IBM ServerGuide Scripting Toolkit

IBM

User's Reference Version 1.2

IBM ServerGuide Scripting Toolkit

User's Reference Version 1.2

IBM

> **Note**
>
> Before using this information and the product it supports, read the information in Appendix D, "Notices," on page 131.

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 1 contains a brief
overview of the ServerGuide Scripting Toolkit features and prerequisites for use.

Chapter 2, "Unpacking files and configuring the source tree," on page 5 contains
information about unpacking the ServerGuide Scripting Toolkit files and working
with the source tree.

Chapter 3, "Customizing Toolkit scenarios," on page 15 contains information about
customizing ServerGuide Scripting Toolkit scenarios.

Chapter 4, "Configuring RAID and Fibre Channel," on page 27 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 33 contains
information about deploying Windows® operating systems.

Chapter 6, "Deploying Linux operating systems," on page 39 contains information
about deploying Linux® operating systems.

Chapter 7, "Performing secure data disposal," on page 43 contains information
about how to use the ServerGuide Scripting Toolkit for secure data disposal.

Chapter 8, "Altiris Deployment Solution support," on page 45 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 59
contains information about how to integrate the ServerGuide Scripting Toolkit with
existing Microsoft® Automated Deployment Services deployment scenarios.

Chapter 10, "IBM Remote Deployment Manager support," on page 69 contains
information about compatibility between the ServerGuide Scripting Toolkit and
IBM Remote Deployment Manager.

Chapter 11, "Using ServerGuide Scripting Toolkit utilities," on page 71 contains
information about the utilities used for the supported deployment scenarios.

Appendix A, "Working with Windows device drivers," on page 117 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 121 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 129 contains information about how to obtain help and service for IBM hardware and software products.

Appendix D, "Notices," on page 131 contains important notice and trademark information.

Notices that are used in this book

This book contains the following notices designed to highlight key information:
- **Notes:** 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 RDM and other deployment and systems-management tools.

IBM Remote Deployment Manager page

http://www.ibm.com/servers/eserver/xseries/systems_management/sys_migration/rdm.h

This Web page provides an overview of IBM Remote Deployment Manager. From this Web page, you can download the latest RDM documentation and white papers. The white papers provide a troubleshooting guide and information about using RDM.

IBM ServerGuide page

http://www.ibm.com/servers/eserver/xseries/systems_management/ sys_migration/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/ sys_migration/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/pc/us/compat/index.html

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/pc/ww/eserver/xseries/systems_management/index.html

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 an overview of the IBM ServerGuide Scripting Toolkit and supported deployment scenarios, a list of prerequisites, and information about supported IBM @server™ and IBM @serverxSeries® 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 @server or IBM @server 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.

The ServerGuide Scripting Toolkit can perform the following tasks.

Note: 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*.

- 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 only)

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.

You can unpack the files to a source system running any of the following operating systems.

Note: 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*.

- Microsoft Windows 2000 (Server and Advanced Server)
- Microsoft Windows Server 2003 (Standard and Enterprise Editions)
- Red Hat®Enterprise Linux AS, Version 2.1
- Red Hat Enterprise Linux AS, Version 3

The ServerGuide Scripting Toolkit requires the following items that are not part of the web-downloadable package:

- Licensed copies of all operating systems to be deployed
- The latest version of the *IBM ServerGuide Setup and Installation* CD that supports the target server
- A source system with a diskette drive
- Blank diskettes (for creating DOS-startable diskettes for deployment scenarios)

The following might also be required for some deployment scenarios:

- CD-R or CD-RW drive (for creating CDs for CD-based installs)
- Blank CDs (for creating DOS-startable or data CDs for CD-based deployment scenarios)
- CD-creation software (for CD-based installs)
- The latest version of the *IBM Director Setup and Installation* CD
- The latest version of the *IBM ServeRAID* CD
- The latest version of the *IBM UpdateXpress* CD
- A source system with a Web browser installed (for Remote Supervisor Adapter II and BladeCenter™ Management Module deployment scenarios)

Supported deployment scenarios

The ServerGuide Scripting Toolkit supports the following deployment scenarios:

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

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

Supported target server hardware and software

The ServerGuide Scripting Toolkit supports deployment of Windows operating systems and Linux operating systems on IBM @server and IBM @server xSeries servers. In general, the ServerGuide Scripting Toolkit provides support for ServerProven® IBM or third-party adapters in the following categories:

- Fast Ethernet
- Fibre Channel
- Gigabit Ethernet
- High-speed Token ring
- IDE and IDE RAID
- SAS and SAS RAID
- SATA and SATA RAID
- SCSI and SCSI RAID (includes Ultra-SCSI)
- Token ring

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

- A list of supported IBM @server and IBM @server 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 @server and IBM @server xSeries servers

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

Chapter 2. Unpacking files and configuring the source tree

This chapter contains information about unpacking the ServerGuide Scripting Toolkit files and configuring the source tree. Before you can use the ServerGuide Scripting Toolkit, you must do the following:

- Unpack the ServerGuide Scripting Toolkit files
- Become familiar with the source tree
- Create additional directories, as required, within the source tree to contain operating-system installation files

Unpacking the files

The ServerGuide Scripting Toolkit is available for download from the IBM Web site as a packaged-for-Web executable file and a compressed ZIP file. The ZIP file only supports scenarios for the Linux version of the Altiris Deployment Solution.

Before you unpack the ServerGuide Scripting Toolkit files to the source system, make sure that you have the following:

- Source system with a supported Windows or Linux operating system installed
- IP address of the source system (for network-based scenarios)
- User account and password created on the source system (for network-based scenarios)

There are different requirements for unpacking the files, depending on how you will use the ServerGuide Scripting Toolkit:

- Most scenarios
- Altiris Deployment Solution for Windows scenarios
- Altiris Deployment Solution for Linux scenarios
- Microsoft Automated Deployment Services scenarios

Default unpacking instructions for most scenarios

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

Important: You must unpack the files from a computer that is running a Windows operating system, even if you are unpacking the files to a computer running another operating system.

1. Download the latest version of the ServerGuide Scripting Toolkit packaged-for-Web executable file from the IBM Systems Management Software: Download/Electronic Support Web page. See "IBM deployment resources on the World Wide Web" on page vi for information.
2. From a computer running a Windows operating system, run the packaged-for-Web executable file that you downloaded in step 1.
3. When the "License Agreement" window opens, read the information; then, select **I accept the terms in the license agreement** and click **Next**.

4. When the "Location to Save Files" window opens, type the fully qualified directory name where you want to unpack the files in the **Save Files in folder** field; then, click **Next**. The source tree (see "Configuring the source tree" on page 8) is created, and the files are unpacked to the location that you specified. The default location is c:\sgshare.

   Notes:

   a. If you are performing a network-based installation, the location that you specify must be the network shared directory for ServerGuide Scripting Toolkit files, operating-system images, and device drivers.

   b. If you are performing a network-based installation, make sure that the directory to which you unpacked the files is a shared network directory.

   c. The name of the share must comply with the DOS 8.3 file naming convention.

   d. When you are unpacking the files to a computer running Linux, the network share must be accessible from the computer you are using to unpack the files. See your Linux operating-system documentation or help files for information about installing and configuring Samba.

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

Altiris Deployment Solution for Windows unpacking instructions

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

Note: The Altiris Deployment Solution for Windows must already be installed on the source system before you unpack the ServerGuide Scripting Toolkit files.

1. Download the latest version of the ServerGuide Scripting Toolkit packaged-for-Web executable file from the IBM Systems Management Software: Download/Electronic Support Web page. See "IBM deployment resources on the World Wide Web" on page vi for information.

2. From a computer running a Windows operating system, run the packaged-for-Web executable file that you downloaded in step 1.

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

4. When the "Location to Save Files" window opens, type

   `c:\`*AltirisPath*`\Altiris\eXpress\Deployment Server`

   where *AltirisPath* is the path where the Altiris Deployment Solution is installed, in the **Save Files in folder** field; then, click **Next**. The source tree (see "Configuring the source tree" on page 8) is created, and the files are unpacked to the location you specified.

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

Altiris Deployment Solution for Linux unpacking instructions

Complete the following steps to download and unpack the latest version of the ServerGuide Scripting Toolkit files for use with the Altiris Deployment Solution for Linux.

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

1. Download the latest version of the ServerGuide Scripting Toolkit ZIP file from the IBM Systems Management Software: Download/Electronic Support Web page. See "IBM deployment resources on the World Wide Web" on page vi for information.
2. Copy the ZIP file you just downloaded to the server where the Altiris Deployment Solution is installed.
3. Use an unzip program to extract the files to the opt/altiris/deployment/adlserver directory. This is the default shared directory within the Altiris Deployment Solution for Linux directory structure.

Microsoft Automated Deployment Services unpacking instructions

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

Note: Microsoft Automated Deployment Services will only install on Windows Server 2003. You can unpack the ServerGuide Scripting Toolkit files to the source system before or after installing Microsoft Automated Deployment Services.

1. Download the latest version of the ServerGuide Scripting Toolkit packaged-for-Web executable file from the IBM Systems Management Software: Download/Electronic Support Web page. See "IBM deployment resources on the World Wide Web" on page vi for information.
2. From the source system, run the packaged-for-Web executable file that you downloaded in step 1.
3. When the "License Agreement" window opens, read the information; then, select **I accept the terms in the license agreement** and click **Next**.
4. When the "Location to Save Files" window opens, type the fully qualified directory name where you want to unpack the files in the **Save Files in folder** field; then, click **Next**. The source tree (see "Configuring the source tree" on page 8) is created, and the files are unpacked to the location you specified.

   Notes:
   a. The location specified here must be the network shared directory for ServerGuide Scripting Toolkit files, operating system images, and device drivers to reside. The default location is c:\sgshare.
   b. The name of the share must comply with the DOS 8.3 file naming convention.
5. When the "InstallShield Wizard Complete" window opens, click **Finish**.

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. See "Source tree directory structure" for information about the default directory structure created for the source tree when you unpack the ServerGuide Scripting Toolkit files.

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

Before using the ServerGuide Scripting Toolkit, you must copy operating-system files, device-driver files, and application files to the source tree. Use the latest version of IBM ServerGuide that supports your server for Windows operating system device drivers.

See "Adding operating-system installation files" on page 10 for information about adding operating-system files to the source tree. See "Adding device-driver files for Windows operating systems" on page 11 for information about adding device-driver files to the source tree. See "Adding IBM Director Agent installation files to the source tree" on page 13 for information about adding application files to the source tree.

Source tree directory structure

When you run the packaged-for-Web executable file and unpack the ServerGuide Scripting Toolkit files to the default location (c:\sgshare), the following directory structure is created.

Notes:

1. When using the Altiris Deployment Solution for Windows, the default network share is c:\Program Files\Altiris\eXpress\Deployment Server.
2. When using the Altiris Deployment Solution for Linux, the default network share is opt/altiris/deployment/adlserver.
3. When using Microsoft ADS, the default network share is c:\sgshare.

