

IBM ServerGuide Scripting Toolkit



User's Reference, Version 1.0

Note: Before using this information and the product it supports, read the general information in Appendix D, “Notices”, on page 87.

First Edition (December 2003)

© Copyright International Business Machines Corporation 2003. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Figures	v
Tables	vii
About this book	ix
How this book is organized.	ix
Notices that are used in this book	x
IBM deployment resources on the World Wide Web.	x
Chapter 1. Introducing ServerGuide Scripting Toolkit	1
Overview of the ServerGuide Scripting Toolkit	1
Deployment scenarios.	1
Supported @server and xSeries servers.	2
Chapter 2. Installing and configuring the ServerGuide Scripting Toolkit.	3
Installing the ServerGuide Scripting Toolkit	3
Creating the source tree	5
Configuring the installation	8
Chapter 3. Using the ServerGuide Scripting Toolkit to deploy Windows	15
Preparing the answer file for an unattended installation	15
Scenarios for deploying Windows	15
Chapter 4. Using the ServerGuide Scripting Toolkit to deploy Linux.	23
Preparing the kickstart file	23
Deploying Linux using a DOS-startable diskette and network share	24
Solving Linux installation problems	25
Chapter 5. Using the ServerGuide Scripting Toolkit with Microsoft ADS	27
Erasing data and configuring RAID controllers	27
Adding device drivers to the Microsoft ADS device-driver set	35
Chapter 6. Using the ServerGuide Scripting Toolkit for server disposal	37
Creating the server disposal diskette	37
Using the server disposal diskette	38
Chapter 7. ServerGuide Scripting Toolkit utilities	41
altboot.exe	41
cfg1030.exe	42
clini.exe and clini32.exe	44
dscan.exe and dscan32.exe	47
fdisk32.com	54
findram.exe	55
format32.com	56
hwdetect.exe	56
hypercfg.exe	60
ipssend.exe	62
Management Processor Command-Line Interface (MPCLI).	65
ramdisk2.exe	66
reboot.com	66
savestat.exe	66
scrub3.exe	67
unattend.exe.	68

Appendix A. Working with device drivers	75
Adding or updating a device driver.	75
Removing a device driver	76
Manually adding device-driver information to the answer file	76
Appendix B. Further automating the deployment process	79
Incorporating BIOS code and firmware updates in the deployment process	79
Incorporating MPCLI in the deployment process.	82
Dynamically updating the answer file for an unattended installation.	83
Appendix C. Getting help and technical assistance	85
Before you call	85
Using the documentation	85
Getting help and information from the World Wide Web	86
Software service and support	86
Hardware service and support	86
Appendix D. Notices	87
Edition notice	88
Trademarks	88
Index	89

Figures

1.	“Location to Save Files” window	4
2.	Source tree displayed in Windows Explorer	7
3.	Installation usrvars.bat file opened in Notepad	8
4.	Installation netvars.bat file opened in Notepad	13
5.	Server disposal usrvars.bat file opened in Notepad	28
6.	“ADS Sequence Editor” window	31
7.	“Save As” window	31
8.	“Template Type” window	32
9.	“Name and Description” window	33
10.	“Command Type” window.	33
11.	“Task Sequence File and Arguments” window	34
12.	“Destination Device Selection” window	34
13.	Server disposal usrvars.bat file opened in Notepad	38
14.	Windows 2000 device-driver directories displayed in Windows Explorer.	47
15.	“UpdateXpress” window	79
16.	“xSeries Firmware Update” window	80
17.	Updates directory of source tree displayed in Windows Explorer	81

Tables

1. Target server configuration and deployment scenarios	2
2. ServerGuide Scripting Toolkit contents	5
3. Example kickstart file statements	23
4. altboot parameters	41
5. altboot return values	42
6. Examples of using the altboot command	42
7. cfg1030 commands and parameters	43
8. Examples of using the cfg1030 command	44
9. clini optional parameters	45
10. clini return values	45
11. Examples of using the clini command	46
12. dscan optional parameters	48
13. dscan return values	49
14. Examples of using the dscan command	49
15. Keywords and values in drvinfo.ini files	51
16. Examples of drvinfo.ini files	53
17. fdisk32 optional parameters	54
18. Examples of using the fdisk32 command	55
19. format32 optional parameters	56
20. hwdetect: Hardware scan functions	56
21. hwdetect: PCI adapter detection functions	57
22. Examples of batch files using the hwdetect hardware scan functions	59
23. Examples of batch files using the PCI adapter detection functions	59
24. Examples of hardware information returned by the hwdetect command	60
25. hypercfg commands and parameters	61
26. Examples of using the hypercfg command	61
27. ipssend commands and parameters	62
28. Examples of using the IPSEND command	64
29. MPCLI commands	65
30. savestat parameters	66
31. Examples of using the savestat command	67
32. scrub3 optional parameters	67
33. Examples of using the scrub3 command	68
34. Optional unattend parameters	69
35. Return values for the unattend command	70
36. Examples of using the unattend command	70
37. Variables used in the [Unattend] section of the unattend initialization file	71
38. Variables used in the [GuiRunOnce] section of the unattend initialization file	72
39. Examples of unattend initialization files	73

About this book

This book provides information about installing and using the IBM® ServerGuide™ Scripting Toolkit. In addition to providing an overview of the ServerGuide Scripting Toolkit, it covers the following topics:

- Installing and setting up the ServerGuide Scripting Toolkit
- Using the ServerGuide Scripting Toolkit to install Microsoft® Windows®
- Using the ServerGuide Scripting Toolkit to install Red Hat Linux
- Integrating the ServerGuide Scripting Toolkit into existing deployment processes
- Using the ServerGuide Scripting Toolkit to securely remove data from a server
- Modifying the device-driver set used for Windows installations
- Further automating the deployment process

It also provides comprehensive reference material concerning the ServerGuide Scripting Toolkit utilities.

How this book is organized

Chapter 1, “Introducing ServerGuide Scripting Toolkit”, on page 1 contains an overview of the ServerGuide Scripting Toolkit.

Chapter 2, “Installing and configuring the ServerGuide Scripting Toolkit”, on page 3 contains information about installing the ServerGuide Scripting Toolkit, setting up a source tree, and configuring the deployment process.

Chapter 3, “Using the ServerGuide Scripting Toolkit to deploy Windows”, on page 15 contains information about using the ServerGuide Scripting Toolkit to perform unattended installations of Microsoft Windows.

Chapter 4, “Using the ServerGuide Scripting Toolkit to deploy Linux”, on page 23 contains information about using the ServerGuide Scripting Toolkit to perform unattended installations of Red Hat Linux.

Chapter 5, “Using the ServerGuide Scripting Toolkit with Microsoft ADS”, on page 27 contains information about incorporating ServerGuide Scripting Toolkit procedures into existing deployment processes, including those that use Microsoft Automated Deployment Services (ADS).

Chapter 6, “Using the ServerGuide Scripting Toolkit for server disposal”, on page 37 contains information about using the ServerGuide Scripting Toolkit to securely erase data from servers before disposal.

Chapter 7, “ServerGuide Scripting Toolkit utilities”, on page 41 contains information about the ServerGuide Scripting Toolkit utilities.

Appendix A, “Working with device drivers”, on page 75 contains information about adding device drivers to or removing device drivers from the device-driver set used for the Windows deployment scenarios. It also contains information about updating the device drivers in the device-driver set.

Appendix B, “Further automating the deployment process”, on page 79 contains information about further automating the deployment process. It includes information about incorporating BIOS code and firmware updates or the Management

Processor Command-Line Interface (MPCLI) in the deployment scenarios. It also includes information about dynamically updating the answer file for an unattended installation of Windows.

Appendix C, “Getting help and technical assistance”, on page 85 contains information about accessing relevant IBM Support Web sites for help and technical assistance.

Appendix D, “Notices”, on page 87 contains product notices and trademarks.

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 IBM deployment and systems-management software.

IBM Remote Deployment Manager page

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

This Web page provides an overview of IBM Remote Deployment Manager.

IBM ServerGuide

<http://www.ibm.com/pc/ww/eserver/xseries/serverguide/index.html>

This Web page provides an overview of ServerGuide.

IBM ServerGuide Scripting Toolkit page

<http://www.ibm.com/pc/support/site.wss/document.do?Indocid=MIGR-53564>

This Web page provides an overview of the ServerGuide Scripting Toolkit.

IBM ServerProven® page

<http://www.ibm.com/pc/us/compat/index.html>

This Web page provides compatibility information about hardware, software, and middleware.

IBM Support page

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

This is the IBM Support Web site for IBM hardware and systems-management software. For systems-management software support, click **Systems management**.

IBM Systems Management Software: Download/Electronic Support page

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

Use this Web page to download IBM systems-management software, including the ServerGuide Scripting Toolkit.

IBM UpdateXpress™

http://www.ibm.com/servers/eserver/xseries/systems_management/sys_migration/xpress.html

This Web page provides an overview of UpdateXpress. It also contains links to pages from which you can download UpdateXpress, order an UpdateXpress CD, or purchase a subscription from a third-party vendor.

IBM xSeries® Systems Management page

http://www.ibm.com/pc/ww/eserver/xseries/systems_management/index.html

This Web page provides an overview of IBM systems-management software.

Chapter 1. Introducing ServerGuide Scripting Toolkit

This chapter contains an overview of the ServerGuide Scripting Toolkit and its deployment scenarios, as well as information about the IBM @server™ or xSeries servers that are supported.

Overview of the ServerGuide Scripting Toolkit

The ServerGuide Scripting Toolkit is a collection of system-configuration tools and installation scripts that you can use to deploy operating systems to your IBM @server or xSeries server in a repeatable, predictable manner. When used with IBM ServerGuide and IBM UpdateXpress, the ServerGuide Scripting Toolkit provides a total solution for deploying IBM @server or xSeries servers in an unattended mode.

You can use the ServerGuide Scripting Toolkit to perform the following tasks automatically:

- Detect hardware
- Configure RAID adapters
- Delete any existing partitions and data from the drives of the target server
- Create a primary operating-system installation partition on the first drive of the target server
- Format the new partition as FAT32
- Install an operating system
- (Windows only) Install device drivers
- Dispose of servers securely

In addition, the ServerGuide Scripting Toolkit saves persistent-state information across system restarts so that it can monitor the deployment process.

Deployment scenarios

The ServerGuide Scripting Toolkit uses the following deployment scenarios:

- DOS-startable diskette and data CD
- DOS-startable CD
- DOS-startable diskette and network share
- Remote Supervisor Adapter II and network share

The system from which you deploy the operating system is the *source system*. The server on which the operating system is installed is the *target server*.

Each of the deployment scenarios requires the following items:

- ServerGuide Scripting Toolkit.
- Licensed copies of the operating system to be installed.
- An UpdateXpress CD.
- A correctly set up source system with a diskette drive.

Note: The Remote Supervisor Adapter II scenario also requires that the source system has a Web browser installed.

Depending on the features of the target server and your network environment, you might prefer to use one deployment scenario rather than another. The following table contains information about the deployment scenarios and the target server configuration that they are designed to support.

Table 1. Target server configuration and deployment scenarios

Target server configuration	Deployment scenario
A diskette and CD drive	DOS-startable diskette and data CD
A CD drive	DOS-startable CD
A diskette drive and access to the network share	DOS-startable diskette and network share
A Remote Supervisor Adapter II and access to the network share	Remote Supervisor Adapter II and network share

You can use any of the scenarios to deploy Windows to a target server. To deploy Linux to a target server, you must use the DOS-startable diskette and network share scenario.

Supported @server and xSeries servers

The ServerGuide Scripting Toolkit readme.txt file contains the following information:

- Supported @server and xSeries servers
- Deployment scenarios that are supported on specific @server and xSeries servers
- Operating systems that can be deployed to specific @server and xSeries servers

The latest version of this file can be downloaded from the ServerGuide Scripting Toolkit Web page. See “IBM deployment resources on the World Wide Web” on page x for more information.

Chapter 2. Installing and configuring the ServerGuide Scripting Toolkit

This chapter contains information about installing the ServerGuide Scripting Toolkit, setting up a source tree, and customizing the installation and batch files.

Installing the ServerGuide Scripting Toolkit

The ServerGuide Scripting Toolkit can be installed on the following operating systems:

- Windows 2000 (Server and Advanced Server)
- Windows Server 2003 (Standard Edition and Enterprise Edition)
- Red Hat Linux, Versions 7.3 and 8.0
- Red Hat Enterprise Linux AS, Version 2.1

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

Installing the ServerGuide Scripting Toolkit on Windows

Complete the following steps to install the ServerGuide Scripting Toolkit on a system running Windows:

1. Download the latest version of the ServerGuide Scripting Toolkit from the IBM Systems Management Software: Download/Electronic Support Web page. See “IBM deployment resources on the World Wide Web” on page x for more information.
2. Double-click the EXE file. The “License Agreement” window opens.
3. Click **I accept the terms in the license agreement**, and then click **Next**. The “Location to Save Files” window opens.

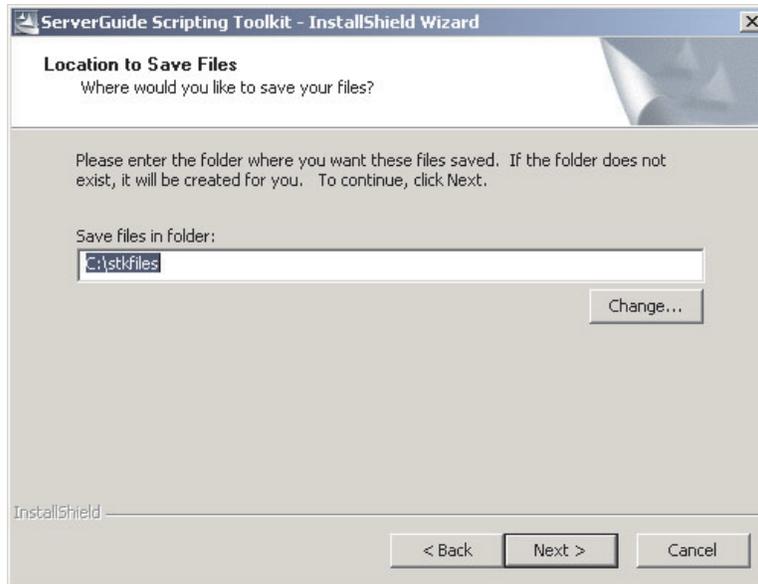


Figure 1. “Location to Save Files” window

4. In the **Save files in folder** field, type the fully qualified directory name where you want to save the ServerGuide Scripting Toolkit files. Click **Next**. The files are extracted.
5. When the extraction process is completed and the “InstallShield Wizard Complete” window opens, click **Finish**.

