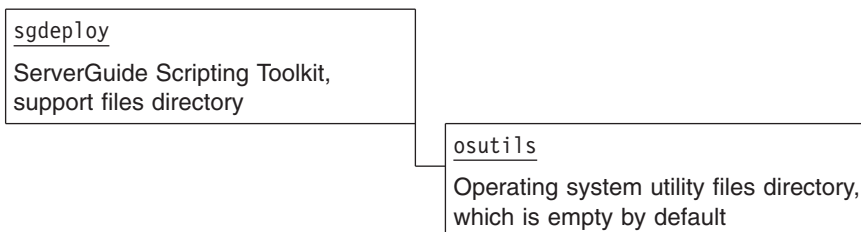
To add the syslinux.com utility to the source tree:
1. Download the syslinux-3.11.zip from the Index of /pub/linux/utils/boot/syslinux Web site at http://www.kernel.org/pub/linux/utils/boot/syslinux/ to a temporary folder.
2. Unpack the archive file to a temporary folder. For example, unpack the ZIP file in the **C:\syslinux** folder.
3. Start the Toolkit Configuration Utility.

4. Click **Add operating system utility files**, then click **Next**.
5. At this time, the **Syslinux utility** is the only utility file and is automatically selected. Click **Next** to continue the installation of the utility.
6. Browse to the folder where you unpacked the syslinux-3.11.zip file and select the syslinux.com file. For example, select **C:\syslinux\dos\syslinux.com**.
7. Optionally change the **Target Folder** value. If you do change the value, edit the USRVARS.BAT file to change the corresponding value.
8. Click **Next**.
9. Click **OK** when the file copy is complete to return to the main menu.
10. Click **Exit** to end the Toolkit Configuration Utility.

Decompress the utility ZIP file before using the Toolkit Configuration Utility to install the utility.

Adding BIOS code and firmware update files to the source tree

In order to use the ServerGuide Scripting Toolkit to update server BIOS code and IBM ServeRAID BIOS and firmware code, you must copy BIOS code and firmware update files into an appropriate directory within the sgdeploy\updates directory of the source tree.

Tip: The Scripting Toolkit Configuration Program does not support adding BIOS code and firmware updates to the source tree.
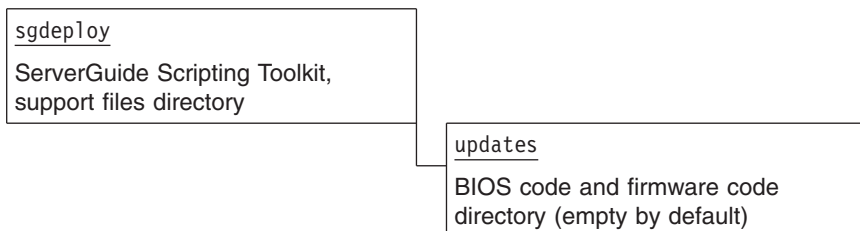
```
sgdeploy
```
ServerGuide Scripting Toolkit, support files directory

```
updates
```
BIOS code and firmware code directory (empty by default)

See "Including BIOS code and firmware updates in a deployment scenario" on page 129 for information about how to add server BIOS code and IBM ServeRAID BIOS and firmware code updates to the source tree.

Chapter 3. Customizing Toolkit scenarios

The ServerGuide Scripting Toolkit comes with a set of diskette images located in the sgdeploy\sgtk\boot\ images directory. Each diskette image provides a complete DOS startup (boot) environment, and is customized to perform a specific ServerGuide Scripting Toolkit scenario.

You can customize the scenarios contained in the image through the use of environment variables contained in two files: netvars.bat and usrvars.bat. The netvars.bat file is only included in diskette images that support use with network-share scenarios. The usrvars.bat file is included in every diskette image, and is already customized for each sample script that comes with the ServerGuide Scripting Toolkit.

Both netvars.bat and usrvars.bat are configured with default values for the defined environment variables, but these default values might not be applicable to all scenarios. You might need to customize these files before you begin using the ServerGuide Scripting Toolkit.

Both netvars.bat and usrvars.bat reference the following global environment variables. These variables are automatically defined during run-time.

Table 2. Global environment variables

Variable name	Description
MACHTYPE	Defines the machine type of the target server
RAMDSK	Defines the RAM drive letter, including a colon (:) character
TKDRV	Defines the ServerGuide Scripting Toolkit drive letter, including a colon (:) character; the default value for this variable is m:

For Altiris users:
- netvars variables are handled by Altiris
- usrvars variables are handled by the individual Altiris jobs

Using helper batch files and creating diskette images

Each ServerGuide Scripting Toolkit supported scenario comes with a helper batch file that creates a DOS-startable (bootable) diskette from a corresponding included diskette image. There are also helper batch files to assist in modifying the netvars.bat and usrvars.bat files.

Most of the helper batch files are located in the sgdeploy\sgtk\boot directory. The ADS helper batch files are located in the sgdeploy\sgtk\ads directory. To use a helper batch file, you must insert a diskette into the diskette drive (a:) of the source system.

Helper batch files in the sgdeploy\sgtk\boot directory: Table 3 contains descriptions of the helper batch files in the sgdeploy\sgtk\boot directory:

Table 3. Descriptions of helper batch files in the sgdeploy\sgtk\boot directory

Helper batch file name	Description
Customize-netvars.bat	*Customizes network environment variables* This file uses notepad.exe to open the a:\netvars.bat file for editing. When the text editor closes, this file calls the Update-Network-Files.bat file to update the necessary network files on the diskette. **Important:** If you do not use this file to modify netvars.bat, you must run the Update-Network-Files.bat file to update the network files on the diskette.

Helper batch file name	Description
Customize-usrvars.bat	*Customizes user environment variables* This file uses notepad.exe to open the a:\usrvars.bat file for editing.
MakeDisk-Disposal.bat	*Creates a secure data disposal diskette* This file uses loaddskf.exe to convert the disposal.img diskette image to a DOS-startable (bootable) diskette in the diskette drive (a:) of the source system.
MakeDisk-Linux-Network-Install.bat	*Creates a Linux network-based installation diskette* This file uses loaddskf.exe to convert the lnetinst.img diskette image to a DOS-startable (bootable) diskette in the diskette drive (a:) of the source system.
MakeDisk-Windows-CD-Install.bat	*Creates a Microsoft Windows CD-based installation diskette* This file uses loaddskf.exe to convert the wcdinst.img diskette image to a DOS-startable (bootable) diskette in the diskette drive (a:) of the source system.
MakeDisk-Windows-Network-Install.bat	*Creates a Microsoft Windows network-based installation diskette* This file uses loaddskf.exe to convert the wnetinst.img diskette image to a DOS-startable (bootable) diskette in the diskette drive (a:) of the source system.
Restore-Diskette-Boot-Record.bat	*Restores the diskette boot record* This file uses altboot.exe to restore the boot record of a startup (boot) diskette that has been altered by altboot.exe.
Save-BootDisk.bat	*Creates an image of the diskette in the source system diskette drive (a:)* This file uses savedskf.exe to convert the diskette in the source system diskette drive (a:) to a file named bootdisk.img in the sgdeploy\sgtk\boot directory of the source tree. This is only for Remote Supervisor Adapter II or BladeCenter Management Module deployment scenarios.
Update-Network-Files.bat	*Updates network files on the diskette with current network environment variables* This file uses the netvars.bat file environment variable settings to update corresponding settings in the following files on the diskette: • a:\bcom\protocol.ini • a:\e1000\protocol.ini • a:\net\network.ini

Helper batch files in the sgdeploy\sgtk\ads directory: Table 4 contains descriptions of the included helper batch files for ADS jobs:

Table 4. Descriptions of helper batch files in the sgdeploy\sgtk\ads directory

Helper batch file name	Description
Modify-ADS-Disposal.bat	This file uses dskimage.exe to copy the provided virtual tk_scrub.vfd floppy diskette image to an editable file on a DOS-startable (bootable) diskette in the diskette drive (a:) of the source system. This file also uses Notepad to open the a:\uservars.bat file for editing.

Helper batch file name	Description
Modify-ADS-RAID.bat	This file uses dskimage.exe to copy the provided virtual tk_raid.vfd floppy diskette image to an editable file on a DOS-startable (bootable) diskette in the diskette drive (a:) of the source system. This helper batch file also uses Notepad to open the a:\uservars.bat file for editing. When you close and save the uservars.bat file, this file automatically starts Notepad again to edit the RAID policy file if one is specified in usrvars.bat.
Modify-ADS-Updates.bat	This file uses dskimage.exe to copy the virtual tk_updt.vfd floppy diskette image to an editable file on a DOS-startable (bootable) diskette in the diskette drive (a:) of the source system. This helper batch file also uses Notepad to open the a:\uservars.bat file for editing. When you close and save the uservars.bat file, this file automatically starts Notepad again to edit the a:\netvars.bat file.
Add-Drivers-To-ADS.bat	Copies device drivers to the correct Microsoft Automated Deployment Services directory This file copies device drivers from a ServerGuide CD to the appropriate location under ADS. See "Adding device drivers to the Microsoft ADS device-driver set" on page 71 for more information.
Makedisk-ADS-RAID.bat	Modifies the pre-created ADS virtual diskette image provided by the ServerGuide Scripting toolkit and places the unzip utility on the diskette image.

Modifying NETVARS.BAT

The netvars.bat file contains environment variable settings that are necessary only for network-share scenarios. Using the guidelines in this section, you can customize the variables in this file for your specific scenario.

To ensure that the network environment variables are properly set in all files on the diskette, use the Customize-netvars.bat helper batch file to make changes to environment variable settings in netvars.bat. Otherwise, you must run the Update-Network-Files.bat file after you finish modifying the netvars.bat file.

Table 5 contains descriptions and default settings for the environment variables you can customize in the netvars.bat file:

Table 5. Environment variables in netvars.bat

Variable name	Description	Default
DPLY_IP	Defines the valid IP address of the system that is the target of the deployment, when USE_DHCP=N	192.168.0.131
DPLY_NAME	Defines the unique boot media name of the system that is the target of the deployment	sgtk1
LOGON	Defines a valid logon name for an account on the system that is the target of the deployment	sgtk0clnt
PWD	Defines a valid logon password for the account specified by LOGON to log on to the target system	sgtk0clnt
SRV_GATE	Defines the gateway IP address of the source server, which the target system uses to connect	192.168.0.2
SRV_IP	Defines the IP address of the source server	192.168.0.25
SRV_MASK	Defines the subnet mask of the source server	255.255.255.0
SRV_NAME	Defines the computer name of the source server	sgtksrv

Table 5. Environment variables in netvars.bat (continued)

Variable name	Description	Default
SRV_SHARE	Defines the shared directory on the source server that contains the files to be deployed	sgshare
USE_DHCP	Defines whether to use Dynamic Host Control Protocol (DHCP) for the target system IP address. Valid values are: • *Y* - Use DHCP and ignore any IP address in the DPLY_IP variable • *N* - Do not use DHCP, but use the value in the DPLY_IP variable instead	N

Modifying USRVARS.BAT

The usrvars.bat file contains environment variable settings that are necessary for all supported scenarios. Using the guidelines in this section, you can customize the variables in this file for your specific scenario.

No usrvars.bat file will contain all of the possible environment variables. Many variables are exclusive to specific deployment scenarios, and are unnecessary for other scenarios. The following functions can be used to group the usrvars.bat environment variables:

- General variables
- Windows deployment variables
- Linux deployment variables
- BIOS code and firmware code update variables
- RAID configuration variables
- Fibre Channel configuration variables
- Scrub configuration variables

Table 6 contains descriptions and default settings for the environment variables you can customize in the usrvars.bat file.

Environment variables:

1. That have values of *Y* or *N* support uppercase or lowercase characters for the variable setting
2. That contain PATH in the name of the variable must contain a fully qualified path that does not end with a forward slash (/) or a backward slash (\)
3. That contain FILE in the name of the variable must only contain a file name, and cannot start with a forward slash (/) or a backward slash (\)
4. That contain DIR in the name of the variable must only contain a directory name, with no preceding path information
5. That contain DRV in the name of the variable must only contain a drive letter followed by a colon (:). These variables cannot end with a forward slash (/) or a backward slash (\)

Table 6. Environment variables in the usrvars.bat file

Variable name	Description	Default
General variables		

Table 6. Environment variables in the usrvars.bat file (continued)

Variable name	Description	Default
DPLY_TYPE	Defines the deployment scenario to be used. Valid values are: • *1* - (For writable media scenarios only) Startup (boot) media is disabled during the deployment process by altboot.exe, to automate the deployment process. • *2* - (For non-writable media scenarios) Startup (boot) media cannot be written to, so you are prompted to remove the media. This is a required setting for DOS-startable (bootable) CDs. • *3* - (For Remote Supervisor Adapter II or BladeCenter Management Module scenarios only) The deployment process exits automatically so that the boot media can unmount. • *4* - ServerGuide Scripting Toolkit is being used with the Altiris Deployment Solution product. • *5* - ServerGuide Scripting Toolkit is being used with the Microsoft Automated Deployment Services product.	2
NOSDIR	Defines the name of the ServerGuide Scripting Toolkit operating-system directory under sgdeploy\sgtk\ examples to be used during the deployment process. Valid values are: *linux* or *windows*	windows
TASDIR	Defines the directory where the ServerGuide Scripting Toolkit files and support files are located. This directory is located in the network-shared directory where you unpack the files.	SGDEPLOY
TAS_PATH	Defines the fully qualified path to the files in TASDIR.	%TKDRV%\%TASDIR% (m:\sgdeploy)
TKDIR	Defines the base directory for the ServerGuide Scripting Toolkit files.	SGTK
TK_PATH	Defines the fully qualified path to the files in TKDIR.	**For CD:** %TKDRV%\%TKDIR% (m:\sgtk) **For network:** %TAS_PATH%\%TKDIR% (m:\sgdeploy\sgtk)
Windows deployment variables		
ANS_FILE	Defines the file name for the unattended installation answer file.	win2000.txt
ANS_PATH	Defines the fully qualified path to the directory that contains ANS_FILE.	%TK_PATH%\examples\%NOSDIR%\ansfile (m:\sgdeploy\sgtk\examples\windows\ansfile)

Table 6. Environment variables in the usrvars.bat file (continued)

Variable name	Description	Default
APPS_DIR	Defines the directory on the target server to which application files are copied during the deployment process.	apps
DAINST	Specifies whether to install IBM Director Agent. Valid values are *Y* or *N*.	N
DA_PATH	Defines the fully qualified path to the IBM Director Agent installation files on the source system.	**For CD:**%TKDRV%\apps\dawin (m:\apps\dawin) **For network:**%TAS_PATH%\apps\dawin (m:\sgdeploy\apps\dawin)
DSET_PATH	Defines the fully qualified path to the IBM ServerGuide device drivers index file, DRVSET.INI	**For CD:** %TKDRV%\w2k_drv (m:\w2k_drv) *or* %TKDRV%\w23_drv (m:\w23_drv) **For network:** %TAS_PATH%\drvs\w2k_drv (m:\sgdeploy\drvs\w2k_drv) *or* %TAS_PATH%\drvs\w23_drv (m:\sgdeploy\drvs\w23_drv)
NOSDRV	Defines the operating-system installation drive on the target system.	C:
OS_PATH	Defines the fully qualified path to the operating-system installation files directory in the source tree.	**For CD:**%TKDRV% (m:) **For network:**%TAS_PATH%\os\w2k_srv (m:\sgdeploy\os\w2ksrv)
PART_SZ	Defines the target server operating-system partition size (in MB) for the primary partition.	8000
WRK_DIR	Defines the working directory on the target server to be used for temporary storage during the installation process. This directory is automatically removed during the deployment process.	wininst
Linux deployment variables		
ANS_FILE	Defines the file name of the kickstart file for Red Hat Enterprise Linux (RHEL) or the file name of the AutoYaST file for SUSE Enterprise Linux Server (SLES) 9.	RHEL: rhas3ks.cfg SLES: sles9.xml
ANS_URL	Defines the location of the file identified in ANS_FILE to the syslinux.com utility. See the operating system documentation for more information.	http://%SRV_IP%/%TASDIR%/%TKDIR%/ examples/linux/redhat/ %ANS_FILE% • RHEL: http://192.168.0.25/sgdeploy/sgtk/ examples/linux/redhat/rhas3ks.cfg • SLES: http://192.168.0.25/sgdeploy/sgtk/ altiris/linux/suse/sles9.xml

Table 6. Environment variables in the usrvars.bat file (continued)

Variable name	Description	Default
DPY_NIC	Defines the network adapter on the target server to use during the deployment process. You must define this variable when more than one network adapter is installed in the target server.	eth0
IRD_FILE	Defines the file name of the network installation RAM disk image file. **Note:** This file name must use the DOS 8.3 file naming convention.	initrd~1.img
IRD_PATH	Defines the fully qualified path to the RAM disk image file specified by IRD_FILE.	%OS_PATH%\images\pxeboot RHEL: m:\sgdeploy\os\rhas3\images\pxeboot SLES: f:\sgdeploy\os\sles9\images\pxeboot)
KRL_FILE	Defines the file name of the network installation kernel file.	vmlinuz
KRL_PATH	Defines the fully qualified path to the file specified by KRL_FILE.	%OS_PATH%\images\pxeboot RHEL: m:\sgdeploy\os\rhas3\images\pxeboot SLES: f:\sgdeploy\os\sles9\images\pxeboot
SL_PATH	Defines the fully qualified path to the syslinux.com utility program.	%TAS_PATH%\osutils (m:\sgdeploy\osutils)
OS_PATH	Defines the fully qualified path to the operating-system installation files directory in the source tree.	RHEL: %TAS_PATH%\os\rhas3 (m:\sgdeploy\os\rhas3) SLES: %TAS_PATH%\os\sles9 (f:\sgdeploy\os\sles9)
OS_TYPE	Defines the manufacturer type of the Linux operating system. Valid values are SUSE or REDHAT.	REDHAT
OS_URL	Defines the fully qualified installation URL for the FTP location for accessing the SUSE Linux Enterprise System (SLES) operating system directory.	ftp://%SRV_IP%/%TASDIR%/os/sles9/
RDSK_SZ	Defines the size (in KB) of the Linux RAM disk to use.	65536
echo nousbstorage > %RAMDSK%\SYSLINUX.TMP	Define any additional kernel parameters you wish to pass to the Linux kernel prior to booting. All kernel options should be separated by a space. For example: echo nousbstorage acpi=noirq > %RAMDISK%\SYSLINUX.TMP	
BIOS code and firmware code update variables		

Table 6. Environment variables in the usrvars.bat file (continued)

Variable name	Description	Default
DO_UPDS	Specifies whether to perform any BIOS code or firmware code updates. Valid values are *Y* or *N*. When set to *N*, this variable overrides the UPD_BIOS and UPD_SR variable settings.	N
UPD_BIOS	Specifies whether to perform BIOS code updates. Valid values are *Y* or *N*.	N
UPD_SR	Specifies whether to perform IBM ServeRAID BIOS code and firmware code updates. Valid values are *Y* or *N*.	N
UPD_BDIR	Defines the base directory that contains the BIOS code and update files in the source tree.	%MACHTYPE%\bios
UPD_SDIR	Defines the base directory that contains the IBM ServeRAID update files in the source tree. Used for adapters 4H, 4Mx, 4Lx, 5i, 6i, 6i+, 6M, 7k.	srvraid
UPD_SV8I	Defines the base directory that contains the IBM ServeRAID update files in the source tree. For adapter 8i.	svraid8i
UPD_PATH	Defines the fully qualified path to the BIOS code and firmware code update files in the source tree.	%TAS_PATH%\updates (m:\sgdeploy\ updates)
RAID configuration variables		
CFG_RD	Specifies whether to perform RAID configuration. Valid values are Y or N.	N
RD_PATH	Defines the fully qualified path to the RAID configuration files in the source tree.	%TK_PATH%\examples\raid (m:\sgdeploy\sgtk\examples\raid)
RD_FILE	Defines the RAID configuration file name (the PRAID policies file) to use when configuring the RAID controller. Leaving this variable blank will run PRAID with the default settings.	No name specified
RD_ERR1	Indicates whether the installation should stop if at least 1 RAID controller cannot be configured because no policy is available. Valid values are: • *N* - do not stop the installation • *Y* - stop the installation	N

Table 6. Environment variables in the usrvars.bat file (continued)

Variable name	Description	Default
RD_ERR2	Indicates whether the installation should stop if no supported RAID controllers are found in the system. Valid values are: • *N* - do not stop the installation • *Y* - stop the installation	N
Fibre Channel configuration variables (for Windows only)		
DO_FIB	Specifies whether to configure Fibre Channel as a boot device on the target server. Valid values are *Y* or *N*.	N
FIB_IO	Defines the input/output (I/O) address of the host bus adapter on the target server to be configured. Valid values are 4-digit numbers. When set to a value of *0000*, the first adapter in the target server scan order (lowest I/O address) is configured.	0000
FIB_LUN	Defines the LUN on the storage device connected to the target server where the operating system is to be installed. Valid values are: 0 through 30.	0
FIB_WWN	Defines the 16-digit hexadecimal world-wide name of the storage device where the boot LUN is located. Valid values are 16-digit hexadecimal numbers. If you set this value to 16 zeros, then the first boot LUN found will be automatically selected. You can find the world-wide node name of the storage device in the controller profile in the IBM Storage Manager software.	0000000000000000
Scrub configuration variables		
DRV_NUM	Defines the drive number (or all drives) to erase. Valid values are: • *n* - Erases a single hard drive on the target server, where *n* is the number of the hard drive. • *ALL* - Erases all hard drives on the target server	ALL

Table 6. Environment variables in the usrvars.bat file (continued)

Variable name	Description	Default
SL	Defines the level of security to be used when cleaning the target server hard drive. Valid values are: • *1* - Limited security: the master boot record and some sectors are overwritten once. • *2* - Medium security: all sectors are overwritten once. • *3* - High security: all sectors are overwritten 4 times. • *4* - US Department of Defense compliant security: all sectors are overwritten 7 times.	1
SMETH	Defines the scrub method that is used to erase the hard drive. Valid values are: • *1* - Use the SL environment variable setting. • *2* - Use the SW environment variable setting.	1
SSIG	Specified whether to write the Scrub signature to the target server hard drive. Valid values are *Y* or *N*.	N
SW	Defines the number of times each sector on the target server hard drive is overwritten. Valid values are any positive integer from 2-99.	2

Automating scenarios

Each boot diskette image provides DOS-based menus to allow interaction with the deployment process. These menus display on every server restart (boot), but you can suppress them. The following examples show these DOS-based menus:

• Network driver selection menu
• Network-based and CD-based Windows selection menu
• Network-based Linux selection menu
• Disposal menu

Table 7. Example DOS-based menus

Network driver selection menu
| ```
PC DOS 7.1 Startup Menu
=======================

1. Load Broadcom (B57) network driver
2. Load IBM Gigabit (E1000) network driver [x206, x306, x345, bc8839]

Enter a choice: 1

F5=Bypass startup files F8=Confirm each line of CONFIG.SYS and AUTOEXEC.BAT [N]
``` |
| **Default menu for network-based and CD-based Windows deployment scenarios** |

Table 7. Example DOS-based menus (continued)

```
PC DOS 7.1 Startup Menu
=======================

1. Run the complete ServerGuide Scripting Toolkit process
2. Perform RAID configuration only
3. Perform Fibre configuration only
4. Perform BIOS updates only
5. Perform IBM ServeRAID BIOS and firmware updates only
6. Boot to a command prompt

Enter a choice: 1

F5=Bypass startup files F8=Confirm each line of CONFIG.SYS and AUTOEXEC.BAT [N]
```

Default menu for network-based Linux deployment scenarios

```
PC DOS 7.1 Startup Menu
=======================

1. Run the complete ServerGuide Scripting Toolkit process
2. Perform RAID configuration only
3. Perform BIOS updates only
4. Perform IBM ServeRAID BIOS and firmware updates only
5. Boot to a command prompt

Enter a choice: 1

F5=Bypass startup files F8=Confirm each line of CONFIG.SYS and AUTOEXEC.BAT [N]
```

Default menu for disposal