The following list shows the structure of the sgshare directory:

- **sgshare**
  Network share
  - **sgdeploy**
    ServerGuide Scripting Toolkit, support files directory
    - **sgtk**
      ServerGuide Scripting Toolkit directory
    - **os**
      Operating-systems directory (empty by default)
    - **drvs**
      ServerGuide device-driver directory (empty by default)
    - **updates**
      BIOS code and firmware code directory (empty by default)
    - **apps**
      Applications directory
      - **diragent**
        IBM Director Agent directory (empty by default)

The following list shows the structure of the sgtk directory:

- **sgtk**
  ServerGuide Scripting Toolkit directory
  - **altiris**
    Altiris Deployment Solution files and sample scripts
    - **linux**
      Linux binary files
      - **redhat**
        Red Hat Linux sample answer files
    - **temp**
      Temporary space for Altiris jobs.
    - **windows**
      Windows binary files
      - **ansfile**
        Windows sample answer files
  - **boot**
    Startup helper files and boot image directory
    - **images**
      Boot images
  - **docs**
    Documentation directory
  - **DOS**
    IBM PC DOS 7.1 directory
    - **cfgfiles**
      DOS configuration files (autoexec.bat and config.sys)
  - **examples**
    Scenario files and sample answer files
    - **linux**
      Linux deployment directory
      - **redhat**
        Red Hat Linux sample answer files
    - **raid**
      RAID sample configuration files
    - **windows**
      Windows deployment directory
      - **ansfile**
        Windows sample answer files
  - **hotfixes**
    Operating-system updates
    - **win2000**
      Windows 2000 directory
      - **KB822624**
        Windows 2000 update files
  - **utils**
    Toolkit utilities

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:

- Windows operating system installation files
- Linux operating system installation files
- Windows device-driver files (for Windows 2000 or Windows Server 2003)
- BIOS code and firmware update files
- IBM Director Agent installation files

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 an appropriate directory within the sgdeploy\os directory of the source tree.

```
sgshare
Network share
```

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

```
os
Operating-systems directory
(empty by default)
```
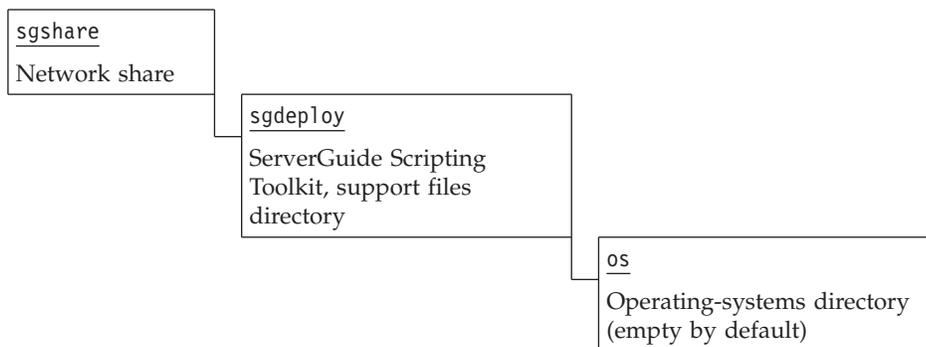
Complete the following steps to add operating-system files to the source tree.

Note: The sgdeploy directory is located in the network shared directory where you unpacked the ServerGuide Scripting Toolkit files.

1. Create directories under the sgdeploy\os directory, as required for the operating systems you will deploy, as shown below.

   Note: If you do not use the directory names shown below, you will have to modify environment variables in the usrvars.bat file. See Chapter 3, "Customizing Toolkit scenarios," on page 15 for information about customizing variables.

Operating system	Directory
Microsoft Windows 2000 Server	w2k_srv
Microsoft Windows 2000 Advanced Server	w2k_adv
Microsoft Windows Server 2003, Standard Edition	w23_std
Microsoft Windows Server 2003, Enterprise Edition	w23_ent
Microsoft Windows Server 2003, Web Edition	w23_web
Red Hat Enterprise Linux AS 2.1	rhas21
Red Hat Enterprise Linux AS 3	rhas3

2. For each operating system, copy the operating-system files from the source media to the applicable directory you just created.

   Notes:

   a. You must maintain the operating-system directory structure when you copy files.

   b. For Windows operating systems, copy the i386 directory and all files it contains. For example, the files for Windows 2000 Server must be in the sgdeploy\os\w2k_srv\i386 directory within the source tree.

   c. For Red Hat Linux, you can copy the contents of the first three installation CDs to the source tree. For Red Hat Enterprise Linux AS 3, there are four installation CDs. See the documentation that comes with your Linux distribution to ensure that you copy all required files.

Adding device-driver files for Windows operating systems

In order to update Windows device drivers using the ServerGuide Scripting Toolkit, you must copy device-driver files into one of the following directories within the source tree:

- sgdeploy\drvs\w2k_drv (for Windows 2000 device drivers)
- sgdeploy\drvs\w23_drv (for Windows Server 2003 device drivers)

Note: This function is supported only for ServerGuide 7.2 or higher.

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

```
drvs
ServerGuide device-driver
directory (empty by default)
```
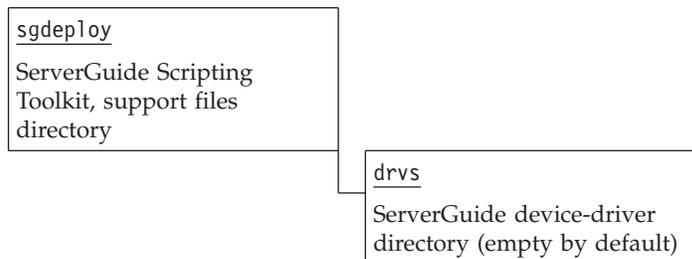
Complete the following steps to add Windows device-driver files to the source tree.

Note: The sgdeploy directory is located in the network shared directory where you unpacked the ServerGuide Scripting Toolkit files.

1. Create directories under the sgdeploy\drvs directory, as required for the operating systems you will deploy, as shown below.

   Note: If you do not use the directory names shown below, you will have to modify environment variables in the usrvars.bat file. See Chapter 3, "Customizing Toolkit scenarios," on page 15 for information about customizing variables.

Operating system	Directory
Microsoft Windows 2000 family	w2k_drv
Microsoft Windows 2003 family	w23_drv

2. Use the following table to determine the location of device drivers on the *IBM ServerGuide Setup and Installation* CD; then, copy the device driver files to the source tree.

Operating system	Copy from: *ServerGuide* **CD directory**	Copy to: **Source tree directory**
Windows 2000	sguide\w2000drv\$oem$\$1\drv*	sgdeploy\drvs\w2k_drv
Windows Server 2003	sguide\w2003drv\$oem$\$1\drv*	sgdeploy\drvs\w23_drv

3. Insert the *IBM ServerGuide Setup and Installation* CD into the CD or DVD drive of a system running a Windows operating system.
4. Wait for the IBM ServerGuide Driver Repository program to start; then, exit the program.
5. Copy the contents of the drv directory (all files and sub-directories it contains, but not the drv directory itself) from the CD to the applicable directory of the source tree (either sgdeploy\drvs\w2k_drv or sgdeploy\drvs\w23_drv, depending on the operating system).

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.
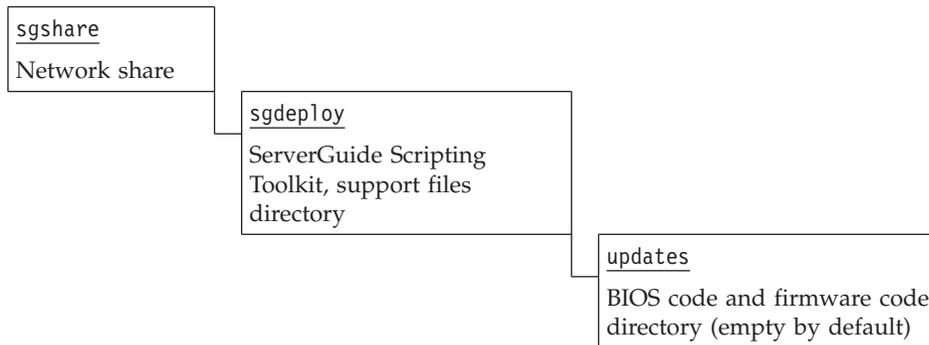
```
sgshare
Network share
```
```
    sgdeploy
    ServerGuide Scripting
    Toolkit, support files
    directory
```
```
        updates
        BIOS code and firmware code
        directory (empty by default)
```

See "Including BIOS code and firmware updates in a deployment scenario" on page 121 for information on how to add server BIOS code and IBM ServeRAID BIOS and firmware code updates to the source tree.

Adding IBM Director Agent installation files to the source tree

In order to install IBM Director Agent using the ServerGuide Scripting Toolkit, you must copy IBM Director Agent installation files into the sgdeploy\apps\diragent directory of the source tree.

```
sgshare
Network share
```

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

```
apps
Applications directory
```

```
diragent
IBM Director Agent
directory (empty by
default)
```
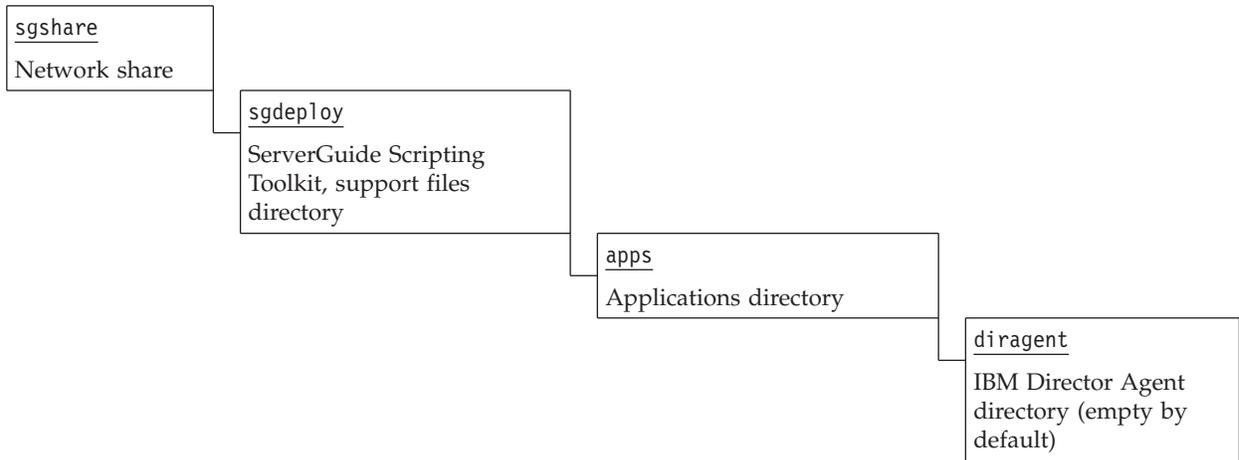
Note: Adding IBM Director Agent installation files to the source tree is only supported for Director Agent 4.20 or higher.

Complete one of the following procedures to add the IBM Director Agent installation files to the source tree:

- If you have a copy of the *IBM Director Setup and Installation* CD, copy the contents of the \director\agent\windows\i386 folder on the CD into the sgdeploy\apps\diragent folder in the source tree:

- If you are getting the Director Agent installation files from the IBM web site, complete the following steps:

  1. Locate and download the IBM Director Agent installation ZIP file (dir4.20_agent_windows.zip for IBM Director Agent 4.20) from the IBM Web site.

  2. Extract the IBM Director Agent installation files from the ZIP file into the sgdeploy\apps\diragent directory in the source tree.

  3. In the sgdeploy\apps\diragent folder of the source tree, rename the installation executable file (for example, IBMDirectorAgent4.20.exe) to IBMDir~1.exe.

  4. In the sgdeploy\apps\diragent folder of the source tree, rename the diragent_windows.xml file to dirage~1.

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 scenarios 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 1. 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:

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.