Installing the ServerGuide Scripting Toolkit on Linux

Note: The Linux network share must be accessible from a system running Windows. Run Samba on the Linux source server and share the exported volume to the system running Windows.

Complete the following steps to install the ServerGuide Scripting Toolkit on a system running Linux:

1. Download the latest version of the ServerGuide Scripting Toolkit to the network share. The ServerGuide Scripting Toolkit can be downloaded from the IBM Systems Management Software: Download/Electronic Support Web page. See “IBM deployment resources on the World Wide Web” on page x for more information.
2. From the system running Windows, connect to the Linux network share.
3. Complete steps 2 through 5 of “Installing the ServerGuide Scripting Toolkit on Windows” on page 3.

ServerGuide Scripting Toolkit directory structure

The root directory of the ServerGuide Scripting Toolkit is named `stkfiles`. The following table provides an overview of the directory contents.

Table 2. ServerGuide Scripting Toolkit contents

Directory	Description
<code>\stkfiles</code>	Root directory
<code>\stkfiles\sg_stk\boot</code>	Files for creating and modifying the DOS-startable (bootable) diskette images You can use the batch files in this directory to perform the following tasks: <ul style="list-style-type: none">• Create the diskettes used in the installation scenarios• Customize and modify the diskettes• Save diskette images
<code>\stkfiles\sg_stk\docs</code>	ServerGuide Scripting Toolkit documentation
<code>\stkfiles\sg_stk\DOS</code>	The IBM PC-DOS files used by the ServerGuide Scripting Toolkit, plus additional PC-DOS files and programs
<code>\stkfiles\sg_stk\examples</code>	The script files that run the deployment scenarios
<code>\stkfiles\sg_stk\examples\linux</code>	Linux-specific installation script files
<code>\stkfiles\sg_stk\examples\linux\redhat</code>	Sample kickstart files
<code>\stkfiles\sg_stk\examples\windows</code>	Windows-specific installation script files
<code>\stkfiles\sg_stk\examples\windows\win2000</code>	Sample answer files for an unattended installation of Windows 2000
<code>\stkfiles\sg_stk\examples\windows\win2003</code>	Sample answer files for an unattended installation of Windows Server 2003
<code>\stkfiles\sg_stk\utils</code>	The non-PC-DOS ServerGuide Scripting Toolkit utilities

Creating the source tree

This section contains information about setting up the source tree. The *source tree* is a set of directories that contain the files that the ServerGuide Scripting Toolkit uses to deploy operating systems to IBM `@server` and xSeries servers. The source tree includes the following files:

- ServerGuide Scripting Toolkit utilities
- (Windows deployment scenarios only) Device drivers
- Operating-system files

The source tree can be located on a shared network directory.

Notes:

1. If the source tree is not on a single hard disk, you must customize the ServerGuide Scripting Toolkit scripts so that they contain the correct path information.
2. Limit the length of directory names and paths when creating the source tree; DOS has a path limit of 127 characters and recognizes file names in 8.3 notation.

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

Complete the following steps to set up the source tree:

1. If necessary, copy the ServerGuide Scripting Toolkit directory to the source-tree system.
2. Add the operating-system installation files to the source tree:
 - a. Create a directory for each operating system you want to deploy.

For Windows 2000 Server	\w2k_srv
For Windows 2000 Advanced Server	\w2k_adv
For Windows Server 2003, Standard Edition	\w23_std
For Windows Server 2003, Enterprise Edition	\w23_ent
For Windows Server 2003, Web Edition	\w23_web
For Red Hat Linux 7.3	\rh73
For Red Hat Linux 8.0	\rh80
For Red Hat Enterprise Linux AS 2.1	\rhas21

You can use a directory-naming convention of your choice. However, the procedures in this document assume that you use the previous directory names.

- b. Using the operating-system installation media, copy the operating-system files into the applicable directories.

Notes:

- 1) Be sure to maintain the directory structure for both Windows and Linux installation files. For example, Windows installation files must be in the default i386 directory structure.
 - 2) For Red Hat Linux, you can copy the contents of the first three Red Hat Linux installation CDs to the source tree.
3. (For Windows deployment scenarios only) Add the device-driver files to the source tree:

Note: You must use an IBM *ServerGuide Setup and Installation* CD, Version 7.2 or later that supports the IBM @server or xSeries server that you want to deploy. The ServerGuide readme.txt file (located in the \Readme directory) lists the servers that are supported. You can download the most recent CD from the IBM ServerGuide Web site. See “IBM deployment resources on the World Wide Web” on page x for more information.

- a. Insert the *ServerGuide Setup and Installation* CD into the CD drive of a system running Windows.
- b. Wait for the IBM ServerGuide Driver Repository to start; then, close the Driver Repository.

- c. Copy the following device-driver directories, including subdirectories, from the *ServerGuide Setup and Installation* CD to the root of the source tree.

For Windows 2000	\w2000drv
For Windows Server 2003	\w2003drv

(The device-driver directories are located in the *sguide* directory on the *ServerGuide Setup and Installation* CD.)

The following figure shows a source tree that might result from the previous steps.

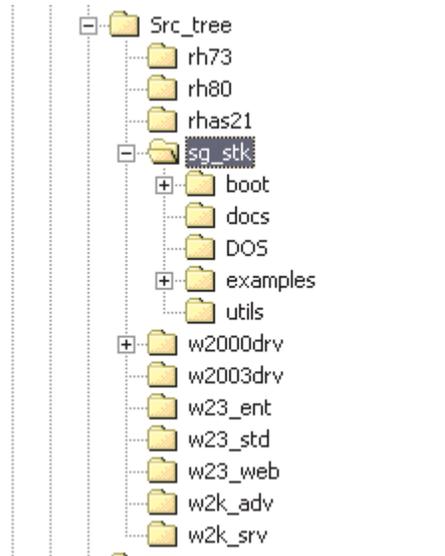


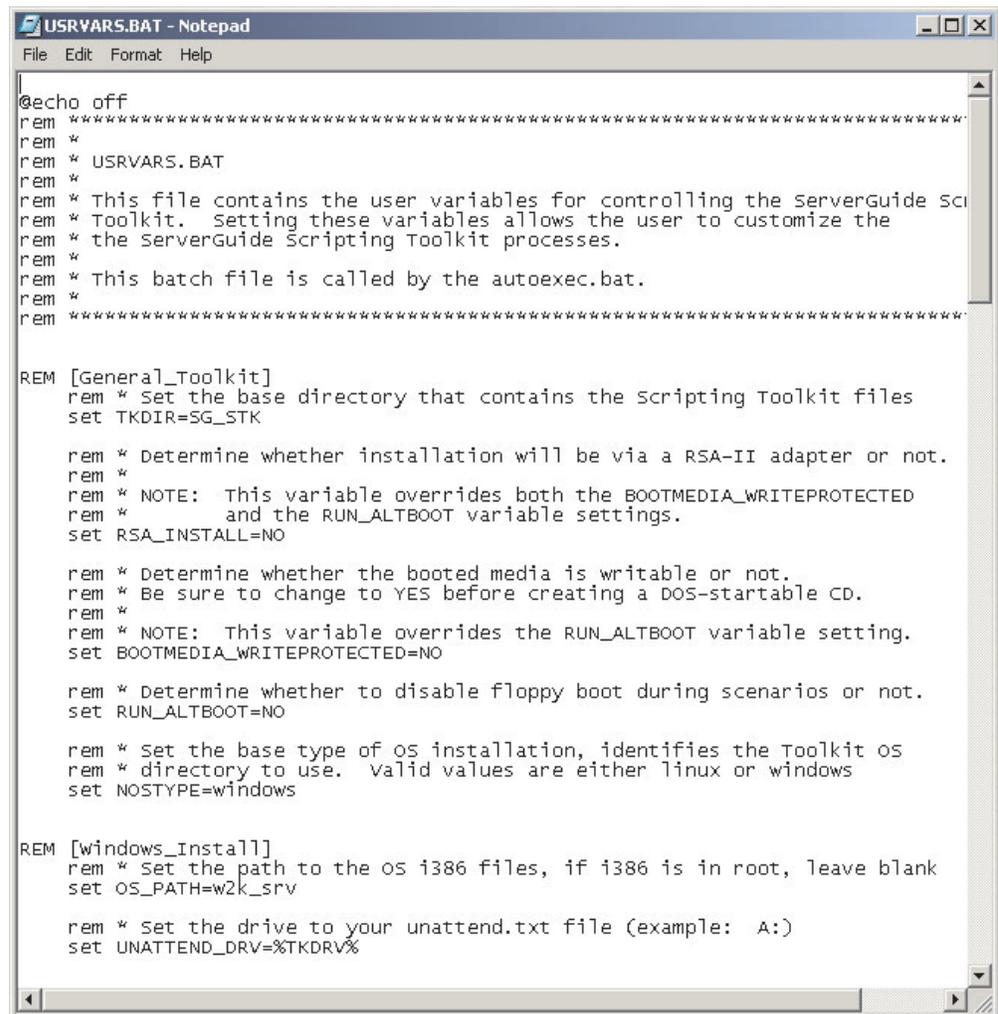
Figure 2. Source tree displayed in Windows Explorer

Configuring the installation

This section contains information about customizing the `usrvars.bat` and `netvars.bat` files.

Customizing the `usrvars.bat` file

During the deployment scenario, the `usrvars.bat` file (located on the DOS-startable media) opens in Notepad.



```
USRVAR.S.BAT - Notepad
File Edit Format Help
@echo off
rem *****
rem *
rem * USRVARS.BAT
rem *
rem * This file contains the user variables for controlling the ServerGuide Sc
rem * Toolkit. Setting these variables allows the user to customize the
rem * the ServerGuide Scripting Toolkit processes.
rem *
rem * This batch file is called by the autoexec.bat.
rem *
rem *****

REM [General_Toolkit]
rem * Set the base directory that contains the scripting Toolkit files
set TKDIR=SG_STK

rem * Determine whether installation will be via a RSA-II adapter or not.
rem *
rem * NOTE: This variable overrides both the BOOTMEDIA_WRITEPROTECTED
rem * and the RUN_ALTBOOT variable settings.
set RSA_INSTALL=NO

rem * Determine whether the booted media is writable or not.
rem * Be sure to change to YES before creating a DOS-startable CD.
rem *
rem * NOTE: This variable overrides the RUN_ALTBOOT variable setting.
set BOOTMEDIA_WRITEPROTECTED=NO

rem * Determine whether to disable floppy boot during scenarios or not.
set RUN_ALTBOOT=NO

rem * Set the base type of OS installation, identifies the toolkit OS
rem * directory to use. Valid values are either linux or windows
set NOSTYPE=windows

REM [windows_Install]
rem * Set the path to the OS i386 files, if i386 is in root, leave blank
set OS_PATH=w2k_srv

rem * Set the drive to your unattend.txt file (example: A:)
set UNATTEND_DRV=%TKDRV%
```

Figure 3. Installation `usrvars.bat` file opened in Notepad

The `usrvars.bat` file contains the following major sections:

- [General_Toolkit]
- [OS_Install], where *OS* is either Windows or Linux
- [Toolkit_Updates]
- [RAID_Configuration]

When setting values of applicable variables to either **yes** or **no**, use upper-case or lower-case notation. Do not use mixed-case notation, for example, **Yes**.

When setting a value for any of the PATH variables such as OS_PATH or UNATTEND_PATH, do not include drive letters. Drive letters are set in variables that include DRV as part of their name (for example, NOSPATh and UNATTEND_PATH). Drive letters set in DRV variables are appended to the values you set using PATH variables. If you set a PATH variable to a value and include a drive letter, the scenario will fail.

Notes:

1. For information about configuring the usrvars.bat file for the server disposal scenarios, see “Creating the Microsoft ADS server disposal diskette” on page 27 and “Creating the server disposal diskette” on page 37.
2. For information about configuring the variables in the [Toolkit_Updates] section, see “Incorporating BIOS code and firmware updates in the deployment process” on page 79.

Configuring the general variables

The general variables are located in the [General_Toolkit] section of the usrvars.bat file. Complete the following steps to configure the general variables:

1. Set TKDIR equal to the root directory of the ServerGuide Scripting Toolkit. By default, this variable is set to SG_STK.
2. Set RSA_INSTALL equal to YES only if you are deploying Windows to a target server using a Remote Supervisor II adapter. By default, this variable is set to NO. This variable overrides the settings of both the BOOTMEDIA_WRITEPROTECTED and RUN_ALTBOOT variables.
3. Set BOOTMEDIA_WRITEPROTECTED equal to YES in the following situations:
 - You are creating a startable CD.
 - You want to write-protect the DOS-startable (bootable) diskette.

By default, this variable is set to NO. This variable overrides the value of the RUN_ALTBOOT variable.

Note: If you are using write-protected DOS-startable (bootable) media and do not set BOOTMEDIA_WRITEPROTECTED to YES, a “write protected error” is displayed when the write-protected DOS-startable (bootable) media is used.

4. Set RUN_ALTBOOT equal to YES to ensure that you will not be prompted to remove the DOS-startable diskette from the diskette drive. By default, this variable is set to NO.
5. Set NOSTYPE equal to either **windows** or **linux**. By default, this variable is set to the applicable operating-system type.

Configuring the Windows installation variables

The Windows installation variables are located in the [Windows_Install] section of the usrvars.bat file. By default, these variables are set for deploying Windows 2000 using a source tree configured as described in “Creating the source tree” on page 5.