```
PC DOS 7.1 Startup Menu
=======================

1. Run the complete ServerGuide Scripting Toolkit process
2. Boot to a command prompt

Enter a choice: 1

F5=Bypass startup files F8=Confirm each line of CONFIG.SYS and AUTOEXEC.BAT [N]
```

The complete Toolkit process is controlled by settings in usrvars.bat, as described in "Modifying USRVARS.BAT" on page 20. At the minimum, the complete process deploys the requested network operating system. In addition, options in usrvars control the following actions:

- Perform selected BIOS and/or IBM ServeRAID BIOS and firmware updates
- Configure Fibre (Windows only)
- Configure RAID

In addition to the menu entry for running the complete Toolkit process, menu selections exist for specific tasks. These tasks also use settings in the usrvars.bat file. However, when you select a menu entry for a specific task, the controlling variable in the usrvars.bat file is ignored and the task is performed.

Preventing DOS-based menus from displaying

Complete the following steps to suppress DOS-based menus from displaying during a network-based deployment process:

1. Open the config.sys file from the diskette in an ASCII text editor.
2. Find the line that contains MENUDEFAULT and refers to the network interface card (NIC) device driver that you want to automatically load; then, uncomment that line. To uncomment a line, remove REM from the beginning of that line. Identify the DOS label for the line you uncomment. The DOS label is located between MENUDEFAULT and the number 0.
3. Find the menu-block section that corresponds to the DOS label from step 2; then, uncomment the line that contains MENUDEFAULT.
4. Save the modified config.sys file to the diskette.

Complete the following steps to suppress DOS-based menus from displaying during a non-network-based deployment process:

1. Open the config.sys file from the diskette in an ASCII text editor.
2. Find the line that contains MENUDEFAULT and uncomment that line. To uncomment a line, remove REM from the beginning of that line.
3. Save the modified config.sys file to the diskette.

Preventing the boot media prompt from displaying

When the boot media is a diskette, set the DPLY_TYPE environment variable equal to 1 to specify using the altboot option during the deployment process. The altboot option allows the server to bypass the diskette drive in the boot sequence after a server restart (boot). Otherwise, you must remove the boot media from the server before the ServerGuide Scripting Toolkit finishes the deployment process and gives control to the operating-system installation process.

See "Using helper batch files and creating diskette images" on page 17 for information about using the Restore-Diskette-Boot-Record.bat helper batch file to restore the diskette boot record. See "Modifying USRVARS.BAT" on page 20 for information about the DPLY_TYPE environment variable.

Preventing the boot media prompt from displaying when using a CD-ROM

When the boot media is a CD, the boot media cannot be altered using the altboot option because the CD is non-writable. To prevent the installation process from stopping and prompting the user to remove the boot media when using a CD, a dummy file must be created on the boot diskette image. This method is available as a base option for IBM PC DOS 7.1 only.

Before saving the boot image to the CD-ROM in your CD creator software, create a zero-byte length file on the boot diskette image called _BOOT_HD.NOZ. When the DOS kernel sees the _BOOT_HD.NOZ file, it displays the `Press the ENTER key to boot from CD or DVD` message and sets a 5-second timer. The user must then press any key, within 5 seconds, to boot from CD-ROM; otherwise, the system defaults to booting from the hard disk drive.

You must then boot from the CD, perform any necessary updates, configure ServeRAID, and create an operating system partition. Once this is done and the Windows files are being copied to the operating system partition, the installation can continue unattended using the _BOOT_HD.NOZ file you created.

Preventing stoppages during Windows deployment scenarios

The ServerGuide Scripting Toolkit provides a sample answer file (win2000.txt for Windows 2000, and win2003.txt for Windows Server 2003) for each supported Windows operating system. By default, the ANS_PATH and ANS_FILE environment variables point to one of these sample files. If you use one of these sample files, make sure that you customize the correct answer file before you begin the deployment process.

For example, if the Windows operating system requires a Product Identification key to be entered during the operating system installation, you must provide this in the answer file or the deployment process will stop and prompt for this information. For this example, you must uncomment the `ProductID=` line in the answer file; then, define the key value as the one provided with the licensed copy of the operating system you are deploying.

Preventing stoppages during Linux deployment scenarios

The ServerGuide Scripting Toolkit provides a sample kickstart file for all supported versions of Red Hat Linux. By default, the ANS_PATH and ANS_FILE environment variables point to one of these sample files. If you use one of these sample files, make sure that you customize the correct kickstart file before you begin the deployment process.

Chapter 4. Configuring RAID and Fibre Channel

This chapter contains information about using the ServerGuide Scripting Toolkit to configure supported RAID adapters and controllers, and Fibre Channel host bus adapters in IBM eServer or xSeries servers. The ServerGuide Scripting Toolkit includes RAID configuration in the deployment scenarios. Fibre Channel configuration is only included in deployment scenarios for Windows operating systems.

You can use the ServerGuide Scripting Toolkit to perform the following tasks:
- Reset supported RAID adapters and controllers in the target server to factory-default settings
- Configure supported RAID adapters and controllers in the target server
- Configure supported Fibre Channel host bus adapters in the target server

The configuration information in this chapter is divided into two main sections, "RAID configuration" and "Fibre Channel configuration" on page 32. These sections contain information about using PRAID to automate configuration of RAID controllers, and Fibre Channel host bus adapters.

RAID configuration

The ServerGuide Scripting Toolkit supports Policy-based RAID configuration and replication using the PRAID.EXE utility. Some features of PRAID.EXE include:
- A single user interface for configuring and replicating RAID controller settings.
- The ability to use the PRAID policies file to describe how your RAID controllers should be configured or replicated.
- Customizable logic to determine the configuration to use with which controllers. This logic can include the machine type of the server, the number of drives connected to the controller, and the RAID controller type.
- An AUTO mode to configure using default settings.
- The ability to configure all RAID controllers in a system with a single program call.
- Features to save useful information about each captured configuration, including machine type, date, and time of capture.
- The ability to restore all controllers to factory-default settings.

You can use the ini files included with PRAID to configure your RAID controllers using default parameters, or you can supply the configuration parameters for each RAID controller using a PRAID policies file. When used in *configure* mode, the policies file directs how PRAID configures the RAID controllers in a system using keywords and values that can be customized. 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. For more information about the PRAID utility, see "PRAID.EXE" on page 101 in the Utilities section.

To use any policy file, you must supply the proper parameters located in the USVARS.BAT file.

The ServerGuide Scripting Toolkit RAID configuration provides support the following types of RAID controllers:
- IBM ServeRAID-4H
- IBM ServeRAID-4Lx
- IBM ServeRAID-4Mx
- IBM ServeRAID-5i
- IBM ServeRAID-6i
- IBM ServeRAID-6i+

- IBM ServeRAID-6M
- IBM ServeRAID-7k
- IBM ServeRAID-7t
- IBM ServeRAID-7e (Adaptec HostRAID) SATA
- IBM ServeRAID-7e (Adaptec HostRAID) SCSI
- IBM ServeRAID-8i
- IBM ServeRAID-8e (Adaptec HostRAID) SAS
- IBM ServeRAID-8e (Adaptec HostRAID) SATA
- LSI 1064x SAS
- LSI 1020 and LSI 1030 chipset
- LSI IDEal RAID

Preparing USRVARS.BAT for RAID configuration

You must update variables in USRVARS.BAT to indicate that you want the ServerGuide Scripting Toolkit to configure the RAID controllers as part of the deployment process.

You can configure all RAID controllers using the default values for arrays, logical drives, and other controller settings, or you can use a policies file to specify the configuration that you want to use for your RAID controllers.

The ServerGuide Scripting Toolkit supplies sample policies files. See "Sample PRAID policies files for RAID configuration" on page 31 for more information.

Configuration using default values

> To configure USRVARS.BAT to use default values for arrays, logical drives, and other controller settings, you must:
> - Set `CFG_RD` to `Y`
> - Leave `RD_FILE` blank.
>
> This causes the ServerGuide Scripting Toolkit process to run the PRAID utility with the /d switch specified, which uses defaults for all controller settings. See Table 8 on page 31 for the list of array and logical drive default settings for your RAID controllers.

Configuration using a specific policies file

> To configure USRVARS.BAT to use a specific PRAID policies file to configure the RAID controllers:
> - Set `CFG_RD` to `Y`.
> - Set `RD_PATH` to the drive letter and path to the PRAID policies file you want to use.
> - Set `RD_FILE` to the file name of the PRAID policies file that you want to use.
>
> This causes the ServerGuide Scripting Toolkit process to run the PRAID utility using the policies file that you specified to configure the RAID controllers.
>
> See "Policies file" on page 108 for additional information about policies files.

Table 8. Default values for arrays and logical drives

Controller	Write caching	Read ahead	Stripe size	Rebuild rate	Number of hot spares	RAID level	Arrays	Logical drives
ServeRAID-4H	On	Adaptive	8 KB	High	• 1 - if there are 4 or more physical drives. The largest physical drive is selected. • 0 - if there are 3 or less physical drives	No. of drives in the array: • 1 - RAID 0 • 2 - RAID 1 • 3 or more - RAID 5	Arrays created using drives that have the same size in MB. Each set of drives with same size will be combined into a single array, up to the maximum allowed for the RAID controller.	1 logical drive created per array using all available space
ServeRAID-4Mx	On	Adaptive	8 KB	High				
ServeRAID-4Lx	Off	Adaptive	8 KB	High				
ServeRAID-5i	On	Adaptive	8 KB	High				
ServeRAID-6i/6i+	On	Adaptive	8 KB	High				
ServeRAID-6M	On	Adaptive	8 KB	High				
ServeRAID-7k	On	Adaptive	8 KB	High				
ServeRAID-8i	Off	On	64 KB	n/a		No. of drives in the array: • 1 - Volume • 2 - RAID 1 • 3 or more - RAID 5		
ServeRAID-8e SAS	n/a	n/a	64 KB	n/a	0	No. of drives in the array: 2 - RAID 1		
ServeRAID-8e SATA	n/a	n/a	64 KB	n/a	0	No. of drives in the array: 2 - RAID 1		
ServeRAID-7t	Off	On	64 KB	n/a	0			
ServeRAID-7e SATA	n/a	n/a	64 KB	n/a	0	No. of drives in the array: • 2 - RAID 1		
ServeRAID-7e SCSI	n/a	n/a	64 KB	n/a	0	No. of drives in the array: • 2 - RAID 1 • 3 - RAID 0		
LSI 1064x SAS	n/a	n/a	n/a	n/a	1 - if 3 or more physical drives avail.	No. of drives in the array: • 1 - no RAID • 2 - RAID 1		
LSI 1020/1030	n/a	n/a	n/a	n/a	1 - if 3 or more physical drives avail.	No. of drives in the array: • 1 - no RAID • 2 - RAID 1		
LSI IDEal RAID	On	n/a	64 KB	n/a	0	No. of drives in the array: • 1 - RAID 0 • 2 - RAID 1		

Sample PRAID policies files for RAID configuration

The ServerGuide Scripting Toolkit supplies some sample PRAID policies files for configuring the RAID controllers using commonly used configurations. These sample PRAID policies files can be used without

modification. You can configure the RAID controllers using these sample PRAID policies files by simply setting the CFG_RD variable to a value of *Y*, and the RD_FILE parameter to the appropriate sample file name.

These sample files are located in `sgdeploy\sgtk\examples\raid`:

RAID1-5.ini

> Creates a RAID-1 array using the first two drives, and a RAID-5 array using the remaining drives connected to the RAID controller for the following RAID controllers: ServeRAID-4H,4Mx,4Lx,5i,6i,6i+,6M,7k

RAID5HSP.ini

> Creates a single hot-spare drive and a RAID-5 array using all remaining drives for the following RAID controllers: ServeRAID-4H,4Mx,4Lx,5i,6i,6i+,6M,7k,7t

Fibre Channel configuration

The ServerGuide Scripting Toolkit automates the preparation of the target server for Windows operating-system installation to a hard disk connected to a supported Fibre Channel host bus adapter. The ServerGuide Scripting Toolkit uses the fibrecfg.exe utility to configure Fibre Channel host bus adapters. See "FIBRECFG.EXE" on page 92 for information about this utility.

The ServerGuide Scripting Toolkit automates the following tasks:

- Selection of the Fibre Channel host bus adapter I/O address that the target server will start (boot), when more than one address is available in the target server
- Enablement of the Fibre Channel host bus adapter in the host bus adapter BIOS code
- Enablement of starting (booting) from the Fibre Channel host bus adapter
- Selection of the primary boot LUN

Fibre Channel configuration is not supported with the Altiris Deployment Solution or Microsoft Automated Deployment Services scenarios.

The following Fibre Channel adapters are supported:

- IBM TotalStorage® FC2-133 Host Bus Adapter
- IBM HS20 Fibre Channel Expansion Card

To automate configuration of Fibre Channel host bus adapters, you must update some environment variables in usrvars.bat so that the ServerGuide Scripting Toolkit will configure the Fibre Channel host bus adapters during the deployment process, as indicated below. See "Modifying USRVARS.BAT" on page 20 for information on the environment variables you can configure.

Configuring Fibre HBAs as a boot device

To configure Fibre HBAs as a boot device during a scripted installation, you must set four variables in USRVARS.BAT:

DO_FIB

> Must be set to a value of *Y* to install the OS to a SAN. The default value is *N*.

FIB_IO

> Defines the I/O address of the HBA to configure. Setting FIB_IO to the default value of ″0000″ will cause the first adapter in the system scan order to be configured. The first adapter in the system scan order is the one with the lowest I/O address.

FIB_WWN

> Defines the 16-digit hexadecimal world-wide name of the storage device where the boot LUN is

located. If you set this value to the default of all zeros: "0000000000000000", the LUN with the lowest loop ID will be used as the boot device, as it will be the first one found.

FIB_LUN

Defines the LUN number on the storage device where the OS should be deployed. Default value is "0".

After setting these variables, the Toolkit installation will prepare the fibre HBA at I/O address *FIB_IO* for OS deployment by enabling the adapter BIOS, enabling selectable boot from the device, and setting LUN *FIB_LUN* at world-wide name *FIB_WWN* as the primary boot LUN.

Chapter 5. Deploying Microsoft Windows operating systems

This chapter contains information about using the ServerGuide Scripting Toolkit to perform unattended installations of Windows operating systems on IBM BladeCenter, IBM eServer, or IBM eServer xSeries servers. See the readme.htm file for a complete list of supported Microsoft Windows operating systems.

If you have not already done so, see "Chapter 2. Installing files and configuring the source tree" on page 7 for preliminary instructions.

The ServerGuide Scripting Toolkit supports the following deployment scenarios for Windows operating systems:

- DOS-startable CD-ROM or DOS-startable diskette and data CD-ROM
- DOS-startable diskette and network share
- Remote Supervisor Adapter II or BladeCenter Management Module and network share

Preparing the answer file for an unattended installation

Two sample answer files, win2000.txt and win2003.txt, come with the ServerGuide Scripting Toolkit. These sample files are located in the sgdeploy\sgtk\examples\windows\ansfile directory of the source tree. Microsoft also includes a sample answer file in the i386 directory of the operating-system installation files.

You must add information to the [UserData] section of the answer file, including a value for the ProductID variable, when required by the operating system you are deploying. You can add information manually to the answer file before you begin the deployment process, or dynamically during the deployment process. See "Updating the answer file dynamically" on page 127 for information about dynamically updating the answer file during the deployment process.

Do not add device-driver information to the answer file. The ServerGuide Scripting Toolkit dynamically adds device-driver information to the answer file during the deployment process.

For information about customizing the answer file, see the Microsoft documentation in the support\tools\deploy.cab file on the operating-system installation CD-ROMs.

Using a DOS-startable CD-ROM or DOS-startable diskette and data CD-ROM

You must have the following items to use this deployment scenario:

- One blank diskette
- One blank CD-ROM
- Source system with properly-configured source tree and a CD-R or CD-RW drive
- CD-creation software
- (Optional) *IBM UpdateXpress* CD

Complete the following steps to deploy a Windows operating system to a target server:

1. Create a DOS-startable (bootable) diskette:
    a. Insert a blank diskette into the diskette drive of the source system.
    b. Navigate to the boot directory in the source tree.
    c. Run the MakeDisk-Windows-CD-Install.bat helper batch file. See "Using helper batch files and creating diskette images" on page 17 for information about this file.

d. Customize the environment variables for this deployment scenario by running the Customize-usrvars.bat helper batch file to open the USRVARS.BAT file for editing. See "Modifying USRVARS.BAT" on page 20 for information on the environment variables you can configure.

   If you are using a DOS-startable (bootable) CD, set the DPLY_TYPE environment variable equal to 2.

e. Customize the appropriate answer file. Sample answer files, win2000.txt and win2003.txt, are located in the sgdeploy\sgtk\examples\windows\ansfile directory of the source tree. For more information on customizing an answer file, see "Preparing the answer file for an unattended installation" on page 35.

f. Label the DOS-startable (bootable) diskette so that you can easily identify it later.

2. Follow these steps to create the data or DOS-startable CD:

   a. Insert a blank CD-ROM into the CD-R or CD-RW drive of the source system.

   b. If you are making a DOS-startable CD-ROM, set the CD-ROM creation software to create a bootable CD-ROM using floppy disk emulation and to generate the image from the diskette you created previously. The image added to the CD-ROM will include the USRVARS.BAT file from the diskette.

   c. Using the CD-ROM creation software, create an image that contains at least the following directories. Remember that:

      1) Directories on the CD-ROM file system cannot be more than 8 levels deep and must comply with DOS 8.3 file naming conventions. Ensure that you enable the use of all DOS 8.3 file names in the CD-ROM creation software.

      2) Ensure that you copy the customized answer file you created previously from the source system to this CD-ROM.

CD-ROM directory	Source tree location	Purpose
apps\dawin	sgdeploy\apps\dawin	To contain IBM Director Agent installation files
OSdevdrv	sgdeploy\drvs\OSdevdr	To contain device-driver files
i386	sgdeploy\os\OS\i386	To contain the Windows operating-system installation files
sgtk	sgdeploy\sgtk	To contain the ServerGuide Scripting Toolkit files
updates	sgdeploy\updates	To contain BIOS and firmware updates

   where:
   - *OSdevdrv* is the operating-system device-driver files directory in the source tree that corresponds to the operating system being deployed. (Can be w2k_drv or w23_drv)
   - *OS* is the name of the operating-system directory in the source tree that contains the operating-system installation files.

   d. Write the image you have created to a CD-ROM and label the CD-ROM so that you can easily identify it later.

When you have completed the CD-ROM, perform these steps on the target server:

1. (Optional) Update the BIOS code and firmware code on the target server:

   a. Start the target server and insert the *IBM UpdateXpress* CD-ROM into the CD-ROM or DVD drive.

   b. When IBM Update*Xpress* starts, follow the instructions on the screen to update the BIOS code and firmware code.

   c. When all updates are completed, remove the *IBM UpdateXpress* CD-ROM from the target server.

2. Install the operating system on the target server:

   a. Insert the DOS-startable (bootable) CD into the CD-ROM or DVD drive of the target server, or insert the DOS-startable diskette into the diskette drive and the data CD-ROM into the CD-ROM or DVD drive. Turn on the server as required.

b. Restart the server so that it can start (boot) from the DOS-startable CD-ROM or diskette and begin the ServerGuide Scripting Toolkit deployment process.

   c. If prompted, remove the DOS-startable CD, or the DOS-startable diskette and data CD-ROM from the target server.

3. (Optional) Update the device drivers to the latest level:

   a. Insert the *IBM UpdateXpress* CD-ROM into the CD-ROM or DVD drive of the target server and wait for Update*Xpress* to start.

   b. Follow the instructions on the screen to update the installed device drivers.

   c. When all updates are completed, remove the *IBM UpdateXpress* CD-ROM from the target server.

Using a DOS-startable diskette and network share

You must have the following items to use this deployment scenario:

- One blank diskette
- Source system with properly-configured source tree
- Network connectivity and access to the source system directory structure from the target server
- (Optional) *IBM UpdateXpress* CD-ROM

Complete the following steps to deploy a Windows operating system to a target server:

1. Create a DOS-startable (bootable) diskette:

   a. Insert a blank diskette into the diskette drive of the source system.

   b. Navigate to the boot directory in the source tree.

   c. Run the MakeDisk-Windows-Network-Install.bat helper batch file. See "Using helper batch files and creating diskette images" on page 17 for information about this file.

   d. Customize the network environment variables for this deployment scenario by running the Customize-netvars.bat helper batch file. See "Modifying NETVARS.BAT" on page 19 for information about the environment variables you can configure.

   e. Customize the environment variables for this deployment scenario by running the Customize-usrvars.bat helper batch file.

       1) To install IBM Director Agent, set DAINST to a value of *Y* and set the DA_PATH environment variable to %TAS_PATH%\apps\dawin (by default, m:\sgdeploy\apps\dawin).

       2) See "Modifying USRVARS.BAT" on page 20 for information on the environment variables you can configure.

   f. Customize the answer file.

   g. Label the DOS-startable (bootable) diskette so that you can easily identify it later.

When you have completed the diskette, perform these steps on the target server:

1. (Optional) Update the BIOS code and firmware code on the target server:

   a. Start the target server and insert the *IBM UpdateXpress* CD into the CD drive.

   b. When IBM Update*Xpress* starts, follow the instructions on the screen to update the BIOS code and firmware code.

   c. When all updates are completed, remove the *IBM UpdateXpress* CD from the target server.

2. Install the operating system on the target server:

   a. Insert the DOS-startable (bootable) diskette into the diskette drive of the target server.

   b. Restart the server so that it can start (boot) from the DOS-startable diskette and begin the ServerGuide Scripting Toolkit deployment process.

   c. When prompted or when the deployment process has completed, remove the DOS-startable diskette from the diskette drive of the target server.

3. (Optional) Update the device drivers to the latest level:

a. Insert the *IBM UpdateXpress* CD into the CD drive of the target server and wait for Update*Xpress* to start.

b. Follow the instructions on the screen to update the installed device drivers.

c. When all updates are completed, remove the *IBM UpdateXpress* CD from the target server.

Using a Remote Supervisor Adapter II or BladeCenter Management Module and network share

You must have the following items to use this deployment scenario:

- One blank diskette
- Source system with properly configured source tree
- Network connectivity to the Remote Supervisor Adapter II or BladeCenter Management Module from the source system.

  The BladeCenter Management Module firmware must be version 1.08 or later.
- Network connectivity and access to the source system directory structure from the target server
- (Optional) *IBM UpdateXpress* CD

Complete the following steps to deploy a Windows operating system to a target server:

1. Create a DOS-startable (bootable) diskette:

   a. Insert a blank diskette into the diskette drive of the source system.

   b. Navigate to the boot directory in the source tree.

   c. Run the MakeDisk-Windows-Network-Install.bat helper batch file. See "Using helper batch files and creating diskette images" on page 17 for information about this file.

   d. Customize the network environment variables for this deployment scenario by running the Customize-netvars.bat helper batch file. See "Modifying NETVARS.BAT" on page 19 for information about the environment variables you can configure.

   e. Customize the environment variables for this deployment scenario by running the Customize-usrvars.bat helper batch file.

      1) You must set the DPLY_TYPE environment variable equal to 3.

      2) To install IBM Director Agent, set DAINST to a value of *Y* and set the DA_PATH environment variable to %TAS_PATH%\apps\dawin (by default, m:\sgdeploy\apps\dawin).

      3) See "Modifying USRVARS.BAT" on page 20 for information on the environment variables you can configure.

   f. Customize the answer file.

   g. Open the config.sys file from the diskette in an ASCII text editor.

   h. Find the line that contains MENUDEFAULT and refers to the network interface card (NIC) device driver that you want to automatically load; then, uncomment that line. To uncomment a line, remove REM from the beginning of that line.

      1) Select a NIC device driver for the Remote Supervisor Adapter II or Blade Center Management Module.

      2) Identify the DOS label for the line you uncomment. The DOS label is located between MENUDEFAULT and the number 0.

   i. Find the menu-block section that corresponds to the DOS label from step 1h; then, uncomment the line that contains MENUDEFAULT.

   j. Save the modified config.sys file to the diskette.

   k. Create a virtual-floppy deployable image from the diskette by running the Save-BootDisk.bat helper batch file. This will create a new disk image file named bootdisk.img in the sgdeploy\sgtk\boot directory.

When you have completed the deployable image, perform these steps on the target server:

1. (Optional) Update the BIOS code and firmware code on the target server:

   a. From the source system, open a Web browser; then, type the host name or IP address of the Remote Supervisor Adapter II or BladeCenter Management Module of the target server in the **Address** or **Location** field and press Enter.

   b. Insert the *IBM UpdateXpress* CD-ROM into the CD-ROM or DVD drive of the source system.

   c. Use the remote control function of the Remote Supervisor Adapter II or BladeCenter Management Module Web interface to mount the CD-ROM or DVD drive that contains the *IBM UpdateXpress* CD.

      For a BladeCenter Management Module, make sure that the KVM owner and media tray owner are set to the blade to which you are deploying the updates.

   d. Use the Web interface to restart (reboot) the target server and start the BIOS code and firmware code update process.

   e. When all updates are completed, unmount the CD-ROM drive from the target server; then, remove the *IBM UpdateXpress* CD-ROM from the source system.

2. Install the operating system on the target server:

   a. Use the remote control function of the Remote Supervisor Adapter II or BladeCenter Management Module Web interface to mount the diskette image you created in step 1k on page 38 on the target server.

   b. Use the Web interface to restart the server so that it can start (boot) from the virtual-floppy diskette image and begin the ServerGuide Scripting Toolkit deployment process.

   c. When prompted, unmount the diskette image from the target server.

   d. To restart (reboot) the target server and complete the operating system installation, press any key in the Remote Console or use the Web interface to restart the server.

3. (Optional) Update the device drivers to the latest level:

   a. Insert the *IBM UpdateXpress* CD-ROM into the CD-ROM or DVD drive of the source system.

   b. Use the Remote Control function of the Web interface to mount the CD-ROM on the target server.

   c. When all updates are completed, unmount the CD-ROM from the target server; then, remove the *IBM UpdateXpress* CD-ROM from the source system.

Chapter 6. Deploying Linux operating systems

This chapter contains information about using the ServerGuide Scripting Toolkit to perform unattended installations of supported Linux operating systems on IBM eServer and IBM eServer xSeries servers. The ServerGuide Scripting Toolkit supports the DOS-startable diskette and network share deployment scenario for the following Linux operating systems:

- Red Hat Enterprise Linux AS 3 32 bit
- Red Hat Enterprise Linux AS 3 64 bit
- Red Hat Enterprise Linux ES 3 32 bit
- Red Hat Enterprise Linux ES 3 64 bit
- Red Hat Enterprise Linux AS 4 32 bit
- Red Hat Enterprise Linux AS 4 64 bit
- Red Hat Enterprise Linux ES 4 32 bit
- Red Hat Enterprise Linux ES 4 64 bit
- SUSE Linux Enterprise Server 9 Service Pack 3 32 bit
- SUSE Linux Enterprise Server 9 Service Pack 3 64 bit

See the readme.htm file for the most current list of supported Linux operating systems. If you have not already done so, see "Chapter 2. Installing files and configuring the source tree" on page 7 for preliminary instructions.

The ServerGuide Scripting Toolkit supports the DOS-startable diskette scenario and the network share deployment scenario for Red Hat Enterprise Linux (RHEL) and SUSE Linux Enterprise Server (SLES) operating systems.

Deploying Red Hat Enterprise Linux

This section describes the procedures required for performing a Red Hat Enterprise Linux (RHEL) deployment. In order to properly deploy RHEL, you must make slight changes to the sample kickstart files provided by the ServerGuide Scripting Toolkit.

Preparing the kickstart file

The following sample kickstart files are provided in the sgdeploy\sgtk\examples\linux\redhat directory of the source tree.

Kickstart file name	Linux distribution
rhas3ks.cfg	Red Hat Enterprise Linux AS 3 32-bit
rhas3x64.cfg	Red Hat Enterprise Linux AS 3 64-bit
rhes3ks.cfg	Red Hat Enterprise Linux ES 3 32-bit
rhes3x64.cfg	Red Hat Enterprise Linux ES 3 64-bit
rhas4ks.cfg	Red Hat Enterprise Linux AS 4 32-bit
rhas4x64.cfg	Red Hat Enterprise Linux AS 4 64-bit
rhes4ks.cfg	Red Hat Enterprise Linux ES 4 32-bit
rhes4x64.cfg	Red Hat Enterprise Linux ES 4 64-bit

You must customize the Installation Media Configuration section in the applicable kickstart file to specify the network protocol and location of the operating-system installation files. See the following example and Table 9 for an example section from the Red Hat Linux AS 3 sample kickstart file that comes with the ServerGuide Scripting Toolkit.