Note: If you are using the ServerGuide Scripting Toolkit with the Altiris Deployment Solution, you will not use any of the helper batch files.

The helper batch files are located in the sgdeploy\sgtk\boot directory. To use a helper batch file, you must insert a diskette into the diskette drive (a:) of the source system. Table 2 contains descriptions of the included helper batch files:

Table 2. Descriptions of helper batch files

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.
Customize-usrvars.bat	*Customizes user environment variables* This file uses notepad.exe to open the a:\usrvars.bat file for editing.
MakeDisk-ADS-Disposal.bat	*Creates a Microsoft Automated Deployment Services secure data disposal diskette* This file uses loaddskf.exe to convert the adsdisp.img diskette image to a DOS-startable (bootable) diskette in the diskette drive (a:) of the source system.
MakeDisk-ADS-Raid-Config.bat	*Creates a Microsoft Automated Deployment Services RAID configuration diskette* This file uses loaddskf.exe to convert the adsraid.img diskette image to a DOS-startable (bootable) diskette in the diskette drive (a:) of the source system.
MakeDisk-ADS-Updates.bat	*Creates a Microsoft Automated Deployment Services updates diskette* This file uses loaddskf.exe to convert the adsupds.img diskette image to a DOS-startable (bootable) diskette in the diskette drive (a:) of the source system.
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.

Table 2. Descriptions of helper batch files (continued)

Helper batch file name	Description
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. **Note:** 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

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

Table 3. Environment variables in netvars.bat

Variable name	Description	Default
DPLY_IP	Defines a valid IP address that can be used during the deployment process by the target server	192.168.0.131
DPLY_NAME	Defines the unique boot media name used during the deployment process by the source system to identify the target server	sgtk1
LOGON	Defines a valid logon name that can be used to connect to the source system during the deployment process	sgtk0clnt
PWD	Defines a valid logon password that can be used with the valid logon name (LOGON) to connect to the source system during the deployment process	sgtk0clnt
SRV_GATE	Defines the gateway IP address used by the target server to access the source system	192.168.0.2
SRV_IP	Defines the IP address of the source system	192.168.0.25
SRV_MASK	Defines the subnet mask used by the source system	255.255.255.0
SRV_NAME	Defines the computer name used by the source system	sgtksrv
SRV_SHARE	Defines the shared directory on the source system that contains the necessary deployment files	sgshare
USE_DHCP	Defines whether to use DHCP for the boot media IP address. Valid values are: • Y - use DHCP • N - do not use DHCP **Note:** Selecting Y for USE_DHCP causes the value in the DPLY_IP variable, if specified, to be ignored.	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 4 on page 20 contains descriptions and default settings for the environment variables you can customize in the usrvars.bat file.

Notes:

1. All environment variables that have values of Y or N support uppercase or lowercase characters for the variable setting.
2. All environment variables 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. All environment variables 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. All environment variables that contain DIR in the name of the variable must only contain a directory name, with no preceding path information.
5. All environment variables 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 4. Environment variables in usrvars.bat

Variable name	Description	Default
General variables		
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
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)
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\diragent (m:\apps\diragent) **For network:** %TAS_PATH%\apps\diragent (m:\sgdeploy\apps\diragent)

Table 4. Environment variables in usrvars.bat (continued)

Variable name	Description	Default
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)
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
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%\w2k_srv (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.	4000
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 filename of the kickstart file.	rhas21ks.cfg
ANS_URL	Defines the kickstart file location that is passed to LOADLIN. See the "Starting a Kickstart Installation" section of the *Red Hat Linux Customization Guide* for information.	http://%SRV_IP%/%TASDIR%/%TKDIR%/examples/%NOSDIR%/redhat/%ANS_FILE% (http://192.168.0.25/sgdeploy/sgtk/examples/linux/redhat/rhas21ks.cfg)
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 RDM 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 file specified by IRD_FILE.	%OS_PATH%\images\pxeboot (m:\sgdeploy\os\rhas21\images\pxeboot)

Table 4. Environment variables in usrvars.bat (continued)

Variable name	Description	Default
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 (m:\sgdeploy\os\rhas21\images\pxeboot)
LL_PATH	Defines the fully qualified path to the LOADLIN program.	%OS_PATH%\dosutils (m:\sgdeploy\os\rhas21\dosutils)
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**	linux
OS_PATH	Defines the fully qualified path to the operating-system installation files directory in the source tree.	%TAS_PATH%\os\rhas21 (m:\sgdeploy\os\rhas21)
RDSK_SZ	Defines the size (in KB) of the Linux RAM disk to use.	8196
BIOS code and firmware code update variables		
UPD_BDIR	Defines 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.	N
UPD_SR	Specifies whether to perform IBM ServeRAID BIOS code and firmware code updates. Valid values are Y or N.	N
DO_UPDS	Specifies whether to perform any BIOS code or firmware code updates. Valid values are Y or N. **Note:** When set to N, this variable overrides the UPD_BIOS and UPD_SR variable settings.	N
UPD_SDIR	Defines the base directory that contains the IBM ServeRAID update files in the source tree.	srvraid
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
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

Table 4. Environment variables in usrvars.bat (continued)

Variable name	Description	Default
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. **Note:** When set to 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. **Note:** You can find the world-wide node name of the storage device in the controller profile in the IBM Storage Manager software, or in the FAStTUti! setup utility for the host-bus adapter.	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
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. See below for example DOS-based menus:

Table 5. 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 non-network-based deployment scenarios
PC DOS 7.1 Startup Menu ====================== 1. Run the ServerGuide Scripting Toolkit scenario 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]

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.

   Note: 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 16 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 19 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.

Because a key must be pressed, a user must be available to boot from a CD, perform updates, configure ServeRAID, and create an operating-system partition. When the installation process reaches the point where the Windows files are being copied to the operating-system partition, the installation can continue unattended because the Scripting Toolkit installation detects the existence of the _BOOT_HD.NOZ file and will not prompt the user to remove the boot media. For all subsequent reboots, the 5-second timer will handle leaving the bootable CD in the drive; it will time out, causing the system to boot from the hard disk drive.

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 @server 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 30. 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 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.

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 93 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
- 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 policy files 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 29 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
> and
> - Leave RD_FILE blank.
>
> This will cause the ServerGuide Scripting Toolkit process to run the PRAID utility with the /d switch specified, which uses defaults for all controller settings. See Table 6 on page 29 for the list of array and logical drive default settings for your RAID controllers.

Configuration using a specific policy 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 filename of the PRAID policies file that you want to use.

This will cause 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 99 for additional information about policies files.

Table 6. 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-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 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 "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,8i.

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,8i.

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 84 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

Note: 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
- FAStT FC-2 Host Bus Adapter

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 19 for information on the environment variables you can configure.

Variable name	Modification needed
DO_FIB	Set to Y for Fibre Channel configuration during the deployment process.
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. **Note:** When set to 0000, the first adapter in the target server scan order (lowest I/O address) is configured.

Variable name	Modification needed
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**
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. **Note:** You can find the world-wide node name of the storage device in the controller profile in the IBM Storage Manager software, or in the FAStTUti! setup utility for the host-bus adapter.

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 ″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 @server or IBM @server xSeries servers. See the readme.txt file for a complete list of supported Microsoft Windows operating systems. If you have not already done so, see Chapter 2, "Unpacking files and configuring the source tree," on page 5 for preliminary instructions.

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

* DOS-startable CD or DOS-startable diskette and data CD
* 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 126 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.

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

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

- One blank diskette
- One blank CD
- Source system with properly-configured source tree and a CD-R or CD-RW drive
- CD-creation software
- (Optional) *IBM UpdateXpress* CD
- (Optional) *IBM Director* 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 16 for information about this file.
   d. Customize the environment variables for this deployment scenario by running the Customize-usrvars.bat helper batch file.

      Notes:
      1) If you are using a DOS-startable (bootable) CD, set the DPLY_TYPE environment variable equal to 2.
      2) To install IBM Director Agent, set DAINST to Y and set the DA_PATH environment variable to %TKDRV%\apps\diragent (by default, m:\apps\diragent).
      3) See "Modifying USRVARS.BAT" on page 19 for information on the environment variables you can configure.
   e. Customize the answer file.
   f. Label the DOS-startable (bootable) diskette so that you can easily identify it later.

2. Create the data or DOS-startable CD:
   a. Insert a blank CD into the CD-R or CD-RW drive of the source system.
   b. When creating a DOS-startable CD, use the CD-creation software to set the boot image to the DOS-startable (bootable) diskette you created in step 1.
   c. Using the CD-creation software, create a data CD that contains at least the following directories.

      Note: The CD file system must be ISO-9660 compliant. Directories cannot be more than 8 levels deep and must comply with DOS 8.3 file naming conventions. Make sure you enable the use of all DOS 8.3 file names in the CD-creation software.

CD directory	Purpose
apps\diragent	To contain IBM Director Agent installation files
OSdevdrv	To contain device-driver files
i386	To contain the Windows operating-system installation files
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\apps\diragent	apps\diragent
sgdeploy\drvs*OSdevdr*	*OSdevdrv*
sgdeploy\os*OS*\i386	i386
sgdeploy\sgtk	sgtk

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. Copy a customized answer file from the source system to the CD; then, update the ANS_PATH and ANS_FILE environment variables with the applicable path and file name information.

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

3. (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 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 from the target server.

4. Install the operating system on the target server:

a. Insert the DOS-startable (bootable) CD into the CD or DVD drive of the target server, or insert the DOS-startable diskette into the diskette drive and the data CD into the CD or DVD 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 ServerGuide Scripting Toolkit deployment process.

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

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

a. Insert the *IBM UpdateXpress* CD into the CD 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 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

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 16 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 18 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.

      Notes:

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

      2) See "Modifying USRVARS.BAT" on page 19 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.

2. (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 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 from the target server.

3. 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.

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

   a. Insert the *IBM UpdateXpress* CD into the CD 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 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.

  Note: 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 16 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 18 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.

      Notes:

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

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

      3) See "Modifying USRVARS.BAT" on page 19 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.

      Notes:

      1) Do not 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 on page 37; 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.

2. (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 into the CD 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 or DVD drive that contains the *IBM UpdateXpress* CD.

> **Note:** 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 drive from the target server; then, remove the *IBM UpdateXpress* CD from the source system.

3. 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 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.

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

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

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

   c. When all updates are completed, unmount the CD from the target server; then, remove the *IBM UpdateXpress* CD 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 Red Hat Linux AS operating systems on IBM @server or IBM @server xSeries servers. See the readme.txt file for a complete list of supported Red Hat Linux operating systems. If you have not already done so, see Chapter 2, "Unpacking files and configuring the source tree," on page 5 for preliminary instructions.

The ServerGuide Scripting Toolkit supports the DOS-startable diskette and network share deployment scenario for Red Hat Linux AS operating systems.

Preparing the kickstart file

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

Note: The sample kickstart file provided for Red Hat Enterprise AS Linux 2.1 is for Quarterly Update 6. You might have to make changes to the kickstart file to support earlier releases of Red Hat Enterprise AS Linux 2.1.