Complete the following steps to configure the Windows installation variables:

1. Set OS_PATH equal to the directory that contains the i386 Windows installation files. By default, this variable is set to w2k_srv.
2. Set UNATTEND_DRV equal to the drive letter of the hard disk drive where the answer file is stored. By default, this variable is set to %TKDRV%, where %TKDRV% is the drive letter of the media where the source tree for the deployment is located.

3. Set UNATTEND_PATH equal to the path of the directory where the unattend.txt file is stored. By default, this variable is set to %TKDIR%\examples\%NOSTYPE%\win2000, where %TKDIR% is the root directory of the ServerGuide Scripting Toolkit and %NOSTYPE% is set to windows.
4. Set UNATTEND_FILENAME equal to the name of the answer file for the unattended installation. By default, this variable is set to unattend.txt.
5. Set PARTITION_SIZE equal to the size of the partition (in MB) that you want to create on the target server. By default, this variable is set to 4000.
6. Set NOSDRV equal to the drive letter of the operating-system partition on the target server. By default, this variable is set to C:.
7. Set SGDD_TARGET_PATH equal to the path of the directory on the operating-system partition where device-driver files are temporarily stored before installation. By default, this variable is set to wininst.
8. Set SGDD_SRC_DRV_PATH equal to the path to the device-driver directory on the source tree. By default, this variable is set to one of the following values:

For DOS-startable diskette with data CD scenario	drv
For DOS-startable CD scenario	drv
For scenarios that use a network share	w2000drv\%oem%\\$1\drv

9. Set SGDD_SRC_TEXTMODE_PATH equal to the path to the textmode device-driver directory on the source tree. By default, this variable is set to one of the following values:

For DOS-startable diskette with data CD scenario	textmode
For DOS-startable CD scenario	textmode
For scenarios that use a network share	w2000drv\%oem%\\$1\textmode

Configuring the Linux installation variables

The Linux installation variables are located in the [Linux_Install] section of the usrvars.bat file. Complete the following steps to configure the Linux installation variables:

1. Set OS_PATH equal to the directory that contains the Linux installation files. By default, this variable is set to rhas21.
2. Set KERNEL_FILE equal to the path and filename of the Linux network installation kernel. By default, this variable is set to %OS_PATH%\images\pxeboot\mlinuz, where %OS_PATH% is the directory that contains the Linux installation files.
3. Set INITRD_FILE equal to the path and filename of the RAM disk image. By default, this variable is set to %OS_PATH%\images\pxeboot\initrd~1.img, where %OS_PATH% is the directory that contains the Linux installation files.

Note: The RAM disk image is named either initrd.img or initrd-everything.img. Under the DOS 8.3 file-naming convention, initrd-everything.img becomes initrd~1.img, depending on the contents of the directory in which the RAM disk image is located.

4. Set KS_FILENAME equal to the name of the kickstart file. By default, this variable is set to rhas21ks.cfg.

5. Set `KS_LOCATION` equal to the location of the kickstart file. By default, this variable is set to `http://%SERVER_IP%/TKDIR%/examples/linux/redhat/%KS_FILENAME%`, where:
 - `%SERVER_IP%` is the IP address of the source system
 - `%TKDIR%` is the root directory of the ServerGuide Scripting Toolkit
 - `%KS_FILENAME%` is the name of the kickstart file

Notes:

- a. If the kickstart file is located anywhere other than on local media, DHCP must be enabled on the source system. See the kickstart file documentation for further information.
 - b. You set the `SERVER_IP` variable when you customize the `netvars.bat` file. See “Customizing the `netvars.bat` file” on page 13.
6. Set `LOADLIN_PATH` equal to the path of the `loadlin.exe` file. By default, this variable is set to `%OS_PATH%\dosutils`, where `%OS_PATH%` is set equal to `%TKDRV%\linux\rhas21` and where `%TKDRV%` is the drive letter of the media where the source tree for the deployment is located.
 7. Set `KS_DEVICE` equal to the network interface card (NIC) of the target server. By default, this variable is set to `eth1`.

Note: Be sure to configure the variable if the target server has more than one NIC.

8. Set `RAMDISK_SIZE` equal to the size (in KB) of the RAM disk image. By default, this variable is set to 8196.

Configuring the RAID variables

The basic RAID configuration variables are located in the [RAID_Configuration] section. Complete the following steps to configure the RAID variables:

1. Set `DO_RAID_CFGS` equal to YES. By default, this variable is set to NO. By default, `DO_CLEAN_FIRST` is set to YES. This ensures that the RAID controller is set to the factory-default settings before it is configured.
2. Go to one of the following procedures:

If the server contains an	Go to
IBM ServeRAID™ controller	“Configuring a ServeRAID controller”
Integrated SCSI controller with RAID capabilities	“Configuring an integrated SCSI controller with RAID capabilities” on page 12
IDE RAID controller	“Configuring an IDE RAID controller” on page 12

Configuring a ServeRAID controller: The ServeRAID configuration variables are located in the [RAID_Configuration_IPSSSEND] section. By default, it contains variables that you can modify to configure as many as three ServeRAID controllers. Using up to eight physical drives, you can create a single array and logical drive for each ServeRAID controller.

Complete the following steps to configure the ServeRAID variables:

1. Modify the [RAID_Configuration_IPSSEND] section so that it contains the following commands:

```
SET CFG_SVRDn=YES
SET SV_ARRAYSIZEn=Size
SET SV_RAIDLEVELn=RAIDlevel
SET SV_1STCHANNELn=Number
SET SV_1STSIDn=SCSIid
SET SV_2NDCHANNELn=Number
SET SV_2NSIDn=SCSIid
...
SET SV_8THCHANNELn=Number
SET SV_8THSIDn=SCSIid
```

where

- *n* is the number of the ServeRAID controller (1-3).
 - *Size* is one of the following values:
 - MAX, which specifies that you want to use all available space
 - The size (in MB) of the logical drive
 - *RAIDlevel* is the RAID level of the logical drive (0, 1, 1E, 5, or 5E).
 - *Number* is the channel number of the physical drive.
 - *SCSIid* is the SCSI ID of the physical drive.
2. Repeat step 1 until you have specified the RAID configuration values for all ServeRAID controllers in the target server.

Configuring an integrated SCSI controller with RAID capabilities: The configuration variables for integrated SCSI controllers with RAID capabilities are located in the [RAID_Configuration_CFG1030] section.

To configure an integrated SCSI controller with RAID capabilities, modify the [RAID_Configuration_CFG1030] section to read as follows:

```
SET CONFIG_SCSI_n=YES
SET SCSI_1STSIDn=SCSIid
SET SCSI_2NSIDn=SCSIid
SET DO_SCSI_HOTSPAREN=Option
SET DO_SCSI_HOTSPARE_SIDn=SCSIid
```

where

- *n* is the number of the integrated SCSI controller with RAID capabilities (1-2).
- *SCSIid* is the SCSI ID of the physical drive.
- *Option* is either YES or NO.

Configuring an IDE RAID controller: The configuration variables for IDE RAID variables are located in the [RAID_Configuration_HYPERCFG] section.

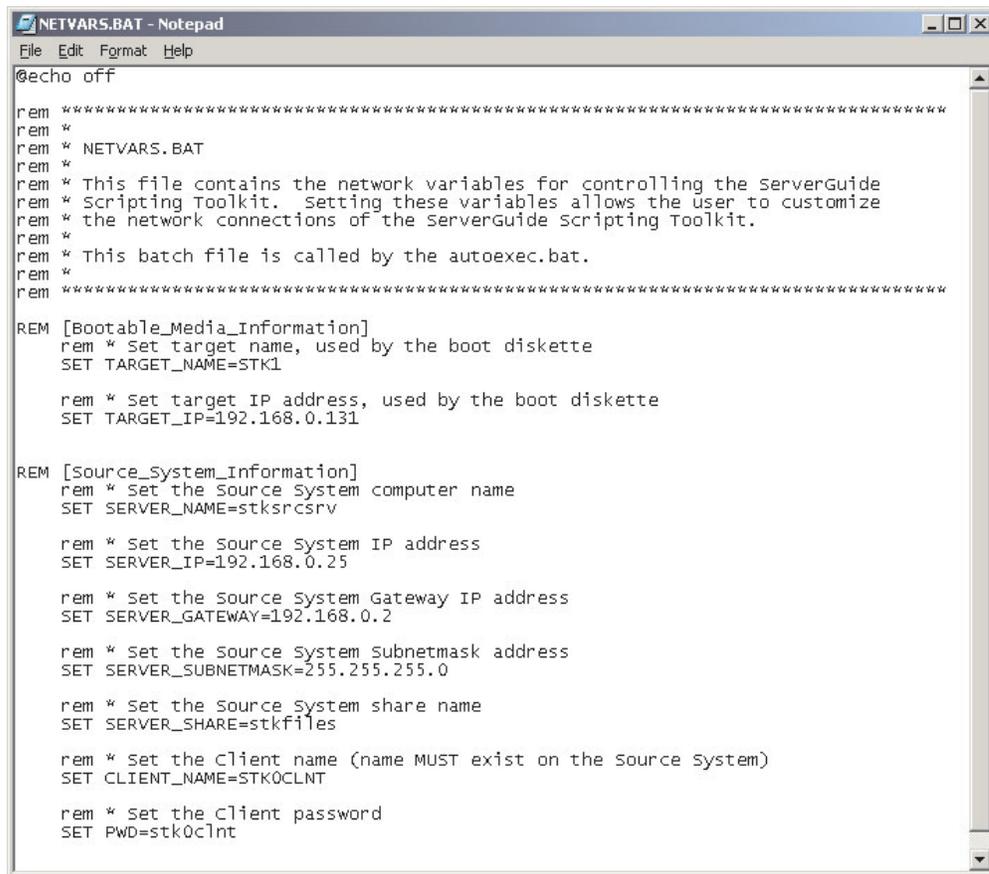
Complete the following steps to configure the IDE RAID variables:

1. Set IDE_CONFIGURE equal to YES. By default, IDE_CONFIGURE is set to NO.
2. Set IDE_STRIPESIZE1 equal to one of the following values:
 - M, which specifies a RAID level-1 logical drive.
 - S *StripeSize*, which specifies a RAID level-0 logical drive. Data is striped across the drive in *StripeSize* (KB) blocks.

By default, IDE_STRIPESIZE1 is set to M.

Customizing the netvars.bat file

During the deployment scenarios, the netvars.bat file (located on the DOS-startable media) might open in Notepad.



```
@echo off

rem *****
rem *
rem * NETVARS.BAT
rem *
rem * This file contains the network variables for controlling the ServerGuide
rem * Scripting Toolkit. Setting these variables allows the user to customize
rem * the network connections of the ServerGuide Scripting Toolkit.
rem *
rem * This batch file is called by the autoexec.bat.
rem *
rem *****

REM [Bootable_Media_Information]
rem * Set target name, used by the boot diskette
SET TARGET_NAME=STK1

rem * Set target IP address, used by the boot diskette
SET TARGET_IP=192.168.0.131

REM [Source_System_Information]
rem * Set the Source System computer name
SET SERVER_NAME=stksrcsrv

rem * Set the Source System IP address
SET SERVER_IP=192.168.0.25

rem * Set the Source System Gateway IP address
SET SERVER_GATEWAY=192.168.0.2

rem * Set the Source System Subnetmask address
SET SERVER_SUBNETMASK=255.255.255.0

rem * Set the Source System share name
SET SERVER_SHARE=stkfiles

rem * Set the Client name (name MUST exist on the Source System)
SET CLIENT_NAME=STK0CLNT

rem * Set the Client password
SET PWD=stk0clnt
```

Figure 4. Installation netvars.bat file opened in Notepad

Complete the following steps to customize the netvars.bat file:

1. Modify the [Bootable_Media_Information] section:
 - a. Set TARGET_NAME equal to the name of the DOS-startable (bootable) diskette. By default, this variable is set to STK1.
 - b. Set TARGET_IP equal to a valid IP address that can be used during the installation process. By default, this variable is set to 192.168.0.131.

Notes:

- 1) The TARGET_NAME and TARGET_IP specified in netvars.bat are used only when the target server starts (boots) from the DOS-startable diskette. The installed operating system uses the host name and IP address specified in the answer or kickstart file for the unattended installation.

- 2) You can use the ServerGuide Scripting Toolkit to simultaneously perform multiple, network-based deployment operations. However, each diskette that is used to connect simultaneously to the same source system *must* have a unique TARGET_NAME and TARGET_IP. You can run the CustomizeNetworkVariables.bat file, which opens the netvars.bat file in Notepad. You then can edit the netvars.bat file. When you close the netvars.bat file, the CustomizeNetworkVariables.bat file automatically updates the variables in the protocol.ini and network.ini files.
2. Modify the [Source_System_Information] section:
 - a. Set SERVER_NAME equal to the NetBIOS name of the source system. By default, this variable is set to stksrsv.
 - b. Set SERVER_IP equal to the IP address of the source system. By default, this variable is set to 192.168.0.25.
 - c. Set SERVER_GATEWAY to the gateway address of the source system. By default, this variable is set to 192.168.0.2
 - d. Set SERVER_SUBNETMASK equal to the subnet mask of the source system. By default, this variable is set to 255.255.255.0.
 - e. Set SERVER_SHARE equal to the shared network directory. By default, this variable is set to stkfiles.
 - f. Set CLIENT_NAME equal to a user name with access to the shared network directory. By default, this variable is set to STK0CLNT.
 - g. Set PWD equal to the password that corresponds to the user name configured in step 2f. By default, this variable is set to stk0clnt.

Note: The user name and password are not encrypted. As a precaution, consider using a user account on the source system that has only the authority to read and run files on the source tree.

Chapter 3. Using the ServerGuide Scripting Toolkit to deploy Windows

This chapter contains information about using the ServerGuide Scripting Toolkit utilities to perform an unattended installation of Microsoft Windows on IBM *@server* or xSeries servers.

You can use the ServerGuide Scripting Toolkit to deploy the following operating systems:

- Windows 2000 (Server and Advanced Server)
- Windows Server 2003 (Standard, Enterprise, and Web Editions)

Preparing the answer file for an unattended installation

You must customize the answer file for an unattended installation. Two sample answer files (each named `unattend.txt`) are located in the `\stkfiles\sg_stk\examples\windows\OS` directory, where *OS* is either `win2000` or `win2003`. In addition, Microsoft includes a sample answer file in the `i386` directory.