```
##########################################################
# Installation Media Configuration
#
# The following lines should be equivalent to:
# HTTP: url --url http://%SERVER_IP%/path/to/OS/files
# NFS:  nfs --server %SERVER_IP% --dir /path/to/OS/files
##########################################################


# nfs --server 192.168.0.25 --dir /var/www/rhas3
url --url http://192.168.0.25/rhas3/
```

Table 9. Kickstart file details

Kickstart file statement	Description
nfs --server 192.168.0.25 --dir /var/www/rhas3	Specifies that the Red Hat Linux installation program uses NFS to download the operating-system installation files. Also specifies that the operating-system installation files are located in the /var/www/rhas3 directory of an NFS server with an IP address of 192.168.0.25.
url --url http://192.168.0.25/rhas3/	Specifies that the Red Hat Linux installation program uses HTTP to download the operating-system installation files. Also specifies that the operating-system installation files are located in the rhas3 directory of a Web server with an IP address of 192.168.0.25.

For information about customizing kickstart files, see the Red Hat Linux documentation on the installation media or the Red Hat Linux Web site http://www.redhat.com. You can also see the kickstart HOWTO available from the Linux Documentation Project Web site http://www.tldp.org.

Using a DOS-startable diskette and network share

You must have the following items to use this deployment scenario:

- One blank diskette
- Source system with properly configured source tree
- (Optional) Network connectivity to the operating system installation files in a separate location on an HTTP, NFS, or FTP server.

  In order to connect to the HTTP server, it might be necessary to change the MIME types allowed by IIS-6. However, you should evaluate your specific security needs before making changes to the allowable MIME types.
- (Optional) *IBM UpdateXpress* CD-ROM

The RHEL operating system installation files must be accessible through HTTP, NFS, or FTP, as required by the RHEL installation program.

The ServerGuide Scripting Toolkit files must be accessible through the Server Message Block/Common Internet File System (SMB/CIFS) protocol. This can be from Samba on a Linux network share or a Windows shared directory.

Attention: The ServerGuide Scripting Toolkit utilities automatically delete the existing partitions on a target server prior to installation. This behavior is changed from the way installations were previously accomplished.

Complete the following steps to deploy a Linux operating system to a target server:

1. Create a DOS-startable (bootable) diskette:
   a. Insert a blank diskette into the diskette drive of the source system, or a Windows computer that you are using to connect to the source system.
   b. Navigate to the boot directory in the Scripting Toolkit source tree.
   c. Run the MakeDisk-Linux-Network-Install.bat helper batch file. See "Using helper batch files and creating diskette images" on page 17 for information about helper batch files.
   d. Run the Customize-netvars.bat helper batch file to customize the network environment variables for this deployment scenario. See "Modifying NETVARS.BAT" on page 19 for information about the environment variables you can configure.
   e. Run the Customize-usrvars.bat helper batch file to customize the environment variables for this deployment scenario. See "Modifying USRVARS.BAT" on page 20 for information about the environment variables you can configure.
   f. Customize the kickstart file for RHEL. See "Preparing the kickstart file" on page 41 for more information.
   g. Label the DOS-startable (bootable) diskette so that you can easily identify it later.

When you have completed the diskette, perform these steps on the target server:
1. (Optional) Update the BIOS code and firmware code on the target server:
   a. Start the target server and insert the *IBM UpdateXpress* CD-ROM into the CD-ROM drive.
   b. When IBM Update*Xpress* starts, follow the instructions on the screen to update the BIOS code and firmware code.
   c. When all updates are completed, remove the *IBM UpdateXpress* CD-ROM from the CD-ROM drive.
2. Install the operating system on the target server:
   a. Insert the DOS-startable (bootable) diskette into the diskette drive of the target server.
   b. Restart the server so that it can start (boot) from the DOS-startable diskette and begin the ServerGuide Scripting Toolkit deployment process.
   c. When prompted or when the deployment process has completed, remove the DOS-startable diskette from the diskette drive of the target server.

Solving Red Hat Linux installation problems

You might have problems deploying Red Hat Enterprise Linux (RHEL) to certain IBM servers, because the Linux distributions do not contain the required device drivers. For example, some Red Hat Linux distributions currently supported by the ServerGuide Scripting Toolkit do not contain the device drivers required by the some of the following devices:

- The IDE RAID controller in the BladeCenter HS20, Machine Type 8832 server and the BladeCenter HS40, Machine Type 8839 server
- IBM ServeRAID-7e, Adaptec HostRAID controller
- IBM ServeRAID-7t, Serial-ATA (SATA) controller
- IBM ServeRAID-8i, SAS controller
- IBM LSI 1064x SAS controller

You might be able to resolve the problem by:
1. Adding a kernel parameter to the usrvars.bat file indicating the location of the driver disk
2. Modifying the kickstart file (RHEL) to include a driver disk for the hardware
3. Modifying the network boot image provided with the Linux distribution

To add a kernel parameter to the usrvars.bat file, follow these steps:
1. Obtain the new device driver diskette.

   You can download device drivers from the IBM Support Web site..

2. Make the device drivers accessible to the RHEL installation program from the network.

   For example, place the files on a Web server.

3. Edit the usrvars.bat file as you normally would with the Customize-usrvars.bat file.

4. In the usrvars.bat file, locate the following:

   ```
   rem echo nousbstorage > %RAMDSK%\SYSLINUX.TMP
   ```

5. To add the device driver to the kernel parameters, use the dd option.

   For example:

   ```
   echo nousbstorage dd=http://host/path/to/dd.img >
   %RAMDSK%\SYSLINUX.TMP
   echo nousbstorage dd=ftp://host/path/to/dd.img >
   %RAMDSK%\SYSLINUX.TMP
   echo nousbstorage dd=nfs:host/path/to/dd.img >
   %RAMDSK%\SYSLINUX.TMP
   ```

Perform the following procedure to modify the kickstart file for RHEL 3 and above, to include a driver disk for the hardware:

1. Obtain the new device driver diskette.

   You can download device drivers from the IBM Support Web site. You must provide device drivers that are compiled for post-installation kernels.

2.
   Copy the device-driver diskette image to a network accessible share.

3. Place a line in the kickstart file that specifies the location of the image.

   For example:

   ```
   driverdisk  --source=ftp://host/path/to/dd.img
   driverdisk  --source=http://host/path/to/dd.img
   driverdisk  --source=nfs:host:/path/to/img
   ```

Complete the following steps to modify the network boot image provided with the Linux distribution:

1. Obtain the new device driver. You can download device drivers from the IBM Support Web site.

2. Update the network boot image RAM disk with the device-driver files.

   The network boot image contains files that must be modified, as well as a gzipped CPIO (Copy Input to Output) archive in which the driver module ultimately resides.

Installing SUSE Linux Enterprise Server

This section describes the procedures required for performing a SUSE Linux Enterprise Server (SLES) deployment. You do not have to change the sample autoYaST files provided by the ServerGuide Scripting Toolkit to properly deploy SLES. You can choose, however, to modify certain aspects of the autoYaST files that are provided by the ServerGuide Scripting Toolkit, to customize your installations.

Preparing the autoYaST file

The following sample SUSE Linux Enterprise Server (SLES) autoYaST files are provided in the sgdeploy\sgtk\examples\linux\suse directory of the source tree:

AutoYaST file name	Linux distribution
sles9.xml	SUSE Linux Enterprise Server 9 SP 3 32-bit
sles9x64.xml	SUSE Linux Enterprise Server 9 SP 3 64-bit

You can customize several sections in the applicable autoYaST file to specify any settings that you intend to change. However, you do not have to modify the autoYaST file to successfully install SLES 9.

The autoYaST file does not require an installation media configuration. The media location is supplied by the OS_URL variable in the usrvars.bat file on the boot media, as described in "Modifying USRVARS.BAT" on page 20.

The ServerGuide Scripting Toolkit supports SLES installations from FTP or NFS servers only. HTTP installations of SLES are not supported currently.

For information about how to customize an autoYaST file, see the SLES documentation on the installation media or the SLES Web site..

Using a DOS-startable diskette and network share

You must have the following items to use this deployment scenario:
- One blank diskette
- A source system with a properly configured source tree
- Network connectivity and access to the source system directory structure from the target server
- (Optional) Network connectivity to the operating system installation files in a separate location on an NFS or FTP server.
- (Optional) *IBM UpdateXpress* CD-ROM

The SLES operating system installation files must be accessible through NFS or FTP, as required by the SLES installation program.

HTTP installations are not supported currently.

The ServerGuide Scripting Toolkit files must be accessible through the Server Message Block/Common Internet File System (SMB/CIFS) protocol. This can be from Samba on a Linux network share or a Windows shared directory.

Attention: The ServerGuide Scripting Toolkit utilities automatically delete the existing partitions on a target server prior to installation. This behavior is changed from the way installations were previously accomplished.

Complete the following steps to deploy a SLES operating system to a target server:
1. Create a DOS-startable (bootable) diskette:
    a. Insert a blank diskette into the diskette drive of the source system, or a Windows computer that you are using to connect to the source system.
    b. Navigate to the boot directory in the Scripting Toolkit source tree.
    c. Run the MakeDisk-Linux-Network-Install.bat helper batch file.

       See "Using helper batch files and creating diskette images" on page 17 for information about helper batch files.
    d. Run the Customize-netvars.bat helper batch file to customize the network environment variables for this deployment scenario.

       See "Modifying NETVARS.BAT" on page 19 for information about the environment variables you can configure.
    e. Run the Customize-usrvars.bat helper batch file to customize the environment variables for this deployment scenario.

       See "Modifying USRVARS.BAT" on page 20 for information about the environment variables you can configure.
    f. Customize the autoYaST file.

       See "Preparing the autoYaST file" on page 44 for more information.
    g. Label the DOS-startable (bootable) diskette so that you can easily identify it later.

The autoYaST file does not require an installation media configuration. The media location is supplied by the OS_URL variable in the usrvars.bat file on the boot media, as described in "Modifying USRVARS.BAT" on page 20.

The ServerGuide Scripting Toolkit supports SLES installations from FTP or NFS servers only. HTTP installations of SLES are not supported currently.

When you have completed the diskette, perform these steps on the target server:

1. (Optional) Update the BIOS code and firmware code on the target server:
   a. Start the target server and insert the *IBM UpdateXpress* CD-ROM into the CD-ROM drive.
   b. When IBM Update*Xpress* starts, follow the instructions on the screen to update the BIOS code and firmware code.
   c. When all updates are completed, remove the *IBM UpdateXpress* CD-ROM from the CD-ROM drive.
2. Install the operating system on the target server:
   a. Insert the DOS-startable (bootable) diskette into the diskette drive of the target server.
   b. Restart the server so that it can start (boot) from the DOS-startable diskette and begin the ServerGuide Scripting Toolkit deployment process.
   c. When prompted or when the deployment process has completed, remove the DOS-startable diskette from the diskette drive of the target server.

Solving SUSE Linux installation problems

You might have problems deploying SUSE Linux Enterprise Server (SLES) to certain IBM servers, because the Linux distributions do not contain the required device drivers. For example, some SLES distributions currently supported by the ServerGuide Scripting Toolkit do not contain the device drivers required by some of the following devices:

- IBM ServeRAID-7e, Adaptec HostRAID Controller
- IBM ServeRAID-8e, Adaptec HostRAID Controller
- Adaptec 9410/9405 SAS Controller

You might be able to resolve the problem by taking the following steps:

1. Add a kernel parameter to prompt for a driver diskette.

   Use the following procedure to add a kernel parameter:
   a. Edit the usrvars.bat file as you normally would with the Customize-usrvars.bat file.
   b. In the usrvars.bat file, locate the following line:

      `rem echo > %RAMDSK%\USRVARS.TMP`

   c. To add the device driver to the kernel parameters, remove the "rem" and use the driverupdate option. For example, change the line to:

      `echo driverupdate=1 > %RAMDSK%\USRVARS.TMP`

   d. Create a driver diskette.
      1) Obtain a blank floppy diskette.
      2) Browse to the IBM Support Web site.
      3) Download the device driver for the operating system that you are installing.
      4) Use a utility such as rawwrite or rawwritewin to extract the disk image to the blank floppy diskette.

   e. Insert the driver diskette into the machine when prompted by the YaST installer.
   f. A message is displayed that indicates that the drivers loaded correctly. Click **OK** to close the confirmation message.
   g. Press **Esc** to continue the installation.
2. Modify the network boot image provided with the Linux distribution.

Complete the following steps to modify the network boot image provided with the Linux distribution:

a. Obtain the new device driver.

   Download device drivers from the IBM Support Web site.

b. Update the network boot image RAM disk with the device-driver files.

   The network boot image contains files that you must modify, as well as a gzipped CPIO (Copy Input to Output) archive in which the driver module ultimately resides.

Chapter 7. Performing secure data disposal

This chapter contains information about using the ServerGuide Scripting Toolkit to perform secure data disposal on IBM eServer or IBM eServer xSeries servers. The ServerGuide Scripting Toolkit can perform the following tasks:

- Reset the target server persistent-state information
- Detect machine type and model, serial number, and other information about the target server hardware
- Reset supported RAID controllers and adapters in the target server to factory-default settings
- Delete all partitions on hard disks that are connected to the target server
- Erase data from one or all hard disks that are connected to the target server

You must have the following items to perform secure data disposal:

- One blank diskette
- One blank CD
- Source system with properly configured source tree and a CD-R or CD-RW drive
- CD creation software

Blade servers and USB-attached CD drives require bootable CDs.

Complete the following steps to securely dispose of data on a target server. You can use the data CD and DOS-startable diskette on all servers that do not require a startable (bootable) CD.

1. Create a DOS-startable (bootable) diskette:
   a. Insert a blank diskette into the diskette drive of the source system.
   b. Navigate to the boot directory in the source tree.
   c. Run the MakeDisk-Disposal.bat helper batch file. See "Using helper batch files and creating diskette images" on page 17 for information about this file.
   d. Customize the environment variables for this deployment scenario by running the Customize-usrvars.bat helper batch file. See "Modifying USRVARS.BAT" on page 20 for information about the environment variables that you can configure.
   e. Label the DOS-startable (bootable) diskette so that you can easily identify it later.
2. Create the data or DOS-startable CD:
   a. When creating a DOS-startable CD, use the CD creation software to set the boot image to the DOS-startable (bootable) diskette that you created previously.
   b. Using the CD creation software, create a data CD that contains at least the following directories. Remember that directories cannot be more than eight levels deep and must comply with DOS 8.3 file-naming conventions. Make sure that you enable the use of all DOS 8.3 file names in the CD creation software.

CD directory	Purpose
sgtk	To contain the ServerGuide Scripting Toolkit files

Copy directories and files from the ServerGuide Scripting Toolkit source tree to the directories you create on the CD as indicated below:

Copy from: Source tree location	Copy to: CD location
sgdeploy\sgtk	sgtk

   c. Label the CD so that you can easily identify it later.

When you have completed the CD, perform these steps on the target server:

1. Perform secure data disposal on the target server:

    a. Insert the DOS-startable (bootable) CD into the CD drive of the target server, or insert the DOS-startable diskette into the diskette drive and the data CD into the CD drive. Turn on the server as required.

    b. Restart the server so that it can start (boot) from the DOS-startable CD or diskette and begin the Scripting Toolkit secure data disposal process.

    c. If prompted, remove the DOS-startable CD or the DOS-startable diskette and data CD from the target server.

More information: See "SCRUB3.EXE" on page 119 for more information about performing secure data disposal on the target server hard disks.

Chapter 8. Altiris Deployment Solution support

This chapter contains information about using the ServerGuide Scripting Toolkit with the Altiris Deployment Solution. See the readme.htm file that comes with the ServerGuide Scripting Toolkit for a current list of servers that are supported by the Altiris Deployment Solution. You can open this file with Internet Explorer. If you have not already done so, see "Chapter 2. Installing files and configuring the source tree" on page 7 for preliminary instructions.

The Altiris Deployment Solution must be installed and configured before you perform any of the tasks described in this chapter.

At the time of this publication, you can install the Altiris Deployment Solution, Windows Version, on the following operating systems:
- Microsoft Windows 2000 Server
- Microsoft Windows Server 2003

At least a basic understanding of how to use the Altiris Deployment Solution is required. See the Altiris Deployment Solution documentation or see the Altiris Web site, http://www.altiris.com, for detailed information.

The ServerGuide Scripting Toolkit can be used with the Altiris Deployment Solution to perform the following tasks:
- Erase data from a hard disk on a target server
- Reset supported RAID adapters or controllers in the target server to factory-default settings
- Configure RAID adapters or controllers in the target server
- Update server BIOS code on the target server
- Perform Scripted Installs of supported Windows operating systems and Linux operating systems
- Perform Image Installs of supported Windows operating systems and Linux operating systems
- Install IBM Director Agent (Windows operating systems only)

The ServerGuide Scripting Toolkit sample jobs for the Scripted Install method automatically install the corresponding Altiris Agent. If you do not use one of the sample jobs, you must copy the applicable installation file to the target server; then, run the file to install the agent. The Altiris Deployment Solution provides aclient.exe for Windows operating systems, and adlagent for Linux operating systems.

Preparing the answer file for an unattended installation

Two sample answer files, win2000.txt and win2003.txt, come with the ServerGuide Scripting Toolkit. These sample files are located in the sgdeploy\sgtk\examples\windows\ansfile directory of the source tree. Microsoft also includes a sample answer file in the i386 directory of the operating-system installation files.

You must add information to the [UserData] section of the answer file, including a value for the ProductID variable, when required by the operating system you are deploying. You can add information manually to the answer file before you begin the deployment process, or dynamically during the deployment process. See "Updating the answer file dynamically" on page 127 for information about dynamically updating the answer file during the deployment process.

Do not add device-driver information to the answer file. The ServerGuide Scripting Toolkit dynamically adds device-driver information to the answer file during the deployment process.

For information about customizing the answer file, see the Microsoft documentation in the support\tools\deploy.cab file on the operating-system installation CDs.

Importing the Altiris-format binary for Windows source servers

You must install and configure the Altiris Deployment Solution before you install the ServerGuide Scripting Toolkit files. When you use the Toolkit InstallShield wizard to integrate with the Altiris Deployment Solution, the wizard imports the Altiris format binary for Windows.

Using the Altiris Tools menu

When you install and configure the Altiris Deployment Solution, the Tools menu does not contain a menu item named IBM Tools. After you install the ServerGuide Scripting Toolkit, the Tools menu in the Altiris Deployment Solution contains a new entry named **IBM Tools**.

The IBM Tools menu contains the following entries:
- **ServerGuide Scripting Toolkit Directory** - Opens an Explorer window of the sgdeploy directory.
- **ServerGuide Scripting Toolkit Configuration Program** - Starts the Toolkit Configuration Program.
- **User's Reference** - Opens this ServerGuide Scripting Toolkit UserGuide.pdf with Abode Acrobat Reader. You receive an error message if Acrobat Reader is not installed.
- **IBM xSeries Support Page** - Opens an Internet Explorer window directly to the IBM Support Page at http://www.ibm.com/pc/support/us.
- **Readme** - Opens the ServerGuide Scripting Toolkit readme.htm using Internet Explorer.
- **IBM Service Processor Discovery** - Opens the IBM Service Processor Discovery Tool (available in Altiris 6.5 only).

Using the ServerGuide Scripting Toolkit sample jobs

By using the sample jobs that come with the ServerGuide Scripting Toolkit, you can perform the following tasks on a target server as part of an Altiris Deployment Solution deployment scenario:
- Configure hardware, including:
  - Configuring RAID and Fibre
  - Capturing and deploying RAID configurations
  - Server disposal, including resetting RAID
- Capture, configure, and deploy supported operating system images.
- Scripted installation of supported operating systems, including configuring RAID.
- System update events:
  - Post-OS Windows updates (from a Windows server only)
  - RemoteUX driver and firmware update (from a Windows server only)
  - Capture CMOS settings to a binary file
  - Deploy CMOS settings from a binary file
  - Perform BIOS and ServeRAID BIOS updates
- Directory test (from a Windows server only)

Customizing ServerGuide Scripting Toolkit sample jobs

The Altiris Scripted Install sample jobs that come with the ServerGuide Scripting Toolkit use many of the same batch files as other ServerGuide Scripting Toolkit scenarios. The Altiris Deployment Solution does not, however, permit the ServerGuide Scripting Toolkit to customize the virtual boot diskette that it uses during the deployment process. This means that you cannot customize the usrvars.bat file with specific settings, as you can with other ServerGuide Scripting Toolkit scenarios.

The sample jobs that come with the ServerGuide Scripting Toolkit contain specific settings for some of the variables that are found in the usrvars.bat file. These settings override the settings from the usrvars.bat file that are included in the virtual boot diskette.

To customize a sample job, you must modify the provided sample jobs from within the Altiris Deployment Solution interface. Each sample job can contain a different setting for the same variable. If you intend to run multiple jobs on the same target server, you might have to modify each sample job to contain the same variable settings.

Attention: If you make changes to an Altiris job or event, the changes are effective for all target servers that run the job or event. Do not make changes without creating a new job or event, unless you are sure that you want to affect all target servers that are scheduled to run that job or event.

Installing IBM Director Agent for Windows

This sample job will perform an installation of the IBM Director Agent on an IBM BladeCenter, IBM eServer, or IBM eServer xSeries server that has an installed Windows operating system.

In order to perform this task using the Altiris Deployment Solution interface, set up the necessary IBM Director Agent files on the Altiris server before beginning the task. See "Adding application (IBM Director Agent for Windows) files" on page 12 for details.

There is no customization available or necessary for this task.

Updating server BIOS code

The sample job for BIOS code updates that come with the ServerGuide Scripting Toolkit is customized for use with IBM eServer and xSeries servers. You can override some of the environment variables in usrvars.bat by defining new values in the Altiris Deployment Solution interface.

See the Altiris Deployment Solution documentation for information about how to modify jobs or events. See "Modifying USRVARS.BAT" on page 20 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Variable name	Description	Default
UPD_BDIR	The base directory that contains the BIOS code and update files in the source tree.	%MACHTYPE%\bios
UPD_BIOS	Specifies whether to perform BIOS code updates. Valid values are Y or N.	Y
UPD_PATH	The fully qualified path to the BIOS code and firmware code update files in the source tree.	%TAS_PATH%\updates (f:\sgdeploy\updates)

Capturing and deploying CMOS settings

The sample jobs for capturing of CMOS settings to a binary file and deploying CMOS settings from a binary file that come with the ServerGuide Scripting Toolkit are customized for use with IBM eServer and xSeries servers. You can override some of the environment variables in usrvars.bat by defining new values in the Altiris Deployment Solution interface.

The capture task gathers the CMOS settings into a binary file whose name and location are determined by the CMOS_FILE and CMOS_PATH variables that are described in the table.

Some xSeries servers that have different machine types use the same BIOS. For example, the following servers might be affected.

Model number	Machine types
xSeries 335	8676, 8830
xSeries 365	8861, 8862

The setup of the system BIOS firmware must be duplicated for these tasks to work correctly. To duplicate the CMOS settings for the two machine types, use either of the following procedures:

- Run the capture task once for each machine type.
- Run the capture task for one machine type, and then manually create the second machine type directory and copy the binary file that the task created into the second machine type directory.

See the Altiris Deployment Solution documentation for information about how to modify jobs or events. See "Modifying USRVARS.BAT" on page 20 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Variable name	Description	Default
CMOS_FILE	The name of the CMOS settings binary file.	cmosinfo.bin File names must be in DOS-compatible 8.3 notation.
CMOS_PATH	The fully qualified path to the CMOS binary file in the source tree.	%TK_PATH%\altiris\temp\%MACHTYPE%
UPD_BDIR	The base directory that contains the BIOS code and update files in the source tree.	%MACHTYPE%\bios
UPD_PATH	The fully qualified path to the BIOS code and firmware code update files in the source tree.	%TAS_PATH%\updates (f:\sgdeploy\updates)

Updating drivers or firmware with RemoteUX (Windows only)

The sample jobs for RemoteUX driver or firmware updates are customized for use with IBM eServer and xSeries servers. You can override some of the environment variables in usrvars.bat by defining new values in the Altiris Deployment Solution interface.

Refer to the Altiris Deployment Solution documentation for information about how to modify jobs or events. You can use the following ServerGuide Scripting Toolkit environment variables in the Altiris script to override processing parameters:

Variable name	Description	Default
LOGON	Defines a user name that can log onto the Altiris source server during the update process. This user does not have to belong to the Administrator user group.	sgtk0clnt
PWD	The password of the logon account on the source server. The password cannot be empty.	sgtk0clnt
ALTSRV	Defines the computer name of the Altiris Deployment Solution server	sgstksrv
UXCD	Defines the name of the shared folder which contains remoteux.exe utility and the necessary update files	uxcd

In addition to the customization variables in the main script, provide a valid user name and password with Administrative privileges for an account on the target server. The Administrator account runs this script on the target machine. Set the Administrator group user ID and password under the **Advanced** tab in the RunScript dialog window.

Once the "Advanced Script Options" window opens, update the following sections under the Security Context (as defined in Altiris Deployment Solution 6.1) section:

- User name
- Password
- Confirm password

Important:

1. The IBM RemoteUX update process requires Administrator-level privileges to complete successfully. Because of this, you must log on using an ID that has administrative privileges. Therefore, by default, the variables as follows:
   - User name = Administrator
   - Password and Confirm password = password (encrypted)
2. The Administrator password must be set.

Prerequisites before using RemoteUX with Altiris

1. Set up the necessary UpdateXpress files on the Altiris server.
2. Install a Windows operating system on the target system that is supported by RemoteUX.
3. Install the Altiris agent and verify that it runs on the target server.
4. Obtain an Administrator group user name and password on the target server.
5. Obtain a user name and password with at least read and execute privileges for the Altiris source server.

Error checking

- The Altiris scripts check for the existence of the RemoteUX utility and display an appropriate message if the utility is not installed.
- The Altiris scripts check the return code RemoteUX runs and display an appropriate message when an error is returned.

See "Using RemoteUX for Windows with Altiris Deployment Solution" on page 133 for information about how to perform your post-installation updates with the Altiris Deployment Solution.

Updating ServeRAID BIOS and firmware

The sample job for IBM ServeRAID BIOS and firmware updates that comes with the ServerGuide Scripting Toolkit is customized for use with IBM eServer and xSeries servers. You can override some of the environment variables in usrvars.bat by defining new values in the Altiris Deployment Solution interface.

See the Altiris Deployment Solution documentation for information about how to modify jobs or events. See "Modifying USRVARS.BAT" on page 20 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Variable name	Description	Default
UPD_PATH	The fully qualified path to the BIOS code and firmware code update files in the source tree.	%TAS_PATH%\updates (f:\sgdeploy\updates)
UPD_SDIR	Defines the base directory that contains the IBM ServeRAID update files in the source tree. Used for adapters 4H, 4Mx, 4Lx, 5i, 6i, 6i+, 6M, 7k	SRVRAID

Variable name	Description	Default
UPD_SR	Specifies whether to perform ServeRAID BIOS and firmware updates. Valid values are Y and N.	Y
UPD_SV8I	Defines the base directory that contains the IBM ServeRAID update files in the source tree. Used for adapter 8i.	svraid8i

Update of the ServeRAID 8i controller is not supported by this sample job for Altiris Deployment Solution for Linux.

Configuring RAID

The RAID configuration is driven by the PRAID utility. See "PRAID.EXE" on page 101 for more information.

You can override some of the environment variables in usrvars.bat by defining new values in the Altiris Deployment Solution interface. See the Altiris Deployment Solution documentation for information about how to modify jobs or events. See "Modifying USRVARS.BAT" on page 20 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Variable name	Description	Default
CFG_RD	Specifies whether to perform RAID configuration. Valid values are Y and N.	Y
RD_PATH	The fully qualified path to the RAID configuration files in the source tree.	%TK_PATH%\examples\raid (f:\sgdeploy\sgtk\examples\raid)
RD_FILE	The name of the PRAID policies file. If no filename is specified, the RAID is configured using the default values. For more information about the PRAID policies file, see "Policies file" on page 108.	No name specified
RD_ERR1	Indicates whether the installation should stop if at least 1 RAID controller cannot be configured because no policy is available. Valid values are: • N - do not stop the installation • Y - stop the installation	N
RD_ERR2	Indicates whether the installation should stop if no supported RAID controllers are found in the system. Valid values are: • N - do not stop the installation • Y - stop the installation	N

Capturing and deploying RAID configurations

The sample jobs for capturing RAID configuration to a Policies file and deploying RAID configuration are customized for use with IBM BladeCenter and xSeries servers.

The Capture RAID configuration task determines how PRAID would configure the target RAID controller and creates a policies file to perform that configuration.

You can use the deploy RAID configuration task to deploy a RAID configuration to a RAID controller or adapter. The two systems must have the same number of hard disk drives attached to the same type of controller.

Refer to the Altiris Deployment Solution documentation for information about how to modify jobs or events. See "Modifying USRVARS.BAT" on page 20 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Variable name	Description	Default
RD_FILE	The name of the RAID configuration file.	raidclon.ini File names must be in DOS-compatible 8.3 notation.
RD_PATH	The fully qualified path to the RAID configuration file.	%TK_PATH%\altiris\temp\%MACHTYPE%
RD_ERR1	Indicates whether the task should stop if at least one RAID controller cannot be configured because no policy is available. Valid values are: • *N* - do not stop the installation • *Y* - stop the installation This variable only applies to the Deploy task.	N
RD_ERR2	Indicates whether the task should stop if no supported RAID controllers are found in the system. Valid values are: • *N* - do not stop the installation • *Y* - stop the installation	Y

Configuring Fibre

This sample job performs a Fibre configuration for the IBM BladeCenter and xSeries servers.

To customize this task, the following usrvars.bat environment variables are provided in the Scripting Toolkit sample job:

Variable name	Description	Default
DO_FIB	Indicates whether to configure Fibre as a boot device. Valid values are: • N - do not perform Fibre configuration • Y - perform Fibre configuration	N
FIB_IO	Specifies the I/O address of the fibre HBA that you want to configure. If you set this value to four zeros (0000) then the first adapter in the system scan order (the one with the lowest I/O address) will be configured. Valid values are 4-digit numbers.	0000
FIB_LUN	Specifies the LUN number on the storage device where the OS should be deployed. Valid values are integers from 0-30.	0
FIB_WWN	Specifies the 16-digit hexadecimal world-wide name of the storage device where the boot LUN is located (case-insensitive). Valid values are 16 digit hexadecimal numbers. If you set this value to 16 zeros, then the first boot LUN found will be automatically selected.	0000000000000000

Customizing Scripted Installs for Windows

The sample jobs for Scripted Installs of Windows operating systems that come with the ServerGuide Scripting Toolkit are customized for the specific version of the operating system that is being deployed. You can override some of the environment variables in usrvars.bat by defining new values in the Altiris Deployment Solution interface.

See the Altiris Deployment Solution documentation for information about how to modify jobs or events. See "Modifying USRVARS.BAT" on page 20 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Variable name	Description	Default
ANS_FILE	The file name for the unattended installation answer file.	win2000.txt For Windows Server 2003 sample jobs, the default is win2003 instead of win2000.
ANS_PATH	The fully qualified path to the directory that contains ANS_FILE.	%TK_PATH%\altiris\%NOSDIR%\ansfile (f:\sgdeploy\sgtk\altiris\windows\ansfile)
DSET_PATH	The fully qualified path to the IBM ServerGuide device drivers index file, DRVSET.INI	%TAS_PATH%\drvs\w2k_drv (f:\drvs\w2k_drv) For Windows Server 2003 sample jobs, the default is w23_drv instead of w2k_drv.
OS_PATH	The fully qualified path to the operating-system installation files directory in the source tree.	%TAS_PATH%\os\w2k_srv (f:\sgdeploy\os\w2ksrv) For Windows Server 2003 sample jobs, the default is w23_std instead of w2k_srv for the above values.
PART_SZ	The target server operating system partition size (in MB) for the primary partition.	8000
DAINST	Specifies whether to install IBM Director Agent. Valid values are Y or N.	N
DA_PATH	Defines the fully qualified path to the IBM Director Agent installation files on the source system.	f:\sgdeploy\apps\dawin

Customizing Scripted Installs for Red Hat Linux

The sample jobs for Scripted Installs of Red Hat Enterprise Linux (RHEL) operating systems that come with the ServerGuide Scripting Toolkit are customized for the specific version of the RHEL AS or ES operating system that is being deployed. You can override some of the environment variables in usrvars.bat by defining new values in the Altiris Deployment Solution interface.

The following sample kickstart files are provided in the sgdeploy\sgtk\examples\linux\redhat directory of the source tree.

Kickstart file name	Linux distribution
rhas3ks.cfg	Red Hat Enterprise Linux AS 3 32-bit
rhas3x64.cfg	Red Hat Enterprise Linux AS 3 64-bit
rhes3ks.cfg	Red Hat Enterprise Linux ES 3 32-bit
rhes3x64.cfg	Red Hat Enterprise Linux ES 3 64-bit

Kickstart file name	Linux distribution
rhas4ks.cfg	Red Hat Enterprise Linux AS 4 32-bit
rhas4x64.cfg	Red Hat Enterprise Linux AS 4 64-bit
rhes4ks.cfg	Red Hat Enterprise Linux ES 4 32-bit
rhes4x64.cfg	Red Hat Enterprise Linux ES 4 64-bit

By default, the example batch files use the kickstart files placed in the sgtk\examples\linux\redhat directory on the source system. Be sure to customize the Installation Media Configuration section within the appropriate kickstart file to specify the network protocol and location of the operating-system files. The Installation Media Configuration section in the example kickstart files is denoted by a comment (line beginning with '#') that contains the text *Installation Media Configuration*, followed by a brief description. For example, the Red Hat Enterprise Linux AS 3 sample kickstart file contains the following text:

```
#########################################################
# Installation Media Configuration
#
# The following lines should be equivalent to:
# HTTP: url --url http://%SERVER_IP%/path/to/OS/files
# NFS:  nfs --server %SERVER_IP% --dir /path/to/OS/files
#########################################################

# nfs --server 192.168.0.25 --dir /var/www/rhas3
url --url http://192.168.0.25/rhas3/
```

Table 10. Server mode statements in a kickstart file

Kickstart statement	Description
nfs --server 192.168.0.22 --dir /var/www/rhas3	The NFS server mode specifies that the Red Hat Linux installation program uses NFS to download the operating system files. Specifies that the operating system files are located in the /var/www/rhas3 directory of an NFS server with IP address 192.168.0.25 The IP address, 192.168.0.22, is the network address of the NFS server. The operating system files are located in the /var/www/rhas3 directory on the server.
url --url http://192.168.0.25/rhas3/	The URL server mode specifies that the RHEL installation program uses HTTP to download the operating system files from a directory on a Web server. The IP address, 192.168.0.25, is the network address of a Web server, such as Microsoft Internet Information Server (IIS). The operating system files are located in the rhas3 directory of the Web server, or in the virtual directory on IIS.

For more information about customizing the kickstart files, see the Red Hat Linux documentation on the installation media or the Red Hat Linux Web site at http://www.redhat.com. You also can refer to the kickstart HOWTO available from the Linux Documentation Project Web site at http://www.tldp.org.

Note: RHEL 32-bit installations install the Altiris Agent. However, RHEL x64-bit installations do not install the Altiris Agent.

See the Altiris Deployment Solution documentation for information about how to modify jobs or events. See "Modifying USRVARS.BAT" on page 20 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Variable name	Description	Default
ANS_FILE	The file name of the kickstart file.	rhas3ks.cfg
ANS_URL	The kickstart file location that is passed to the syslinux utility. See the "Starting a Kickstart Installation" section of the *Red Hat Linux Customization Guide* for more information.	http://%SRV_IP%/%TASDIR%/ %TKDIR%/altiris/linux/redhat/ %ANS_FILE% (http://192.168.0.25/ sgdeploy/sgtk/altiris/linux/redhat/ rhas3ks.cfg)
DPY_NIC	The network adapter on the target server to use during the deployment process. You must define this variable when more than one network adapter is installed in the target server.	eth0
IRD_FILE	The file name of the network installation RAM disk image file. **Note:** File names must be in DOS-compatible 8.3 notation.	initrd~1.img
IRD_PATH	The fully qualified path to the file specified by IRD_FILE.	%OS_PATH%\images\pxeboot (f:\sgdeploy\os\rhas3\images\ pxeboot)
KRL_FILE	The file name of the network installation kernel file.	vmlinuz
KRL_PATH	The fully qualified path to the file specified by KRL_FILE.	%OS_PATH%\images\pxeboot (f:\sgdeploy\os\rhas3\images\ pxeboot)
SL_PATH	The fully qualified path to the syslinux utility.	%TAS_PATH%\osutils (m:\sgdeploy\osutils)
OS_PATH	The fully qualified path to the operating-system installation files directory in the source tree.	%TAS_PATH%\os\rhas3 (f:\sgdeploy\os\rhas3) %TAS_PATH%\os\rhes3 %TAS_PATH%\os\rhas4 %TAS_PATH%\os\rhes4
OS_TYPE	The manufacturer type of the Linux operating system.	SUSE *or* **REDHAT**
OS_URL	The fully qualified URL to the RHEL operating system directory.	ftp://%SRV_IP%/%TASDIR%/os/ rhas3/
RDSK_SZ	Defines the size (in KB) of the Linux RAM disk to use.	65536
echo nousbstorage > %RAMDSK%\SYSLINUX.TMP	Defines any additional kernel parameters to pass to the Linux kernel prior to booting. All kernel options must be separated by a space.	echo nousbstorage > %RAMDISK%\SYSLINUX.TMP

Customizing scripted installations for SUSE Linux

Scripting Toolkit sample jobs for scripted installations of SUSE Linux Enterprise Server (SLES) operating systems deploy the 32-bit or 64-bit SLES 9 operating system. The 32-bit installation can also install the Altiris Agent. You can use the Altiris Deployment Solution interface to override some of the environment variables specified in the Scripting Toolkit usrvars.bat file.

The following sample SUSE Linux Enterprise Server (SLES) autoYaST files are provided in the sgdeploy\sgtk\examples\linux\suse directory of the source tree:

AutoYaST file name	Linux distribution
sles9.xml	SUSE Linux Enterprise Server 9 SP 3 32-bit
sles9x64.xml	SUSE Linux Enterprise Server 9 SP 3 64-bit

Note: SLES 32-bit installations install the Altiris Agent. However, SLES x64-bit installations do not install the Altiris Agent. To successfully install the Altiris Agent during 32-bit SLES 9 installations, identify the correct IP address of your Altiris Deployment Server. Replace `192.168.0.1` with the correct IP address in each of the following files. When both files are correct, the scripted installation installs the Altiris Agent after installing the operating system files:

- `sgdeploy\sgtk\altiris\linux\suse\sles9.xml`

  Locate and change the IP address in the `sles9.xml` file:

  ```
  #!/bin/sh
  cd /mnt/etc/rc.d/rc5.d
  wget http://192.168.0.1/sgdeploy/sgtk/altiris/linux/suse/S99adlagent.sh
  chmod 755 S99adlagent.sh
  cd /
  ```

- `sgdeploy\sgtk\altiris\linux\suse\S99adlagent.sh`

  Locate and change the IP address in the `S99adlagent.sh` file:

  ```
  SERVERIP=192.168.0.1
  ```

See the Altiris Deployment Solution documentation for information about how to modify jobs or events. You can override the following ServerGuide Scripting Toolkit `uservars.bat` environment variables.

Variable name	Description	Default
ANS_FILE	The file name of the autoYaST file.	sles9.xml
ANS_URL	The autoYaST file location that is passed to the syslinux utility.	http://%SRV_IP%/%TASDIR%/%TKDIR%/altiris/linux/suse/%ANS_FILE% (http://192.168.0.25/sgdeploy/sgtk/altiris/linux/suse/sles9.xml)
DPY_NIC	The network adapter on the target server to use during the deployment process. You must define this variable when more than one network adapter is installed in the target server.	eth0
IRD_FILE	The file name of the network installation RAM disk image file. **Note:** File names must be in DOS-compatible 8.3 notation.	initrd~1.img
IRD_PATH	The fully qualified path to the file specified by IRD_FILE.	%OS_PATH%\images\pxeboot (f:\sgdeploy\os\sles9\images\pxeboot)
KRL_FILE	The file name of the network installation kernel file.	vmlinuz
KRL_PATH	The fully qualified path to the file specified by KRL_FILE.	%OS_PATH%\images\pxeboot (f:\sgdeploy\os\sles9\images\pxeboot)
SL_PATH	The fully qualified path to the syslinux utility.	%TAS_PATH%\osutils (m:\sgdeploy\osutils)

Variable name	Description	Default
OS_PATH	The fully qualified path to the operating-system installation files directory in the source tree.	%TAS_PATH%\os\sles9 (f:\sgdeploy\os\sles9)
RDSK_SZ	Defines the size (in KB) of the Linux RAM disk to use.	65536
echo nousbstorage > %RAMDSK%\SYSLINUX.TMP	Defines any additional kernel parameters to pass to the Linux kernel prior to booting. All kernel options must be separated by a space.	echo nousbstorage > %RAMDISK%\SYSLINUX.TMP

Performing Image Installs

The Image Install sample jobs that come with the ServerGuide Scripting Toolkit capture an operating-system image from a donor server and deploy that image to a target server. You might have to perform some configuration, such as RAID configuration, on the target server before you can deploy an operating system to it.

To perform an Image Install, the Altiris Agent must be installed on the server from which you will clone the installation (donor server). The ServerGuide Scripting Toolkit sample jobs for the Scripted Install method automatically install the corresponding Altiris Agent. If you do not use one of the sample jobs, you must copy the applicable installation file to the target server; then, run the file to install the agent. The Altiris Deployment Solution provides aclient.exe for Windows operating systems and adlagent for Linux operating systems.

Remember:

1. When you deploy a target server by using the Image Install method, it will have the Altiris Agent installed, because this is a requirement of the donor server.

2. By default, the Linux clone script will create a file named `%COMPNAME%.img` in the images directory on the deployment server. Because the Altiris cloning utility runs under DOS, you might encounter errors while cloning if the computer name is a string that is not compliant with DOS file naming conventions, such as `localhost.localdomain`. One possible workaround is to set the host name of your system to a value that adheres to the DOS convention via the kickstart/autoYaST files. For more information, see the sample Altiris kickstart file provided in the `sgdeploy\sgtk\altiris\linux\redhat` directory or the Altiris autoYaST file provided in the `sgdeploy\sgtk\altirisl\linux\suse\` directory, both of which determine the host name of the installed machine.

Sample jobs are included for imaging Windows operating systems and Linux operating systems. All versions of those operating systems that are supported by the ServerGuide Scripting Toolkit are supported by the sample jobs for Altiris-based deployment scenarios.

The Linux imaging sample jobs utilize a sector-based imaging technique, so the target server must have identical hardware to that of the donor server.

Cloning of multiple disks in a single pass is not supported by Altiris RapiDeploy. You can work around this issue by placing all Linux partitions on a single disk during installation by modifying the example kickstart files to included the `--ondisk` directive to the partition configuration *part* lines please refer to the Red Hat documentation for further information).

To specify the disk to be cloned, you must add an addition command-line parameter to the Deploy Linux Image and Capture Linux Image ServerGuide Scripting Toolkit for Altiris Deployment Solutions jobs. For example, to clone the operating system image on disk 2, you would add the following parameter to the command-line options for the job:

`-d2`

Support for Altiris system tokens

System tokens are variables that you can insert in scripts or answer files to extract information from the Altiris Deployment Database. System tokens are case-sensitive. The Altiris Deployment Solution provides the following sample answer files, which contain examples of how to use system tokens:

- unattend.txt for Windows, located in the *AltirisPath*\Altiris\eXpress\Deployment Server\Samples\Scripted OS Install\windows directory, where *AltirisPath* is the path to where the Altiris Deployment Solution is installed
- kickstart for Red Hat Enterprise Linux (RHEL), located in the *AltirisPath*/altiris/express/Deployment Server/Samples/Scripted OS Install/redhat directory

The ServerGuide Scripting Toolkit sample answer files (win2000.txt for Windows 2000, and win2003.txt for Windows Server 2003) and kickstart files for Red Hat Enterprise Linux (RHEL) also contain examples of how to use system tokens.

The ServerGuide Scripting Toolkit sample files can use system tokens in the Windows and Linux Scripted Install methods. See the *Altiris Deployment Solution Product Guide* for a complete listing of supported, predefined system tokens.

Remember:

1. System tokens are not supported with the Image Install method.
2. System tokens are substituted only in the operating system answer files (for Windows) and kickstart files for RHEL.
3. System tokens are not supported for autoYaST files for SUSE Linux Enterprise Server (SLES).

Chapter 9. Microsoft Automated Deployment Services support

This chapter contains information about using the ServerGuide Scripting Toolkit with existing deployment processes based on Microsoft Automated Deployment Services (ADS) on IBM eServer or IBM eServer xSeries servers. If you have not already done so, see "Chapter 2. Installing files and configuring the source tree" on page 7 for preliminary instructions.

Microsoft ADS runs only under Microsoft Windows Server 2003 operating systems.

The ServerGuide Scripting Toolkit must be installed on the same source system as Microsoft ADS.

The ServerGuide Scripting Toolkit can be used with Microsoft ADS to perform the following tasks:
- Erase data from a hard disk on a target server.
- Configure supported RAID adapters and controllers in the target server.
- Update Server BIOS code and IBM ServeRAID BIOS and firmware code on the target server.
- Install IBM Director Agent.
- Add device drivers to the Microsoft ADS device-driver set.

Performing secure data disposal

Secure data disposal performs the following tasks:
- Deletes all partitions
- Erases data on all hard drives
- Performs medium level (level 2) security clean up

More information: See "SCRUB3.EXE" on page 119 for more information about performing secure data disposal on the target server hard disks.

Complete the following steps to perform server disposal using ADS:
1. Open the ADS Management Console.
2. Run the pre-created job template labeled IBM Scripting Toolkit Disposal.

The above job template runs on the target system independently and does not require user intervention during the disposal process. This job causes the target server to reboot once before the disposal process is completed.

The disposal job template uses a sample virtual floppy image that is installed with the ServerGuide Scripting Toolkit. You can edit the virtual floppy image with the Modify-ADS-Disposal.bat file. However, you should edit the file only if you must change default usrvar.bat values that control one of the following settings:
- Security method
- Security level or the number of writes
- Enabling the scrub signature
- Specifying a particular drive number to clean

To edit the sample virtual floppy image you need:
- A blank diskette
- A Windows Server 2003 Enterprise system with Service Pack 1 installed running Microsoft ADS
- The ServerGuide Scripting Toolkit installed on the same system as Microsoft ADS with the sample disks installed and a properly configured source tree

65

Perform the following procedure to modify the pre-created job template labeled IBM Scripting Toolkit Disposal and the sample virtual floppy disk:

1. Insert a blank diskette into the diskette drive of the source system.

2. Navigate to the sgtk\ads directory in the source tree.

3. Run the Modify-ADS-Disposal.bat helper batch file. See "Using helper batch files and creating diskette images" on page 17 for information about settings in this file.

4. A notepad window displays the usrvars.bat file in edit mode. Edit the settings for the disposal job to change the default behavior.

5. Close the notepad window and save the file onto the diskette to also replace the file in the virtual floppy disk with the version of the file that you just created.

6. Remove the diskette.

7. Open the ADS Management Console.

8. Run the recently updated job template labeled IBM Scripting Toolkit Disposal.

Configuring supported RAID adapters and controllers

This procedure configures all RAID controllers using default settings for all supported controllers found in the system.

You need the following items for this scenario:
- A Windows Server 2003 Enterprise system with Service Pack 1 installed running Microsoft ADS.
- The ServerGuide Scripting Toolkit installed on the same system as Microsoft ADS.

See "Chapter 4. Configuring RAID and Fibre Channel" on page 29 for information on how to create and customize files in the applicable directories of the source tree for specific RAID controllers.

Complete the following steps to perform a default RAID configuration using ADS:

1. Download the Info-ZIP unzip utility from the info-ZIP Web site: ftp://ftp.info-zip.org/pub/infozip/MSDOS/

2. Place the unzip utility in the `../sgdeploy/osutils/infozip/` directory.

3. Run the `Makedisk-ADS-RAID.bat` helper file located in the `../sgdeploy/sgtk/ads/` directory to configure the diskette image.

4. Open the ADS Management Console.

5. Run the pre-created job template labeled *IBM Scripting Toolkit RAID Configuration*.

The *IBM Scripting Toolkit RAID Configuration* job template uses a sample virtual floppy disk that erases any existing RAID configuration before setting up RAID to default settings according to the hardware configuration of the system.

You can, however, customize the environment variables on the provided virtual boot disk, and perform a different task with the modified diskette. Perform the following procedure to modify the pre-created job template labeled *IBM Scripting Toolkit RAID Configuration* and the sample virtual boot disk only if you must change usrvars settings that control:
- Specifying a RAID policies file
- Changing error conditions (ERR 1 or ERR 2)
- Specifying the RAID policies file path

Verify that the following items are available before editing the sample virtual boot image:
- A blank diskette
- A Windows Server 2003 Enterprise system with Service Pack 1 installed running Microsoft ADS.
- The ServerGuide Scripting Toolkit installed on the same system as Microsoft ADS.

If you must change the usrvars.bat file or the RAID policy file:

1. If you have not already run the `Makedisk-ADS-RAID.bat` helper file, perform this task first. See "Using helper batch files and creating diskette images" on page 17 for information about this helper batch file.

2. Create a customized policy file and save the file in the source tree, such as in the `\sgtk\examples\raid` directory.

3. Insert a blank diskette into the diskette drive of the source system.

4. Navigate to the sgtk\ads directory in the source tree.

5. Run the Modify-ADS-RAID.bat helper batch file. See "Using helper batch files and creating diskette images" on page 17 for information about this helper batch file.

6. A Notepad window displays the usrvars.bat file in edit mode. Edit the settings for the RAID configuration job, and save and close the file. See "Modifying USRVARS.BAT" on page 20 for information about the environment variables you can configure.

7. If you specify a policy file in the usrvars.bat file, a Notepad window opens displaying the contents of the policy file that is specified in the `usrvars.bat` file. If the policy file does not already exist, Notepad creates and opens a new file.

8. Delete the entire contents of the policy file that is open in the Notepad window.

9. Copy the entire contents of the customized policy file that you saved earlier.

10. Paste the contents into the policy file that you are editing in the Notepad window.

11. Close the Notepad window and save the file so that the virtual boot disk files are updated with the version of the files that you just edited.

12. Remove the diskette.

13. Open the ADS Management Console.

14. Run the pre-created job template labeled "IBM Scripting Toolkit RAID Configuration".

Updating server BIOS code and IBM ServeRAID BIOS and firmware code

This procedure updates the system BIOS and performs an update of the IBM ServeRAID BIOS and firmware code for all supported ServeRAID controllers.

Complete the following steps to perform a BIOS update and IBM ServeRAID BIOS and firmware updates using ADS:

1. Insert a blank diskette into the diskette drive of the source system.

2. Navigate to the sgtk\ads directory in the source tree.

3. Run the Modify-ADS-Updates.bat helper batch file.

   See "Using helper batch files and creating diskette images" on page 17 for information about this file.

4. A Notepad window displays the netvars.bat file in edit mode. Edit the network settings and save and close the file.

   See "Modifying NETVARS.BAT" on page 19 for information about the environment variables you can configure.

5. Another Notepad window displays the usrvars.bat file in edit mode. Edit the settings for the update environment variables for this deployment.

   See "Modifying USRVARS.BAT" on page 20 for information about the update environment variables that you can configure.

6. Save and close the usrvars.bat file to replace the virtual boot disk files with the version of the files that you just edited. The floppy boot disk files are modified. Then the ADS virtual boot image files are updated on the server.

7. Remove the diskette.

8. Open the ADS Management Console.
9. Run the newly updated job template labeled "IBM Scripting Toolkit Updates" against the target server.

The above job template uses a virtual floppy disk to perform a system BIOS update and the update of all supported ServeRAID controllers found on the target server.

Requirements: You must have the following items to use this scenario:

- A Windows Server 2003 Enterprise system with Service Pack 1 installed running Microsoft ADS
- The ServerGuide Scripting Toolkit installed on the same system as Microsoft ADS with the sample disks installed and a properly configured source tree
- A blank diskette

Also, you must set up the necessary BIOS files in the appropriate directories on the source server. See "Including BIOS code and firmware updates in a deployment scenario" on page 129 for information on how to include the necessary BIOS code update files in the source

Warning: Do not remove the diskette from the diskette drive during the update.

You can create two different virtual diskette images, one for BIOS updates and another for ServeRAID BIOS and firmware updates. Customize the environment variables in usrvars.bat to create the two versions of the diskette.

Should you choose to create different virtual boot disk images for each update type, manually create a task sequence and job template. See "Creating a Microsoft ADS sequence" on page 70 and "Creating a Microsoft ADS job template" on page 70 for more information.

Tip: In some cases, updating the server BIOS code or IBM ServeRAID BIOS and firmware code requires more conventional memory (RAM) than is available after loading Microsoft ADS and the IBM ServerGuide Scripted Toolkit. If this occurs, you cannot update the server BIOS code or the IBM ServeRAID BIOS and firmware code.

See the readme.htm file that comes with the ServerGuide Scripting Toolkit for a list of verified server BIOS code and IBM ServeRAID BIOS and firmware code update scenarios.

Installing IBM Director Agent for Windows

The ServerGuide Scripting Toolkit can optionally install IBM Director Agent 5.1 or higher, automatically on the target server. To have the Scripting Toolkit automatically install Director Agent:

- Obtain the IBM Director Agent installation files.
- Set up the source tree correctly.

See "Adding application (IBM Director Agent for Windows) files" on page 12 for more information.

Microsoft ADS does not directly support application installations. The ServerGuide Scripting Toolkit can, however, copy the application installation files to the target server and modify the target server system registry so that the application installation runs the next time the server restarts (reboots).

Complete the following steps to deploy IBM Director Agent to a target server:

1. Add the IBM Director Agent installation files to the source tree.
   See "Adding application (IBM Director Agent for Windows) files" on page 12 for more information.
2. Click **Start > All Programs > Microsoft ADSSequence Editor**.
3. Type a description in the **Description** field, such as `Install IBM Director Agent`.
4. Click **Actions > Insert > Download File**.

5. In the **Controller File** field, type the full path to the first file in the \sgshare\sgdeploy\apps\dawin directory on the Microsoft ADS server.

6. Type the following line in the **Target File** field:

   ```
   \device\harddisk0\partition1\WinOS\Temp\file_name_of_first_file
   ```

   WinOS is the directory where the operating system is installed on the target server. The *file_name_of_first_file* parameter is the name of the file identified in step 5.

7. Repeat step 4 on page 68 to step 6 for all files in the \sgshare\sgdeploy\apps\dawin directory.

8. Click **Actions > Add > Configure Registry**.

9. Click the **Load hive** button.

10. Type the following line in the field just to the right of the **Load hive** button:

    ```
    \device\harddisk0\partition1\WinOS\System32\config\software
    ```

    WinOS is the directory where the operating system is installed on the target server.

11. Type

    ```
    DASetup
    ```

    in the **Name** field.

12. Type

    ```
    Microsoft\Windows\CurrentVersion\RunOnce
    ```

    in the **Registry Key** field.

13. Select **REG_SZ** from the **Type** drop-down list.

14. Type the following in the **Value** field:

    ```
    WinDRIVE:\WinOS\Temp\dir5.10_agent_windows.exe /s /a
     UNATTENDED rsp="WinDRIVE:\WinOS\Temp\diragent.rsp"
    ```

    Type the string on one line instead of the two lines illustrated in this example.

    WinOS is the directory where the operating system is installed on the target server. *WinDRIVE* is the drive letter of the disk the operating system is installed on the target server. For example,

    ```
    C:\Windows\Temp\dir5.10_agent_windows.exe /s/a
     UNATTENDED rsp="C:\Windows\Temp\diragent.rsp"
    ```

15. Click **Actions > Add→Script Command**.

16. Type

    ```
    C:\Program Files\Microsoft ADS\tools\adsdevice.wcf
    ```

    in the **Command** field.

17. Type the following lines in the **Parameters** field:

    ```
    -e
    $Device.System.Name$
    -jobtemplate
    boot-to-hd
    ```

18. Select **Controller** from the **Run at** drop-down list.

19. Click **Actions > Insert > Reboot Computer**.

20. Save this new Microsoft ADS Sequence.

21. Create a Microsoft ADS job template that calls the Microsoft ADS sequence that you just created.

    See "Creating a Microsoft ADS job template" on page 70.

22. Select a target server to run the job template against.

The IBM Director Agent installation process requires a restart and requires administrator-level access to complete successfully after the restart. Therefore, log on with an account that has administrator-level access the first time that the server restarts after performing this procedure.

Verify the installation of IBM Director Agent on the target server by opening the Services option under the Administrative Tools option in the Control Panel. Look for the following services:

- IBM Director Agent SLP Attributes (Started, Automatic)
- IBM Director Agent WMI CIM Server (Started, Automatic)
- IBM Director CIM Listener (Started, Automatic)
- IBM Director Support Program (Started, Automatic)

Creating a Microsoft ADS sequence

Complete the following steps to create a Microsoft ADS sequence:

1. Click **Start→All Programs→Microsoft ADS→Sequence Editor**.
2. Type a description for the new sequence in the **Description** field.

   If the target server is currently active and running the Microsoft ADS Administration Agent, complete step 3. Otherwise continue with step 4

3. Click **Actions→Insert→Reboot Computer**.
4. Click **Actions→Add→Boot Virtual Floppy**.
5. In the left pane, click **Boot Virtual Floppy**.
6. Type a description in the **Description** field.
7. In the **Floppy path** field, type the file name of the Microsoft ADS-formatted virtual diskette image file you created for the BIOS deployment scenario.

   See "Updating server BIOS code and IBM ServeRAID BIOS and firmware code" on page 67.

8. Select the **Reboots** check box.
9. Click **File→Save As**; then navigate to the Program Files\Microsoft ADS\Samples\Sequences directory.
10. In the **File name** field, type a descriptive name for the Microsoft ADS sequence; then, click **Save**.

Creating a Microsoft ADS job template

Complete the following steps to create a Microsoft ADS job template:

1. Click **Start→All Programs→Microsoft ADS→ADS Management**.
2. Click **Actions→New Job Template**. The New Job Template wizard starts and the Welcome to the New Job Template Wizard window opens.
3. Click **Next**. The Template Type window opens.
4. Click **An entirely new template**, then click **Next**. The Name and Description window opens.
5. In the **Name** field, type a descriptive name for the new job template.
6. In the **Description** field, type an optional description; then, click **Next**. The Command Type window opens.
7. Click **Task Sequence**, then click **Next**. The Task Sequence File and Arguments window opens.
8. To select the task sequence file, click **Browse** located to the right of the **Task sequence file** field. Locate the applicable task sequence file you created in "Creating a Microsoft ADS sequence" or "Installing IBM Director Agent for Windows" on page 68 and click **Next**. The Destination Device Selection window opens.
9. Select **None**, then click **Next**. The Completing the New Job Template Wizard window opens.
10. Click **Finish**.

You can now run the ServerGuide Scripting Toolkit disposal process or the ServerGuide Scripting Toolkit hardware configuration process by running the Microsoft ADS job template.

For information about creating a multistep job template, such as a template that would both configure a RAID controller and install a cloned image of a server, see the Microsoft ADS documentation.

Adding device drivers to the Microsoft ADS device-driver set

Microsoft Automated Deployment Services requires the latest device drivers for network interface cards (NICs) and storage controllers.

For more information, search the Microsoft ADS help for the *Overview* subtopic in the *Deployment Agent Builder service* topic.

You must have the *IBM ServerGuide Setup and Installation* CD, version 7.3 or higher, to use this scenario.

Complete the following steps to add device drivers to the Microsoft ADS device-driver set:

1. Insert the ServerGuide Setup and Installation CD into the CD drive of the ADS server.
2. Open Windows Explorer and browse to the Scripting Toolkit installation directory.
3. Navigate to the ads directory in the source tree.
4. Run the Add-Drivers-To-ADS.bat batch file located in the directory to automatically copy the drivers to the ADS server.

   See "Using helper batch files and creating diskette images" on page 17 for information about this file.
5. Press any key to exit the command window once the copy is completed.
6. Open the Windows Control panel and open the Administrative Tools group.
7. Run the Services Administrative Tool and right-click **ADS Deployment Agent Builder**.
8. Select **Stop** from the menu.
9. Right-click the **ADS Deployment Agent Builder**, then select **Start** from the menu.

   ADS then dynamically creates images that include the new device drivers.

Chapter 10. ServerGuide Scripting Toolkit utilities

This chapter contains information about the utilities that come with the ServerGuide Scripting Toolkit. For each utility there is a description of parameters, along with examples. All utilities are located within the sgdeploy\sgtk\utils directory of the source tree.

The command-line syntax examples in this documentation use the following conventions:
- Variables are shown in *italics*
- Required parameters are shown within <> brackets
- Optional parameters are shown within [] brackets
- Required or optional parameters from which you must make a unique choice are separated by a vertical bar (|) character

You must enter all parameters for a utility on a single command line, even when the information in this documentation is shown on multiple lines.

ACU.EXE

The acu.exe utility configures an IBM ServeRAID 7t controller. The utility obtains configuration information from a donor server, optionally modifies the configuration, and deploys the configuration onto one or more target servers. See the documentation that comes with the IBM ServeRAID 7t controller for more information about the utility.

The acu.exe utility that comes with the ServerGuide Scripting Toolkit has the following command-line syntax:

```
acu </Rscript_file_name|/Pscript_file_name|/DCcontainer_number|SHUT>
[/ALL|/Ccontroller_number] [/I] [/S] [/Llog_file_name]
[BUILDALL] [ALARM=Enable|Disable]
```

Parameter	Description
/Pscript_file_name	Uses playback mode to read the specified script file and create arrays.
/Rscript_file_name	Uses record mode to scan the controller configuration and save the information in the specified script file.
/DCcontainer_number	Deletes the specified container number.
SHUT	Shuts down the controller kernel.
/ALL	Runs the acu.exe utility on all of the controllers in the system.
/Ccontroller_number	Specifies the optional controller number. In systems with multiple controllers, this number identifies the controller on which to operate.
/I	Specifies not to wait for a build process or a clear process to complete while configuring a controller in playback mode.
/S	Indicates silent mode, when processing does not display status during the configuration process.
/Llog_file_name	Uses the optional log mode to log activity and errors in the specified file.
BUILDALL	Starts auto-synchronization over all Quick Init Mode Arrays.
ALARM=Enable|Disable	Enables or disables the alarm.

The following examples illustrate acu.exe utility usage.

Example	Description
`acu /Ra:\cfg1.txt`	Records the controller configuration in a file named cfg1.txt
`acu /Pa:\cfg1.txt`	Uses the cfg1.txt file to configure a controller

This program is called by PRAID to configure the IBM ServeRAID-7t controller.

The ACUSAS.EXE utility for configuring IBM ServeRAID 8i controllers was renamed from ACU.EXE to ACUSAS.EXE to avoid collisions with this ACU.EXE configuration utility for the ServeRAID-7t controller.

ACUAHCI.EXE

The acuahci.exe utility configures an IBM ServeRAID 8e SATA controller. The utility obtains configuration information from a donor server, optionally modifies the configuration, and deploys the configuration onto one or more target servers. See the documentation that comes with the IBM ServeRAID 8e SATA controller for more information about the utility.

The acuahci.exe utility that comes with the ServerGuide Scripting Toolkit has the following command-line syntax:

```
acuahci </Pscript_file_name|/Rscript_file_name|/D[ALL|array_number]>
[/Ccontroller_number] [/Llog_file_name] [LIST]
[BUILDALL] [CLEARALL]
```

Parameter	Description
/P*script_file_name*	Uses playback mode to read the specified script file and create arrays.
/R*script_file_name*	Uses record mode to scan the controller configuration and save the information in the specified script file.
/D[ALL|*array_number*]	Deletes the array, all arrays, or the specific array identified by the array number parameter.
/C*controller_number*	Specifies the optional controller number. In systems with multiple controllers, this number identifies the controller on which to operate.
/L*log_file_name*	Uses the optional log mode to log activity and errors in the specified file.
LIST	Lists all controllers.
BUILDALL	Initiates the Build task on the arrays with BUILD state.
CLEARALL	Initiates the Clear task on the arrays with CLEAR state.

The following examples illustrate acuahci.exe utility usage.

Example	Description
`acuahci /Ra:\cfg1.txt`	Records the controller configuration in a file named cfg1.txt
`acuahci /Pa:\cfg1.txt`	Uses the cfg1.txt file to configure a controller

This program is called by PRAID to configure the IBM ServeRAID 8e SATA controller.

ACUICHSV.EXE

The acuichsv.exe utility configures an IBM ServeRAID 7e (Adaptec HostRAID) SATA controller. Configuration information is obtained from a donor server and, after optional modifications, deployed onto one or more target servers. See the documentation that comes with the ServeRAID 7e SATA controller for more information about the utility.

The acuichsv.exe utility that comes with the ServerGuide Scripting Toolkit has the following command-line syntax:

```
acuichsv </Pscript_file_name|/Rscript_file_name|/D[ALL|array_number|/?]>
[/ALL|/Ccontroller_number] [/I] [/S] [/Llog_file_name][LIST]
```

Parameter	Description
/Pscript_file_name	Uses playback mode to read the specified script file and create arrays.
/Rscript_file_name	Uses record mode to scan the controller configuration and save the information in the specified script file.
/D[ALL|array_number]	Deletes the array, all arrays, or the specific array identified by the array number parameter.
/Ccontroller_number	Specifies the optional controller number. In systems with multiple controllers, this number identifies the controller on which to operate.
/S	Indicates silent mode, when processing does not display status during the configuration process.
/Llog_file_name	Uses the optional log mode to log activity and errors in the specified file.
LIST	Lists all controllers in the system.
/?	Displays the list of available options.

The following examples illustrate acuichsv.exe utility usage.

Example	Description
acuichsv /Ra:\cfg1.txt	Records the controller configuration in a file named cfg1.txt
acuichsv /Pa:\cfg1.txt	Uses the cfg1.txt file to configure the controller

This program is called by PRAID to configure the IBM ServeRAID-7e SATA controller.

ACUSAS.EXE

The acusas.exe utility configures an IBM ServeRAID 8i controller. Configuration information is obtained from a donor server and, after optional modifications, deployed onto one or more target servers. See the documentation that comes with the IBM ServeRAID 8i controller for more information about the utility.

The acusas.exe utility that comes with the ServerGuide Scripting Toolkit has the following command-line syntax:

```
acusas </Rscript_file_name|/Pscript_file_name|/DCcontainer_number|SHUT>
[/ALL|/Ccontroller_number] [/I] [/S] [/Llog_file_name]
[BUILDALL] [ALARM=Enable|Disable]
```

Parameter	Description
/Rscript_file_name	Uses record mode to scan the controller configuration and save the information in the specified script file.

Parameter	Description
/P*script_file_name*	Uses playback mode to read the specified script file and create arrays.
/DC*container_number*	Deletes the specified container number.
SHUT	Shuts down the controller kernel.
/ALL	Runs the acu.exe utility on all of the controllers in the system.
/C*controller_number*	Specifies the optional controller number. In systems with multiple controllers, this number identifies the controller on which to operate.
/I	Specifies not to wait for a build process or a clear process to complete while configuring a controller in playback mode.
/S	Indicates silent mode, when processing does not display status during the configuration process.
/L*log_file_name*	Uses the optional log mode to log activity and errors in the specified file.
BUILDALL	Starts auto-synchronization over all Quick Init Mode Arrays.
ALARM=Enable|Disable	Enables or disables the alarm.

The following examples illustrate acusas.exe utility usage.

Example	Description
acu /Ra:\cfg1.txt	Records the controller configuration in a file named cfg1.txt
acu /Pa:\cfg1.txt	Uses the cfg1.txt file to configure a controller

This program is called by PRAID to configure the IBM ServeRAID-8i controller.

The ACUSAS.EXE utility was renamed from ACU.EXE to ACUSAS.EXE to avoid collisions with the configuration utility for the ServeRAID-7t controller, which is named ACU.EXE.

ACUSAS8E.EXE

The acusas8e.exe utility configures an IBM ServeRAID 8e SAS controller. Configuration information is obtained from a donor server and, after optional modifications, deployed onto one or more target servers. See the documentation that comes with the IBM ServeRAID 8e SAS controller for more information about the utility.

The acusas8e.exe utility that comes with the ServerGuide Scripting Toolkit has the following command-line syntax:

```
acusas8e </Pscript_file_name|/Rscript_file_name|>
[/Ccontroller_number] [/Llog_file_name] [LIST]
```

Parameter	Description
/P*script_file_name*	Uses playback mode to read the specified script file and create arrays.
/R*script_file_name*	Uses record mode to scan the controller configuration and save the information in the specified script file.
/L*log_file_name*	Uses the optional log mode to log activity and errors in the specified file.
/C*controller_number*	Specifies the optional controller number. In systems with multiple controllers, this number identifies the controller on which to operate.

Parameter	Description
LIST	Lists all controllers.

The following examples illustrate acusas8e.exe utility usage.

Example	Description
acusas8e /Ra:\cfg1.txt	Records the controller configuration in a file named cfg1.txt
acusas8e /Pa:\cfg1.txt	Uses the cfg1.txt file to configure a controller

This program is called by PRAID to configure the IBM ServeRAID 8e SAS controller.

ALTBOOT.EXE

The altboot.exe utility permits the ServerGuide Scripting Toolkit to bypass startup (boot) of the diskette drive in the target server during the deployment process. This is done by modifying the boot sector on the diskette so that the diskette is ignored or the first active partition on the first hard disk is explicitly started (booted).

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

altboot </h|/r|/?> [/f:*filename*]

Parameter	Description
/?	Displays all parameters
/h	Saves a backup image of the boot sector on the diskette, and forces the target server to book from the hard disk regardless of the startup (boot) sequence configured in the target server BIOS
/r	Restores the boot sector on the diskette from the diskette image
/f:*filename*	Specifies a file name for the backup image of the boot sector, where *filename* is the name of the file. The default file name is bootsec.img. This parameter is only valid when the /b or /r parameter is used.

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

Value	Description
0	Success
1	File error (invalid boot-sector image, file access errors)
2	Diskette drive or diskette error
3	Command-line parameter error

The following examples illustrate altboot.exe utility usage.

Example	Description
altboot /f:a:\saveboot.sec	Saves an image of the boot sector as a:\saveboot.sec
altboot /r	Restores the boot sector on the diskette from the default file name a:\bootsec.img

Example	Description
`altboot /r /f:a:\test.img`	Restores the boot sector on the diskette from the a:\test.img file
`altboot /h /f:a:\floppy.img`	Explicitly starts (boots) the target server from the hard disk and saves an image of the boot sector as a:\floppy.img

CFG1030.EXE

This program is called by PRAID to configure LSI-1020 and LSI-1030 integrated RAID controllers.

The cfg1030.exe utility configures the RAID controller in IBM eServer and xSeries servers that have an integrated SCSI controller with RAID capabilities based on the LSI-1020 or LSI-1030 chip set. Configuration information is obtained from a donor server and, after optional modifications, deployed onto one or more target servers.

The cfg1030.exe utility, version 1.04.00 comes with the ServerGuide Scripting Toolkit. See the documentation that comes with the server for information about this utility. The following are examples of cfg1030.exe utility usage:

Example	Description
cfg1030 create 1 logicaldrive NEWARRAY 52071 1 0 1 0 2 qsync	Creates a new mirrored logical drive of 52 GB on channel 0 and SCSI ID 1 and on channel 0 and SCSI ID2 on controller 1, using quick synchronization
cfg1030 create 1 logicaldrive NEWARRAY MAX 1 0 1 cfg1030 setstate 1 0 1 HSP	Creates a logical drive on controller 1, using all available space on the drive, sets RAID1 for channel 0 and SCSI ID 1; then, sets the state to hot spare

CFGGEN.EXE

This program is called by PRAID to configure LSI 1064x SAS controllers.

The cfggen.exe utility configures the RAID controller in IBM eServer and xSeries servers that have an integrated SCSI controller with RAID capabilities based on the LSI 1064x chip set. Configuration information is obtained from a donor server and, after optional modifications, deployed onto one or more target servers.

The cfggen.exe utility, version 2.00.04 comes with the ServerGuide Scripting Toolkit. See the documentation that comes with the server for more information about the utility.

The cfggen.exe utility that comes with the ServerGuide Scripting Toolkit has the following command-line syntax:

`cfggen controller_number command parameters`

Parameter	Description
`controller_number`	The controller number is an integer between 0 and 255.

Parameter	Description
command	The following commands are valid: **CREATE** Creates an IR volume. **DEFAULTS** Sets controller configuration to default settings. **DISPLAY** Displays controller, volume and physical device info. **FORMAT** Performs a low-level format of a SCSI hard disk drive. **HOTSPARE** Makes the drive a hot spare. **STATUS** Displays current volume status info. **SETOFFLINE** Takes a hard disk off line. **VSIME** Updates the IMEVolumeSettings parameter in Mfg Pg 4 or in the existing IME volume.
parameters	Issue a command using the following syntax to see parameter help for the command: `cfggen controller_number command`

The following examples show cfggen.exe utility usage:

Example	Description
cfggen 1 create logicaldrive NEWARRAY 52071 1 0 1 0 2 qsync	Creates a new mirrored logical drive of 52 GB on channel 0 and SCSI ID 1 and on channel 0 and SCSI ID2 on controller 1, using quick synchronization
cfggen 1 create logicaldrive NEWARRAY MAX 1 0 1 cfggen setstate 1 0 1 HSP	Creates a logical drive on controller 1, using all available space on the drive, sets RAID1 for channel 0 and SCSI ID 1; then, sets the state to hot spare

CLINI.EXE

The Command Line INI utility can perform the following functions:

- Write information to an INI file:
  - Add new sections, items, or values
  - Remove sections, items, or values
  - Change existing sections, items, or values
- Read information from an INI file:
  - Read items and store all or part of the value as an environment variable
  - Read items and check all or part of the value for strings, substrings, or tokens

Three versions of the Command Line INI utility come with the ServerGuide Scripting Toolkit:
- A 16-bit version for DOS.

- A 32-bit version for use on Microsoft Windows 32-bit operating systems and the 32-bit version of Windows Preinstallation Environment (Windows PE) 2005. The 32-bit version was formerly named clini32.exe.
- A 64-bit version for Windows x64 operating systems and for Windows PE 2005 (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.

In addition to setting values, the clini.exe program 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.exe program checks the number of characters on the command line against the current limits of 127 characters for the DOS version and 255 characters for the Windows version. A message is displayed if the characters exceed the limit. The /O parameter overrides character-limit checking.

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

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

Parameter	Description
filename	Defines the fully qualified path to the INI file to process
/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*	Specifies the value to append to the existing item in the INI file. The /I parameter is required to use the /A:*value* parameter.
/U:*value*	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 /U:*value* parameter.
/E:*variable*	Specifies the environment variable used to store the value of the item from the INI file. The /I parameter is required to use the /E:*variable* 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 clini.exe. If the environment variable exists on the system, it is deleted when the batch file runs.
/=:*string*	Verifies that the value of the item is equal to *string*, returning a value of 0 if true and 100 if false.
/C:*string*	Verifies that value of the item has *string* as a substring, returning a value of 0 if true and 100 if false.
/CT:*string*	Verifies that the value of the item has *string* 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.
/B:*filename*	Defines the fully qualified path and file name of the batch file to create for setting the environment variable. The default is CLIniSet.bat if no file name is specified for this parameter. This parameter is only valid when the /E parameter is used.

Parameter	Description
/D:*delimiter*	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.
/T:*n*	Specifies the token in a delimited value to set as the specified environment variable, where n is greater than or equal to 1. The default delimiter is a comma unless otherwise specified with the /D parameter. This parameter is only valid with the /E parameter.
/R	Removes the specified section, item, or value from the INI file. Removing the last item in a section also removes the section.
/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.exe utility concatenates spaces around ″=″ when writing items.
/O	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 DOS version is limited to 127 characters, while the Windows version is limited to 255 characters.

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

Value	Description
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 Command Line INI utility usage.

Example	Description
`clini info.ini /S:Hardware /I:Machine Type /V:8549 /N`	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"
`clini .\info.ini /S:Hardware /I:Machine Name /V:Server1`	Adds the item Machine Name, with a value of Server1, to the existing Hardware section of the info.ini file

`clini info.ini /S:Hardware /I:Machine Type /E:MachineType`
 	
`call CLIniSet.bat`
	Reads the Machine Type value from the info.ini file, and stores it as an environment variable called MachineType

`clini info.ini /S:Hardware /I:Machine Type2 /V:%MachineType%`	Writes the value of the environment variable MachineType to the INI file named info.ini, using section Hardware and item Machine Type2

Example	Description

`clini info.ini /S:Hardware /I:Machine Type2`
`/E:MachineType2 /B:d:\EnvSet1.bat`
 	
`call d:\EnvSet1.bat`
	Reads the Machine Type value from the info.ini file and stores it as an environment variable called MachineType2 using a custom path and name for the batch file created to set the environment variable

After running the first five examples above, in sequence, the info.ini file would contain the following information:
 	
`[Hardware]`
`Machine Type = 8549`
`Machine Type2 = 8549`
`Machine Name = Server1`
 	
Also, two new environment variables would be created as indicated below:
 	
`MachineType = 8549`
`MachineType2 = 8549`	

`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:" "`
	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:
	
	`[User]`
	`Name = Toolkit User`

`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:,`
	Creates a new file named info.ini with a comma delimited list of values. The resulting info.ini file contains:
	
	`[Section]`
	`Item = Value1,Value2,Value3,Value2`

`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:,`
	Creates a new file named info.ini with a comma delimited list of unique values. The resulting info.ini file contains:
	
	`[Section]`
	`Item = Value1,Value2,Value3`

`clini info.ini /S:Section /I:Item /E:MyEVariable`
`/T:2`
 	
or
 	
`clini info.ini /S:Section /I:Item /E:MyEVariable`
`/T:2 /D:,`
	Reads information from the info.ini file created in the previous example, and sets the second value of the item to the MyEVariable environment variable. The resulting CLIniSet.bat file contains:
	
	`Set MyEVariable=Value2`

Content of doit.bat:
 	
`@Echo off`
`clini info.ini /S:Secton /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`
	This example creates a file called info.ini with the following content:
	
	`[Section]`
	`Item = Value1`
	
	Then it checks to see if the value of Item in `[Section]` is equal to `Value1` and displays a message.
	
	After running doit.bat, the follow message is displayed:
	
	`It's true`

Example	Description

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
``
	This example creates a file called info.ini with the following content:
	
	``
	[Section]
	Item = Value1
	``
	
	Then it checks to see if the value of Item in `[Section]` contains substring `alu` and displays a message.
	
	After running doit.bat, the follow message is displayed:
	
	`It's true`

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
``
	This example creates a file called info.ini with the following contents:
	
	``
	[Section]
	Item = V1,V2,V3
	``
	
	Then it checks to see if the value of Item in `[Section]` contains token `V2` in a comma delimited list and displays a message.
	
	After running doit.bat, the follow message is displayed:
	
	`It's true`

DDCOPY.EXE

The Device Driver Copy (ddcopy.exe) utility can perform the following functions:
- Copy only those drivers in the driver set that support a specific machine.
- Copy network drivers required for Microsoft ADS.

Three versions of the utility come with the ServerGuide Scripting Toolkit:
- A 16-bit version for DOS
- A 32-bit version for Windows 32-bit operating systems and for the Windows Preinstallation Environment (Windows PE) 2005 (32-bit)
- A 64-bit version for Windows x64 operating systems and for Windows PE 2005 (x64).

The following graphic shows the Microsoft required directory structure for unattended installation:

Figure 1. Microsoft directory structure for unattended installation

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:\w2k_drv\$oem$\$1\drv` directory.

When you issue a ddcopy command against a directory that contains device-driver directories (for example, c:\w2k_drv\$oem$\$1\drv), 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 32-bit version of ddcopy does not run on DOS. The 16-bit version of the ddcopy command runs on DOS as well as on Windows 32-bit operating systems, but does not run on 32-bit Windows PE.

You can use Windows long file names with the 32-bit version of ddcopy, but you must use file names that follow the DOS 8.3 format with the 16-bit version of ddcopy.

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

`ddcopy <source_path> <destination_path> [/M:machine_types] | /ADS] [/V:n] [/?]`

Parameter	Description
source_path	Defines the fully qualified path to the directory that contains the device-driver directories and the drvset.ini file.
destination_path	Specifies the fully qualified path of the target directory for copying the device drivers.
/M:machine_types	Specifies machine types to limit the number of drivers that are copied. Multiple machine types are allowed when delimited by commas.

Parameter	Description
/ADS	Copies only those drivers required for Microsoft Automated Deployment Services (ADS).
/V:*n*	Specifies the verbose level used to report status during the deployment process. Valid values for n are: • **0** - quiet mode • **3** - default • **5** - maximum information

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

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

The following examples illustrate ddcopy.exe utility usage.

Example	Description
`ddcopy d:\drivers\$oem$\$1\drv` `c:\wininst\$oem$\$1\drv`	Copies all the drivers from the d:\drivers\$oem$\$1\drv directory to the c:\wininst\$oem$\$1\drv directory
`ddcopy d:\drivers\$oem$\$1\drv` `  c:\wininst\$oem$\$1\drv /M:8832`	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 c:\wininst\$oem$\$1\drv\ drvset.ini file
`ddcopy d:\drivers\$oem$\$1\drv` `c:\wininst\$oem$\$1\drv /M:8832,8865`	Copies the drivers for machine type 8832 and machine type 8865 from the d:\drivers\$oem$\$1\drv directory to the c:\wininst\$oem$\$1\drv directory and updates the supported systems field in the c:\wininst\$oem$\$1\drv\drvset.ini file

`ddcopy d:\drivers\$oem$\$1\drv`
`"c:\program files\microsoft ads\nbs\repository\user\presystem"`
`/ADS`
	Copies all the drivers for ADS from the d:\drivers\$oem$\$1\drv directory to the c:\program files\microsoft ads\nbs\repository\user\presystem directory.

DRVINST.EXE

The drvinst.exe utility is a 32-bit Windows application that displays a message stating that it will take approximately ten minutes for the remaining executable device drivers to be installed. The ServerGuide Scripting Toolkit calls this utility during the deployment process automatically, when required. The utility stops after a 30-second countdown, or you can exit it by clicking **OK**.

DSCAN.EXE

The Driver Scan utility can perform the following functions:

- Scan 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.exe command during the installation of Windows operating systems.
- Create 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.

Three versions of the utility come with the ServerGuide Scripting Toolkit:

- A 16-bit version for DOS
- A 32-bit version for Windows 32-bit operating systems and for the Windows Preinstallation Environment (Windows PE) 2005 (32-bit)
- A 64-bit version for Windows x64 operating systems and for Windows PE 2005 (x64).

The following graphic shows the Microsoft required directory structure for unattended installation:

Figure 2. Microsoft directory structure for unattended installation

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:\w2k_drv\$oem$\$1\drv` directory.

When you issue a dscan command against a directory that contains device-driver directories (for example, c:\w2k_drv\$oem$\$1\drv), 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, that are used by the unattend utility.

The 32-bit version of dscan does not run on DOS. The 16-bit version of dscan runs on DOS as well as Windows 32-bit operating systems, but not on Windows PE 32-bit systems. You can use Windows long-file names with the 32-bit version of dscan, but you must use file names that follow the DOS 8.3 format with the 16-bit version of dscan.

The Driver Scan utility stores information in an INI file named drvinfo.ini in the device-driver directory, for use by the unattend.exe utility. If the drvinfo.ini file already exists for the device driver, it is left unchanged. See "DRVINFO.INI" on page 89 for information about the drvinfo.ini file. See "UNATTEND.EXE" on page 120 for information about the unattend.exe utility.

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 Driver Scan 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.exe utility has the following command-line syntax:

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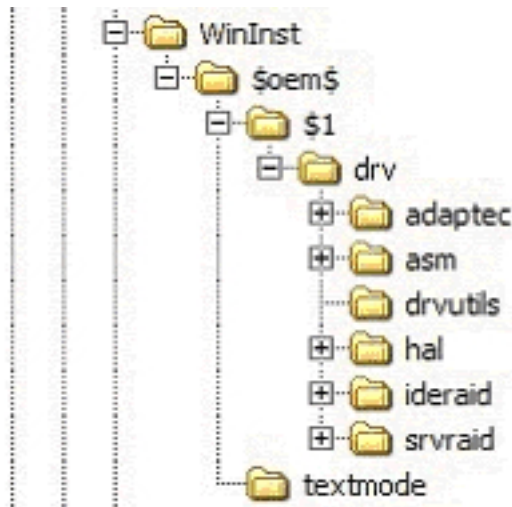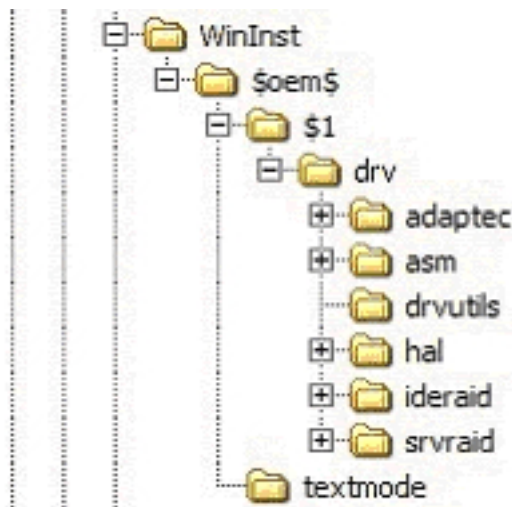dscan <driver_path> [/S|/SS|/T[:path]] [/M:machine_type]
[/OW] [/V:n] [/W:n] [/O:file_name] [/?]
```

Parameter	Description
driver_path	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 *driver_path* has oem in the path, the Driver Scan utility creates the oem\textmode 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	Specifies to scan device drivers and create drvinfo.ini files only, if necessary. Text mode device drivers are not merged when this parameter is used.
/SS	Specifies to scan a single device driver and create the drvinfo.ini file only, if necessary. Text mode device drivers are not merged when this parameter is used.
/T[:*path*]	Specifies to build the text mode device drivers only. Other device drivers are not scanned, and drvinfo.ini files are not created when this parameter is used. If *path* is specified, the text mode device drivers are merged to the specified path. Otherwise, the *driver_path* parameter must have oem in the path so that the text mode device drivers are merged into the oem\textmode directory.
/M:*machine_type*	Specifies a machine type, where *machine_type* is the machine type 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 the /M:*machine_type* parameter.

Parameter	Description
/OW	Overwrites the text-mode drivers without deleting and recreating the text-mode directory. This parameter is not valid 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 *n* are: • **0** - quiet mode • **3** - default • **5** - maximum information
/W:*n*	Specifies the version of Microsoft Windows for the device drivers: • **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:

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

The following examples illustrate Driver Scan utility usage.