Kickstart file name	Linux distribution
rhas21ks.cfg	Red Hat Enterprise Linux AS 2.1
rhas3ks.cfg	Red Hat Enterprise Linux AS 3

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 7 on page 40 for an example section from the Red Hat Linux AS 2.1 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/rhas21
url --url http://192.168.0.25/rhas21/
```

Table 7. Kickstart file details

Kickstart file statement	Description
nfs --server 192.168.0.25 --dir /var/www/rhas21	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/rhas21 directory of an NFS server with an IP address of 192.168.0.25.
url --url http://192.168.0.25/rhas21/	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 rhas21 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
- 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 HTTP, NFS, or FTP server.
- (Optional) *IBM UpdateXpress* CD

Notes:

1. The operating system installation files must be accessible through HTTP, NFS, or FTP, as required by the Red Hat Linux Installation program
2. 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 do not delete the partitions on a target server automatically. However, if you use the sample kickstart files that come with the ServerGuide Scripting Toolkit, the Red Hat Linux Installation program will delete all partitions.

Complete the following steps to deploy a Red Hat 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 source tree.

    c. Run the MakeDisk-Linux-Network-Install.bat helper batch file. See "Using helper batch files and creating diskette images" on page 16 for information about helper batch files.

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

f. Customize the kickstart file. (See "Preparing the kickstart file" on page 39 for information.)

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

2. (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 CD drive.

3. 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 Linux installation problems

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

- The IDE RAID controller in the BladeCenter HS20, Machine Type 8832 server and the BladeCenter HS40, Machine Type 8839 server
- Broadcom NetXtreme Gigabit Ethernet NIC present in many IBM @server and xSeries servers
- IBM ServeRAID-7e, Adaptec HostRAID controller
- IBM ServeRAID-7t, Serial-ATA (SATA) controller

You can either modify the network boot image provided with the Linux distribution, or you can replace the hardware in question. For example, you can install an IBM 10/100 Ethernet adapter and disable the on-board Broadcom adapter.

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

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 will ultimately reside. For information, see the kickstart HOWTO available from the Linux Documentation Project Web site at http://www.tldp.org.

3. Make the device drivers accessible to the Red Hat Linux installation program (anaconda) or install them manually. You must provide device drivers that are compiled for the post-installation (non-BOOT) kernels (depending on your hardware configuration, these are typically standard, smp, and bigmem): this can be accomplished in a number of ways depending on your Red Hat Linux version.

   For Red Hat Enterprise Linux {A, E}S 3 and above, you must copy the device-driver diskette image to a network accessible share; then, 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

   Note: This assumes that you have a network adapter that is recognized by the default drivers present in the ramdisk image used to install Red Hat Linux.

Chapter 7. Performing secure data disposal

This chapter contains information about using the ServerGuide Scripting Toolkit to perform secure data disposal on IBM @server or IBM @server 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

Complete the following steps to securely dispose of data on a target server.

Note: You can use the data CD and DOS-startable diskette on all servers that do not require a startable (bootable) CD. This deployment scenario is not supported on blade servers.

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 16 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 19 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 in step 1.
   b. Using the CD creation software, create a data CD that contains at least the following directories.

      Note: The CD file system must be ISO-9660 compliant. 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.

3. 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.

Chapter 8. Altiris Deployment Solution support

This chapter contains information about using the ServerGuide Scripting Toolkit with the Altiris Deployment Solution. See the readme.txt file that comes with the ServerGuide Scripting Toolkit for a current list of servers that are supported by the Altiris deployment solution. If you have not already done so, see Chapter 2, "Unpacking files and configuring the source tree," on page 5 for preliminary instructions.

At the time of this publication, you can install the Altiris Deployment Solution on the following operating systems:

- Microsoft Windows 2000 Server
- Microsoft Windows Server 2003
- Red Hat Enterprise Linux AS 2.1

Note: You must install and configure the Altiris Deployment Solution before you unpack the ServerGuide Scripting Toolkit files.

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)

Note: 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.

Overview

The ServerGuide Scripting Toolkit provides sample Altiris jobs (events) that you can import into an existing Altiris Deployment Solution installation. These sample files are packaged as Altiris-format binary files. The sample jobs perform basic hardware configuration and operating-system deployment and can also be used to create custom Altiris jobs for use with IBM @server and xSeries servers.

Two Altiris-format binary files come with the ServerGuide Scripting Toolkit. One binary file is for use only with the Altiris Deployment Solution for Windows, and the other is for use only with the Altiris Deployment Solution for Linux. See the readme.txt file that comes with the ServerGuide Scripting Toolkit for a list of the supported versions of the Altiris Deployment Solution.

Note: You cannot use the binary file that is intended for the Altiris Deployment Solution for Windows with the Altiris Deployment Solution for Linux, and vice versa. You also cannot use the binary files with earlier versions of the Altiris Deployment Solution.

Unpacking the ServerGuide Scripting Toolkit files

You must unpack the ServerGuide Scripting Toolkit files to the default shared directory of the Altiris Deployment Solution directory structure. See Chapter 2, "Unpacking files and configuring the source tree," on page 5 for information about how to unpack the ServerGuide Scripting Toolkit files and details about the source tree.

Note: In addition to unpacking the files, you must copy applicable operating-system installation files, device-driver sets, BIOS and firmware code updates, and application source files (such as IBM Director Agent) to the source tree.

There is a corresponding Altiris-format binary file for Windows operating systems and Linux operating systems. You must import the applicable Altiris-format binary file, depending upon the version of the Altiris Deployment Solution, that comes with the ServerGuide Scripting Toolkit to use the provided sample jobs.

Importing the Altiris-format binary for Windows

Complete the following steps to import the binary file into the Altiris Deployment Solution for Windows:

1. From the Altiris Deployment Solution GUI, click **File→Import/Export→Import Jobs**. The "Import Job" window opens.
2. Make sure that the **Import to Job Folder** check box is not selected; then, click **Browse**. The "Open" window opens.
3. Navigate to the *AltirisPath*\Altiris\eXpress\Deployment Server\sgdeploy\sgtk\altiris directory, where *AltirisPath* is the path to where the Altiris Deployment Solution is installed.
4. Select the Altiris-format binary file, ServerGuideToolkit-AltirisDSForWindows.bin; then, click **Open**.
5. Click **OK** to complete the import process.

   Note: The Altiris Deployment Solution for Windows prompts you to specify an account under which to run the Altiris server service. If this account has an empty password, the Altiris Agent for Linux installation might fail.

The ServerGuide Scripting Toolkit sample jobs have the following structure, within the Altiris Deployment Solution GUI. A plus (+) sign indicates a folder, and a minus (-) sign indicates a script.

```
+ ServerGuide Toolkit - IBM BladeCenter / xSeries Support
 + Applications (Post-OS Installs)
        - IBM Director Agent for Windows Install
```

```
   + Hardware Configuration Events
         - Capture RAID configuration
         - Configure Fibre
         - Configure RAID
         - Deploy RAID configuration
         - Server Disposal and reset RAID
         - Server Disposal without RAID
   + Imaging Events
         - Capture Linux image
         - Capture Windows image
         - Configure RAID and deploy Linux image
         - Configure RAID and deploy Windows image
         - Deploy Linux image
         - Deploy Windows image
 + Scripted OS Installs Events
    + Microsoft Windows 2000 OS Installation Events
             - Configure RAID and Install Windows 2000 Server
             - Configure RAID and Install Windows 2000 Advanced Server
             - Install Windows 2000 Advanced Server
             - Install Windows 2000 Server
    + Microsoft Windows 2003 OS Installation Events
             - Configure RAID and Install Windows 2003 - Enterprise
             - Configure RAID and Install Windows 2003 - Standard
             - Configure RAID and Install Windows 2003 - Web
             - Install Windows Server 2003 û Enterprise Edition
             - Install Windows Server 2003 û Standard Edition
             - Install Windows Server 2003 û Web Edition
    + Red Hat Enterprise Linux AS OS Installation Events
             - Configure RAID and Install Red Hat Enterprise Linux AS 2.1
             - Configure RAID and Install Red Hat Enterprise Linux AS 3
             - Install Red Hat Enterprise Linux AS 2.1
             - Install Red Hat Enterprise Linux AS 3
  + System Update Events
       + Windows Post-OS Updates
           - Run RemoteUX Drivers Update
           - Run RemoteUX Firmware Update
    - Capture CMOS setting to a binary file
    - Deploy CMOS setting from a binary file
    - Perform BIOS update
    - Perform IBM ServeRAID BIOS/firmware update
  + Test
    - Directory Test
```

Importing the Altiris-format binary for Linux

Complete the following steps to import the binary file into the Altiris Deployment
Solution for Linux:

1. From the Altiris Deployment Solution GUI, in the "Jobs" pane, select **Import Jobs** from the **Select Action** list. The "Import Job" page is displayed.

2. Make sure that the **Import to Job Folder** check box is not selected; then, click **Browse**. The "Open" window opens.

3. Navigate to the
   AltirisPath/adlserver/sgdeploy/sgtk/altiris directory, where *AltirisPath* is the
   path to where the Altiris Deployment Solution is installed.

4. Select the Altiris-format binary file, ServerGuideToolkit-AltirisDSForLinux.bin;
   then, click **Open**.

5. Click **OK** to complete the import process.

By default, the Altiris Deployment Solution for Linux creates a user named
"altiris" with an empty password that is used to access to the eXpress share (the F
drive: from within the DOS environment). If this user account has an empty
password, the Altiris Deployment Solution for Linux installation might fail.

If the Altiris Deployment Solution for Linux installation fails, complete the following steps:

1. Change the values of the smbuser and smbpassword entries in the file answerfile in the installation directory from smbuser=altiris smbpasswd= to smbuser=sgtk0clnt smbpasswd=sgtk0clnt.

2. Make the same changes to the user name and password information in the Altiris Kickstart file (rhas3ks.cfg and rhas21ks.cfg in the sgdeploy/sgtk/altiris/linux/redhat directory). For example:

```
mount -t smbfs -o username=sgtk0clnt,password=sgtk0clnt
//192.168.0.25/eXpress /mnt/altiris
```

> **Note:** This entry is shown with a line break after sgtk0clnt. In your file, the entire entry must all be on one line.

This example is shown using default installation values. If the customer creates an account with another user name and password, this information must be modified accordingly.

The ServerGuide Scripting Toolkit sample jobs have the following structure, within the Altiris Deployment Solution GUI. A plus (+) sign indicates a folder, and a minus (-) sign indicates a script.