You must add information to the `[UserData]` section of the answer file, including a value for the `ProductID` keyword, if it is required by the operating system you are deploying. This information either can be added manually to the answer file before the deployment scenario is started or dynamically added during the deployment process. You do not need to add device-driver information to the answer file; the ServerGuide Scripting Toolkit process dynamically adds device-driver information to the answer file at run-time.

For more information about customizing the answer file, see the Microsoft documentation on the Microsoft Windows 2000 or Windows Server 2003 installation CDs. The documentation is located in the `\support\tools\deploy.cab` file. For information about dynamically customizing an answer file, see “Dynamically updating the answer file for an unattended installation” on page 83.

Scenarios for deploying Windows

This section provides information about the following ServerGuide Scripting Toolkit scenarios that you can use to deploy Windows to target servers:

- DOS-startable (bootable) diskette and data CD
- DOS-startable (bootable) CD
- DOS-startable (bootable) diskette and a network share
- Remote Supervisor Adapter II and a network share

Each deployment scenario requires different combinations of installation media and network access. The Remote Supervisor Adapter II scenario can be used only to deploy Windows to a target server that contains a Remote Supervisor Adapter II.

Each deployment scenario assumes the ServerGuide Scripting Toolkit is installed on the source system in the default directory. If the ServerGuide Scripting Toolkit is not installed in this location, change the directory names and paths as necessary.

Using a DOS-startable diskette and data CD

You need the following items for this scenario:

- A blank diskette and a blank CD
- A source system that contains a properly set up source tree
- CD creation software and a system with a CD-RW drive
- UpdateXpress CD

Complete the following steps to deploy Windows to an IBM @server or xSeries server:

1. Create the DOS-startable (bootable) diskette:
 - a. Insert a blank diskette into the diskette drive of the source system.
 - b. To change to the ServerGuide Scripting Toolkit directory, from a command prompt, type the following command and press Enter:
`cd \stkfiles\sg_stk\boot`
 - c. To expand the DOS-startable (bootable) diskette image onto the blank diskette, type the following command and press Enter:
`MakeWindowsCDInstallToolkitDisk.bat`
 - d. To open the `usrvars.bat` file on the diskette in Notepad, type the following command and press Enter:
`CustomizeUserVariables.bat`

Edit the file to customize the installation variables. For more information, see “Customizing the `usrvars.bat` file” on page 8.
 - e. Save the answer file for the unattended installation to the source system. For more information, see “Preparing the answer file for an unattended installation” on page 15.
 - f. Label the DOS-startable (bootable) diskette.
2. Create the data CD:
 - a. Using CD creation software, create a data CD that contains at least the following directories:

Directory	Location in the source tree	What it contains
<code>sg_stk</code>	<code>d:\sg_stk</code>	ServerGuide Scripting Toolkit
<code>i386</code>	<code>d:\OSi386</code>	Windows installation files
<code>drv</code>	<code>d:\OSdevdrv\%oem%\\$1\drv</code>	Device drivers
<code>textmode</code>	<code>d:\OSdevdrv\%oem%\textmode</code>	Text mode device drivers

where

- *d* is the drive letter of the hard disk.
- *OS* is one of the following values:
 - `w23_ent`
 - `w23_std`
 - `w23_web`
 - `w2k_adv`
 - `w2k_srv`
- *OSdevdrv* is either `w2000drv` or `w2003drv`.

To work under DOS, the directories on the data CD cannot be nested more than eight levels deep.

- b. Copy the answer file from the source system to the root directory of the data CD.
 - c. Label the data CD.
3. Update the BIOS code and firmware:
 - a. Start the target server and insert the IBM Update*Xpress* CD into the CD drive.
 - b. Restart the target server. IBM Update*Xpress* starts. Follow the instructions on the screen to update the BIOS code and firmware.
 - c. When the update is completed, remove the IBM Update*Xpress* CD from the CD drive.
4. Install the operating system:
 - a. Insert the DOS-startable (bootable) diskette into the diskette drive of the target server.
 - b. Insert the data CD into the CD drive.
 - c. Restart the target server. The server starts from the DOS-startable (bootable) diskette and performs the ServerGuide Scripting Toolkit installation.
 - d. Either when prompted or when the installation is completed, remove the DOS-startable (bootable) diskette and data CD from the target server.
5. (Optional) Upgrade the device drivers to the most current versions:
 - a. Insert the IBM Update*Xpress* CD into the CD drive of the target server. IBM Update*Xpress* starts automatically.
 - b. Follow the instructions on the screen to update the device drivers.
 - c. When the update is completed, remove the IBM Update*Xpress* CD from the CD drive.

Using a DOS-startable CD

You need the following items for this scenario:

- A blank diskette and a blank CD
- A source system that contains a properly set up source tree
- CD creation software and a system with a CD-RW drive
- Update*Xpress* CD

Complete the following steps to deploy Windows to an IBM @server or xSeries server:

1. Create the DOS-startable (bootable) diskette:
 - a. Insert a blank diskette into the diskette drive of the source system.
 - b. To change to the ServerGuide Scripting Toolkit directory, from a command prompt, type the following command and press Enter:


```
cd \stkfiles\sg_stk\boot
```
 - c. To expand the DOS-startable (bootable) diskette image onto the blank diskette, type the following command and press Enter:


```
MakeWindowsCDInstallToolkitDisk.bat
```

- d. To open the `usrvars.bat` file on the diskette in Notepad, type the following command and press Enter:

```
CustomizeUserVariables.bat
```

Edit the file to customize the installation variables. Be sure to set `BOOTMEDIA_WRITEPROTECTED` equal to YES. For more information, see “Customizing the `usrvars.bat` file” on page 8.

- e. Save the answer file for the unattended installation to the source system. For more information, see “Preparing the answer file for an unattended installation” on page 15.
 - f. Label the DOS-startable (bootable) diskette.
2. Create the data CD:
 - a. Using CD creation software, create a data CD that contains at least the following directories:

Directory	Location in the source tree	What it contains
<code>sg_stk</code>	<code>d:\sg_stk</code>	ServerGuide Scripting Toolkit
<code>i386</code>	<code>d:\OS\i386</code>	Windows installation files
<code>drv</code>	<code>d:\OSdevdrv\%oem%\\$1\drv</code>	Device drivers
<code>textmode</code>	<code>d:\OSdevdrv\%oem%\textmode</code>	Textmode device drivers

where

- *d* is the drive letter of the hard disk.
- *OS* is one of the following values:
 - `w23_ent`
 - `w23_std`
 - `w23_web`
 - `w2k_adv`
 - `w2k_srv`
- *OSdevdrv* is either `w2000drv` or `w2003drv`.

To work under DOS, the directories on the data CD cannot be nested more than eight levels deep.

- b. Copy the answer file from the source system to the root directory of the data CD.
 - c. Using the CD creation software, set the boot image to the DOS-startable (bootable) diskette.
 - d. Label the data CD.
3. Update the BIOS code and firmware:
 - a. Start the target server and insert the IBM Update*Xpress* CD into the CD-ROM drive.
 - b. Restart the target server. IBM Update*Xpress* starts. Follow the instructions on the screen to update the BIOS code and firmware.
 - c. When the update is completed, remove the IBM Update*Xpress* CD from the CD-ROM drive.
 4. Install the operating system:
 - a. Insert the DOS-startable (bootable) CD into the CD-ROM drive of the target server.
 - b. Restart the target server. The server starts from the DOS-startable (bootable) CD and performs the ServerGuide Scripting Toolkit installation.

- c. When prompted, remove the CD from the CD-ROM drive.
 - d. Press any key to complete the installation.
5. (Optional) Upgrade the device drivers to the most current versions:
 - a. Insert the IBM Update*Xpress* CD into the CD-ROM drive of the target server. IBM Update*Xpress* starts automatically.
 - b. Follow the instructions on the screen to update the device drivers.
 - c. When the update is completed, remove the IBM Update*Xpress* CD from the CD-ROM drive.

Using a DOS-startable diskette and network share

You need the following items for this scenario:

- A blank diskette
- A source system that contains a properly set up source tree
- Network connectivity and access to the source-system directory
- Update*Xpress* CD

Complete the following steps to deploy Windows to an IBM @server or xSeries server:

1. Configure the network share:
 - a. Make the source-tree directory accessible to the network.
 - b. To prevent the source tree from being accidentally deleted, configure the source-tree directory as read-only.
 - c. Create a user ID and password for the source-tree directory.

Note: The user name and password in the netvars.bat file are not encrypted. As a precaution, consider using a user account on the source system that has only the authority to read and run files on the source tree.

2. Create the DOS-startable (bootable) diskette:
 - a. Insert a blank diskette into the diskette drive of the source system.
 - b. To change to the ServerGuide Scripting Toolkit directory, from a command prompt, type the following command and press Enter:


```
cd \stkfiles\sg_stk\boot
```
 - c. To expand the DOS-startable (bootable) diskette image onto the blank diskette, type the following command and press Enter:


```
MakeWindowsNetworkInstallToolkitDisk.bat
```
 - d. To open the usrvars.bat file on the diskette in Notepad, type the following command and press Enter:


```
CustomizeUserVariables.bat
```

Edit the file to customize the installation variables. For more information, see “Customizing the usrvars.bat file” on page 8.

- e. To open the netvars.bat file on the diskette in Notepad, type the following command and press Enter:


```
CustomizeNetworkVariables.bat
```

Edit the file to customize the network variables. For more information, see “Customizing the netvars.bat file” on page 13.

- f. Save the answer file for the unattended installation to the source system. For more information, see “Preparing the answer file for an unattended installation” on page 15.
 - g. Label the DOS-startable (bootable) diskette.
3. Update the BIOS code and firmware:
 - a. Start the target server and insert the IBM Update*Xpress* CD into the CD-ROM drive.
 - b. Restart the target server. IBM Update*Xpress* starts. Follow the instructions on the screen to update the BIOS code and firmware.
 - c. When the update is completed, remove the IBM Update*Xpress* CD from the CD-ROM drive.
4. Install the operating system:
 - a. Insert the DOS-startable (bootable) diskette into the diskette drive of the target server.
 - b. Restart the target server. The server starts from the DOS-startable (bootable) diskette and performs the ServerGuide Scripting Toolkit installation.
 - c. Either when prompted or when the installation is completed, remove the DOS-startable (bootable) diskette from the target server.
5. (Optional) Upgrade the device drivers to the most current versions:
 - a. Insert the IBM Update*Xpress* CD into the CD-ROM drive of the target server. IBM Update*Xpress* starts automatically.
 - b. Follow the instructions on the screen to update the device drivers.
 - c. When the update is completed, remove the IBM Update*Xpress* CD from the CD-ROM drive.

Using a Remote Supervisor Adapter II and network share

You need the following items for this scenario:

- A blank diskette and a blank CD
- A source system that contains a properly set up source tree
- Network connectivity to the Remote Supervisor Adapter II and access to the source-tree directory from the source system
- Update*Xpress* CD

Note: Make sure that you have updated the Remote Supervisor Adapter II firmware level. You can download the updates from the IBM Support Web site. For more information, see “IBM deployment resources on the World Wide Web” on page x.

Complete the following steps to deploy Windows to an IBM *@server* or xSeries server using the Remote Supervisor Adapter II installed in the target server:

1. Configure the network share:
 - a. Make the source-tree directory accessible to the network.
 - b. To prevent the source tree from being accidentally deleted, configure the source-tree directory as read-only.
 - c. Create a user ID and password for the source-tree directory.

Note: The user name and password in the netvars.bat file are not encrypted. As a precaution, consider using a user account on the source system that has only the authority to read and run files on the source tree.

2. Create the DOS-startable (bootable) diskette:

- a. Insert a blank diskette into the diskette drive of the source system.
- b. To change to the ServerGuide Scripting Toolkit directory, from a command prompt, type the following command and press Enter:

```
cd \stkfiles\sg_stk\boot
```

- c. To expand the DOS-startable (bootable) diskette image onto the blank diskette, type the following command and press Enter:

```
MakeWindowsNetworkInstallToolkitDisk.bat
```

- d. To open the usrvars.bat file on the diskette in Notepad, type the following command and press Enter:

```
CustomizeUserVariables.bat
```

Edit the file to customize the installation variables. Be sure to set RSA_INSTALL equal to YES. For more information, see “Customizing the usrvars.bat file” on page 8.

- e. To open the netvars.bat file on the diskette in Notepad, type the following command and press Enter:

```
CustomizeNetworkVariables.bat
```

Edit the file to customize the network variables. For more information, see “Customizing the netvars.bat file” on page 13.

- f. Save the answer file for the unattended installation to the source system. For more information, see “Preparing the answer file for an unattended installation” on page 15.

- g. Type the following command and press Enter:

```
SaveRSABootDisk.bat
```

Issuing this command runs a batch file that creates an image from the DOS-startable (bootable) diskette. By default, the new disk image is named network.img.

- h. Remove the diskette from the diskette drive of the source system.

3. Update the BIOS code and firmware of the target server:

- a. From the source system, open a Web browser.
- b. In the **Address** or **Location** field, type the host name or IP address of the Remote Supervisor Adapter II.
- c. Insert the IBM Update*Xpress* CD into the CD drive of the source system.
- d. Use the Remote Control function of the Remote Supervisor Adapter II Web interface to mount the CD drive that contains the Update*Xpress* CD on the target server.
- e. Use the Remote Supervisor Adapter II Web interface to restart (reboot) the target server and start the BIOS code and firmware update.
- f. When the update is completed, unmount the CD drive from the target server.
- g. Remove the IBM Update*Xpress* CD from the CD drive on the source system.