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

DRVINFO.INI

The drvinfo.ini file contains information specific to each device driver. The unattend.exe utility uses this information to add device-driver information to the answer file for Windows deployment scenarios. You can 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 valid variables:

Variable name	Description
Automatically Reboots	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	Specifies the installation method for the device driver. Valid values are: Executable, Manual, PnP, or Textmode. If set to *Manual*, the unattend.exe utility does not install the device driver.
Order Before	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	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	Defines any required command-line parameters required by the executable device driver. This variable is only valid when Installation Mode is set to *Executable*.
Path	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: • **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
PCIVenDevID	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 *Executable*. Entries must be in the same format as those in the txtsetup.oem file, with multiple entries delimited by commas. For example: `PCIVenDevID = PCI\VEN_1002&DEV_5159&SUBSYS_029A1014`

Variable name	Description
Reboot Required	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 *Executable*. Valid values are True or False.
	To use this variable, the drvutils directory must contain the Holdit.exe and Reboot.exe utilities.
	This variable is only supported for Windows 2000.
Required for ADS	Specifies that this driver is required for Microsoft Automated Deployment Services (ADS). The value can be set to either True or False. The default is False.
Supported Locales	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 ProdSpec.ini file from the i386 directory of the applicable operating system installation directory.
	You cannot use both Supported Systems and Unsupported Systems in the same drvinfo.ini file.
Supported Systems	Specifies the servers supported by this device driver. This value can be All, None, or a comma-delimited list of server machine types.
	You cannot use both Supported Systems and Unsupported Systems in the same drvinfo.ini file.
Unsupported Locales	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 ProdSpec.ini file from the i386 directory of the applicable operating system installation directory.
	You cannot use both Supported Locales and Unsupported Locales in the same drvinfo.ini file.
Unsupported Systems	Specifies servers that are not supported by this device driver. This value must be a comma-delimited list of server machine types.
	You cannot use both Supported Systems and Unsupported Systems in the same drvinfo.ini file.

The following examples illustrate drvinfo.ini file contents.

Example	Description
| ```
[Driver Information]
Installation Mode=PnP
Path=
Parameters=
Automatically Reboots=
Reboot Required=
PCIVenDevID=
Order Before=
Supported Systems=All
;Unsupported Systems=
Supported Locales=
;Unsupported Systems=
``` | Supports a plug-and-play device driver with INF files in the root of the device driver directory, and supports all target servers |

Example	Description
| ```
[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=
``` | 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 |
| ```
[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=
``` | 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 |
| ```
[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=
``` | Supports an executable device-driver installation using setup.exe, with a -Q 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. |

FDISK32.EXE

The fdisk32.exe utility manages and configures partitions of hard disks in the target server. The ServerGuide Scripting Toolkit uses this utility during the deployment process to configure hard disks before installing operating-system files.

The fdisk32.exe utility has the following command-line syntax:

```
fdisk32 [/status|d</pri:n|/ext:n|/log:n|/mbr|/delete:all>]
```

Parameter	Description
/status	Displays the status of all fixed drives in the target server
d	Specifies the hard disk number to be configured, where *d* can be any positive integer.
/pri:*n*	Creates a primary DOS partition of size *n* in MB on the specified hard disk
/ext:*n*	Creates an extended DOS partition of size *n* in MB on the specified hard disk
/log:*n*	Creates a logical drive of size *n* in MB in the extended partition on the specified hard disk

Parameter	Description
`/mbr`	Writes a new master boot record to the hard disk
`/delete:all`	Deletes all partitions on the specified hard disk

The following examples illustrate fdisk32.exe utility usage.

Example	Description
`fdisk32 1 /delete:all`	Deletes all partitions on the hard disk 1
`fdisk32 1 /pri:4096`	Creates a 4096 MB (4 GB) primary DOS partition on hard disk 1
`fdisk32 1 /ext:4000`	Creates a 4000 MB extended DOS partition on hard disk 1
`fdisk32 1 /log:2000`	Creates a 2000 MB logical drive in the extended partition on hard disk 1

FIBRECFG.EXE

The fibrecfg.exe utility configures the boot settings of a Fibre Channel host bus adapter, or enables you to view current configuration and firmware level information.

The fibrecfg.exe utility supports the following fibre channel host bus adapters:
- IBM TotalStorage FC2-133 Host Bus Adapter
- IBM HS20 Fibre Channel Expansion Card

The following supplemental files must be in the current directory or the search path. These files come with the ServerGuide Scripting Toolkit and are located in the sgdeploy\sgtk\utils directory within the source tree.
- flasutil.exe
- nvram22.def
- nvram23.def
- ibmutil.exe

The fibrecfg.exe utility uses information from the above files for viewing information about and configuring Fibre Channel host bus adapters in a target server during the deployment process. See "FLASUTIL.EXE" on page 94 for information about the flasutil.exe utility.

The fibrecfg.exe utility creates several temporary files that require approximately 20 KB of available storage in the current directory. This means you cannot run the utility from a location that is read-only or has no free space available. Before exiting, the utility deletes all temporary files that it created.

If you use the fibrecfg.exe utility to modify BIOS code settings or startup (boot) settings on the Fibre Channel host bus adapter, you must restart (reboot) the target server before the changes are complete for the rest of the deployment process.

The fibrecfg.exe utility has the following command-line syntax:
`fibrecfg [/Eaddr|/Daddr] [/BEn addr wwn lun|/BD addr] [/I|/P]`

Parameter	Description
`/Eaddr`	Enables the Fibre Channel host bus adapter BIOS, where *addr* is the 4-digit I/O address of the adapter to configure. When set to 0000, the first adapter found in the target server scan order (lowest I/O address) is enabled.

Parameter	Description
/D*addr*	Disables the Fibre Channel host bus adapter BIOS, where *addr* is the 4-digit I/O address of the adapter to configure. When set to 0000, the first adapter found in the target server scan order (lowest I/O address) is enabled.
/BE*n addr wwn lun*	Enables the selectable boot option for the Fibre Channel host bus adapter, and specifies a LUN as a selectable startup (boot) device, where: *n* is the selectable boot device, with valid values from 0-9 *addr* is the 4-digit I/O address of the adapter to configure; when set to 0000, the first adapter found in the target server scan order (lowest I/O address) is enabled. *wwn* is the 16-character hexadecimal world-wide name of the remote storage device that contains the startup (boot) LUN. (You can find the world-wide node name of the storage device in the controller profile in the IBM Storage Manager software.) *lun* is the LUN number that you are setting as the startup (boot) LUN
/BD *addr*	Disables the selectable boot option for the Fibre Channel host bus adapter, where addr is the 4-digit I/O address of the adapter to configure. When set to 0000, the first adapter found in the target server scan order (lowest I/O address) is disabled.
/I	Displays information about all Fibre Channel host bus adapters in the target server in an INI-file format. You can use the > DOS output-redirect option to save the output to an output file.
/P	Displays information about all Fibre Channel host bus adapters in the target server in a variable=value format. You can use the > DOS output-redirect option to save the output to an output file.

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

Value	Description
0	Success
Non-zero	Error

The following examples illustrate fibrecfg.exe utility usage.

Example	Description
fibrecfg /E 2400	Enables the BIOS on the Fibre Channel host bus adapter at I/O address 2400 in the target server
fibrecfg /BE0 0000 210000E0B06BF9A 0	Enables the selectable-boot option on the first adapter in the target server, and sets the primary boot device to LUN 0 on the RAID device with world-wide name 210000E0B06BF9A
fibrecfg /I >myfibre.ini	Displays all information about Fibre Channel host bus adapters in the target server The >myfibre.ini parameter is a DOS output-redirect option that causes the output from the fibrecfg.exe utility to be saved in the specified file.

Example	Description
Below is an example of what the myfibre.ini file created in the above example might contain: ``` [Fibre] TWG_TOTAL_NUM_ADAPTERS=2 TWG_FIBRE_ADAPTER.1.IOAddress=2400 TWG_FIBRE_ADAPTER.1.PortWWN=210000096b360016 TWG_FIBRE_ADAPTER.1.Model=QLA2312 TWG_FIBRE_ADAPTER.1.Manufacturer=Qlogic TWG_FIBRE_ADAPTER.1.FirmwareVersion=1.29 TWG_FIBRE_ADAPTER.1.BiosStatus=D TWG_FIBRE_ADAPTER.1.BootStatus=D TWG_FIBRE_ADAPTER.1.BootWWN=200b00a0b8075501 TWG_FIBRE_ADAPTER.1.BootLUN=0 TWG_FIBRE_ADAPTER.2.IOAddress=2600 TWG_FIBRE_ADAPTER.2.PortWWN=210000096b360017 TWG_FIBRE_ADAPTER.2.Model=QLA2312 TWG_FIBRE_ADAPTER.2.Manufacturer=Qlogic TWG_FIBRE_ADAPTER.2.FirmwareVersion=1.29 TWG_FIBRE_ADAPTER.2.BiosStatus=D TWG_FIBRE_ADAPTER.2.BootStatus=D TWG_FIBRE_ADAPTER.2.BootWWN=200b00a0b8075501 TWG_FIBRE_ADAPTER.2.BootLUN=0 ``` The format returned when the /P parameter is used, has "Fibre_" preceding each of the above values, as in the following example: ``` Fibre_TWG_FIBRE_ADAPTER.1.IOAddress=2400 ```	

FINDRAM.EXE

The findram.exe utility determines the drive letter of the RAM drive that was created by IBM PC DOS 7.1 when the target server started (booted). The ServerGuide Scripting Toolkit calls this utility during the deployment process automatically, when required. The findram.exe utility returns a numerical value from 3-26, which corresponds to the letters C-Z.

FLASUTIL.EXE

The flasutil.exe utility is called by the fibrecfg.exe utility during a deployment process. See "FIBRECFG.EXE" on page 92 for information on the fibrecfg.exe utility. The ServerGuide Scripting Toolkit calls this utility during the deployment process automatically, when required.

FORMAT32.EXE

The format32.exe utility formats a FAT32 partition on the hard disk of a target server for use with IBM PC DOS 7.1. The ServerGuide Scripting Toolkit uses this utility during the deployment process to format hard disks before installing operating-system files.

The format32.exe utility has the following command-line syntax:

```
format32 <drive:> [/v:label] [/q] [/autotest]
```

Parameter	Description
drive	Specifies the drive to format on the target server
/v:label	Specifies an optional volume label for the specified drive
/q	Performs a quick-format of the specified drive
/autotest	Does not display any prompts during the formatting process

The following examples illustrate format32.exe utility usage.

Example	Description
`format32 c: /q /autotest`	Formats the c: drive on the target server using a quick-format and requiring no interaction

HWDETECT.EXE, HWDETECT

The hwdetect.exe utility performs basic hardware detection functions and more advanced PCI-device detection functions on the target server. You can store the information returned by the hwdetect.exe utility in an output file. You can also return values that set the errorlevel DOS environment variable.

Table 11 lists the operating systems supported by versions of hwdetect.exe.

Table 11. Operating systems supported by the versions of hwdetect.exe

Version	Operating system
16-bit	DOS only
32-bit	Microsoft Windows 2000 Server, Windows Server 2003, and the 32-bit version of Windows Preinstallation Environment 2005
Windows x64	64-bit version Windows Preinstallation Environment 2005
Linux	Red Hat Enterprise Linux (RHEL) AS/ES 3 U6 (32bit and x64), RHEL AS/ES 4 U2 (32bit and x64), and SLES 9.0 SP3 (32-bit and x64) operating systems

The Linux version of hwdetect requires the following packages:
- glibc-2.2.4-32.18 or higher
- libstdc++-2.96-129.7.2 or higher
- compat-2004.7.1 for SLES 9 SP3

You can only use the hwdetect.exe 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.

The hwdetect.exe utility has the following 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]
```

Parameter	Description
Basic hardware scan functions	
`/s`	Determines if the target server is an IBM eServer or IBM eServer xSeries server. The return values are: • **0** for an IBM eServer or IBM eServer xSeries server • **1** for a non-IBM eServer or IBM eServer xSeries server
`/p`	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.
`/i`	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.

Parameter	Description
/m:*type*	Compares the machine type of the target xSeries server to the specified machine type, *type*. The return values are: • **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	
/vid:*vendor_id*	Determines whether there is a PCI adapter in the target server that matches the specified vendor ID, where *vendor_id* is a hexadecimal value. The return value is the number of matching adapters in the target server, or 0 if there are no matches.
/did:*device_id*	Determines whether there is a PCI adapter in the target server that matches the specified device ID, where *device_id* is a hexadecimal value. The return value is the number of matching adapters in the target server, or 0 if there are no matches.
/svid:*sub_vendor_id*	Determines whether there is a PCI adapter in the target server that matches the specified sub-vendor ID, where *subvendor_id* is a hexadecimal value. The return value is the number of matching adapters in the target server, or 0 if there are no matches.
/sdid:*sub_device_id*	Determines whether there is a PCI adapter in the target server that matches the specified sub-device ID, where *subdevice_id* is a hexadecimal value. The return value is the number of matching adapters in the target server, or 0 if there are no matches.
/bn:*bus_number*	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 /vid, /did, /svid, or /sdid parameter is specified on the command line.
/dn:*device_number*	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 /bn parameter is specified on the command line.
/add:*num*	Adds an integer number, *num*, 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 *num*.

The following examples illustrate hwdetect.exe utility usage.

Example	Description

```hwdetect /s```
```if errorlevel 1 goto NONIBM```
```if errorlevel 0 goto IBM```
 	
```:NONIBM```
```rem Perform non-IBM equipment specific steps here```
```goto FINISH```
 	
```:IBM```
```rem Perform IBM equipment specific steps here```
 	
```:FINISH```
	Determines if the target server is an IBM server or not, and branches accordingly to perform equipment-specific steps

Example	Description
| ```
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
``` | 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 non-support for other machine types |
| ```
------------- myscript.sh -----------------
#!/bin/sh

./hwdetect /m:8676

# Check the return code of the last command,
# stored in '$?'
if [ "$?" -eq "1" ]; then
   # Perform 8676 specific action here...
   echo "Inside machine type 8676 section"
else
   ./hwdetect /m:8669

   # Check the return code of the last command,
   # stored in '$?'
   if [ "$?" -eq "1" ]; then
      # Perform 8669 specific action here...
      echo "Inside machine type 8669 section"
   else
      # Requested machine type not found.
      echo "System not supported!"
   fi
fi
``` | This is a Linux shell script that determines if the target server is either a machine type 8676 or machine type 8669 server, and branches accordingly to call a system-specific action or to display a message of non-support for other machine types. |
| ```
hwdetect /i>hwdetect.out
clini hwdetect.out /S:CI /I:Vendor_ID.0 /E:Vendor
CLIniSet.bat
``` | Creates an output file that lists the hardware configuration for the target server, so that the clini.exe utility can search for a specific PCI adapter from a vendor and set en environment variable accordingly |
| ```
hwdetect /vid:0x9005 /did:0x0250
if errorlevel 1 call 6Mstuff.bat
``` | 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 |

Example	Description

`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% IBM ServeRAID adapters in`
`this system`
	Determines the total number of IBM ServeRAID adapters in the target server, assuming there are no more than three of each type: IBM ServeRAID 4, IBM ServeRAID 5, and IBM ServeRAID 6/6i/6M

`hwdetect /i>hwdetect.out`	Displays hardware configuration information about the target server. The >hwdetect.out parameter is a DOS output-redirect option that causes the output from the hwdetect.exe utility to be saved in the specified file.

Below is an example of the hwdetect.out file that the last example might create:

```
[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=IBM
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
...
```

Notes:

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.

HYPERCFG.EXE

This program is called by PRAID to configure the LSI-IDEal RAID controller where available on BladeCenter HS20 systems.

The hypercfg.exe utility configures an LSI IDEal RAID controller in an IBM eServer BladeCenter HS20. The ServerGuide Scripting Toolkit uses this utility during the deployment process to configure the LSI IDEal RAID controller. The raidmgr.bat batch file that comes with the ServerGuide Scripting Toolkit contains examples of advanced usage of the hypercfg.exe utility. The idecfg1.bat file that comes with the ServerGuide Scripting Toolkit contains sample configuration information.

The hypercfg.exe utility that comes with the ServerGuide Scripting Toolkit has the following command-line syntax. See the *IBM BladeCenter HS20 LSI IDEal RAID User's Guide* for information about this utility.

```
hypercfg
```

Parameter	Description
/R*filename*	Scans the controller for configuration information and saves to the specified file
/P*filename*	Uses the specified file to configure the controller
/D[ALL|*n*]	Deletes configuration information, where ALL specifies to delete all arrays, or *n* is the specific array to be deleted
/ALL	Specifies that all controllers are to be configured

Parameter	Description
/C*n*	Specifies a controller number, *n*, when more than one controller is installed in the target server
/I	Specifies not to wait for a build or delete process to complete when configuring a controller
/S	Processes without displaying status during the configuration process
/L*filename*	Creates a log file, with the specified file name, and records any errors encountered during the configuration process

The following examples illustrate hypercfg.exe utility usage.

Example	Description
hypercfg /AM /C1 /D01	Configures controller number 1 in the target server for RAID1 and sets channel 0 as a secondary drive

IBMUTIL.EXE

This is a supplementary program called by FIBRECFG. See "FIBRECFG.EXE" on page 92 for details.

IPSSEND.EXE

The ipssend.exe utility is called by PRAID to configure the following IBM ServeRAID controllers:

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

The ipssend.exe utility can restore a controller to factory-default settings, or create and configure a RAID array.

The ipssend.exe utility, version 7.10.18 comes with the ServerGuide Scripting Toolkit. See the documentation that comes with the RAID controller for information about this utility. The following are examples of ipssend.exe utility usage.

The ipsraspi.sys DOS device driver must be loaded before you call the ipssend.exe utility in a ServerGuide Scripting Toolkit scenario. If you use the provided sample deployment scenarios, the loadraid.bat file automatically does this.

Example	Description
ipssend create 1 logicaldrive NEWARRAY MAX 5 noprompt	Creates a new RAID5 array on the first controller, using the maximum available space
ipssend backup 1 c:\myraid.cfg noprompt	Creates a backup of the configuration settings on the first controller to a file named myraid.cfg
ipssend restore 1 c:\myraid.cfg noprompt	Uses a file named myraid.cfg to restore configuration settings to the first controller

Example	Description
`ipssend setconfig 1 default`	Resets the first controller to factory-default settings

You must load the ipsraspi.sys DOS device driver before calling the IPSSEND.EXE utility. In the provided sample deployment scenarios, this is done by the loadraid.bat file.

IPSSENDL.EXE

IPSSENDL.EXE is a utility that implements a subset of the functions available in IPSSEND.EXE for configuring ServeRAID adapters, while using less conventional memory.

See the description of IPSSEND.EXE for more information.

LINECOMB.EXE

The linecomb.exe utility is a 16-bit DOS executable that combines the lines of a file into one single line. This is used in Linux-scripted installations to create a compatible syslinux.cfg file for the deployment scenarios.

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

`linecomb source_file output_file`

Parameter	Description
`source_file`	Specifies the name of the source file to be scanned.
`output_file`	Specifies the name of the output file.

The following example illustrates linecomb.exe utility usage.

Example	Description
`linecomb mltiline.txt sngl.txt`	This command takes each individual line in mltiline.txt and combines them into one line, each separated by spaces. The output file name is sngl.txt.

The linecomb utility issues the following return codes to indicate status:
- **1** - Source file not found
- **2** - Unable to create target file

LOADDSKF.EXE

The loaddskf.exe utility uses a diskette image file to create a DOS-startable (bootable) diskette. The file must contain a diskette image. The destination diskette is checked to make sure that it is the same format as the one from which the diskette image was originally created. The destination diskette must be blank. The ServerGuide Scripting Toolkit calls this utility during the deployment process automatically, when required.

PRAID.EXE

The ServerGuide Scripting Toolkit supports Policy-based RAID configuration and replication using the PRAID.EXE utility. Some features of PRAID.EXE include:
- A single user interface for configuring and replicating RAID controller settings.

- The ability to use the PRAID policies file to describe how your RAID controllers should be configured or replicated.
- Customizable logic to determine what configuration to use with which controllers. This logic can include the machine type of the server, the number of drives connected to the controller, and the RAID controller type.
- An AUTO mode to configure using default settings.
- The ability to configure all RAID controllers in a system with a single program call.
- Features to save useful information about each captured configuration, including machine type, date, and time of capture.
- The ability to restore all controllers to factory-default settings.

When used to configure RAID controllers, PRAID accepts a PRAID policies file as input and uses this information to create an output-script file that is capable of configuring all supported RAID controllers with the specified parameters. Optionally, if you do not wish to use a policies file, you can instruct PRAID to configure the RAID controllers using default values for arrays and logical drives.

When used to capture RAID controller settings, PRAID creates or appends to a PRAID policies file that contains the RAID controller settings which can later be used to configure RAID controllers with the same hardware configuration.

PRAID has three modes of operation:

Configure mode
for scripted configuration of RAID controllers. Run the output-script file created by PRAID to configure the RAID controllers. See "Output-script file" on page 117 for details.

Attention: When used in configure mode, PRAID restores all RAID controllers in the server to factory-default settings before configuring any of the RAID controllers. If you do not have a backup of data for all drives, it is recommended that you create backups before running PRAID on your server.

Capture mode
For replicating RAID controller settings.

Restore-defaults mode
For resetting RAID controllers to factory default settings only.

Environment requirements

PRAID works by scripting the interfaces of other RAID-configuration utilities. In order to accomplish this, several utilities must be available to PRAID when it is running:

- IPSSEND: ServeRAID-4H, 4Mx, 4Lx, 5i, 6M, 6i, 6i+, 7k
- ACU: ServeRAID-7t

  ServeRAID-7t, ServeRAID-8i, and ServeRAID-8e-SAS require a utility called ACU. However, different versions of this utility are required to support each controller. Therefore, if using a ServeRAID-8i, you must rename the utility to ACUSAS to differentiate between the three versions of ACU. If using a ServeRAID-8e-SAS, you must rename the utility to ACUSAS8E to differentiate between the three versions of ACU.

- ACUAHCI; ServeRAID-7t
- ACUSAS: ServeRAID-8i
- ACUSAS8E: ServeRAID-8e SAS
- ACUAHCI: ServeRAID-8e SATA
- ACUICHSV: ServeRAID-7e SATA
- RAIDSEL: ServeRAID-7e SCSI

- CFG1030: LSI 1020/1030 chipset
- HYPERCFG: IDEal RAID
- CFGGEN: LSI 1064x SAS

These utilities must be in the system search path or their location must be specified before invoking PRAID using the /p flag. All of these utilities are included with the ServerGuide Scripting Toolkit.

In addition, you must load the ipsraspi.sys (version 7.10.18) DOS ServeRAID driver before using PRAID if you are using a ServeRAID-4H, 4Mx, 4Lx, 5i, 6M, 6i, 6i+, or 7k controller, since IPSSEND requires this driver. This driver is included with the ServerGuide Scripting Toolkit. You can load this driver using either of the following methods:

- If you have booted into the ServerGuide Scripting Toolkit DOS environment, you can run the loadraid.bat script file located in the `sgdeploy\sgtk\examples` directory.
- Add the ipsraspi.sys file to your bootable DOS diskette, and place following line in your config.sys file: `'DEVICE=IPSRASPI.SYS /F'`

To run PRAID.EXE, you must have a small amount of space available in the working directory, or specify a directory with available space using the /t parameter.

Due to DOS limitations, PRAID leaves a single, zero-byte file named PRAIDTMP.BAT in the directory where temporary space is specified. When PRAID has executed, you can delete this file.

Usage

Each mode of PRAID operation requires a different syntax, as shown:

Configure mode
> `PRAID.EXE /f:policies /d /o:outScript /p:path /t:temp /a /e1 /e2 /e3 /s:l,n /v:n /y /b`

Capture mode
> `PRAID.EXE /c[:p] /f:policies /p:path /t:temp /e2 /e3 /v:n`

Restore defaults mode
> `PRAID /r /e2 /v:n /y`

In addition, you can supply the PRAID parameters in a parameters file, PRAID @parameters_file, instead of using the command line.

Table 12. Configure mode parameters

Parameter	Description
`/d`	Configures all controllers in the system using default settings for arrays and logical drives instead of using a policies file. The default settings used are the same as the default settings for the policies file. See Table 8 on page 31 for a list of default values for each RAID controller. You cannot use this parameter with the /f parameter.
`/o:outscript`	Names the full path and file name of the output-script file that you would like PRAID to create. This script will contain the RAID configuration commands to configure the RAID controllers as specified, and must be named with a `.bat` extension to indicate that it is a DOS batch file. If you do not supply the name of this file PRAID will use the default name praidout.bat. See "Output-script file" on page 117 for more information.
`/a`	Runs the output-script specified by the /o parameter automatically on exit. If you choose not to specify this parameter, you must manually run the output-script file to configure the RAID controllers.

Table 12. Configure mode parameters (continued)

Parameter	Description
/e1	Returns an error code of 1 if one or more controllers are not configured because there was no policy found to configure the controllers. This parameter cannot be used with the /a parameter.
/s:*l,n*	Calls the SAVESTAT program in the output-script file error checking. This call will cause the value n to be written to SAVESTAT location l anytime one of the RAID configuration utilities in the output-script file returns an error condition. To use this parameter, SAVESTAT must be located in the system search path.
/b	Automatically create commands in the output-script file that will build the arrays if it is necessary before writing to the drives. PRAID makes the building process optional because it can take up to 45 minutes. If you do not supply the /b parameter, PRAID will prompt you to see if you want the array built as part of the output-script file. If you supply the /b parameter and the arrays do not need to be built before using them, then this parameter has no effect. RAID-1 arrays created on ServeRAID-7e SCSI controllers must be built before they can be used for data.
/f:*policies*	Required for configure (unless /d is used) and capture. The path and file name of the policies file: • In configure mode, this points to the policies for PRAID to use when configuring the RAID controllers. • In capture mode, this points to the file where you would like the captured configurations to be written. If the file does not exist, PRAID will create it. If the file does exist, PRAID will append to the end of it.
/y	Do not prompt before resetting controllers to factory-default settings. PRAID always resets all controllers to factory-default settings before configuring them. If you do not supply this parameter, PRAID will pause to warn you before resetting the RAID controllers to factory-default settings.
/p:*path*	The full path to where the RAID configuration utilities are located (IPSSEND, CFG1030, RAIDSEL, ACU, ACUSAS, ACUICHSV, and HYPERCFG). You do not need to specify this parameter if all of the RAID configuration utilities are already in the system search path, which is recommended.
/t: *path*	The full or relative path where PRAID might write temporary files. PRAID requires temp space in order to run. If you do not specify this parameter, PRAID will write its temporary files to the current directory, which is recommended.
/e2	Returns an error code of 2 if there are no supported RAID controllers found in the system. By default, PRAID does not return an error if no controllers are found in the system.
/e3	Returns an error code of 3 if at least one controller is found with no drives attached. By default, PRAID does not return an error if no drives are attached to a RAID controller. This parameter cannot be used with the /a parameter.
/v:*n*	The verbosity level, where n is 0 (quiet), 3 (default), or 5 (maximum).
@*parameters_file*	The path and file name of a file containing the command line parameters. Parameters should be separated by space characters. If you use this parameter, any other parameters that you specify on the command line will override the parameters that you supply in the params file. This parameter is useful when the command-line parameters are long and can exceed the 127-character command line limit in DOS.