```
+ ServerGuide Toolkit û IBM BladeCenter / xSeries support
 + Hardware Configuration Events
  - Capture RAID configuration
  - Configure Fibre
  - Configure RAID
  - Deploy RAID configuration
  - Server Disposal
  - Server Disposal with RAID
 + Scripted OS Installs Events
  + Red Hat Enterprise Linux AS OS Installation Events
   - Configure RAID and Install Red Hat Enterprise Linux AS 2.1
   - Configure RAID and Install Red Hat Enterprise Linux AS 3
   - Install Red Hat Enterprise Linux AS 2.1
   - Install Red Hat Enterprise Linux AS 3
  + Imaging Events
   - Capture Linux image
   - Configure RAID and deploy Linux image
   - Deploy Linux image
  + System Update Events
   - Capture CMOS setting to a binary file
   - Deploy CMOS setting from a binary file
   - Perform BIOS update
   - Perform IBM ServeRAID BIOS/firmware update
```

The default shared directory of the Altiris Deployment Solution directory structure for Linux has read and execute permissions assigned only to members of the altgrp group, as listed in the /etc/group file. If source files are installed in the recommended directory, network server daemons will not be able to access these files unless their user or group IDs are assigned read, write, and execute permissions to the default shared directory. For example, in the /etc/group file, change the line altrgrp:x:503:sgtk0clnt,root to altrgrp:x:503:sgtk0clnt,root,apache.

> **Note:** Do not create a Web alias to the root Altiris Deployment Solution directory. This directory contains files that should not be accessible to arbitrary users.

For example, you can create an alias named sgdeploy by adding the following lines to the /etc/httpd/conf/httpd.conf file:

```
Alias /sgdeploy/ "/opt/altiris/deployment/adlserver/sgdeploy/"
<Directory "/opt/altiris/deployment/adlserver/sgdeploy/">
    Options Indexes Multiviews
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
```

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:

- Applications (Post-OS installs)
  - IBM Director Agent for Windows Install
- Hardware configuration
  - Capture RAID configuration
  - Configure Fibre
  - Configure RAID (only configures RAID and then stops)
  - Deploy RAID configuration
  - Server disposal (erases data on the target server hard disks, optionally resets RAID controller to factory-default settings, and then stops)
- Scripted Installs
  - Windows 2000
    - Configure RAID and install Windows 2000 Server or Advanced Server
    - Install Windows 2000 Server or Advanced Server
  - Windows 2003
    - Configure RAID and install Windows 2003 - Enterprise, Standard, or Web
    - Install Windows 2003 - Enterprise, Standard, or Web
  - Red Had Enterprise Linux
    - Configure RAID and install Red Hat Enterprise Linux AS 2.1 or 3
    - Install Red Hat Enterprise Linux AS 2.1 or 3
- Image Installs
  - Capture Windows or Linux image (Altiris Deployment Solution for Linux supports only Linux imaging)
  - Configure RAID and deploy Windows or Linux image
  - Distribute Windows or Linux image (Altiris Deployment Solution for Linux supports only Linux imaging: optionally configures RAID on the target server and then distributes the image)
- System updates
  - Run RemoteUX to update device drivers
  - Run RemoteUX to update firmware
  - Capture CMOS settings to a binary file
  - Deploy captured CMOS settings from a binary file
  - Update server BIOS code
  - Perform IBM ServeRAID BIOS and firmware update

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.

Note: Some variables in the Scripted Install sample jobs for the Altiris Deployment Solution have different default values than for other ServerGuide Scripting Toolkit scenarios.

The usrvars.bat file refers to the following global environment variables. These variables are automatically defined during runtime.

Table 8. Global environment variables for Altiris Deployment Solution scenarios

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

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 to an IBM BladeCenter or xSeries server which has an existing Windows OS is installed. In order to perform this task using the Altiris Deployment Solution interface, the necessary IBM Director Agent files must be setup on the Altiris server before beginning the task. See "Adding IBM Director Agent installation files to the source tree" on page 13 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 @server and xSeries servers. You can

override some of the environment variables in usrvars.bat by defining new values in the Altiris Deployment Solution interface.

Note: The required BIOS files must be set up on the Altiris server before you can run this task.

See the Altiris Deployment Solution documentation for information about how to modify jobs or events. See "Modifying USRVARS.BAT" on page 19 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Note: Some variables in the sample jobs for the Altiris Deployment Solution have different default values than for other ServerGuide Scripting Toolkit scenarios.

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 @server 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.

Note: The required BIOS files must be set up on the Altiris server before you run these tasks. See "Adding BIOS code and firmware update files to the source tree" on page 12 for information.

Some xSeries servers that have different machine types use the same BIOS. The following servers are 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 19 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Note: Some variables in the sample jobs for the Altiris Deployment Solution have different default values than for other ServerGuide Scripting Toolkit scenarios.

Variable name	Description	Default
CMOS_FILE	The name of the CMOS settings binary file.	cmosinfo.bin **Note:** 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 @server and xSeries servers. You can override some of the environment variables in usrvars.bat by defining new values in the Altiris Deployment Solution interface.

Notes:

1. This job calls RemoteUX which is supplied on the RemoteUX CD. Therefore, you must set up the contents of the RemoteUX CD on the Altiris server before you can run this task. See "Performing post-installation updates with RemoteUX" on page 127 for information.
2. The Windows Altiris Agent must be installed on the target server before you run this task.

Note:

See the Altiris Deployment Solution documentation for information about how to modify jobs or events. See "Modifying USRVARS.BAT" on page 19 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Note: Some variables in the sample jobs for the Altiris Deployment Solution have different default values than for other ServerGuide Scripting Toolkit scenarios.

Variable name	Description	Default
LOGON	The name of the account.	Administrator
PWD	The password of the account. **Note:** The password cannot be empty.	password
UXDRV	The drive where the remoteux.exe utility is located.	C:
UX_PATH	The fully qualified path to the remoteux.exe utility.	%ALT_PATH%\updates\uxcd

Variable name	Description	Default
ALT_PATH	The base path to the Altiris Deployment Solution files.	"Program Files"\Altiris\eXpress\"Deployment Server"

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 @server and xSeries servers. You can override some of the environment variables in usrvars.bat by defining new values in the Altiris Deployment Solution interface.

Note: The required BIOS and firmware files must be set up on the Altiris server before you can run this task.

See the Altiris Deployment Solution documentation for information about how to modify jobs or events. See "Modifying USRVARS.BAT" on page 19 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Note: Some variables in the Scripted Install sample jobs for the Altiris Deployment Solution have different default values than for other ServerGuide Scripting Toolkit scenarios.

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	The base directory for the ServeRAID update files.	SRVRAID
UPD_SR	Specifies whether to perform ServeRAID BIOS and firmware updates. Valid values are Y and N.	Y

Configuring RAID

The RAID configuration is driven by the PRAID utility. See "PRAID.EXE" on page 93 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 19 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Note: Some variables in the sample jobs for the Altiris Deployment Solution have different default values than for other ServerGuide Scripting Toolkit scenarios.

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)

Variable name	Description	Default
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 99.	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.

Note: The two systems must have the same number of hard disk drives attached to the same type of controller.

See the Altiris Deployment Solution documentation for information about how to modify jobs or events. See "Modifying USRVARS.BAT" on page 19 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Note: Some variables in the Scripted Install sample jobs for the Altiris Deployment Solution have different default values than for other ServerGuide Scripting Toolkit scenarios.

Variable name	Description	Default
RD_FILE	The name of the RAID configuration file.	raidclon.ini **Note:** 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 **Note:** This variable only applies to the Deploy task.	N

Variable name	Description	Default
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 Install 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 19 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Note: Some variables in the sample jobs for the Altiris Deployment Solution have different default values than for other ServerGuide Scripting Toolkit scenarios.

Variable name	Description	Default
ANS_FILE	The file name for the unattended installation answer file.	win2000.txt **Note:** 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) **Note:** 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) **Note:** 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.	4000
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\diragent

Customizing Scripted Installs for Linux

The sample jobs for Scripted Install of Linux 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 19 for information about the usrvars.bat environment variables. You can override the following ServerGuide Scripting Toolkit environment variables.

Notes:

1. Some variables in the sample jobs for the Altiris Deployment Solution have different default values than for other ServerGuide Scripting Toolkit scenarios.
2. Be careful if you edit the Altiris example kickstart file. This file must contain UNIX® style EOL characters; otherwise, the agent installation script in the kickstart post-installation section (%post) will fail.

Variable name	Description	Default
ANS_FILE	The filename of the kickstart file.	rhas21ks.cfg

Variable name	Description	Default
ANS_URL	The kickstart file location that is passed to LOADLIN. See the "Starting a Kickstart Installation" section of the *Red Hat Linux Customization Guide* for information.	http://%SRV_IP%/%TASDIR%/%TKDIR%/altiris/%NOSDIR%/redhat/%ANS_FILE% (http://192.168.0.25/sgdeploy/sgtk/altiris/linux/redhat/rhas21ks.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 RDM 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\rhas21\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\rhas21\images\pxeboot)
LL_PATH	The fully qualified path to the LOADLIN program.	%OS_PATH%\dosutils (f:\sgdeploy\os\rhas21\dosutils)
OS_PATH	The fully qualified path to the operating-system installation files directory in the source tree.	%TAS_PATH%\os\rhas21 (f:\sgdeploy\os\rhas21)
RDSK_SZ	The size (in KB) of the Linux RAM disk to use.	8196

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.

Notes:

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 hostname of your system to a value that adheres to the DOS convention via

the `network` line. For more information, see the sample Altiris Kickstart file provided in the `sgdeploy/sgtk/altiris/linux` directory that makes use of system tokens, including `network`.

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.

Note: 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.

Note:

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 Linux, 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 Linux) 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.

Notes:

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) or kickstart files (for Linux)

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 @server or IBM @server xSeries servers. If you have not already done so, see Chapter 2, "Unpacking files and configuring the source tree," on page 5 for preliminary instructions.

Notes:

1. Microsoft ADS runs only under Microsoft Windows Server 2003 operating systems.
2. 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
- Reset supported RAID adapters and controllers in the target server to factory-default settings
- 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

Performing secure data disposal

You must have the following items to use this 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

There are several options for performing secure data disposal on a server. Each option requires a unique diskette image file, a unique Microsoft ADS sequence, and a unique Microsoft ADS job template. For each secure data disposal option that you wish to perform on one or more target servers, complete the following steps:

1. Create a ServerGuide Scripting Toolkit Microsoft ADS disposal 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-ADS-Disposal.bat helper batch file. See "Using helper batch files and creating diskette images" on page 16 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 18 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. See "Modifying USRVARS.BAT" on page 19 for information about the environment variables you can configure.

f. Label the diskette so that you can easily identify it later.

2. From a command prompt, change to the Microsoft ADS tftproot directory (Program Files\Microsoft ADS\tftproot by default).

3. Create a Microsoft ADS-formatted virtual diskette image by typing:

   `dskimage.exe a: DiskName`

   where *a* is the diskette drive on the source system and *DiskName* is a DOS 8.3 file name for the diskette image file.

4. Create a Microsoft ADS sequence that starts (boots) the target server using this diskette image file. See "Creating a Microsoft ADS sequence" on page 64.

5. Create a Microsoft ADS job template that calls the Microsoft ADS sequence you just created. See "Creating a Microsoft ADS job template" on page 65.

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

This procedure causes ADS to reboot the target server to a DOS-based menu, allowing you to select which NIC driver to use and then continue with the task or exit to a DOS prompt. To further automate the task, modify the CONFIG.SYS file on the boot diskette (see "Preventing DOS-based menus from displaying" on page 24); then, re-create the Microsoft ADS-formatted virtual-diskette image. Running the job template with the new virtual-diskette image will cause the task to complete without user intervention at the target server.

Configuring supported RAID adapters and controllers

You must have the following items to use this 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

There are several options for configuring supported RAID adapters and controllers. Each option requires a unique diskette image file, a unique Microsoft ADS sequence, and a unique Microsoft ADS job template. For each RAID adapter or controller configuration that you wish to perform on one or more target servers, complete the following steps:

1. See Chapter 4, "Configuring RAID and Fibre Channel," on page 27 for information on how to create and customize files in the applicable directories of the source tree for specific RAID controllers.

2. Create a ServerGuide Scripting Toolkit Microsoft ADS RAID configuration 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-ADS-Raid-Config.bat helper batch file. See "Using helper batch files and creating diskette images" on page 16 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 18 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. See "Modifying USRVARS.BAT" on page 19 for information about the environment variables you can configure.

f. Label the diskette so that you can easily identify it later.

3. From a command prompt, change to the Microsoft ADS tftproot directory (Program Files\Microsoft ADS\tftproot by default).

4. Create a Microsoft ADS-formatted virtual diskette image by typing:

   `dskimage.exe a: DiskName`

   where *a* is the diskette drive on the source system and *DiskName* is a DOS 8.3 file name for the diskette image file.

5. Create a Microsoft ADS sequence that starts (boots) the target server using this diskette image file. See "Creating a Microsoft ADS sequence" on page 64.

6. Create a Microsoft ADS job template that calls the Microsoft ADS sequence you just created. See "Creating a Microsoft ADS job template" on page 65.

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

This procedure causes ADS to reboot the target server to a DOS-based menu, allowing you to select which NIC driver to use and then continue with the task or exit to a DOS prompt. To further automate the task, modify the CONFIG.SYS file on the boot diskette (see "Preventing DOS-based menus from displaying" on page 24); then, re-create the Microsoft ADS-formatted virtual-diskette image. Running the job template with the new virtual-diskette image will cause the task to complete without user intervention at the target server.

Updating server BIOS code and IBM ServeRAID BIOS and firmware code

You must have the following items to use this 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

Note: 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 will not be able to update the server BIOS code or IBM ServeRAID BIOS and firmware code. See the readme.txt 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.

There are several options for updating the server BIOS code and IBM ServeRAID BIOS and firmware code. Each option requires a unique diskette image file, a unique Microsoft ADS sequence, and a unique Microsoft ADS job template. For each server BIOS code and for each IBM ServeRAID firmware update that you wish to perform on one or more target servers, complete the following steps:

1. See "Including BIOS code and firmware updates in a deployment scenario" on page 121 for information on how to include the necessary BIOS code update files in the source tree.

2. 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-ADS-Updates.bat helper batch file. See "Using helper batch files and creating diskette images" on page 16 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 18 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. See "Modifying USRVARS.BAT" on page 19 for information about the environment variables you can configure.

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

3. From a command prompt, change to the Microsoft ADS tftproot directory (Program Files\Microsoft ADS\tftproot by default).

4. Create a Microsoft ADS-formatted virtual diskette image by typing:

   `dskimage.exe a: DiskName`

   where *a* is the diskette drive on the source system and *DiskName* is a DOS 8.3 file name for the diskette image file.

5. Create a Microsoft ADS sequence that starts (boots) the target server using this diskette image file. See "Creating a Microsoft ADS sequence" on page 64.

6. Create a Microsoft ADS job template that calls the Microsoft ADS sequence you just created. See "Creating a Microsoft ADS job template" on page 65.

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

This procedure causes ADS to reboot the target server to a DOS-based menu, allowing you to select which NIC driver to use and then continue with the task or exit to a DOS prompt. To further automate the task, modify the CONFIG.SYS file on the boot diskette (see "Preventing DOS-based menus from displaying" on page 24); then, re-create the Microsoft ADS-formatted virtual-diskette image. Running the job template with the new virtual-diskette image will cause the task to complete without user intervention at the target server.

Installing IBM Director Agent

Microsoft ADS does not directly support application installations. The ServerGuide Scripting Toolkit can, however, be used with Microsoft ADS to copy the application installation files to the target server during the deployment process; then, modify the target server system registry so that the application installation runs the next time the server restarts (reboots). See Chapter 4, "Configuring RAID and Fibre Channel," on page 27 for more information.

Note: You must have a source system with a properly configured source tree to use this scenario.

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

1. See "Adding IBM Director Agent installation files to the source tree" on page 13 for information on adding the IBM Director Agent installation files to the source tree.
2. Click **Start→All Programs→Microsoft ADS→Sequence Editor**.
3. In the **Description** field, type a description 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\diragent directory on the Microsoft ADS server.
6. In the **Target File** field, type

   `\device\harddisk0\partition1\`*WinOS*`\Temp\`*Filename*

   where *WinOS* is the directory where the operating system is installed on the target server and *Filename* is the name of the same file as in step 5.
7. Repeat step 4 to step 6 for all files in the \sgshare\sgdeploy\apps\diragent directory.
8. Click **Actions→Add→Configure Registry**.
9. On the "Configure Registry" page, complete the following steps:
   a. Click the **Load hive** button.
   b. Type

      `\device\harddisk0\partition1\`*WinOS*`\CurrentVersion\RunOnce`

      where *WinOS* is the directory where the operating system is installed on the target server, in the field just to the right of the **Load hive** button.
   c. Type

      `DASetup`

      in the **Name** field.
   d. Type

      `Microsoft\Windows\CurrentVersion\RunOnce`

      in the **Registry Key** field.
   e. Select **REG_SZ** from the **Type** drop-down list.

f. Check the contents of the \sgshare\sgdeploy\apps\diragent folder on your ServerGuide Toolkit.

- If the ibmsetup.exe file is in the folder, type the following in the **Value** field:

```
\WinOS\Temp\ibmsetup.exe UNATTENDED rsp="\WinOS\Temp\diragent.rsp"
```

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

- If the IBMDirectorAgent4.*xx*.exe file is in the folder, type the following in the **Value** field:

```
\WinOS\Temp\IBMDirectorAgent4.xx.exe -s -a SILENT
```

   where *WinOS* is the directory where the operating system is installed on the target server and *xx* is the minor version number.

10. Click **Actions→Add→Script Command**.

11. On the "Script Command" page, complete the following steps:

   a. Type

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

   in the **Command** field.

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

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

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

12. Click **Actions→Insert→Reboot Computer**.

13. Save this new Microsoft ADS Sequence.

14. Create a Microsoft ADS job template that calls the Microsoft ADS sequence you just created. See "Creating a Microsoft ADS job template" on page 65.

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

   Note: The IBM Director Agent installation process requires administrator-level access. The first time the server restarts (reboots) after this task completes, you must logon with an account that has administrator-level access.

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. In the **Description** field, type a description for the new sequence.

3. Click **Actions→Insert→Reboot Computer**.

4. Click **Actions→Add→Boot Virtual Floppy**.

5. In the left pane, click **Boot Virtual Floppy**.

6. In the **Description** field, type a description.

7. In the **Floppy path** field, type the file name of the Microsoft ADS-formatted virtual diskette image file you created for the deployment scenario.

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**.
11. Continue with "Creating a Microsoft ADS job template."

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. From the Microsoft ADS Console, expand the **Automated Deployment Services** tree.
3. Right-click **Job Templates➔New Job Template**. The New Job Template wizard starts and the "Welcome to the New Job Template Wizard" window opens.
4. Click **Next**. The "Template Type" window opens.
5. Select **An entirely new template** and click **Next**. The "Name and Description" window opens.
6. In the **Name** field, type a descriptive name for the new job template.
7. In the **Description** field, type an optional description; then, click **Next**. The "Command Type" window opens.
8. Select **Task Sequence** and click **Next**. The "Task Sequence File and Arguments" window opens.
9. To select the task sequence file, click the **Browse** button located to the right of the **Task sequence file** field. Locate the applicable task sequence file you created in "Creating a Microsoft ADS sequence" on page 64 or "Installing IBM Director Agent" on page 63 and click **Next**. The "Destination Device Selection" window opens.
10. Select **None** and click **Next**. The "Completing the New Job Template Wizard" window opens.
11. Click **Finish**.

Adding device drivers to the Microsoft ADS device-driver set

Microsoft ADS requires the latest device drivers for network interface cards (NICs) and storage controllers. This section contains information about adding the following specific device drivers to the ADS device-driver set:

- Broadcom NIC device driver
- Intel™ NIC device driver
- Mass-storage device drivers

Attention: Microsoft ADS version 1.0 might fail to process some device-driver INF files. This is a known Microsoft ADS problem, and might prevent the ADS Deployment Agent Builder from starting. See Microsoft Knowledge Base articles 829053 and 830413 for information and to obtain the latest Hotfixes related to this problem. A future Microsoft Windows Server 2003 Service Pack might include these Hotfixes. Alternatively, you can change the device-driver INF files to solve this problem for specific devices.

You must have the following items to use this scenario:

- *IBM ServerGuide Setup and Installation* CD, version 7.3 or higher.

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

1. Copy the contents of the following directories from the *IBM ServerGuide Setup and Installation* CD to the Microsoft ADS\nbs\repository\user\PreSystem directory on the Microsoft ADS server.

Device	Copy from: the *IBM ServerGuide Setup and Installation* CD
Broadcom NIC	\sguide\w2003drv\$oem$\$1\drv\bc
Intel NIC	\sguide\w2003drv\$oem$\$1\drv\e1 and \sguide\w2003drv\$oem$\$1\drv\e2
Mass-storage	\sguide\w2003drv\oem\textmode

2. If the Microsoft ADS Console is running, shut it down.
3. Click **Start → All Programs → Administrative Tools → Services**. The "Services" window opens.
4. Right-click **ADS Deployment Agent Builder**; then, click **Stop**.
5. If you have not installed the Microsoft ADS 1.0 Hot Fixes or a Microsoft ADS update that includes the fixes for processing device driver INF files, complete the steps in "Adding Broadcom Ethernet controller device drivers," "Adding Intel Ethernet controller device drivers," and "Adding mass-storage device drivers" on page 67.
6. Right-click **ADS Deployment Agent Builder**; then, click **Start**.
7. Close the "Services" window.
8. Start the Microsoft ADS Console.

Adding Broadcom Ethernet controller device drivers

Complete the following steps if you have a Broadcom Ethernet controller in the target server:

1. Open the b57win32.inf file in an ASCII text editor. This file is in the Microsoft ADS\nbs\repository\user\PreSystem directory on the Microsoft ADS server.
2. Find the section that contains the following text:

   ```
   [Manufacturer]
   %BRCM%=Broadcom, NTx86.5.1
   ```

3. Change the information above to the following text:

   ```
   [Manufacturer]
   %BRCM%=Broadcom.NTx86.5.1
   ```

   Note: Make sure that the value of the %BRCM% variable matches the text contained in the section heading that follows.

4. Save and close the b57win32.inf file.

Adding Intel Ethernet controller device drivers

Complete the following steps if you have an Intel Ethernet controller in the target server:

1. Open the e1000325.inf file in an ASCII text editor. This file is in the Microsoft ADS\nbs\repository\user\PreSystem directory on the Microsoft ADS server.

2. Find the section that contains the following text:

```
[Manufacturer]
%Intel%=Intel,NTx86,NTIA64
```

3. Change the information above to the following text:

```
[Manufacturer]
%Intel%=Intel.NTx86
```

Note: Make sure that the value of the %Intel% variable matches the text contained in the section heading that follows.

4. Save and close the e1000325.inf file.

5. Open the e100b325.inf file in an ASCII text editor. This file is in the Microsoft ADS\nbs\repository\user\PreSystem directory on the Microsoft ADS server.

6. Find the section that contains the following text:

```
[Manufacturer]
%V_INTEL%=Intel, NTx86
```

7. Change the information above to the following text:

```
[Manufacturer]
%V_INTEL%=Intel.NTx86
```

Note: Make sure that the value in the %V_INTEL% variable matches the text contained in the section heading that follows.

8. Save and close the e100b325.inf file.

Adding mass-storage device drivers

Complete the following steps if you have at least one of the following IBM ServeRAID controllers installed in the target server:

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

1. Open the nfrd960.inf file in an ASCII text editor. This file is in the sguide\w2003drv\oem\textmode directory on the ServerGuide CD.

2. Find the section that contains the following text:

```
[Manufacturer]
%IBM%=IBM,NTx86.5.2
```

3. Change the information above to the following text:

```
[Manufacturer]
%IBM%=IBMNTx86.5.2
```

Note: Make sure that the value of the %IBM% variable matches the text contained in the section heading that follows.

4. Save and close the nfrd960.inf file.

Chapter 10. IBM Remote Deployment Manager support

This chapter contains information about using the ServerGuide Scripting Toolkit with IBM Remote Deployment Manager on IBM @server or IBM @server xSeries servers. See the readme.txt file that comes with the ServerGuide Scripting Toolkit for information about the supported versions of Remote Deployment Manager.

Note: At the time of publication, only the ServerGuide Scripting Toolkit scenarios that deploy Windows operating systems have been tested for compatibility with IBM Remote Deployment Manager.

Overview

IBM Remote Deployment Manager gives administrators centralized control over many servers, workstations, desktop computers, point-of-sale systems, and mobile computers that are connected to a local-area network (LAN) or wide-area network (WAN). Using Remote Deployment Manager tasks in IBM Director Console, you can remotely deploy operating systems, applications, server BIOS code updates, and other firmware updates.

See the documentation that comes with Remote Deployment Manager for information on hardware and operating-system support, and how to use the program. See "IBM deployment resources on the World Wide Web" on page vi for Web links to related information.

Using ServerGuide Scripting Toolkit with Remote Deployment Manager

You can use Remote Deployment Manager to clone an image from a target server that was originally deployed by a ServerGuide Scripting Toolkit Windows deployment scenario. The Get Donor task in Remote Deployment Manager will create an image of the server configuration that you can use with the Windows Clone Install task to deploy to other similarly configured servers.

This is useful when you have more than one target server that requires the same operating-system configuration. You can deploy once using the ServerGuide Scripting Toolkit and ensure that you have the latest device drivers on the target server; then, use Remote Deployment Manager to automate and schedule the deployment to other target servers.

Note: Do not clone IBM Director Agent code. If the target server has IBM Director Agent code installed, you must uninstall the IBM Director Agent code before you clone the image. After you clone the image, you can install the IBM Director Agent code on the servers.

See the documentation that comes with IBM Remote Deployment Manager for information on how to use the program to clone a target server.

Chapter 11. Using 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.

To use these utilities, select **Go to command prompt** from the ServerGuide Scripting Toolkit menu. The utilities are located in the `utils` directory.

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

Note: 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. 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 controller for information about this utility.

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

```
acu </Rfilename|/Pfilename|/D[ALL|n]> [/ALL|/Cn] [/I] [/S] [/Lfilename]
```

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
/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 acu.exe utility usage.

Example	Description
`acu /Ra:\cfg1.txt`	Records the controller configuration to 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.

Note: The ACU.EXE utility for configuring the ServeRAID-7t has been renamed to ACUSAS.EXE to avoid naming collisions.

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 server for information about this utility.

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

`acuichsv </Rfilename|/Pfilename|/D[ALL|n]> [/ALL|/Cn] [/I] [/S] [/Lfilename]`

Parameter	Description
`/Rfilename`	Scans the controller for configuration information and saves to the specified file
`/Pfilename`	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
`/Cn`	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
`/Lfilename`	Creates a log file, with the specified file name, and records any errors encountered during the configuration process

The following examples illustrate acuichsv.exe utility usage.

Example	Description
`acuichsv /Ra:\cfg1.txt`	Records the controller configuration to a file named cfg1.txt
`acuichsv /Pa:\cfg1.txt`	Uses the cfg1.txt file to configure a 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 8icontroller. 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 controller for information about this utility.

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

```
acusas </Rfilename|/Pfilename|/D[ALL|n]> [/ALL|/Cn] [/I] [/S] [/Lfilename]
```

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
/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 acusas.exe utility usage.

Example	Description
`acu /Ra:\cfg1.txt`	Records the controller configuration to 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.

Note: This utility has been renamed. The original name of the utility is ACU.EXE. The configuration utility for the ServeRAID-7t controller is also named ACU.EXE, so it has been renamed to avoid naming collisions.

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 </b|/h|/r|/?> [/f:filename]`

Parameter	Description
/?	Displays all parameters
/b	Saves a backup image of the boot sector on the diskette, and bypasses the diskette drive in the startup (boot) sequence **Note:** This parameter is not supported on all target servers. See the readme.txt file for information.
/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 /b /fa:\saveboot.sec`	Bypasses the diskette drive and 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
`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 @server 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

CLINI.EXE, CLINI32.EXE

The Command Line INI utility can perform the following functions:
- Write information to an INI file:
  - Add new sections, items, or values
  - Change existing sections, items, or values
- Read a value from an INI file and store it as an environment variable

Two versions of the Command Line INI utility come with the ServerGuide Scripting Toolkit: A 16-bit version for DOS (clini.exe) and a 32-bit version for Windows (clini32.exe).

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.

Note: If the batch file already exists, it is deleted and recreated with the new information.

The clini.exe and clini32.exe utilities have 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:filename] [/D:delimiter] [/T:n] [/N] [/O]
```

Parameter	Description
filename	Defines the fully qualified path to the INI file that the Command Line INI utility will 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.
/B:*filename*	Defines the fully qualified path and file name of the batch file to create to be used 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.
/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 will result 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 when the /E parameter is used.
/=:*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.
/N	If the specified INI file exists, it is deleted and a new INI file is created. This parameter is not valid if the /E parameter is used.
/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 Command Line INI utility returns the following values to indicate status:

Value	Description
0	Success
1	Syntax error
2	Program error

Value	Description
3	Destination is read-only
4	Current working directory is read-only
5	File not found

The following examples illustrate Command Line INI utility usage.

Example	Description
`clini info.ini /S:Hardware /I:Machine Type /V:8549 /N`	Creates a new INI file named info.ini with a section called Hardware and one item, Machine Type, is set equal to "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
`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`

Example	Description

`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 vale of the item to the MyEVariable environment variable. The resulting CLIniSet.bat file contains:
	`Set MyEVariable=Value2`

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 the **OK** button.

DSCAN.EXE, DSCAN32.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.
- Create a text mode directory, copy all text mode device drivers into that directory, and then dynamically create a master txtsetup.oem file containing all information that is in the individual txtsetup.oem files. Each entry in the master txtsetup.oem file is unique, and known unattended installation defects are automatically addressed.

Two versions of the Driver Scan utility come with the ServerGuide Scripting Toolkit: a 16-bit version for DOS (dscan.exe) and a 32-bit version for Windows (dscan32.exe).

In the directory structure required for that all device driver files are located in device-driver directories. These directories are named for the device driver they contain.

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 dscan32 command runs only on Windows; the dscan command runs on both DOS and Windows. You can use Windows long file names with dscan32, but you must use file names that follow the DOS 8.3 format with 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 81 for information about the drvinfo.ini file. See "UNATTEND.EXE" on page 112 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 and dscan32.exe utilities have the same command-line syntax, as shown in the following example:

```
dscan <driverpath> [/S|/SS|/T[:path]] [/M:machinetype] [/V:n] [/W:n] [/?]
```

Parameter	Description
/?	Displays all parameters
driverpath	Defines the fully qualified path to the directory to scan for device drivers. Each driver is assumed to be in a separate sub-directory within this path. If *driverpath* has oem in the path, the Driver Scan utility will create the oem\textmode directory and merge 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 only scan device drivers and create drvinfo.ini files, if necessary. Text mode device drivers are not merged when this parameter is used.
/SS	Specifies to only scan a single device driver and create the drvinfo.ini file, if necessary. Text mode device drivers are not merged when this parameter is used.
/T[:*path*]	Specifies to only build the text mode device drivers. 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 *driverpath* parameter must have oem in the path so that the text mode device drivers are merged into the oem\textmode directory.
/M:*machinetype*	Specifies a machine type, where *machinetype* is the machine type of the target server, 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:*machinetype*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. Valid values are 0 for Windows 2000 and 1 for Windows Server 2003.

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

Value	Description
0	Success
1	Syntax error

Value	Description
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.
	Notes:
	1. To use this variable, the drvutils directory must contain the Holdit.exe and Reboot.exe utilities.
	2. 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 specific device driver. This variable is only valid when Installation Mode is set to "PnP". Valid values are the name of the directory where the device driver is located, or All. If more than one device driver has this value set to "All", the device drivers are installed before those that do not have any other 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".

Variable name	Description
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`
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. **Notes:** 1. To use this variable, the drvutils directory must contain the Holdit.exe and Reboot.exe utilities. 2. This variable is only supported for Windows 2000.
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. **Note:** You cannot use both Supported Locales and Unsupported Locales 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. **Note:** 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. **Note:** 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. **Note:** 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

`[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 needs to 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 only supports 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
/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
- FAStT FC-2 Host Bus Adapter

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 86 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.

Note: 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
/E*addr*	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.
/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, or in the FAStTUti! setup utility for the host-bus adapter.) *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. **Note:** 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. **Note:** 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 **Note:** 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.

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
```

Note: 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 84 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

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.

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 xSeries server. The return values are: **0** for an IBM xSeries server **1** for a non-IBM xSeries server
/p	Displays all hardware information for the target server in a variable=value format. The return value is 0 when successful. **Note:** You can use the > DOS 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. **Note:** You can use the > DOS 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
```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
```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 **Note:** 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.

Example	Description
Below is an example of what the hwdetect.out file created in the above example might contain:	

```
[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
Total_Enabled_Memory_Mb=256
ROM_Diagnostics_Build_Level=EME112A
ISMP_Build_Level=BR8T30A
RSA_Build_Level=GEE834A

[PCI]
Total_Number_Devices=10
Bus_Number.0=0
Device_Number.0=1
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
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

...
```

Notes:

1. The date is listed in mm/dd/yyyy format.

2. There will be an entry for Processor_Family and Processor_Speed_MHz for each microprocessor in the server.

3. The ROM_Diagnostics_Build_Level will be empty for servers that don't support this.

4. PCI devices are listed in the order they are scanned.

5. PCI devices are listed in the *Value.n* format, where *Value* is the variable name and *n* is the nth PCI device scanned.

6. The format returned when the /p parameter is used, has "System_" or "PCI_", as applicable, preceding each of the above values, as in the following example:

```
System_Machine_Type=8674
System_Model_Number=42X

PCI_Total_Number_Devices=10
PCI_Bus_Number.0=0
```

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 @server 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
/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 84 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.

Note: 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
`ipssend setconfig 1 default`	Resets the first controller to factory-default settings

Note: 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.

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 108 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

> **Note:** Both ServeRAID-8i and ServeRAID-7t require a utility called ACU. However, each controller requires a different version of this utility. If you

are using a ServeRAID-8i you must therefore rename the utility to ACUSAS to differentiate between the two versions of ACU.

- ACUSAS: ServeRAID-8i
- ACUICHSV: ServeRAID-7e SATA
- RAIDSEL: ServeRAID-7e SCSI
- CFG1030: LSI 1020/1030 chipset
- HYPERCFG: IDEal RAID

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'`

Notes:

1. 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.
2. 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 9. 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 6 on page 29 for a list of default values for each RAID controller. **Note:** You cannot use this parameter with the /f parameter.

Table 9. Configure mode parameters (continued)

Parameter	Description
/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 108 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.
/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. **Note:** 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 may 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).

Table 9. Configure mode parameters (continued)

Parameter	Description
`@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 may exceed the 127-character command line limit in DOS.

Table 10. 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 may 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).

Table 10. Capture mode parameters (continued)

Parameter	Description
@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 may exceed the 127-character command line limit in DOS.

Table 11. 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 may 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 may exceed the 127-character command line limit in DOS.

Usage examples

Table 12. 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.
`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 13. 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 14. 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 15. 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.

Note: 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" 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 16. 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 102 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 102 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 103 for additional information.
RebuildRate	No	HIGH	Specifies the rebuild rate that should be applied to the RAID controller. See "RebuildRate" on page 103 for additional information.
StripeSize	No	• 8 (for ServeRAID 4H, 4Mx, 4Lx, 5i, 6i, 6i+, 6M, and 7k) • 64 (for ServeRAID 7t, 8i, 7e-SCSI, 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 103 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 104 for additional information.
Array_Defaults	No	• 0%:1 for 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 104 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 105 for additional information.
Hotspares	No	None	Defines a list of specific physical drives to designate as hot-spare drives. See "Hotspares" on page 105 for additional information.

Table 16. Policy file parameters (continued)

Keyword	Required?	Default	Description
Logical_Mode	No	AUTO	Defines the logical-drive creation policy to use when creating logical drives. See "Logical_Mode" on page 105 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 106 for additional information.
Logical.*num*	No	None	Lets you specify how many logical drives are created and the specific parameters for each logical drive. See "Logical.*num*" on page 106 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.

  Note: 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
- LSI-SCSI-RAID
- LSI-IDEal-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. Note that 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 107 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 107 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 107 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.

> **Note:** 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 may 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.

  Note: 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* may be in any of the following formats:
- A positive integer – specifies the size in MB.
- A percentage – specifies that a percentage of the total space should be used.
- FILL – indicates that all available space on the array should be used.

Raidlevel specifies the RAID level for the logical drive. See "Supported settings for RAID controllers" on page 107 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" on page 107 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" on page 106.

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 17 lists the supported settings for each RAID controller when using PRAID.

Note: 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 17. 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-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 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 18 shows the default RAID values that PRAID will use when AUTO is specified for *raidlevel*.

Table 18. Default RAID levels

Number of drives in array	Controller types	RAID level
1	ServeRAID-4H,4Mx,4Lx,5i,6i,6M,7k,8i,LSI-IDEal	0
1	ServeRAID-7t	VOLUME
2	All	1
3	ServeRAID-7t	5
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.

Note: 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.

Values are returned using the errorlevel DOS environment variable so that you can create a batch file to branch according to the value returned. The savestat.exe utility, version 1.01 that comes with the ServerGuide Scripting Toolkit has the following command-line syntax:

`savestat </setn=value|/getn|/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

Parameter	Description
/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
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 **Note:** 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.

Note: 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

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.

There are three types of device drivers that the unattend.exe utility processes:
- 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 [/?] <*filename*|/U:*file* /D:*path* /H:*file* /I:*path*> [/U:*file*] [/D:*path*] [/H:*filename*] [/I:*path*] [/S:*drive*] [/C] [/T] [/P] [/E] [/V:*n*]

Parameter	Description
/?	Displays all parameters
filename	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. **Notes:** 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. **Note:** 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

Value	Description
5	Specified file name does not exist
6	Specified path does not exist

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.

Notes:

1. None of the commands can cause the server to restart (reboot).
2. 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. **Note:** 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.

Variable name	Description
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. **Note:** 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. **Note:** 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.
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 |

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=

[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.txt 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)

   Note: 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.

   Note: 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 "DSCAN.EXE, DSCAN32.EXE" on page 78 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 are already configured for use with this utility.

If you do not use the unattend.exe utility along with the configured device drivers, you must 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.

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.

Including BIOS code and firmware updates in a deployment scenario

The ServerGuide Scripting Toolkit sample script files are configured to automate server 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 post-installation updates with RemoteUX" on page 127 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.

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

3. (Optional) If the file that you downloaded from the IBM Support Web site is in compressed format, you must uncompress the file before you can use it in a deployment scenario.

4. Double-click the executable file. A 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 19 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.

```
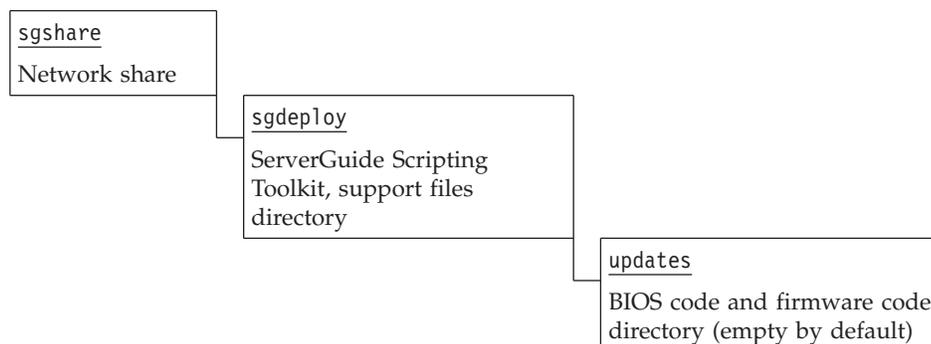sgshare
Network share
```
```
    sgdeploy
    ServerGuide Scripting
    Toolkit, support files
    directory
```
```
        updates
        BIOS code and firmware code
        directory (empty by default)
```

For IBM ServeRAID updates, create a srvraid directory in the sgdeploy\updates directory.

Note: 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 biosfw*number*.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. Insert a diskette into the diskette drive of your local system.

    b. Extract the first biosfw*number*.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 biosfw*number*.img 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 biosfw*number*.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 19 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 UpdateXpress* CD contains two types of server BIOS code and IBM ServeRAID BIOS and firmware code updates:

- DOS-based
- Combined DOS and Windows

You must 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.

```
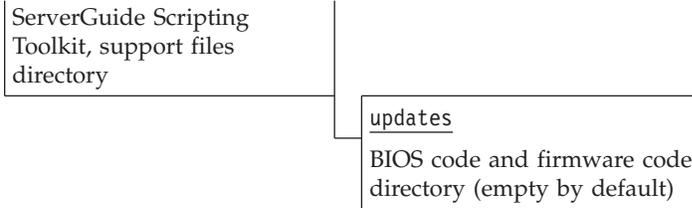sgshare
```
Network share

```
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.

Note: 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. In a system running Windows, insert the *IBM UpdateXpress* CD in the CD or
DVD drive. If the IBM Update*Xpress* program starts, exit the program.

4. Navigate to the CD and open the index.htm file in the root directory of the
IBM UpdateXpress CD.

5. In the left pane, click the server for which you want 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 on page 125.

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-Web executable file format that you will have to extract later.

19. After the files are extracted, click **Finish**.
20. Repeat step 5 on page 124 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 19 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 post-installation updates with RemoteUX

You can use the RemoteUX program to push BIOS code updates, firmware code updates, and device-driver file updates to a remote target server that has an operating system installed. When running the RemoteUX program, you must insert the *IBM UpdateXpress* 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.

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

Complete the following steps to use RemoteUX to perform post-installation updates to a target server:

1. Insert the *IBM UpdateXpress* CD into the CD drive of the source system.
2. From a command prompt, change directories to the CD drive.
3. Run the remoteux.exe program with the applicable parameters to update the target server.

   For example, to update the BIOS code and firmware code on a remote server named remote:

   ```
   remoteux.exe \\remote -user:administrator -pwd:password -firmware -a -r
   ```

   To update the device drivers on a remote server with an IP address of 192.168.0.150:

   ```
   remoteux.exe \\192.168.0.150 -user:fred -pwd:password -drivers
   ```

   Valid parameters are displayed if you run remoteux.exe without any parameters. See the documentation that comes with Update*Xpress* for information.

Using RemoteUX with Altiris Deployment Solution for Windows to perform post-installation updates

Complete the following steps to use RemoteUX to perform post-installation updates with Altiris Deployment Solution to a target server:

1. Navigate to the
   C:*AltirisPath*\Altiris\eXpress\Deployment Server\sgdeploy\sgtk\updates
   directory, where *AltirisPath* is the path to where the Altiris Deployment
   Solution is installed.
2. Add a directory named uxcd to the updates directory.
3. Insert the *IBM UpdateXpress* CD into the CD drive of the source system.
4. From a command prompt, copy the entire contents of the *IBM UpdateXpress*
   CD, including subdirectories, to the uxcd directory that you created in step 2.
5. Remove the *IBM UpdateXpress* CD from the CD drive of the source system.
6. Customize the applicable Altiris job. See "Updating drivers or firmware with
   RemoteUX (Windows only)" on page 52 for information.
7. Run the job.

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 information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

> IBM Director of Licensing
> IBM Corporation
> North Castle Drive
> Armonk, NY 10504-1785
> U.S.A.

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product, and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Some software may differ from its retail version (if available) and may not include all user manuals or all program functionality.

IBM makes no representations or warranties regarding third-party products or services.

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
@server	ServerGuide
IBM	ServerProven
IBM (logo)	xSeries

Adaptec and HostRAID are trademarks of Adaptec, Inc., in the United States, other countries, or both.

Intel, MMX, 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 Windows NT are trademarks of Microsoft Corporation in the United States, other countries, or both.

Red Hat, the Red Hat "Shadow Man" logo, and all Red Hat-based trademarks and logos are trademarks or registered trademarks of Red Hat, Inc., in the United States and other countries.

Other company, product, or service names may 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 may 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 may differ from its retail version (if available), and may not include user manuals or all program functionality.

Index

A

ACU.EXE 71
ACUICHSV.EXE 72
ACUSAS.EXE 73
adding files to the source tree 10
ALTBOOT.EXE 74
Altiris deployment, unpack instructions 6
answer file
    updating 118
    updating dynamically 126
automating scenarios 24

B

batch files,helper 16
BIOS and firmware updates 12, 121
boot-media prompt, prevent display 24
boot-media prompt, prevent display with CD-ROM 25

C

CFG1030.EXE 75
CLINI.EXE 75
CLINI32.EXE 75
configuration files
    fibre channel 30
configuring
    fibre channel 27
    RAID 27
    source tree 5, 8
custom HAL, device drivers 112
customer support vi
customizing toolkit scenarios 15

D

deployment scenarios
    enhancing 121
    supported 2
device drivers
    adding 117
    custom HAL 112
    executable 112, 120
    HAL, custom 119
    plug-and-play 112, 119
    removing 118
    scanning 78
    text mode 112, 119
    Windows-based operating systems 11, 117
DOS-based menus, preventing display of 24
DRVINFO.INI 81
DRVINST.EXE 78
drvutils directory 78
DSCAN.EXE 78
DSCAN32.EXE 78
dynamically updating the answer file 126

E

eFixes vi
executable device drivers 112
executable, device drivers 120

F

FDISK32.EXE 83
features 1
fibre channel
    adapters supported 30, 84
    configuration 27, 30
FIBRECFG.EXE 84
files
    unpacking 5

G

global environment variables 15

H

HAL
    custom 119
hardware
    supported 3
help vi
helper batch files 16
Holdit.exe 78

I

IBM Web site
    RDM resources vi
    Remote Deployment Manager vi
    ServerGuide vi
    ServerGuide Scripting Toolkit vi
    ServerProven vii
    Support vii
    Systems Management Software vii
    UpdateXpress vi
    xSeries Systems Management vii
IBMUTIL.EXE 91
IDE RAID controller 41
introduction 1

L

Linux unpack instructions 7

M

Microsoft Automated Deployment Services 7
modifying
    NETVARS.BAT 18
    USRVARS.BAT 19

IBM ®

Part Number: 31R1049

Printed in USA