4. Install the operating system:
 - a. Use the Remote Control function of the Remote Supervisor Adapter II Web interface to mount the diskette image that you created in step 2g on page 21 on the target server.
 - b. Use the Remote Supervisor Adapter II Web interface to restart (reboot) the target server and start the ServerGuide Scripting Toolkit installation. You can use the remote console function of the Remote Supervisor Adapter II Web interface to view the progress of the installation.
 - c. When prompted, unmount the diskette image.
 - d. Use the Remote Supervisor Adapter II Web interface to restart (reboot) the target server to complete the operating system installation.
5. (Optional) Upgrade the device drivers to the most current versions:
 - a. Insert the IBM Update*Xpress* CD into the CD-ROM drive on the source system.
 - b. Use the Remote Control function of the Remote Supervisor Adapter II Web interface to mount the CD-ROM drive containing the Update*Xpress* CD on the target server.
 - c. When the update is completed, unmount the CD-ROM drive from the target server.
 - d. Remove the IBM Update*Xpress* CD from the CD-ROM drive on the source system.

Chapter 4. Using the ServerGuide Scripting Toolkit to deploy Linux

This chapter contains information about using the ServerGuide Scripting Toolkit utilities to perform an unattended installation of Red Hat Linux on IBM *e*server and xSeries servers.

You can use the ServerGuide Scripting Toolkit to deploy the following operating systems:

- Red Hat Linux, Versions 7.3 and 8.0
- Red Hat Enterprise Linux AS, Version 2.1

Preparing the kickstart file

The ServerGuide Scripting Toolkit provides the following sample kickstart files:

Filename	Linux distribution
rh73ks.cfg	Red Hat Linux, Version 7.3
rh80ks.cfg	Red Hat Linux, Version 8.0
rhas21ks.cfg	Red Hat Enterprise Linux AS, Version 2.1

These files are located in the `\stkfiles\sg_stk\examples\linux\redhat` directory of the ServerGuide Scripting Toolkit directory structure.

Note: The sample kickstart file provided for Red Hat Enterprise Linux AS, Version 2.1 is for Red Hat Enterprise Linux AS 2.1, Quarterly Update 2. It might need modifications if you are deploying versions of Red Hat Enterprise Linux AS 2.1 earlier than Quarterly Update 2.

By default, the example batch files use the kickstart files placed in the `\stkfiles\sg_stk\examples\linux\redhat` directory on the source system. Be sure to customize the Installation Media Configuration section to specify the network protocol and location of the operating-system files. The following table contains examples of statements that the Installation Media Configuration section might contain:

Table 3. Example kickstart file statements

Statement	What it does
<code>url --url http://192.168.0.25/rh73/</code>	<ul style="list-style-type: none">• Specifies that the Red Hat Linux installation program uses HTTP to download the operating-system files.• Specifies that the operating-system files are located in the <code>rh73</code> directory of a Web server with IP address 192.168.0.25.
<code>nfs --server 192.168.0.25 --dir /var/www/rh73</code> Note: This example is shown with a line break after <code>dir</code> . In your file, the entire command must be all on one line.	<ul style="list-style-type: none">• Specifies that the Red Hat Linux installation program uses NFS to download the operating-system files.• Specifies that the operating-system files are located in the <code>/var/www/rh73</code> directory of an NFS server with IP address 192.168.0.25.

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

Attention: The ServerGuide Scripting Toolkit utilities do not delete the partitions on the target servers automatically; however, if you use the default kickstart files that are provided by the ServerGuide Scripting Toolkit, the Red Hat Linux installation files delete all partitions.

Deploying Linux using a DOS-startable diskette and network share

This section provides information about deploying Red Hat Linux using a DOS-startable (bootable) diskette and a network share.

The installation scenario assumes the ServerGuide Scripting Toolkit is installed on the source system in the default directory. If the ServerGuide Scripting Toolkit is not installed in this location, change the directory names and paths as needed.

You need the following items for this scenario:

- A blank diskette.
- A system running Windows that contains the ServerGuide Scripting Toolkit files. This system also can contain the source tree, but it must contain the ServerGuide Scripting Toolkit files. Otherwise, the batch files will not run correctly.
- Network connectivity and access to the source-tree directory.

Note: The operating-system installation files must be accessible with either HTTP, Network File System (NFS) or File Transfer Protocol (FTP), as required by the Red Hat Installation program. The ServerGuide Scripting Toolkit files must be accessible via the Server Message Block/Common Internet File System (SMB/CIFS) protocol, which can be accomplished with either Samba in Linux, or a network share directory in Windows.

- Update*Xpress* CD.

Complete the following steps to deploy Linux to an IBM *@server* and xSeries server:

1. Configure the network share:
 - a. Make the source-tree directory accessible to the network.
 - b. To prevent the source tree from being accidentally deleted, configure the source-tree directory as read-only.
 - c. Create a user ID and password for the source-tree directory.

Note: The user name and password are not encrypted. As a precaution, consider using a user account on the source system that has only the authority to read and run files on the source tree.

2. Create the DOS-startable (bootable) diskette:
 - a. Insert a blank diskette into the diskette drive of the source system or the system running Windows that you are using to connect to the source system.
 - b. To change to the ServerGuide Scripting Toolkit directory, from a command prompt, type the following command and press Enter:

```
cd stkfiles\sg_stk\boot
```

- c. To expand the DOS-startable (bootable) diskette image onto the blank diskette, type the following command and press Enter:
`MakeLinuxNetworkInstallToolkitDisk.bat`
 - d. To open the `usrvars.bat` file on the diskette in Notepad, type the following command and press Enter:
`CustomizeUserVariables.bat`

 Edit the file to customize the installation variables. For more information, see “Customizing the `usrvars.bat` file” on page 8.
 - e. To open the `netvars.bat` file on the diskette in Notepad, type the following command and press Enter:
`CustomizeNetworkVariables.bat`

 Edit the file to customize the network variables. For more information, see “Customizing the `netvars.bat` file” on page 13.
 - f. Save the answer file for the unattended installation to the source system. For more information, see “Preparing the kickstart file” on page 23.
 - g. Label the DOS-startable (bootable) diskette.
3. Update the BIOS code and firmware:
 - a. Start the target server and insert the IBM Update*Xpress* CD into the CD-ROM drive.
 - b. Restart the target server. IBM Update*Xpress* starts. Follow the instructions on the screen to update the BIOS code and firmware.
 - c. When the update is completed, remove the IBM Update*Xpress* CD from the CD-ROM drive.
 4. Install the operating system:
 - a. Insert the DOS-startable (bootable) diskette into the diskette drive of the target server.
 - b. Restart the target server. The server starts from the DOS-startable (bootable) diskette and performs the ServerGuide Scripting Toolkit installation.
 - c. Either when prompted or when the installation is completed, remove the DOS-startable (bootable) diskette from 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
- Broadcom NetXtreme Gigabit Ethernet NIC present in many IBM @server and xSeries servers

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 more information, see the kickstart HOWTO available from the Linux Documentation Project Web site at <http://www.tldp.org>.

Chapter 5. Using the ServerGuide Scripting Toolkit with Microsoft ADS

This chapter contains information about using the ServerGuide Scripting Toolkit with existing deployment processes based on Microsoft Automated Deployment Services (ADS). You can use ServerGuide Scripting Toolkit and Microsoft ADS to perform the following tasks:

- Erase data from a server hard disk and restore ServeRAID controllers and integrated SCSI adapters with RAID capabilities to the factory-default settings
- Configure RAID controllers
- Add device drivers to the Microsoft ADS device-driver set

The ServerGuide Scripting Toolkit must be installed on the same server as Microsoft ADS. The server must be running Windows Server 2003, Enterprise Edition.

Erasing data and configuring RAID controllers

To securely erase data from a server hard disk or configure RAID controllers, you first create the applicable ServerGuide Scripting Toolkit diskette; then, you create a Microsoft ADS sequence and Microsoft ADS new job template.

To perform the following task	Go to
Securely erase data from a server	"Creating the Microsoft ADS server disposal diskette"
Configure a RAID controller	"Creating the RAID configuration diskette" on page 29

Creating the Microsoft ADS server disposal diskette

Notes:

1. You will need a blank diskette for this procedure.
2. This procedure assumes that the ServerGuide Scripting Toolkit is installed on the source system in the default directory. If the ServerGuide Scripting Toolkit is not installed in this location, change the names and directory paths as needed.

Complete the following steps to create the DOS-startable (bootable) Microsoft ADS server disposal diskette:

1. Insert a blank diskette into the diskette drive of the source system.
2. To change to the Microsoft ADS binary directory, from a command prompt, type the following command and press Enter:

```
cd \Program Files\Microsoft ADS\bin
```

3. To run the Microsoft-provided dskimage program and expand the vfloppy image file onto the blank diskette, type the following command and press Enter:

```
dskimage.exe vfloppy.vfi a: /f
```

where *a* is the drive letter of the diskette drive.

4. To copy the vfreboot.com program to the source system, type the following command and press Enter:

```
copy a:\vfreboot.com d:\
```

where *a* is the drive letter of the diskette drive, and *d* is the drive letter of the hard disk drive.

5. To change to the root directory of the ServerGuide Scripting Toolkit, type the following command and press Enter:

```
cd \stkfiles\sg_stk\boot
```

6. To expand the DOS-startable (bootable) diskette image onto the diskette, type the following command and press Enter:

```
MakeADSDisposalToolkitDisk.bat
```

7. To copy the vfreboot.com program to the diskette, type the following command and press Enter:

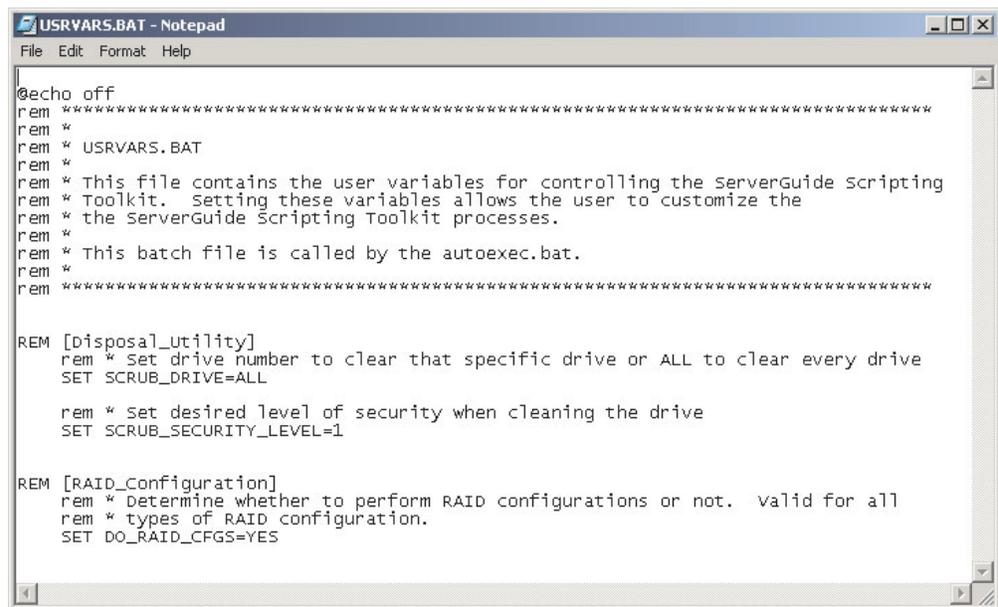
```
copy d:\vfreboot.com a:\
```

where *d* is the drive letter of the hard disk drive and *a* is the drive letter of the diskette drive.

8. Type the following command and press Enter:

```
CustomizeUserVariables.bat
```

The usrvars.bat file opens in Notepad.



```
USRVARS.BAT - Notepad
File Edit Format Help

@echo off
rem *****
rem *
rem * USRVARS.BAT
rem *
rem * This file contains the user variables for controlling the ServerGuide Scripting
rem * Toolkit. Setting these variables allows the user to customize the
rem * the ServerGuide Scripting Toolkit processes.
rem *
rem * This batch file is called by the autoexec.bat.
rem *
rem *****

REM [Disposal_Utility]
rem * Set drive number to clear that specific drive or ALL to clear every drive
SET SCRUB_DRIVE=ALL

rem * Set desired level of security when cleaning the drive
SET SCRUB_SECURITY_LEVEL=1

REM [RAID_Configuration]
rem * Determine whether to perform RAID configurations or not. Valid for all
rem * types of RAID configuration.
SET DO_RAID_CFGS=YES
```

Figure 5. Server disposal *usrvars.bat* file opened in Notepad

9. Complete the following steps to configure the disposal variables:
 - a. Set SCRUB_DRIVE equal to one of the following variables:
 - ALL, which specifies that all hard disk drives are erased
 - *n*, which specifies that the *n*th hard disk drive is erased, where *n* is a positive integer

By default, this variable is set to ALL.

- b. Set SCRUB_SECURITY_LEVEL equal to one of the following variables:

Value	Meaning
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.

By default, this variable is set to 1.

- c. Set DO_RAID_CFGS equal to YES to restore ServeRAID controllers and integrated SCSI controllers with RAID capabilities to the factory-default settings. By default, this variable is set to YES.
10. To change to the Microsoft ADS tftproot directory, type the following command and press Enter:
- ```
cd \Program Files\Microsoft ADS\tftproot
```
11. To create a Microsoft ADS-formatted virtual diskette image from the diskette, type the following command and press Enter:
- ```
diskimage.exe a: DiskName
```

where *a* is the drive letter of the diskette drive, and *DiskName* is a DOS 8.3 format name of the diskette.

12. Remove the DOS-startable (bootable) Microsoft ADS server disposal diskette from the diskette drive and label it.

Go to “Creating a Microsoft ADS sequence” on page 30.

Creating the RAID configuration diskette

Notes:

1. You will need a blank diskette for this procedure.
2. This procedure assumes that the ServerGuide Scripting Toolkit is installed on the source system in the default directory. If the ServerGuide Scripting Toolkit is not installed in this location, change the names and directory paths as needed.

Complete the following steps to create the DOS-startable (bootable) RAID configuration diskette:

1. Insert a blank diskette into the diskette drive of the source system.
2. To change to the Microsoft ADS binary directory, from a command prompt, type the following command and press Enter:

```
cd \Program Files\Microsoft ADS\bin
```

3. To run the Microsoft-provided diskimage program and expand the vfloppy image file onto the blank diskette, type the following command and press Enter:

```
diskimage.exe vfloppy.vfi a: /f
```

where *a* is the drive letter of the diskette drive.

4. To copy the vfreboot.com program to the source system, type the following command and press Enter:

```
copy a:\vfreboot.com d:\
```

where *a* is the drive letter of the diskette drive, and *d* is the drive letter of the hard disk drive.