Table 13. Capture mode parameters

Parameter	Description
`/c:p`	Indicates capture mode. The *:p* portion is optional. If you do not include the optional portion, then *:p* will assume the default value: *t,d*. You can use :p to provide a list of parameters describing the *AppliesTo* that should be created when capturing the parameters to a policy. :p is a list containing any of the following: • m - use the machine type of the system in the AppliesTo.1 entry for the policy. • s - use the serial number of the system in the AppliesTo.1 entry for the policy • 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.
`/f:policies`	Required for configure (unless /d is used) and capture. The path and file name of the policies file: • In configure mode, this points to the policies for PRAID to use when configuring the RAID controllers. • In capture mode, this points to the file where you would like the captured configurations to be written. If the file does not exist, PRAID will create it. If the file does exist, PRAID will append to the end of it.
`/y`	Do not prompt before resetting controllers to factory-default settings. PRAID always resets all controllers to factory-default settings before configuring them. If you do not supply this parameter, PRAID will pause to warn you before resetting the RAID controllers to factory-default settings.
`/p:path`	The full path to where the RAID configuration utilities are located (IPSSEND, CFG1030, RAIDSEL, ACU, ACUSAS, ACUICHSV, and HYPERCFG). You do not need to specify this parameter if all of the RAID configuration utilities are already in the system search path, which is recommended.
`/t: path`	The full or relative path where PRAID might write temporary files. PRAID requires temp space in order to run. If you do not specify this parameter, PRAID will write its temporary files to the current directory, which is recommended.
`/e2`	Returns an error code of 2 if there are no supported RAID controllers found in the system. By default, PRAID does not return an error if no controllers are found in the system.
`/e3`	Returns an error code of 3 if at least one controller is found with no drives attached. By default, PRAID does not return an error if no drives are attached to a RAID controller. This parameter cannot be used with the /a parameter.
`/v:n`	The verbosity level, where n is 0 (quiet), 3 (default), or 5 (maximum).
`@parameters_file`	The path and file name of a file containing the command line parameters. Parameters should be separated by space characters. If you use this parameter, any other parameters that you specify on the command line will override the parameters that you supply in the params file. This parameter is useful when the command-line parameters are long and can exceed the 127-character command line limit in DOS.

Table 14. Restore defaults mode parameters

Parameter	Description
`/r`	Restores all RAID controllers to factory-default settings and then returns immediately. No RAID configuration is done if you use this parameter.
`/f:policies`	Required for configure (unless /d is used) and capture. The path and file name of the policies file: • In configure mode, this points to the policies for PRAID to use when configuring the RAID controllers. • In capture mode, this points to the file where you would like the captured configurations to be written. If the file does not exist, PRAID will create it. If the file does exist, PRAID will append to the end of it.
`/y`	Do not prompt before resetting controllers to factory-default settings. PRAID always resets all controllers to factory-default settings before configuring them. If you do not supply this parameter, PRAID will pause to warn you before resetting the RAID controllers to factory-default settings.
`/p:path`	The full path to where the RAID configuration utilities are located (IPSSEND, CFG1030, RAIDSEL, ACU, ACUSAS, ACUICHSV, and HYPERCFG). You do not need to specify this parameter if all of the RAID configuration utilities are already in the system search path, which is recommended.
`/t: path`	The full or relative path where PRAID might write temporary files. PRAID requires temp space in order to run. If you do not specify this parameter, PRAID will write its temporary files to the current directory, which is recommended.
`/e2`	Returns an error code of 2 if there are no supported RAID controllers found in the system. By default, PRAID does not return an error if no controllers are found in the system.
`/e3`	Returns an error code of 3 if at least one controller is found with no drives attached. By default, PRAID does not return an error if no drives are attached to a RAID controller. This parameter cannot be used with the /a parameter.
`/v:n`	The verbosity level, where n is 0 (quiet), 3 (default), or 5 (maximum).
`@parameters_file`	The path and file name of a file containing the command line parameters. Parameters should be separated by space characters. If you use this parameter, any other parameters that you specify on the command line will override the parameters that you supply in the params file. This parameter is useful when the command-line parameters are long and can exceed the 127-character command line limit in DOS.

Usage examples

Table 15. Configure mode examples

Example	Description
`PRAID /d /y /b /a`	• Creates a script to configure all RAID controllers in the system using default settings. • Does not prompt before setting controllers to the factory-default settings. • Performs drive synchronization, when required, without prompting. • Runs the output-script file automatically on exit. This example is especially useful for unattended scripted installs.

Table 15. Configure mode examples (continued)

Example	Description
`PRAID /f:policies.ini /v:5 /e1`	• Configures the RAID controllers in the system using the policies file `policies.ini`. • Sets the verbosity to maximum. • Returns an error code of there are no matching policies for one or more controllers. See the "Return codes" section for more information.

Table 16. Capture mode examples

Example	Description
`PRAID /c /f:c:\mydata\policies.ini`	Captures the configuration of all RAID controllers into the file `c:\mydata\policies.ini`. The `c:\mydata\policies.ini` file must exist before running this command.
`PRAID /c:m,t /f:policies.ini /t:\temp`	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. Uses the `\temp` directory for temp space. The `\temp` directory must exist before running this command.

Table 17. Restore defaults mode examples

Example	Description
`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.

Table 18. Running PRAID using a parameters file

Example	Description
`PRAID @c:\sgtk\params.txt`	Runs PRAID using the parameters located in the `c:\sgtk\params.txt` file.

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** - Syntax error on the command line.
- **5** - Syntax error in the policies file or the policy file could not be opened.
- **6** - Error creating the output script file or writing to temp space.
- **7** - Error resetting a controller to the default settings.
- **8** - Error gathering information about a controller.
- **9** - For all other errors.

In addition, if there are errors running the output-script file, PRAID will set the environment variable *PRDERR* to the return code of the failing configuration command. The output-script file will stop execution immediately if an error is encountered.

Due to DOS limitations, if you use the /a parameter to run the output-script file automatically, the PRDERR variable will not be set.

Checking the return code under MSDOS

The ServerGuide Scripting Toolkit runs under PCDOS. If you run PRAID in an MSDOS environment, PRAID will not be able to set the DOS errorlevel properly under some conditions.

When you run PRAID, it creates a small batch file called PRDERR.BAT located in the directory where temporary space is specified. Running this batch file will set the environment variable PRDERR to the return code of PRAID if the DOS errorlevel could not be set properly. Under MSDOS, you must check both the return code of PRAID and the value of PRDERR after running the PRDERR.BAT file to determine the correct return code of PRAID.

If you are running PRAID under PCDOS you do not need to run the PRDERR.BAT file, and you can delete PRDERR.BAT after PRAID has executed.

Policies file

When used in configure mode, the policies file directs how PRAID configures the RAID controllers in a system 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.

A policies file can be created using any of the following methods:
1. Run PRAID in capture mode to create a policies file from an already-configured RAID controller.
2. Use one of the example policies files provided with the ServerGuide Scripting Toolkit, and customize it to configure your RAID controllers. The example files are located in the sgdeploy\sgtk\examples\raid directory.
3. Use an ASCII text editor to create a new policies file.

The policies file is an ASCII text file that is organized in INI-file format. Each INI-file section name indicates the start of a new policy for configuring RAID controllers.

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

Some examples of legal section names are: [`Policy.1`], [`Policy.mypolicy`], and [`Policy.My-RAID5-config`]. Each section in the policies file represents a single policy for configuring RAID controllers. You can have up to 50 policies in a single policies file.

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 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 109 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, that 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 the 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. All values are case-insensitive.

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

In addition to this reference, the ServerGuide Scripting Toolkit also provides two example policies files that you can modify for your own use. These example policies files are located in the *sgdeploy\sgtk\examples\ raid* directory:

- RAID1-5.ini Creates a RAID-1 array using the first two drives, and a RAID-5 array using the remaining drives. Valid for ServeRAID-4H, 4Mx, 4Lx, 5i, 6i, 6M, 6i+, 7k, 8i
- RAID5HSP.ini Creates a single RAID-5 array with a single hot-spare drive using all available drives. Valid for ServeRAID-4H, 4Mx, 4Lx, 5i, 6i, 6i+, 6M, 7k, 7t, 8i.
- template.ini Provides a policies file template containing all parameters with details about each parameter.
- syntax.txt Provides a syntax specification for the polices file.

Table 19. Policy file parameters

Keyword	Required?	Default	Description
Policy.*name*	Yes	None	This header designates the start of a new policy. See "Policy.*name*" on page 111 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*" on page 111 for additional information.
ReadAhead	No	• ADAPTIVE (for ServeRAID 4H, 4MX, 4Lx, 5i, 6i, 6i+, 6M, and 7k) • On (for ServeRAID 7t and 8i)	Specifies the read ahead setting that should be applied to the RAID controller. See "ReadAhead" on page 112 for additional information.
RebuildRate	No	HIGH	Specifies the rebuild rate that should be applied to the RAID controller. See "RebuildRate" on page 112 for additional information.

Table 19. Policy file parameters (continued)

Keyword	Required?	Default	Description
StripeSize	No	• 8 (for ServeRAID 4H, 4Mx, 4Lx, 5i, 6i, 6i+, 6M, and 7k) • 64 (for ServeRAID 7t, 8i, 7e-SCSI, 7e-SATA, 8e-SATA, 8e-SAS and LIS-IDEal-RAID) • 32 (for ServeRAID 7e-SATA)	Specifies the stripe-unit size in KB that the controller should use for its arrays. See "StripeSize" on page 112 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" on page 112 for additional information.
Array_Defaults	No	• 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" on page 113 for additional information.
Array.*letter*	No	None	Lets you specify exactly how many arrays are created and the exact physical drives that you would like in each array. See "Array.*letter*" on page 113 for additional information.
Hotspares	No	None	Defines a list of specific physical drives to designate as hot-spare drives. See "Hotspares" on page 114 for additional information.
Logical_Mode	No	AUTO	Defines the logical-drive creation policy to use when creating logical drives. See "Logical_Mode" on page 114 for additional information.
Logical_Defaults	No	FILL:AUTO:AUTO	Defines the default logical drive settings that should be used when creating logical drives. See "Logical_Defaults" on page 115 for additional information.

Table 19. Policy file parameters (continued)

Keyword	Required?	Default	Description
Logical.*num*	No	None	Lets you specify how many logical drives are created and the specific parameters for each logical drive. See "Logical.*num*" on page 115 for additional information.

Policy.name:
Description

This header designates the start of a new policy. You can specify *name* 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.

Examples

`[Policy.RAID-5-Hotspare]`

AppliesTo.n:
Description

Use this parameter to describe when the current policy should be chosen to configure the RAID controllers. 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 containing one or more of the following parameters:

- m:*mtype*, where *mtype* is the four digit machine type of an IBM eServer or xSeries server.
- s:*serial*, where *serial* is the serial number of an IBM eServer 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 following descriptive names :
  - SERVERAID-4H
  - SERVERAID-4Mx
  - SERVERAID-4Lx
  - SERVERAID-5i
  - SERVERAID-6i (for ServeRAID-6i and 6i+)
  - SERVERAID-6M
  - SERVERAID-7t
  - SERVERAID-7e-SCSI
  - SERVERAID-7e-SATA
  - SERVERAID-7k
  - SERVERAID-8i
  - ServeRAID-8e-SATA
  - ServeRAID-8e-SAS
  - LSI-SCSI-RAID
  - LSI-IDEal-RAID

    – `LSI-SAS-RAID`

- d:*drives*, where *drives* is an integer value specifying the number of drives connected to the controller. Only drives in a **Ready** state after resetting the controller to factory-default settings are counted.
- ALL. Indicates that this policy should be used for all RAID controllers. This parameter is good to use if you declare a default policy that is not covered by any of the other policies.

Examples

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:
Description

Specifies the read ahead setting that should be applied to the RAID controller. If this parameter is not applicable for a RAID controller, then it will be ignored. See "Supported settings for RAID controllers" on page 115 for the list of ReadAhead settings supported by PRAID for each RAID controller. Possible settings are:
- `Adaptive`
- `On`
- `Off`

Examples
```
ReadAhead = On
```

RebuildRate:
Description

Specifies the rebuild rate that should be applied to the RAID controller. If this parameter is not applicable for a RAID controller, then it will be ignored. See "Supported settings for RAID controllers" on page 115 for the list of RebuildRate settings supported by PRAID for each RAID controller.
- `High`
- `Medium`
- `Low`

Examples
```
RebuildRate = High
```

StripeSize:
Description

Specifies the stripe-unit size in KB that the controller should use for its arrays. If this parameter is not applicable for a RAID controller, then it will be ignored. See "Supported settings for RAID controllers" on page 115 for the list of StripeSize settings supported by PRAID for each RAID controller. Possible values are any stripe size supported by the controller.

Examples
```
StripeSize = 32
```

Array_Mode:

Description

Defines the array-creation policy to use when selecting physical disk drives to include in an array. Possible values are:

`Auto` Creates arrays using drives that have the same size in MB. This is the default. Each set of drives with same size on 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.

Examples

```
Array_mode = CUSTOM
```

Array_Defaults:
Description

Defines the default values to use for the variance and number of hot-spare drives when `AUTO` is specified for `Array_Mode`. This 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 equal size in MB will be combined into a single array.
- `5%` - All drives within 5% size in MB will be combined into a single array.
- `10%` - All drives within 10% size in MB will be combined into a single array.
- `100%` - All drives, regardless of size in MB, will be 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 will not create any hot-spare drives.

Examples

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

Array.letter:
Description

Lets you specify exactly how many arrays are created and the exact physical drives that you would like in each array. You can specify the physical drives using any of the following methods:

- 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 *n*th 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, `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.

Examples

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:
Description

Defines a list of specific physical drives to designate as hot-spare drives. You can specify the physical drives using any one of these methods:

- 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 *n*th 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.

Examples

Example using channel number and SCSI ID:
```
Hotspares = 1:12,2:14
```

Example using integer value:
```
Hotspares = 12, 13
```

Logical_Mode:
Description

Defines the logical-drive creation policy to use when creating logical drives. Possible values 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 using the AUTO (default) scheme
- Write-cache mode is set using the default value for the controller.

        You can adjust these default values 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 using the `Logical.`*num* parameter.

Examples

```
Logical_Mode = CUSTOM
```

Logical_Defaults:
Description

Defines the default logical drive settings that should be used when creating logical drives. This parameter is only valid when `AUTO` is specified for `Logical_Mode`. Values for this parameter are expressed in the format: *size:raidlevel:writecmode*, where:

Size specifies the size of each logical drive. One logical drive will be created on each array using the given size. *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" 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 will be ignored. See "Supported settings for RAID controllers" 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.) This is the default value if writecmode is not specified.

Examples

```
Logical_Defaults = FILL:5:ON
```

Logical.*num:*
Description

Lets you specify how many logical drives are created and the specific parameters for each logical drive. 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."

Examples

```
Logical.1 = A:FILL:1:AUTO
Logical.2 = B:50%:5:AUTO
Logical.3 = B:50%:5:AUTO
```

Supported settings for RAID controllers

Table 20 on page 116 lists the supported settings for each RAID controller when using PRAID.

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

Table 20. Supported settings for each RAID controller when using PRAID

Controller	Rebuild Rate	Read Ahead	Stripe Size	RAID Levels[*]	Write-cache Mode
ServeRAID-4H	• HIGH • MEDIUM • LOW	• ADAPTIVE • ON • OFF	8,16,32,64	0,1,1E,5E AUTO (RAID-x0 not supported)	• ON • OFF • AUTO
ServeRAID-4Mx ServeRAID-4Lx	• HIGH • MEDIUM • LOW	• ADAPTIVE • ON • OFF	8,16,32,64	0,1,1E,5,5E,5EE AUTO (RAID-x0 not supported)	• ON • OFF • AUTO
ServeRAID-6i/6i+ ServeRAID-6M ServeRAID-7k	• HIGH • MEDIUM • LOW	• ADAPTIVE • ON • OFF	8,16,32,64	0,1,1E,5,5EE AUTO (RAID-x0 not supported)	• ON • OFF • AUTO
ServeRAID-5i	• HIGH • MEDIUM • LOW	• ADAPTIVE • ON • OFF	8,16,32,64	0,1,1E,5 AUTO (RAID-x0 not supported)	• ON • OFF • AUTO
ServeRAID-7t	[n/a]	• ON • OFF	16,32,64	0,1,5,10, VOLUME AUTO	• ON • OFF • AUTO
ServeRAID-8i	[n/a]	• ON • OFF	16, 32, 64, 128, 256, 512	0,1,1E,5,5EE, 50,10,6,60, VOLUME	• ON • OFF • AUTO
ServeRAID-8e SAS	[n/a]	[n/a]	16, 32, 64	0, 1, AUTO	[n/a]
ServeRAID-8e SATA	[n/a]	[n/a]	16,32, 64	0, 1, VOLUME, AUTO	[n/a]
ServeRAID-7e SATA	[n/a]	[n/a]	16,32,64	0, 1, AUTO	[n/a]
ServeRAID-7e SCSI	[n/a]	[n/a]	16,32,64	0, 1 AUTO	[n/a]
LSI-1064x SAS	[n/a]	[n/a]	[n/a]	1, AUTO	[n/a]
LSI 1020/1030 chipset	[n/a]	[n/a]	[n/a]	1, AUTO	[n/a]
LSI IDEal RAID	[n/a]	[n/a]	32,64,128,256, 512,1024,2048, 4096	0, 1, AUTO	• ON • OFF • AUTO

[*] RAID Levels 5E and 5EE only support one logical drive per array.

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. Table 21 on page 117 shows the default RAID values that PRAID will use when AUTO is specified for *raidlevel*.

Table 21. Default RAID levels

Number of drives in array	Controller types	RAID level
1	ServeRAID-4H,4Mx,4Lx,5i,6i,6M,7k,7t,8i,LSI-IDEaI	0
1	ServeRAID-7t,8e-SATA	VOLUME
2	All	1
3	ServeRAID-7t	5
3 or more	ServeRAID-7e-SATA,8e-SATA,8e-SAS	1
3 or more	ServeRAID-7e-SCSI	0
3 or more	ServeRAID-4H,4Mx,4Lx,5i,6i,6M,7k,8i	5
4	ServeRAID-7t	10

Output-script file

When used in configure mode, PRAID generates an output-script file using the file name specified by the `/o` switch, or with the default file name, `praidout.bat` if the `/o` switch is not specified. This file contains the necessary commands to configure the RAID controllers in the system based on the policies that you created in the policies file.

After PRAID completes, run the output-script file from the command line to configure the RAID controllers. Alternatively, if you specified the `/a` switch, PRAID will run the output-script file automatically upon exit. For example:

```
C:\PRAID /f:mypolicy.ini /o:myscript.bat
C:\myscript.bat
```

is equivalent to:

```
C:\PRAID /f:mypolicy.ini /a
```

The output script file is created in an organized fashion with comments describing what each script file command will accomplish. For each RAID controller that is configured, the comments in the script file will tell which policy from the policies file was used.

The logic of the output script file checks the error codes of each command to make sure an error does not occur while running the commands. If an error occurs, the script file will display an error message and set the `PRDERR` environment variable to the return code of the failing utility, unless the `/a` switch was specified. In addition, if you specified the `/s` switch, the `SAVESTAT` program will be called to set the error condition in a ServerGuide Scripting Toolkit process.

Some RAID configuration commands require a separate response file as input in order to configure the RAID. If a response file is required for a command, it is created in the same directory as the output script file. The response file is named *command.$num*. For example, the response file would be named `ACUICHSV.$1` for the first ServeRAID-7e SATA. In addition, a note will be placed in the output script file reminding the user that the additional response file is required.

RAIDSEL.EXE

This program is called by PRAID to configure IBM ServeRAID-7e SCSI controllers.

The raidsel.exe utility configures an IBM ServeRAID 7e (Adaptec HostRAID) SCSI controller.

The raidsel.exe utility comes with the ServerGuide Scripting Toolkit. See the documentation that comes with the server for information about this utility.

The following examples illustrate raidsel.exe utility usage.

Example	Description
`raidsel -L 1 -g d0b0t410 d0b0t510`	Creates a mirror of drives (RAID1) on controller 0, channel 0, using SCSI IDs 4 and 5, both on logical unit number 0
`raidsel -L 0 -g d0b0t410 d0b0t510 -z 32KB`	Creates a RAID0 array on controller 0, channel 0, using SCSI IDs 4 and 5, both on logical unit number 0 with a stripe size of 32KB
`raidsel -L all`	Lists all information for the controller

REBOOT.COM

The reboot.com utility restarts (reboots) the target server. The Scripting Toolkit uses this utility during the deployment process to restart the target server, when required. This utility does not perform a disk reset or flush hard disk cache. There are no command line parameters for the reboot.com utility.

SAVEDSKF.EXE

The savedskf.exe utility creates a diskette image file from a DOS-startable (bootable) diskette. The destination is checked to make sure that there is enough free space to save the diskette image. Sectors above the highest sector containing data, as indicated by the file-allocation table, are not written into the diskette image. The ServerGuide Scripting Toolkit calls this utility during the deployment process automatically, when required.

SAVESTAT.EXE

The savestat.exe utility enables you to store and retrieve up to five values in CMOS, using persistent-state information, on the target server. This utility is useful to pass information to the deployment process after a restart (reboot) occurs, such as where in the deployment process to continue after the restart.

The 16-bit version of savestat.exe runs only in DOS. The 32-bit version of savestat.exe is for use on Microsoft Windows 2000 Server, Windows Server 2003, and the 32-bit version of Windows Preinstallation Environment 2005. The x64 version of savestat.exe is for use on the 64-bit version of Windows Preinstallation Environment 2005.

Values are returned using the errorlevel DOS environment variable or the return code in Windows so that you can create a batch file to branch according to the value returned. The savestat.exe utility that comes with the ServerGuide Scripting Toolkit has the following command-line syntax:

`savestat </set`*n*`=`*value*`|/get`*n*`|/reset>`

Parameter	Description
`/set`*n*`=`*value*	Saves an integer value, *value*, to the *n*th location in persistent-storage memory, where *n* can be any number 1-5 and *value* can be any number 0-254. The return values are: **0** if successful **1** if not successful
`/get`*n*	Retrieves a value currently set in the *n*th location in persistent-storage memory, where *n* can be any number 1-5. The return value is the number stored, or 255 if not successful.
`/reset`	Resets all persistent-storage memory to zero values. The return values are: **0** if successful **1** if not successful

The following examples illustrate savestat.exe utility usage.

Example	Description
`savestat /set2=100`	Stores the value 100 in the second persistent-storage memory location

`savestat /get2`
 	
`if errorlevel 100 goto end`
`if errorlevel 1 goto level1`
 	
`:level1`
`call level1.bat`
 	
`:end`
	Retrieves the value of the second persistent-storage memory location and branches in the batch file according to the value returned

SCRUB3.EXE

The scrub3.exe utility performs secure data disposal on the target server hard disks. This utility erases all data on one or more hard disks.

Attention: You cannot retrieve data from a hard disk after running this utility on the target server, so make sure you do not need the data anymore before including this utility in a deployment scenario.

The scrub3.exe utility has the following command-line syntax:

`scrub3 [/?] [/Q=NO] [/S=NO] </D=<drive|ALL>> </L=level|/W=writes>`

Parameter	Description
`/?`	Displays all parameters
`/Q=NO`	Causes the utility to display a maximum number of messages during the process
`/S=NO`	Prevents the scrub signature from being written to the hard disk that is being erased
`/D=<drive|ALL>`	Specifies the *drive* on the target server to erase, or all drives when ALL is specified
`/L=level`	Defines the *level* of security to be used when cleaning the target server hard disk. Valid values are: **1** - (Limited security) The master boot record and some sectors are overwritten once. **2** - (Medium security) All sectors are overwritten once. **3** - (High security) All sectors are overwritten four times. **4** - (U.S. Department of Defense-compliant security) All sectors are overwritten seven times.
`/W=writes`	Defines the number of times each sector on the target server hard disk is overwritten. Valid values for *writes* are any positive integer.

The following examples illustrate scrub3.exe utility usage.

Example	Description
`scrub3 /L=1 /D=ALL`	Overwrites the master boot record, the first 100 sectors of each partition, and the last two sectors on every hard disk in the target server with a 0x0000 pattern

Example	Description
scrub3 /Q=NO /D=2 /L=2	Overwrites every sector on hard disk number 2 with a pattern ox 0x0000, but does not erase any other hard disk on the target server, while displaying status messages on the screen
scrub3 /D=ALL /L=3	Overwrites every byte on every sector of every hard disk in the target server four times This is a very secure method for data disposal, but will take a long time to complete.

SLEEP.EXE

The sleep.exe utility pauses the deployment process on the target server for a specified amount of time. After the time-interval elapses, the target server resumes the deployment process.

You can exit before the time-interval elapses by pressing any key on the target server.

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

sleep [n]

Parameter	Description
n	Specifies n in seconds for the target server to pause before resuming the deployment process

The following examples illustrate sleep.exe utility usage.

Example	Description
sleep 10	Pauses for 10 seconds before resuming the deployment process

TSHUDWN.EXE

The tshudwn.exe utility is a 16-bit DOS shutdown utility.

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

tshudwn

The following examples illustrate tshudwn.exe utility usage.

Example	Description
tshudwn	Shuts down the server.

UNATTEND.EXE

The unattend.exe utility adds device-driver specific information to the Microsoft Windows unattended installation answer file for a deployment scenario. The ServerGuide Scripting Toolkit uses the unattend.exe utility to dynamically add server-specific device-driver information to the answer file for an unattended installation. The device drivers on the *IBM ServerGuide Setup and Installation* CD are already configured for use with this utility.

The 16-bit version of unattend.exe runs only in DOS. The 32-bit version of unattend.exe is for use on the 32-bit version of Windows Preinstallation Environment 2005.

Three versions of the utility come with the ServerGuide Scripting Toolkit:
- A 16-bit version for DOS
- A 32-bit version for Windows 32-bit operating systems and for the Windows Preinstallation Environment (Windows PE) 2005 (32-bit)
- A 64-bit version for Windows x64 operating systems and for Windows PE 2005 (x64).

The unattend.exe utility processes three types of device drivers:
- Text mode device drivers
- Plug-and-play device drivers
- Executable device drivers

The unattend command adds the device-driver information to the answer file in one or more locations, depending on the type of device-driver:

Text mode
> Text mode device-driver information is added to the [MassStorageDevices] and [OemBootFiles] sections. Entries are not duplicated; existing entries are not changed.

Hardware abstraction layer (HAL)
> HAL device-driver information is assigned to the ComputerType keyword in the [Unattended] section. Any value previously assigned to this keyword is overwritten.

Plug and Play
> The OemPnPDriversPath 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.

Executable
> The executable device-driver information is added to the [GUIRunOnce] section. Existing entries are not changed.

Path to the oem directory
> The OemFilesPath keyword in the [Unattended] section is set to the path to the oem directory. Any value previously assigned to this keyword is overwritten.

For text mode device drivers, the unattend.exe utility uses the information in the hwdetect.ini file and the txtsetup.oem file (located in the \oem\textmode 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 txtsetup.sif file from the i386 directory of the Windows operating-system installation source files.

For plug-and-play and executable device drivers, the unattend.exe utility uses information from the hwdetect.ini file (the output of the hwdetect.exe utility) and the drvinfo.ini 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.

The unattend.exe 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]
```

Parameter	Description
/?	Displays all parameters

Parameter	Description
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 127-character limit. 1. Any settings for the /U, /D, /H, or /I parameters you place on the command line will override settings in the unattend.ini file. 2. If you do not specify a setting for *filename*, 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 *path* does not include oem in the path, 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 *n* are: **0** - quiet mode **3** - default **5** - maximum information

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

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

The following examples illustrate unattend.exe utility usage.

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 the 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 the text mode device drivers from 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 before the executable device drivers are installed, or after they are completed.

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 valid 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 or 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 or 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 or 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: **0** - quiet mode **3** - default **5** - maximum information
[GUIRunOnce] section	
*name*_Command	Specifies the *name* of the command to run
*name*_Supported_Systems	Specifies the servers on which to run the command, *name*. This value can be All, None, or a comma-delimited list of server machine types. You cannot use both *name*_Supported_Systems and *name*_Unsupported_Systems in the same unattend.ini file.
*name*_Unsupported_Systems	Specifies the servers on which not to run the command, *name*. This value must be a comma-delimited list of server machine types. You cannot use both *name*_Supported_Systems and *name*_Unsupported_Systems 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 *name*_Command variable and either a *name*_Supported_Systems or *name*_Unsupported_Systems variable defined.

Variable name	Description
Before Drivers	Specifies a comma-delimited list of commands to run before the executable device drivers are installed. Each command must have a *name*_Command variable and either a *name*_Supported_Systems or *name*_Unsupported_Systems variable defined.

The following are examples of unattend.ini file contents:

Example	Description
| ```
[Unattend]
UnattendTxt=c:\unattend.txt
Drivers Path=c:\w2\$oem$\$1\drv
HWDetectIni=c:\hwdetect.ini
I386 Path=c:\i386
System Drive=
Textmode=
PnP=
Executable=
Verbose Level=
``` | Adds plug-and-play device drivers from c:\w2\$oem$\$1\drv and text mode device drivers from c:\w2\$oem$\textmode to the answer file and deletes device drivers not specific to the target server |
| ```
[Unattend]
UnattendTxt=c:\unattend.txt
Drivers Path=c:\w2\$oem$\textmode
HWDetectIni=c:\hwdetect.ini
I386 Path=c:\i386
System Drive=
Textmode=True
PnP=
Executable=
Verbose Level=
``` | Adds only the text mode device drivers from c:\w2\$oem$\textmode directory to the answer file |
| ```
[Unattend]
UnattendTxt=c:\unattend.txt
Drivers Path=c:\w2\$oem$\$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

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
``` | 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 |

Appendix A. Working with Windows device drivers

This chapter contains information about working with the Windows device drivers in the ServerGuide Scripting Toolkit source tree that are copied from the *IBM ServerGuide Setup and Installation* CD. Using the ServerGuide Scripting Toolkit, you can add, update, or remove Windows device drivers within the source tree.