5. To change to the root directory of the ServerGuide Scripting Toolkit, type the following command and press Enter:

```
cd \stkfiles\sg_stk\boot
```

6. To expand the DOS-startable (bootable) diskette image onto the diskette, type the following command and press Enter:

```
MakeADSToolkitHardwareConfig.bat
```

7. To copy the vfreboot.com program to the diskette, type the following command and press Enter:

```
copy d:\vfreboot.com a:\
```

where *d* is the drive letter of the hard disk drive and *a* is the drive letter of the diskette drive.

8. To open the usrvars.bat file in Notepad, type the following command and press Enter:

```
CustomizeUserVariables.bat
```

Edit the file to customize the installation variables. For more information, see “Configuring the RAID variables” on page 11.

9. To change to the Microsoft ADS tftproot directory, type the following command and press Enter:

```
cd \Program Files\Microsoft ADS\tftproot
```

10. To create a Microsoft ADS-formatted virtual diskette image from the diskette, type the following command and press Enter:

```
diskimage.exe a: DiskName
```

where *a* is the drive letter of the diskette drive, and *DiskName* is a DOS 8.3 format name of the diskette.

11. Remove the DOS-startable (bootable) Microsoft ADS RAID configuration diskette from the diskette drive and label it.

Go to “Creating a Microsoft ADS sequence”.

Creating a Microsoft ADS sequence

Complete the following steps to create a new Microsoft ADS sequence:

1. Click **Start** → **All Programs** → **Microsoft ADS** → **Sequence Editor**. The Microsoft ADS Sequence Editor opens.
2. In the **Description** field, type a description.
3. Click **Actions** → **Insert** → **Reboot Computer**.
4. Click **Actions** → **Add** → **Boot Virtual Floppy**.
5. In the left pane, click **Boot Virtual Floppy**.

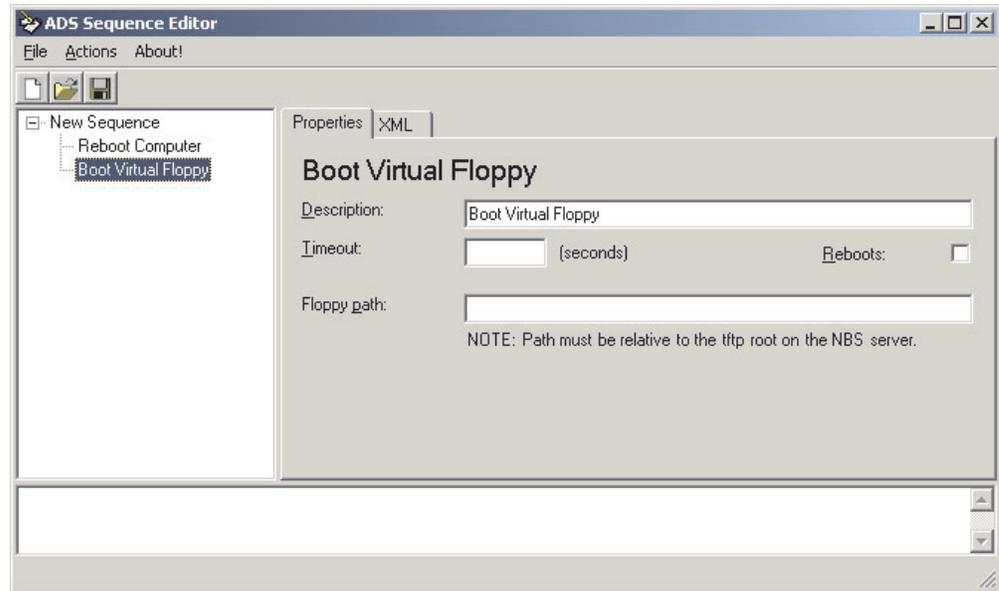


Figure 6. “ADS Sequence Editor” window

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 that you created in “Creating the Microsoft ADS server disposal diskette” on page 27 or in “Creating the RAID configuration diskette” on page 29.
8. Click **File** → **Save As**. The “Save As” window opens.
9. Navigate to the \Program Files\Microsoft ADS\Samples\Sequences directory.

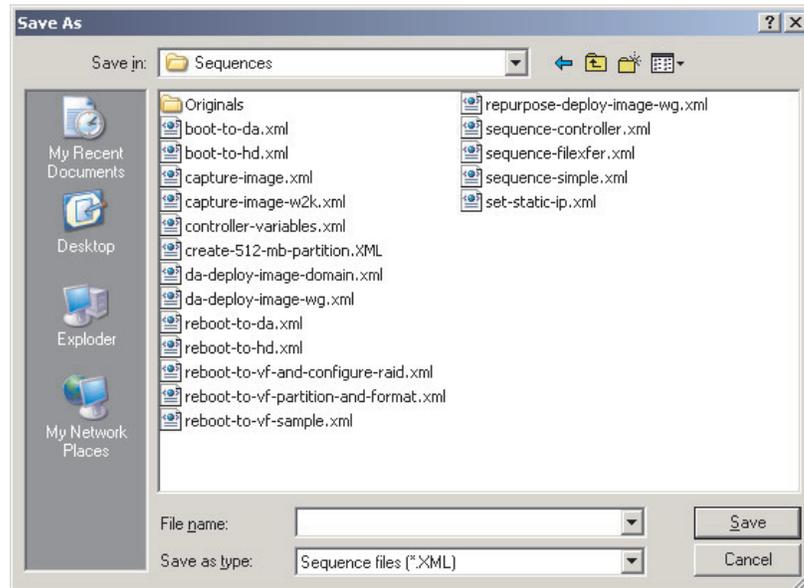


Figure 7. “Save As” window

10. In the **File name** field, type a descriptive name for the Microsoft ADS sequence.
11. Click **Save**.

Go to “Creating a Microsoft ADS job template”.

Creating a Microsoft ADS job template

Complete the following steps to create a new job template:

1. Click **Start** → **All Programs** → **Microsoft ADS** → **ADS Management**. The Microsoft ADS Console starts.
2. Click **Action** → **New Job Template**. The New Job Template wizard starts, and the “Welcome to the New Job Template Wizard” window opens.
3. Click **Next**. The “Template Type” window opens.

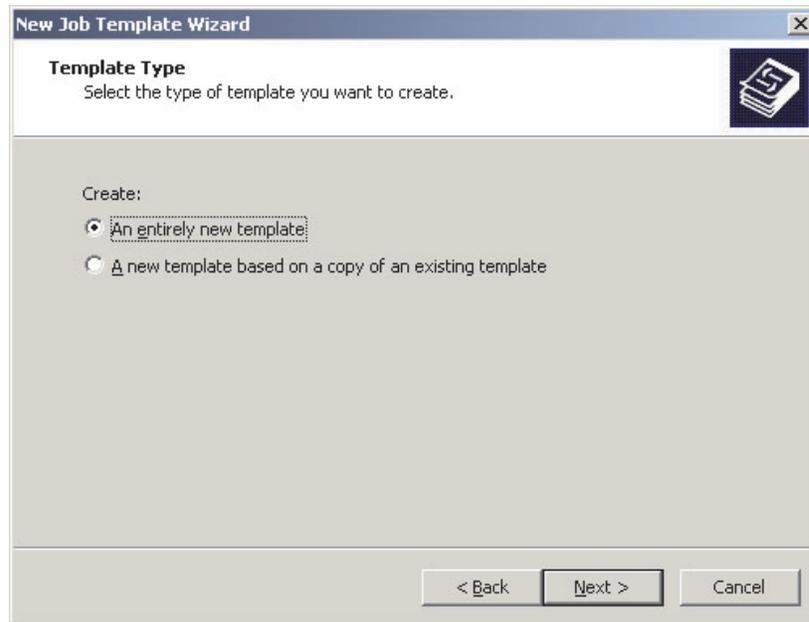


Figure 8. “Template Type” window

4. Click **An entirely new template**, and then click **Next**. The “Name and Description” window opens.

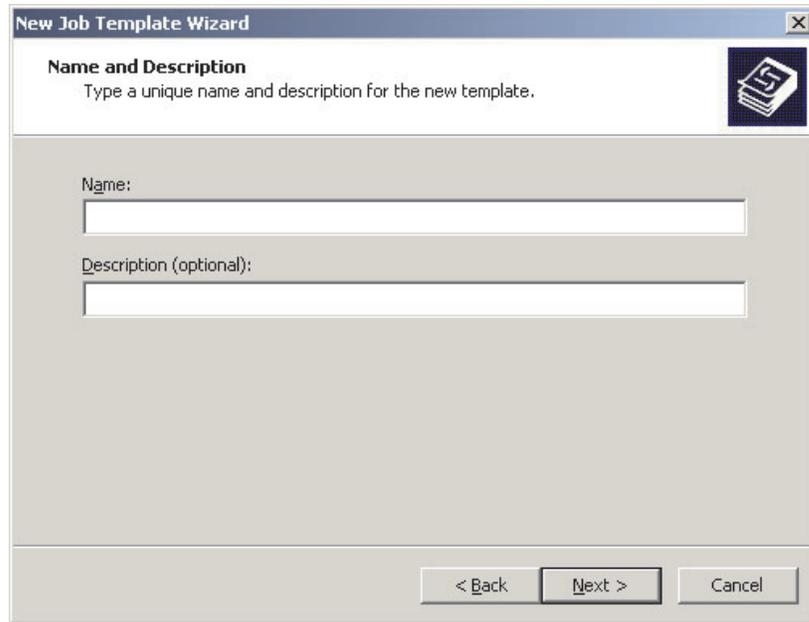


Figure 9. "Name and Description" window

5. In the **Name** field, type a descriptive name. In the **Description** field, type an optional description, and then click **Next**. The "Command Type" window opens.

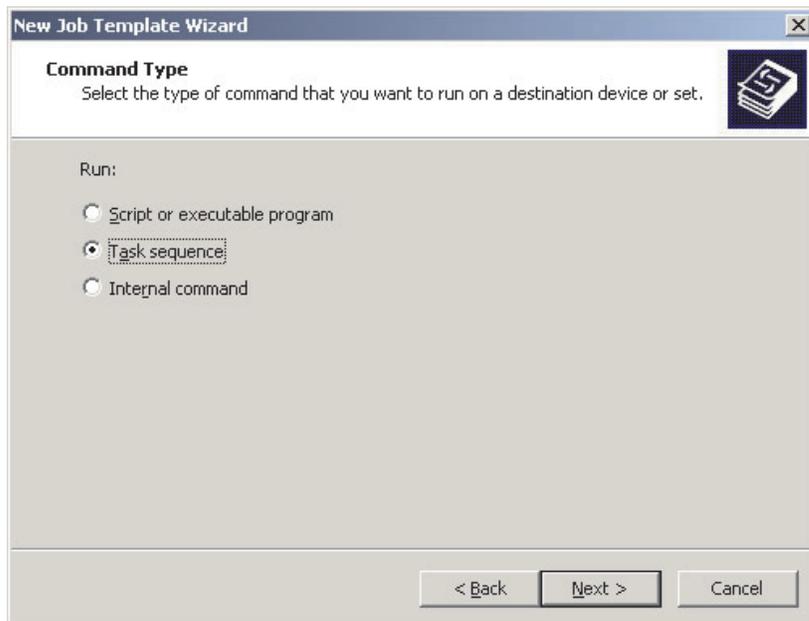


Figure 10. "Command Type" window

6. Click **Task Sequence**, and then click **Next**. The "Task Sequence File and Arguments" window opens.

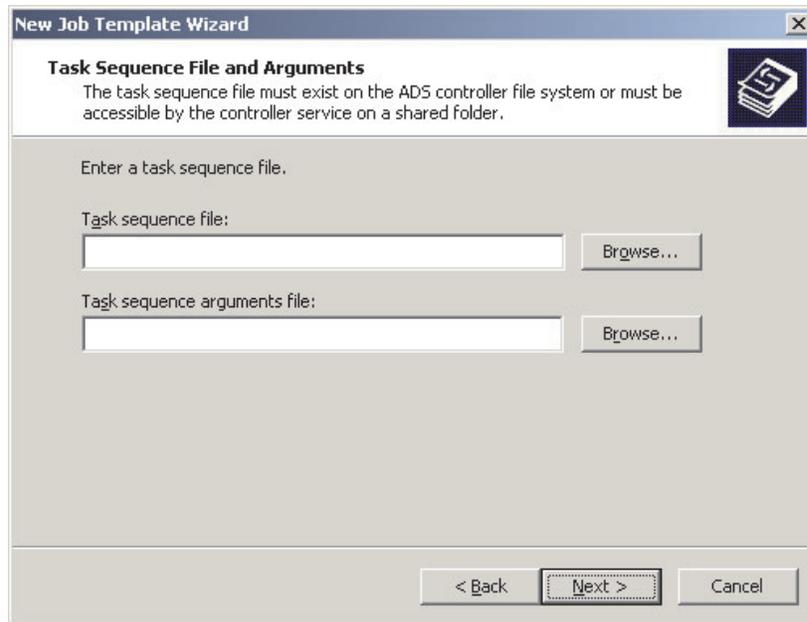


Figure 11. “Task Sequence File and Arguments” window

7. To select the task sequence file, click **Browse**. Locate the applicable task sequence file you created in “Creating a Microsoft ADS sequence” on page 30, and then click **Next**. The “Destination Device Selection” window opens.

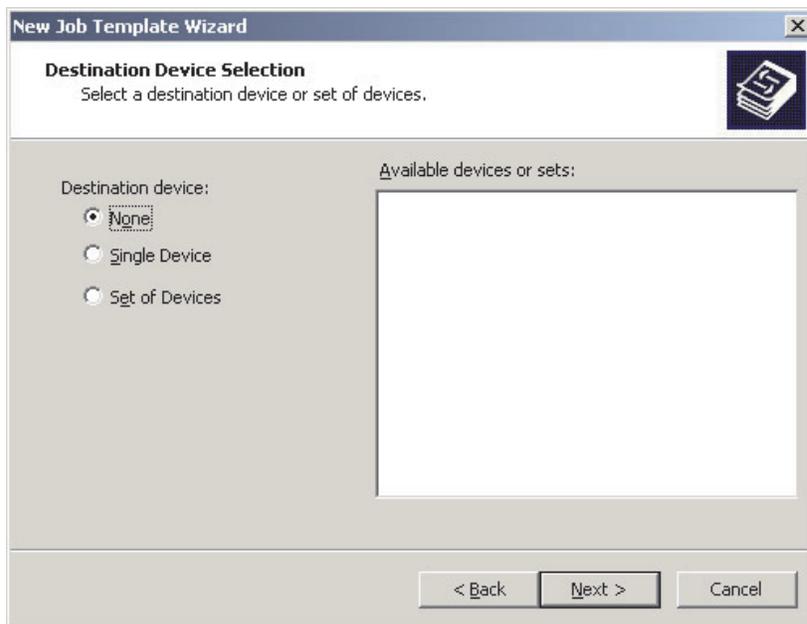


Figure 12. “Destination Device Selection” window

8. Click **None**, and then click **Next**. The “Completing the New Job Template Wizard” window opens.
9. Click **Finish**.

You now can securely erase a server hard disk drive or configure the RAID controller in a server by running the Microsoft ADS job template.

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

Adding device drivers to the Microsoft ADS device-driver set

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

- Broadcom NIC device drivers
- IBM hard disk drive device drivers

For more information, search the ADS help for the topic “Deployment Agent Builder service” and the subtopic “overview.”

Notes:

1. You need a *ServerGuide Setup and Installation CD*, Version 7.2 or later.
2. Microsoft ADS, Version 1.0, cannot correctly parse some INF files. Make sure that you have installed the latest Microsoft ADS updates.

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

1. Copy the following applicable device driver files to the PreSystem directory on the server that is running Microsoft ADS:

Device drivers	Location on the ServerGuide CD
BroadCom NIC (b57xp32 ini and b57xp32.sys)	\sguide\w2003drv\soem\$\1\drv\bc
IBM hard disk drive	\sguide\w2003drv\soem\$\textmode

If you installed Microsoft ADS in the default location, the PreSystem directory is located at *d*:\Program Files\Microsoft ADS\nbs\repository\user\Presystem, where *d* is the drive letter of the hard disk drive.

2. Shut down Microsoft ADS Console if it is running.
3. Click **Start** → **All Programs** → **Administrative Tools** → **Services**. The “Services” window opens.
4. Right-click **ADS Deployment Agent Builder**, and then click **Stop**.
5. Right-click **ADS Deployment Agent Builder**, and then click **Start**.
6. Close the “Services” window.
7. Start Microsoft ADS Console.

Chapter 6. Using the ServerGuide Scripting Toolkit for server disposal

You can use the ServerGuide Scripting Toolkit to automate the disposal of IBM @server and xSeries servers. The ServerGuide Scripting Toolkit disposal process can perform the following operations:

- Reset the server persistent-state information
- Detect hardware
- Set ServeRAID and integrated SCSI controllers with RAID capabilities to the factory-default settings
- Delete all partitions
- Permanently erase data from one or all drives

Note: You cannot use the server disposal process to set IDE RAID controllers to the factory-default settings.

Creating the server disposal diskette

This scenario assumes the ServerGuide Scripting Toolkit is installed on the source system in the default directory. If the ServerGuide Scripting Toolkit is not installed in this location, change the directory names and paths as needed.

Complete the following steps to create a DOS startable (bootable) server disposal diskette:

1. Insert a blank diskette into the diskette drive of the source system.
2. To change to the boot directory of the ServerGuide Scripting Toolkit, from a command prompt, type the following command and press Enter:
`cd \stkfiles\sg_stk\boot`
3. To expand the DOS-startable (bootable) diskette image onto the blank diskette, type the following command and press Enter:
`MakeDisposalToolkitDisk.bat`
4. Type the following command and press Enter:
`CustomizeUserVariables.bat`

The usrvars.bat file opens in Notepad.

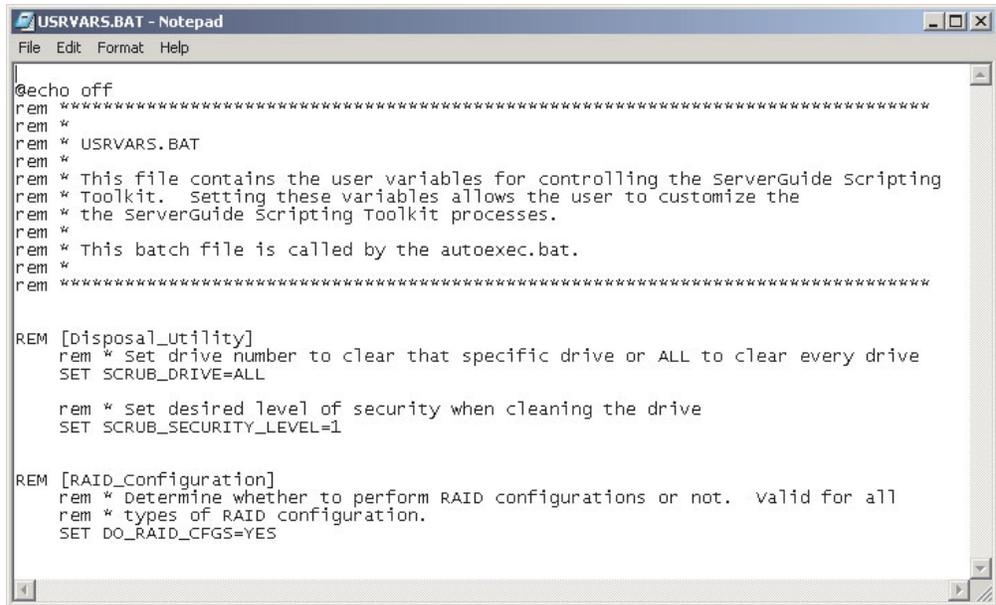


Figure 13. Server disposal usrvs.bat file opened in Notepad

5. Complete the following steps to configure the disposal variables:
 - a. Set SCRUB_DRIVE equal to one of the following values:
 - ALL, which specifies that all hard disk drives are erased
 - *n*, which specifies that the *n*th hard disk drive is erased, where *n* is a positive integer

By default, this variable is set to ALL.
 - b. Set SCRUB_SECURITY_LEVEL equal to one of the following values:

Value	Meaning
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.

- By default, this variable is set to 1.
- c. Set DO_RAID_CFGS equal to YES to restore ServeRAID controllers and integrated SCSI controllers with RAID capabilities to the factory-default settings. By default, this variable is set to YES.
 6. Label the DOS-startable (bootable) server disposal diskette.

Using the server disposal diskette

Complete the following steps to securely erase data from a server:

1. Insert the DOS-startable (bootable) server disposal diskette into the diskette drive of the target server.
2. Start the server.
3. When the operation is completed, remove the diskette from the diskette drive.

You now can securely dispose of the server, or you can redeploy an operating system to the server.

Chapter 7. ServerGuide Scripting Toolkit utilities

The command-line syntax in this document uses the following conventions:

- Commands are shown in lowercase letters.
- Variables are shown in italics and explained immediately afterward.
- Optional commands or variables are enclosed in brackets.
- When you must type one of two or more alternative parameters, the parameters are separated by vertical bars.
- Default values are underlined>.
- Repeatable parameters are enclosed in braces.

altboot.exe

You can use the altboot command to perform the following tasks:

- Bypass the diskette drive in the boot order
- Force a system to start (boot) from the hard disk drive
- Back up boot sector information
- Restore previously-saved boot sector information

Syntax

An altboot command uses the following syntax:

```
altboot /b [/f:filename] | /h [/f:filename] | /r [/f:filename]
```

The diskette must not be write-protected for the altboot command to run correctly. Using the altboot command with a write-protected diskette results in a DOS write-protect error.

The following table contains information about the altboot parameters.

Table 4. altboot parameters

Function	What it does	Syntax
Bypass	Bypasses the diskette drive in the boot order.	<pre>altboot /b [/f:filename]</pre> <p>where <i>[/f:filename]</i> is an optional parameter that specifies that the diskette drive boot sector information be written to an alternative file, and <i>filename</i> is the fully qualified name of the alternative file.</p> <p>If <i>[/f:filename]</i> is not specified, by default the boot sector information is written to <i>a:\bootsect.bin</i>, where <i>a</i> is the drive letter of the first diskette drive (drive 0).</p>
Hard disk	Bypasses the diskette drive and forces system to boot from the hard disk drive.	<pre>altboot /h [/f:filename]</pre> <p>where <i>[/f:filename]</i> is an optional parameter that specifies that the diskette drive boot sector information be written to an alternative file, and <i>filename</i> is the fully qualified name of the alternative file.</p> <p>If <i>[/f:filename]</i> is not specified, by default the boot sector information is written to <i>a:\bootsect.bin</i>, where <i>a</i> is the drive letter of the first diskette drive (drive 0).</p>

Table 4. *altboot* parameters (continued)

Restore	Restores the diskette drive boot sector.	<p><code>altboot /r [/f:filename]</code></p> <p>where <code>[/f:filename]</code> is an optional parameter that specifies that the diskette drive boot sector information be read from an alternative file, and <code>filename</code> is the fully qualified name of the alternative file.</p> <p>If <code>[/f:filename]</code> is not specified, by default the boot sector information is read from <code>a:\bootsect.bin</code>, where <code>a</code> is the drive letter of the first diskette drive (drive 0).</p>
Help	Prints usage information.	<code>altboot /?</code>

The following table shows the return values and their meanings.

Table 5. *altboot* return values

Value	Meaning
0	Successful completion
1	File error, such as an invalid boot sector image or a file read/write error
2	Diskette drive error
3	Command-line argument error

Examples of using the `altboot` command

The following table contains examples of using the `altboot` command.

Table 6. Examples of using the `altboot` command

Command	What it does
<code>altboot /b</code>	Bypasses the diskette drive in the boot order and saves a copy of the diskette drive boot sector information to <code>a:\bootsect.bin</code>
<code>altboot /b /f:a:\saveboot.bin</code>	Bypasses the diskette drive in the boot order and saves a copy of the diskette drive boot sector information to <code>a:\saveboot.bin</code>
<code>altboot /r</code>	Restores the diskette drive boot sector information from <code>a:\bootsect.bin</code>
<code>altboot /r /f:a:\test.bin</code>	Restores the diskette drive boot sector information from <code>a:\test.bin</code>
<code>altboot /h /f:a:\floppy.bin</code>	Starts (boots) the system from the hard disk drive and saves a copy of the diskette drive boot sector information to <code>a:\floppy.bin</code>

cfg1030.exe

You can use the `cfg1030` utility to configure an integrated SCSI controller with RAID capabilities. You can perform the following tasks:

- Create a logical drive and assign it RAID level-1
- Back up the configuration settings to a file
- Generate information about the controller and its RAID configuration
- Create a hot-spare drive
- Restore a controller to the factory-default settings

Syntax

The cfg1030 utility uses the following syntax:

`cfg1030 command parameters`

The following table contains information about the cfg1030 commands and parameters.

Table 7. *cfg1030 commands and parameters*

Function	What it does	Syntax
Create	Creates a logical drive and assigns it RAID level-1.	<p><code>cfg1030 create controller logicaldrive newarray size 1 {1 drive} [qsync]</code></p> <p>where</p> <ul style="list-style-type: none"> <code>controller</code> is the number of the SCSI controller. <code>size</code> is one of the following values: <ul style="list-style-type: none"> The size of the logical drive in MB MAX, which specifies that you want to use all available space in the array <code>{1 drive}</code> is a repeatable parameter that specifies the hard disk drives included in the logical drive, and <code>drive</code> is the SCSI ID of the hard disk drive. <code>[qsync]</code> is an optional parameter that specifies that the logical drive is quick synchronized. The first 32 KB of the physical drives in the logical drive are cleared to 0.
Back up	Backs up the configuration settings to a file.	<p><code>cfg1030 backup controller filename</code></p> <p>where</p> <ul style="list-style-type: none"> <code>controller</code> is the number of the SCSI controller. <code>filename</code> is the filename. The filename can be fully qualified. If you provide only a filename the file is saved to the local directory.
Get configuration	Lists information about the controller.	<p><code>cfg1030 getconfig controller [option]</code></p> <p>where</p> <ul style="list-style-type: none"> <code>controller</code> is the number of the SCSI controller. <code>[option]</code> is an optional parameter that specifies the information that is returned, and <code>option</code> is one of the following values: <ul style="list-style-type: none"> AD, which specifies the controller information LD, which specifies the logical drive information PD, which specifies the physical drive information ALL, which specifies all information <p>If an option is not specified, by default the getconfig command returns all information.</p>
Restore	Configures a controller using information and settings stored in a file.	<p><code>cfg1030 restore controller filename</code></p> <p>where</p> <ul style="list-style-type: none"> <code>controller</code> is the number of the SCSI controller. <code>filename</code> is the filename. The filename can be fully qualified. If you provide only a filename the <code>cfg1030</code> command searches for the file in the local directory.

Table 7. *cfg1030* commands and parameters (continued)

Function	What it does	Syntax
Hot spare	Creates a hot-spare drive.	<code>cfg1030 setstate controller 1 drive hsp</code> where <ul style="list-style-type: none"> • <i>controller</i> is the number of the SCSI controller. • <i>drive</i> is the SCSI ID of the hard disk drive.
Erase	Deletes logical drives and restores a controller to the factory-default settings.	<code>cfg1030 setconfig controller default</code> where <i>controller</i> is the number of the SCSI controller.

Examples of using the *cfg1030* command

The following table contains examples of using the *cfg1030* utility.