Adding or updating a device driver

Complete the following steps to add or update a Windows device driver in the source tree:

1. Obtain the new device driver from one of the following locations:
   - IBM Support Web site http://www.ibm.com/pc/support/
   - *IBM UpdateXpress* CD
   - Other equipment manufacturer distributions

2. If you are adding a new device driver, create a unique directory for the device driver in the applicable sgdeploy\drvs*OSdevdrv* directory in the source tree, where *OSdevdrv* is the operating system device-driver files directory in the source tree that corresponds to the operating system being deployed. (Can be w2k_drv or w23_drv)

   The length of the path (including directory and file names) must not exceed 127 characters and must comply with the DOS 8.3 file naming convention.

3. If you are updating a device driver, find the applicable sgdeploy*drvs\OSdevdrv* directory in the source tree that contains the current files; then, delete the files and directories. Do not delete the empty device-driver directory.

4. Copy the device-driver files to the new, or empty, directory. Make sure that you copy all files and maintain the same directory structure.

5. From a command prompt on the source system, type the following command and press enter:

   `SHARE\sgdeploy\sgtk\utils\dscan32.exe` *drivepath* `/s`

   where *SHARE* is the path to the sgdeploy directory on the source system, and *drivepath* is the fully qualified name of the directory that contains the device-driver directories. The value of *drivepath* is typically *SHARE*\sgdeploy\sgtk\drvs\w2k_drv for Windows 2000 drivers and *SHARE*\sgdeploy\sgtk\drvs\ w23_drv for Windows 2003 drivers; however, you must use the path where the device-driver set is copied on your system.

   This command scans the device-driver files, determines the installation mode of the new or updated device driver, and creates a drvinfo.ini file for the new or updated device driver in the device-driver directory.

6. Open the drvinfo.ini file in an ASCII text editor and add specific device-driver information, such as supported systems, required command-line parameters, etc. See "DRVINFO.INI" on page 89 for information about the drvinfo.ini file.

7. If the new or updated device driver is a text mode device driver, the ServerGuide Scripting Toolkit automatically merges the text mode driver at run-time on the target server during the deployment process. To make sure that no problems occur, you can merge the text mode drivers in advance by typing the following command from a command prompt and pressing enter:

   `SHARE\sgdeploy\sgtk\utils\dscan32.exe` *drivepath* `/t`

   where *SHARE* is the path to the sgdeploy directory on the source system, and *drivepath* is the fully qualified name of the directory that contains the device-driver directories.

   This command deletes and recreates the textmode directory and the txtsetup.oem file used by the ServerGuide Scripting Toolkit.

Removing a device driver

Complete the following steps to remove a Windows device driver from the source tree:

1. Find the applicable sgdeploy *drvs**OSdevdrv* directory in the source tree that contains the files; then, delete the device-driver directory and all files and directories it contains.

2. If the removed device driver is a text mode device driver, the ServerGuide Scripting Toolkit automatically removes the text mode driver at run-time on the target server during the deployment process. To make sure that no problems occur, you can remove the text mode drivers in advance by typing the following command from a command prompt and pressing enter:

   SHARE\sgdeploy\sgtk\utils\dscan32.exe *drivepath* /t

   where *SHARE* is the path to the sgdeploy directory on the source system, and *drivepath* is the fully qualified name of the directory that contains the device-driver directories.

   This command deletes the textmode directory and updates the txtsetup.oem file used by the ServerGuide Scripting Toolkit.

Updating the answer file

The ServerGuide Scripting Toolkit uses the unattend.exe utility to dynamically add server-specific device-driver information to the answer file for an unattended installation. The device drivers on the *IBM ServerGuide Setup and Installation* CD-ROM are already configured for use with this utility.

If you do not manually use the unattend.exe utility along with the configured device drivers, you must manually add the device-driver information to the answer file before performing an unattended installation of a Windows operating system.

This section contains information about the following types of device drivers:

- Text mode device drivers
- Custom hardware abstraction layer (HAL) device drivers
- Plug-and-play device drivers
- Executable device drivers

Text mode device drivers

Complete the following steps to add text mode device drivers to the answer file:

1. Open the answer file in an ASCII text editor.

2. In the [MassStorageDrivers] section, add the description of the device driver. Use Microsoft-specified nomenclature for this description as shown in the following example:

   ```
   [MassStorageDrivers]
   "LSI Logic PCI SCSI/FC MPI Driver (Server 2003 32-bit" = "OEM"
   "IDE CD-ROM (ATAPI 1.2)/PCI IDE Controller" = "RETAIL"
   ```

3. Add the device driver file names to the [OemBootFiles] section, as shown in the following examples:

   ```
   [OemBootFiles]
   symmpi.sys
   symmpi.inf
   mpi2k332.cat
   txtsetup.oem
   ```

4. Save and close the answer file.

Custom HAL device drivers

Complete the following steps to add custom HAL device drivers to the answer file:

1. Open the answer file in an ASCII text editor.

2. In the [Unattended] section, set the ComputerType variable to the name of the device driver. Use Microsoft-specified nomenclature for this name as shown in the following example:

```
[Unattended]
ComputerType = "IBM eServer xSeries 440/445 (Windows 2000 HAL)", OEM
```

3. Add the device driver file names to the [OemBootFiles] section, as shown in the following example:

```
[OemBootFiles]
halx44n.dll
halx44n.cat
halx44n.inf
txtsetup.oem
```

4. Save and close the answer file.

Plug and Play device drivers

Complete the following steps to add plug-and-play device drivers to the answer file:

1. Open the answer file in an ASCII text editor.

2. In the [Unattended] section, set the OemFilesPath variable to the fully qualified path of the oem directory in the source tree.

3. Set the OemPnPDrivesPath variable to a semi-colon delimited list of the directories that contain the INF file. See the following example:

```
[Unattended]
OemFilesPath = C:\oem_path\$oem$
OemPnPDrivesPath = drv\ACT;drv\ASF;drv\BC;drv\E1;drv\E2;drv\E3
```

4. Save and close the answer file.

Executable device drivers

Executable device drivers typically require command-line parameters to run in unattended mode. You must specify these device drivers and command-line parameters in the [GUIRunOnce] section of the answer file or in the cmdlines.txt file. See Microsoft operating system documentation for information about either of these methods.

The following examples show how to include commands in the [GUIRunOnce] section of the answer file:

```
[GUIRunOnce]
Command0 = "CMD.EXE /C RMDIR C:\WinInst /s /q"
Command1 = "CMD.EXE /C C:\drv\ASM\WIN2000\SETUP.EXE -Q"
Command2 = "CMD.EXE /C C:\drv\ATI3\SETUP.EXE -s -A /K"
```

Including `cmd.exe /c` before each command ensures that each command is completed before the next command starts. The Microsoft unattended installation process automatically copies the device-driver files from the \oem directory structure to the root of the target server.

Commands in the [GUIRunOnce] section must include paths to the location where the files are copied on the target server. For example, if the setup.exe file is in C:\wininst\$oem$\$1\drv\mydriver, you must specify the path for the command as C:\drv\mydriver\setup.exe.

Updating the answer file dynamically

You can use ServerGuide Scripting Toolkit commands to dynamically assign user information in the answer file for an unattended Windows operating system deployment scenario.

A sample batch file with the name of custunat.bat is provided in the sgshare\sgdeploy\sgtk\examples\ windows directory. The Toolkit scenarios do not call this batch file automatically; however, a commented-out call to this batch file is included in the instos.bat file.

Complete the following steps to use this feature:

1. Use an ASCII text editor to customize the custunat.bat file.

2. Use an ASCII text editor to uncomment the `call custunat.bat` line in the instos.bat file.

Appendix B. Enhancing deployment scenarios

This appendix contains information that you can use to further enhance and automate ServerGuide Scripting Toolkit deployment scenarios. It also contains information about how to use IBM Update*Xpress* to perform post-installation updates on a target server.

None of the deployment scenarios in this section are supported by the Toolkit Configuration Program.

Including BIOS code and firmware updates in a deployment scenario

The ServerGuide Scripting Toolkit sample script files are configured to automate system BIOS code and IBM ServeRAID BIOS and firmware code updates during a deployment scenario. You do not have to use IBM Update*Xpress* to perform these updates.

Notes:
1. The ServeRAID BIOS and firmware code updates are supported on the following IBM ServeRAID controllers: ServeRAID-4H, ServeRAID-4Mx, ServeRAID-4Lx, ServeRAID-5i, ServeRAID-6i, ServeRAID-6i+, ServeRAID-6M, and ServeRAID-7k.
2. You might have to update the remaining firmware code for other devices post-installation, using either the IBM UpdateXpress CD or the RemoteUX function. See "Performing Updates Post-Install Using RemoteUX for Windows" on page 132 for information about how to use the RemoteUX function.

Including BIOS code and firmware updates from the IBM Support Web site in a deployment scenario

Complete the following steps to incorporate server BIOS code and IBM ServeRAID BIOS and firmware code updates in a deployment scenario:

1. Edit the source tree directory:

```
sgshare
Network share
```

```
sgdeploy
ServerGuide Scripting Toolkit, support
files directory
```

```
updates
BIOS code and firmware
code directory (empty by
default)
```

   a. For IBM ServeRAID updates, create a srvraid directory in the sgdeploy\updates directory.

   Do not create machine-type-named directories for IBM ServeRAID updates. These update files are not machine-type specific even though you search for them using a server machine type.

   b. Complete the following steps to edit the source tree directory for server BIOS code updates:

   1) Create a directory in sgdeploy\updates for each different server.

   Use the machine type of the IBM server as the name of the directory, for example, 8671 for an xSeries 235, machine type 8671. See the readme.txt file that comes with the ServerGuide Scripting Toolkit for a list of supported servers and machine types.

   2) Create a bios directory in the machine type directory, for example, 8671\bios.

2. Obtain the latest level of server BIOS code, IBM ServeRAID BIOS code, and IBM ServeRAID firmware code from the IBM Support Web site..

3. (Optional) If the file that you downloaded from the IBM Support Web site is in compressed format, you must decompress the file before you can use it in a deployment scenario.

4. Double-click the executable file. An MSDOS window opens.

5. Press **ENTER** 14 times.

6. Press **Y**; then, press **ENTER**.

7. Insert a diskette into the diskette drive of your local system.

8. Press **ENTER**. The files are extracted to the diskette.

9. When the window shows that the extraction is complete, press **N**. The MSDOS window closes.

10. Copy the entire contents of the update package, including any readme.txt files, into the applicable directory that you created in the source tree.

11. Repeat step 2 through step 10 for each server.

12. Customize the usrvars.bat file for the deployment scenario to perform the updates for the target server, using the Customize-usrvars.bat helper batch file.

See "Modifying USRVARS.BAT" on page 20 for information about the environment variables you can configure.

Including ServeRAID BIOS code and firmware updates from an IBM ServeRAID CD in a deployment scenario

Complete the following steps to incorporate server BIOS code and IBM ServeRAID BIOS and firmware code updates in a deployment scenario:

1. Obtain the latest level of server BIOS code, IBM ServeRAID BIOS code, and IBM ServeRAID firmware code from the IBM ServeRAID CD.

2. Edit the source tree directory:

```
sgshare
Network share
```
```
sgdeploy
ServerGuide Scripting Toolkit, support
files directory
```
```
updates
BIOS code and firmware
code directory (empty by
default)
```

For IBM ServeRAID updates, create a srvraid directory in the sgdeploy\updates directory.

Do not create machine-type-named directories for IBM ServeRAID updates. These update files are not machine-type specific even though you search for them using a server machine type.

3. In a system running Windows, insert the IBM ServeRAID CD in the CD or DVD drive.

4. Navigate to the CD and open the \diskette\scsi directory of the IBM ServeRAID CD. The update files are diskette format image files and are named biosfwnumber.img, where number is an integer equal to or greater than 1.

5. Complete the following steps to extract the BIOS and firmware updates from the image files:

    a. a. Insert a diskette into the diskette drive of your local system.

b. Extract the first biosfwnumber.img file to the diskette. You can do this using loaddskf, which is shipped with the Scripting Toolkit in the sgdeploy\sgtk\utils directory or the image utility of your choice.

   c. Copy the contents of the diskette to the applicable directory in the source tree directory.

   d. Extract the next image file to the diskette; then, copy the contents of the diskette to the applicable directory in the source tree directory.

   e. Copy the entire contents of the update package, including any readme.txt files, into the applicable directory that you created in the source tree.

   f. Repeat step 5d and 5e for all biosfwnumber.img files.

6. Customize the usrvars.bat file for the deployment scenario to perform the updates for the target server, using the Customize-usrvars.bat helper batch file. See "Modifying USRVARS.BAT" on page 20 for information about the environment variables that you can configure.

Including BIOS code and firmware updates from an IBM UpdateXpress CD in a deployment scenario

The IBM Update*Xpress* CD contains two types of server BIOS code and IBM ServeRAID BIOS and firmware code updates: v DOS-based v Combined DOS and Windows

Extract the DOS portion from the combined DOS and Windows updates.

Complete the following steps to incorporate server BIOS code and IBM ServeRAID BIOS and firmware code updates in a deployment scenario:

1. Obtain the latest level of server BIOS code, IBM ServeRAID BIOS code, and IBM ServeRAID firmware code from the IBM UpdateXpress CD.

2. Edit the source tree directory:

```
sgshare
Network share
```

```
sgdeploy
ServerGuide Scripting Toolkit, support
files directory
```

```
updates
BIOS code and firmware
code directory (empty by
default)
```

   a. For IBM ServeRAID updates, create a srvraid directory in the sgdeploy\updates directory.

       Do not create machine-type-named directories for IBM ServeRAID updates. These update files are not machine-type specific even though you search for them using a server machine type.

   b. Complete the following steps to edit the source tree directory for server BIOS code updates:

       1) Create a directory in sgdeploy\updates for each different server.

           Use the machine type of the IBM server as the name of the directory, for example, 8671 for an xSeries 235, machine type 8671. See the readme.txt file that comes with the ServerGuide Scripting Toolkit for a list of supported servers and machine types.

       2) Create a bios directory in the machine type directory, for example, 8671\bios.

3. 3. In a system running Windows, insert the IBM UpdateXpress CD in the CD drive. If the IBM UpdateXpress program starts, exit the program.

4. Navigate to the CD and open the index.htm file in the root directory of the IBM Update*Xpress* CD.

5. In the left pane, click the server for which you intend to obtain BIOS code and firmware code updates. The available updates are displayed in the right pane.

6. Select the updates and click **Download Now**. The "File Download" window opens.

7. Select Run this program from the current location and click **OK**.

   Note: A "Security Warning" window might open after you click **OK**. If the "Security Warning" window opens, click **Yes**.

8. If an "IBM License Agreement" window opens, go to step 17.

9. Select Extract to floppy.

10. Click Next. The "Insert Disk" window opens.

11. Insert a diskette into the diskette drive of your local system.

12. Click **OK**. The "Extracting Image" window opens.

13. Click **OK**.

14. After the files are extracted, click **Exit**.

15. Copy the entire contents of each update package, including any readme.txt files, into the applicable directory that you created in the source tree.

16. Go to step 20.

17. Read the terms and conditions of the agreement and click **Accept** on the "IBM License Agreement" window. The "xSeries Firmware Update" window opens.

18. Select **Extract** to folder and type the fully qualified path where you want to extract the files; then, click **Next** twice.

    Note: Save the files into the applicable directory that you created in the source tree. Most files will be in a compressed or packaged-for-the-Web executable file format that you will have to extract later.

19. After the files are extracted, click **Finish**.

20. Repeat step 5 through step 19 for each different server.

21. Customize the usrvars.bat file for the deployment scenario to perform the updates for the target server, using the Customize-usrvars.bat helper batch file. See "Modifying USRVARS.BAT" on page 20 for information about the environment variables that you can configure.

Updating the answer file dynamically

You can use ServerGuide Scripting Toolkit commands to dynamically assign user information in the answer file for an unattended Windows operating system deployment scenario.

A sample batch file with the name of custunat.bat is provided in the sgshare\sgdeploy\sgtk\examples\ windows directory. The Toolkit scenarios do not call this batch file automatically; however, a commented-out call to this batch file is included in the instos.bat file.

Complete the following steps to use this feature:

1. Use an ASCII text editor to customize the custunat.bat file.

2. Use an ASCII text editor to uncomment the call custunat.bat line in the instos.bat file.

Performing Updates Post-Install Using RemoteUX for Windows

You can use the RemoteUX program to push BIOS code updates, firmware updates, and device driver updates to a remote target server that has an installed operating system. When running the RemoteUX program, the UpdateXpress CD is located in the CDROM drive of your source server, while the updates are deployed to the remote target server. The RemoteUX program is available in the root directory of the IBM UpdateXpress CD.

When running the RemoteUX program, you must insert the IBM Update*Xpress* CD in the CD drive of the source system while the updates are being deployed to the target server. The RemoteUX program is in the root directory of the CD.

To run RemoteUX (either standalone or integrated through Altiris), you must have an Administrator group user and password. You cannot leave the password blank.

RemoteUX supports the same Windows operating systems that are supported by the ServerGuide Scripting Toolkit. Both the source system and the target server must be running one of these supported Windows operating systems.

For BIOS code and firmware code updates, RemoteUX performs the following tasks:
- Restarts (reboots) the target server
- Verifies the BIOS code and firmware code levels on the target server and updates all downlevel code
- Restarts (reboots) the target server again when the updates are completed

For device-driver file updates, Remote UX updates the downlevel device drivers. You can configure RemoteUX to automatically restart the server after the updates are completed.

RemoteUX keeps log files to track information about the updates performed. These log files are saved in the %systemdrive%\uxlog\ux.log file on the target server. Remote UX provides commands to retrieve the log files and check the status of updates.

Using RemoteUX to perform post-installation updates for Windows systems

To run RemoteUX, you must have an Administrator group user and password on the target system. You cannot leave the password blank.

Complete the following steps to use RemoteUX to perform post-installation updates:
1. Insert the IBM Update*Xpress* CD into the CD-ROM drive of your source server.
2. At a command prompt, change directories to the CD-ROM drive.
3. Run the remoteux.exe program using the appropriate parameters to update the target server. For example:
   - To use RemoteUX to update the BIOS code and firmware on a remote server named ″remote″:

     ```
     remoteux.exe \\remote -user:administrator -pwd:password -firmware -a -r
     ```
   - To use RemoteUX to update the drivers on a remote server with IP address 192.168.0.150:

     ```
     remoteux.exe \\192.168.0.150 -user:fred -pwd:password -drivers
     ```
   - To use RemoteUX to update BIOS code and firmware by running RemoteUX locally on the target server:

     ```
     remoteux.exe \\127.0.0.1 -user:administrator -pwd:password -firmware -a -r
     ```
   - List valid parameters by running remoteux.exe without parameters at the command prompt.

     More help on RemoteUX exists in the index.html file located in the root directory of the UpdateXpress CD.

Using RemoteUX for Windows with Altiris Deployment Solution

To run RemoteUX, you must have an Administrator group user and password on the target system. You cannot leave the password blank.

Complete the following steps to use RemoteUX to perform your post-installation updates with the Altiris Deployment Solution:
1. Add the contents of the UpdateXpress CD to the source tree:

a. Add a ″uxcd#″ directory to the ′*Altiris_Path*\sgdeploy\sgtk\updates directory.
   Where:
   - uxcd# is created as follows:
     – For non-Blade servers create: uxcd1
     – For Blade servers create: uxcd3
   - The AltirisPath is the ″Program Files″\Altiris\eXpress\″Deployment Server″ directory by default.
b. Change the permissions on the ″uxcd#″ directory to make it a network shared folder.

2. Insert your IBM UpdateXpress CD into the CDROM drive of your source server.
   a. For non-Blade servers insert UpdateXpress CD 1
   b. For Blade servers insert UpdateXpress CD3

3. At a command prompt, xcopy the entire contents of the UpdateXpress CD, including subdirectories to the ″uxcd#″ directory under ″\Program Files″\Altiris\eXpress\Deployment Server\sgdeploy\sgtk\updates″
   a. For non-Blade servers copy the contents of UpdateXpress CD 1 to uxcd1
   b. For Blade servers copy the contents of the UpdateXpress CD 3 to uxcd3

Note: The supplied Altiris sample jobs should work without copying the entire contents of the UpdateXpress CD to the source tree, but instead running it directly off the CD. However, an UpdateXpress CD left in the CD drive of the source server has the potential to update the source server during a boot event of the source server.

Appendix C. Getting help and technical assistance

If you need help, service, or technical assistance or just want more information about IBM products, you will find a wide variety of sources available from IBM to assist you. This appendix contains information about where to go for additional information about IBM and IBM products, what to do if you experience a problem with your xSeries or IntelliStation® system, and whom to call for service, if it is necessary.

Before you call

Before you call, make sure that you have taken 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 IBM *xSeries Documentation* CD or in the IntelliStation *Hardware Maintenance Manual* at the IBM Support Web site.
- Go to the IBM Support Web site at http://www.ibm.com/pc/support/ 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 IBM 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 IBM xSeries or IntelliStation system and preinstalled software, if any, is available in the documentation that comes with your system. That documentation includes printed 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. IBM maintains pages on the World Wide Web where you can get the latest technical information and download device drivers and updates. To access these pages, go to http://www.ibm.com/pc/support/ and follow the instructions. Also, you can order publications through the IBM Publications Ordering System at http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi.

Getting help and information from the World Wide Web

On the World Wide Web, the IBM Web site has up-to-date information about IBM xSeries and IntelliStation products, services, and support. The address for IBM xSeries information is http://www.ibm.com/eserver/xseries/. The address for IBM IntelliStation information is http://www.ibm.com/pc/intellistation/.

You can find service information for your IBM products, including supported options, at http://www.ibm.com/pc/support/.

Software service and support

Through IBM Support Line, you can get telephone assistance, for a fee, with usage, configuration, and software problems with xSeries servers, IntelliStation workstations, and appliances. For information about which products are supported by Support Line in your country or region, go to http://www.ibm.com/services/sl/products/.

For more information about Support Line and other IBM services, go to http://www.ibm.com/services/, or go to http://www.ibm.com/planetwide/ for support telephone numbers. In the U.S. and Canada, call 1-800-IBM-SERV (1-800-426-7378).

Hardware service and support

You can receive hardware service through IBM Services or through your IBM reseller, if your reseller is authorized by IBM to provide warranty service. Go to http://www.ibm.com/planetwide/ for support telephone numbers, or in the U.S. and Canada, call 1-800-IBM-SERV (1-800-426-7378).

In the U.S. and Canada, hardware service and support is available 24 hours a day, 7 days a week. In the U.K., these services are available Monday through Friday, from 9 a.m. to 6 p.m.

Appendix D. Notices

This book contains the following notices designed 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 is placed just before the instruction or situation in which damage could occur.

Edition notice

Trademarks

The following terms are trademarks of International Business Machines Corporation in the United States, other countries, or both:

BladeCenter	IntelliStation
e-business logo	ServeRAID
eServer	ServerGuide
IBM	ServerProven
IBM (logo)	xSeries
TotalStorage	

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a trademark of The Open Group in the United States, other countries, or both.

Other company, product, or service names might be trademarks or service marks of others.

Important notes

When referring to processor storage, real and virtual storage, or channel volume, KB stands for approximately 1000 bytes, MB stands for approximately 1 000 000 bytes, and GB stands for approximately 1 000 000 000 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 might vary depending on operating environments.

IBM makes no representation or warranties regarding non-IBM products and services that are ServerProven, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. These products are offered and warranted solely by third parties.

IBM makes no representations or warranties with respect to non-IBM products. Support (if any) for the non-IBM products is provided by the third party, not IBM.

Some software can differ from its retail version (if available), and may not include user manuals or all program functionality.

Index

A

ACU.EXE   73
ACUAHCI.EXE   74
ACUICHSV.EXE   75
ACUSAS.EXE   75
ACUSAS8E.EXE   76
adding
   application files   12
   device driver files   13
   Director Agent files   12
   operating system installation files   14
   operating system utility files   15
   syslinux files   15
adding files to the source tree   10
ADS   9
ALTBOOT.EXE   77
Altiris
   RemoteUX for Windows   133
Altiris deployment, installation instructions   10
answer file
   updating   126
   updating dynamically   127, 132
application files
   adding   12
application files to the source tree, adding   12
Automated Deployment Services (ADS)   9
automating scenarios   26

B

batch files,helper   17
BIOS and firmware post-install updates
   using RemoteUX for Windows   132
BIOS and firmware updates   16, 129
   from IBM Support Web site   129
BIOS code and firmware updates
   from an IBM UpdateXpress CD   131
boot-media prompt, prevent display   28
boot-media prompt, prevent display with CD-ROM   28

C

CFG1030.EXE   78
CFGGEN.EXE   78
CLINI.EXE   79
configuration files
   fibre channel   32
configuration program   11
configuring
   fibre channel   29
   RAID   29
   source tree   7, 10
custom HAL, device drivers   121
customer support   2
customizing toolkit scenarios   17

D

default
   source tree directories   14
deployment scenarios
   enhancing   129
   supported   4
device drivers
   adding   125
   custom HAL   121
   executable   121, 127
   HAL, custom   126
   plug-and-play   121, 127
   removing   126
   scanning   86
   text mode   121, 126
   Windows-based operating systems   13, 125
directories, source tree
   configuring   10
DOS-based menus, preventing display of   27
DRVINFO.INI   89
DRVINST.EXE   86
drvutils directory   87
DSCAN.EXE   86
dynamically updating the answer file   127

E

eFixes   2
executable device drivers   121
executable, device drivers   127

F

FDISK32.EXE   91
features   3
fibre channel
   adapters supported   32, 92
   configuration   29, 32
FIBRECFG.EXE   92
file
   InstallShield   5
   ZIP   5
files
   installing   7

G

global environment variables   17

H

HAL
   custom   126
hardware
   supported   4
help   2

139

T

text mode, device drivers   121, 126
TKConfig.exe   11
Toolkit Configuration Utility   11
   default OS directories   14
   SLES 9 considerations   15
toolkit utilities, using   73
trademarks   137

U

uninstall an existing installation   7
updating
   answer file   126
   answer file, dynamically   127
using, toolkit utilities   73
USRVARS.BAT
   modifying   20

V

variables
   global environment   17

W

Web site
   IBM RDM resources   2
   IBM ServerGuide   2
   IBM ServerGuide Scripting Toolkit   2
   IBM ServerProven   2
   IBM Support   2
   IBM Systems Management Software   2
   IBM UpdateXpress   2
   IBM xSeries Systems Management   2
where to find the Toolkit   5
Windows-based operating systems
   adding device driver files   13
   device drivers, working with   125
   installation instructions   10
working with device drivers, Windows   125

IBM ®

Part Number: 41Y2977

Printed in USA

(1P) P/N: 41Y2977