Table 8. Examples of using the *cfg1030* command

Command	What it does
<pre>cfg1030 create 1 logicaldrive newarray max 1 1 0 1 1 qsync</pre> <p>Note: This example is shown with a line break after <code>max</code>. In your file, the entire command must be all on one line.</p>	Performs the following actions: <ul style="list-style-type: none"> • Creates a new array and logical drive using controller 1 • Uses all the available space in the array for the logical drive • Assigns the logical drive RAID level-1 • Uses the physical drives at channel 1, SCSI ID 0 and SCSI ID 1 • Specifies that a quick synchronization is performed
<code>cfg1030 setstate 1 0 1 HSP</code>	Designates the physical drive at channel 0, SCSI ID 1 as a hot spare drive for controller 1.

clini.exe and clini32.exe

You can use the *clini* and *clini32* commands to perform the following tasks:

- Create, add, or change values in an INI file
- Read a value from an INI file and store it as an environment variable

The *clini32* command runs only on Windows; the *clini* command runs on both DOS and Windows. You can use Windows long-file names with *clini32*, but you must use file names that follow the DOS 8.3 format with *clini*.

Note: When using *clini* with fully qualified path names, remember that DOS has a path limit of 127 characters.

To store the value as an environment variable, *clini* creates a batch file that contains a command to set the environment variable. By default, this batch file is named `cliniset.bat`. If a `cliniset.bat` file already exists, it is deleted and re-created. You must run the batch file to set the environment variable.

Syntax

The `clini` and `clini32` commands use the following syntax:

```
clini filename /s:section [option]
```

where:

- *filename* is the fully qualified name of the INI file.
- *section* is the name of the section in the INI file.

The following table contains information about the `clini` optional parameters.

Table 9. *clini* optional parameters

Function	What it does	Syntax
Item	Specifies the name of the item in the INI file. If the item is not specified, the value is written in the section.	<i>/i:item</i> where <i>item</i> is the item name.
Value	Specifies the value of the item in the INI file.	<i>/v:value</i> where <i>value</i> is the item value.
Environment variable	Creates a batch file that contains a command to set the specified value as an environment variable. If the batch file already exists, it is deleted and re-created.	<i>/e:environmentvariable [/b:filename]</i> where: <ul style="list-style-type: none"> • <i>environmentvariable</i> is the name of the environment variable. • <i>[/b:filename]</i> is an optional parameter that specifies the name and location of the batch file, and <i>filename</i> is the fully qualified name of the batch file. By default, the batch file is named <code>cliniset.bat</code> and is located in the current working directory.
New	Creates a new INI file. If a file exists with the specified name, it is deleted and a new file created.	<i>/n</i> Note: This parameter is not valid if the <i>/e</i> parameter is issued.

The following table shows the return values and their meanings.

Table 10. *clini* return values

Value	Meaning
0	Success.
1	Syntax error.
2	Program error.
3	The destination is read-only.
4	The current working directory is read-only.
5	File not found.

Examples of using the clini command

The following table contains examples of using clini commands.

Table 11. Examples of using the clini command

Command	What it does
<code>clini info.ini /s:Hardware /i:Machine Type /v:8549 /n</code>	Creates an info.ini file containing the following string in the [Hardware] section: Machine Type = 8549 If an info.ini file already exists, the file is deleted and re-created, containing only the section and string as described.
<code>clini info.ini /s:Hardware /i:Machine Name /v:Server1</code>	Adds the following string to the [Hardware] section of an existing info.ini file: Machine Name = Server1 If an info.ini file does not exist, it is created.
<code>clini info.ini /s:Hardware /i:Machine Type /e:MachineType</code>	Retrieves the Machine Type information from the [Hardware] section of the info.ini file. Creates a default batch file in the current working directory; this file contains a command to set an environment variable named MachineType to the value retrieved from the info.ini file. By default, the batch file is named cliniset.bat. To set the environment variable, you must run the batch file by issuing the following command: cliniset.bat
<code>clini info.ini /s:Hardware /i:Machine Type2 /v:%MachineType%</code> Note: This example is shown with a line break. In your file, the entire command must be all on one line.	Adds the following string to the [Hardware] section of the info.ini file: Machine Type2 = <i>EnvironVar</i> where <i>EnvironVar</i> is the value of the environment variable named MachineType.
<code>clini info.ini /s:Hardware /i:Machine Type2 /e:MachineType2 /b:d:\EnvSet1.bat</code> Note: This example is shown with a line break. In your file, the entire command must be all on one line.	Retrieves the Machine Type2 information from the [Hardware] section of the info.ini file. Creates a batch file with the fully qualified name of d:\EnvSet1.bat; this file contains a command to set an environment variable named MachineType2 to the value retrieved from the info.ini file. To set the environment variable, you must run the batch file by issuing the following command: d:\EnvSet1.bat

After the preceding commands are run, the info.ini file contains the following information:

```
-----info.ini-----
[Hardware]
Machine Type = 8549
Machine Name = Server1
Machine Type2 = 8549
-----
```

In addition, two batch files exist that contain commands to set two environment variables MachineType and MachineType2.

dscan.exe and dscan32.exe

Note: The device drivers provided on the *ServerGuide Setup and Installation CD* already are configured for use with ServerGuide Scripting Toolkit; you do not need to run the dscan utility against these device drivers.

You can use the dscan and dscan32 commands to perform the following tasks:

- 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 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 dynamically create a master txtsetup.oem file that contains all the information from the individual txtsetup.oem files. Each entry in the master txtsetup.oem file is unique, and known unattended installation defects are automatically addressed.

The following figure shows the directory structure of the Windows 2000 device-driver directories copied from the *ServerGuide Setup and Installation CD*:

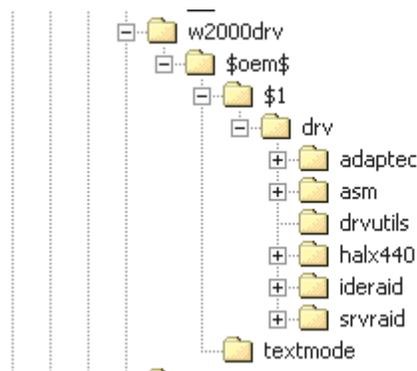


Figure 14. Windows 2000 device-driver directories displayed in Windows Explorer

Note that all device-driver files are located in individual directories. The directories that contain the files for an individual device driver are called *device-driver directories*. Figure 14 contains the following device-driver directories:

- \adaptec
- \asm
- \halx440
- \ideraid
- \srvraid

When you issue a dscan command against a directory that contains device-driver directories (for example, c:\Src_tree\w2000drv\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.

Syntax

The dscan command uses the following syntax:

```
dscan driverpath [options]
```

where *driverpath* is the fully qualified name of a directory that contains device-driver directories.

If you issue the dscan command without any optional parameters, the utility automatically performs all of the following tasks:

- Scans all device drivers and creates an drvinfo.ini file for each device driver. The drvinfo.ini files are located in each of the device-driver directories.
- (If *driverpath* includes \$oem\$) Creates a textmode directory, copies any text mode device-driver files and txtsetup.oem files to the textmode directory, and dynamically creates a master txtsetup.oem file. The master txtsetup.oem file is located in the textmode directory, which is by default located at \ \$oem\$ \textmode.

When a dscan command is run that creates a textmode directory, an existing textmode directory is deleted and recreated. When a dscan command is run that generates a drvinfo.ini file, if a drvinfo.ini file already exists in the device-driver directory, the file is not overwritten.

You can add parameters to the dscan command to make sure that dscan performs only one of the following tasks:

- Scans all device drivers
- Scans a single device driver
- Creates the textmode directory, copies the text mode device-driver files into the textmode directory, and dynamically creates the master txtsetup.oem file.

The following table contains information about the dscan optional parameters.

Table 12. dscan optional parameters

Function	What it does	Syntax
Scan	Scans device drivers and creates drvinfo.ini files.	dscan <i>driverpath</i> /s
Scan Single	Scans a single device driver and creates the drvinfo.ini file.	dscan <i>driverpath</i> /ss

Table 12. *dscan* optional parameters (continued)

Textmode	Creates the textmode directory, copies the text mode device drivers into the textmode directory, and dynamically creates the master txtsetup.oem file.	<p><code>dscan driverpath /t[:path] [/m:MachineType]</code></p> <p>where:</p> <ul style="list-style-type: none"> • <code>[:path]</code> is an optional parameter that specifies an alternative location for the textmode directory. If this directory already exists, it is deleted and re-created. • <code>[/m:MachineType]</code> is an optional parameter that specifies that only device-driver files that pertain to a specific server are copied to the textmode directory, and only txtsetup.oem files that pertain to a specific server are included in the master txtsetup.oem file. <i>MachineType</i> is a machine type. <p>Note: If <i>path</i> is not specified, <i>driverpath</i> must include \$oem\$.</p>
Verbose	Specifies the verbose level.	<p><code>dscan driverpath /v:VerboseLevel</code></p> <p>where <i>VerboseLevel</i> is an integer in the 0-5 range. 0 is quiet, 3 is the default, and 5 is the maximum.</p>
Windows version	Specifies the Windows version for which the device drivers are processed.	<p><code>dscan driverpath /w:WindowsVersion</code></p> <p>where <i>WindowsVersion</i> is one of the following values:</p> <ul style="list-style-type: none"> • 0, which specifies Windows 2000 • 1, which specifies Windows 2003
Help	Shows the usage information.	<code>dscan /?</code>

The following table shows the return values and their meanings.

Table 13. *dscan* return values

Value	Meaning
0	Successful completion.
1	Syntax error.
2	Program error.
3	The destination is read-only.

dscan.exe examples

The following table contains examples of using the *dscan* command.

Table 14. Examples of using the *dscan* command

Command	What it does
<code>dscan c:\insttemp\%oem%\\$1\drv</code>	<ul style="list-style-type: none"> • Scans the device drivers in the <code>c:\insttemp\%oem%\\$1\drv</code> directory and any of the subdirectories that it contains. Creates <code>drvinfo.ini</code> files for each device driver found. • Creates a <code>c:\insttemp\%oem%\textmode</code> directory, copies the text mode device drivers to the <code>c:\insttemp\%oem%\textmode</code> directory, and dynamically creates the master <code>txtsetup.oem</code> file.

Table 14. Examples of using the dscan command (continued)

Command	What it does
dscan c:\drivers /s	Scans the device drivers in the c:\drivers directory and any of the subdirectories that it contains. Creates drvinfo.ini files for each device driver found.
dscan c:\drivers\mydriver /ss	Scans the single device driver located in the c:\drivers\mydriver directory. Creates a drvinfo.ini file for the device driver.
dscan c:\w2\%oem%\\$1\drv /t	Creates a c:\w2\%oem%\textmode directory, copies the text mode device-driver files to the c:\w2\%oem%\textmode directory, and creates the master txtsetup.oem file. This dscan action is performed against all text mode device-driver files that are contained in the c:\w2\%oem%\\$1\drv directory and any subdirectories that it contains. (You must have previously scanned the device drivers and created the drvinfo.ini files.)
dscan c:\drivers /t:c:\other\textmode	Creates a c:\other\textmode directory, copies the text mode device-driver files to the c:\other\textmode directory, and creates the master txtsetup.oem file. This dscan action is performed against all text mode device-driver files that are contained in the c:\drivers directory and any subdirectories that it contains. (You must have previously scanned the device drivers and created the drvinfo.ini files.)
dscan c:\drivers /t:c:\other\textmode /m:8832	Creates a c:\other\textmode directory, copies the text mode device-driver files to the c:\other\textmode directory, and creates the master txtseup.oem file. This dscan action is performed against text mode device-driver files that meet both of the following criteria: <ul style="list-style-type: none"> • Located in the c:\drivers directory • Pertain to the machine type 8832 server You must have previously scanned the device drivers and created the drvinfo.ini files. In addition, you must have modified the value of the Supported Systems keyword in the drvinfo.ini files to indicate which device drivers are supported on the BladeCenter HS20, machine type 8832 server.

drvinfo.ini files

Each drvinfo.ini file contains information that pertains to a specific device driver. You can generate the drvinfo.ini file by running the dscan command; you also can write or edit the drvinfo.ini file using an ASCII text editor. The unattend.exe command uses the drvinfo.ini file to add device-driver information to the answer file for the unattended installation.

If a drvinfo.ini file is generated by running dscan, the Installation Mode and Path keywords are assigned values automatically. You might want to modify the drvinfo.ini file to specify values for other keywords.

The following table contains information about the keywords and values that are used in a drvinfo.ini file.

Table 15. Keywords and values in drvinfo.ini files

Keyword	Value	What it does
Installation Mode	<p>One of the following values:</p> <ul style="list-style-type: none"> • PnP (Plug and Play) • Executable • Textmode • Manual 	<p>Specifies the installation mode of the device driver.</p> <p>Note: If Installation Mode is set to Manual, the unattend.exe command will not install the device driver.</p>
Path	<p>One of the following values:</p> <ul style="list-style-type: none"> • Path to the directory where the INF file is located (Plug and Play device driver) • Path to the directory where the EXE file is located (executable device driver) • Path to the directory where the txtsetup.oem file is located (text mode device driver). • Blank, meaning the device-driver file is located at the root of the device-driver directory. <p>Each value assigned to Path is relative to the path of the device-driver directory.</p>	<p>Specifies the location of the device-driver installation file.</p> <p>Note: If Installation Mode is set to Manual, the Path keyword is ignored.</p>
Parameters	<p>Command-line parameters required by an executable device driver.</p>	<p>Specifies command-line parameters required by an executable device driver.</p>
Automatically Reboots	<ul style="list-style-type: none"> • True • False 	<p>When set to True, this keyword informs the ServerGuide Scripting Toolkit that the executable-device-driver installation program automatically restarts (reboots) the server.</p> <p>If an executable-device-driver installation program restarts the server, and this keyword is set to False, then any remaining installation procedures are not completed.</p> <p>By default, this keyword is set to False.</p> <p>Notes:</p> <ol style="list-style-type: none"> 1. To use this keyword, the drvutils directory (copied from the <i>ServerGuide Setup and Installation CD</i>) must be located on the source tree. 2. This keyword is supported only on servers running Windows 2000. 3. You must use the unattend.exe utility to add device-driver information to the unattend.txt file.

Table 15. Keywords and values in drvinfo.ini files (continued)

Keyword	Value	What it does
Reboot Required	<ul style="list-style-type: none"> • True • False 	<p>When set to True, this keyword specifies that ServerGuide Scripting Toolkit restarts (reboots) the server immediately after an executable device driver is installed.</p> <p>By default, this keyword is set to False.</p> <p>Notes:</p> <ol style="list-style-type: none"> 1. To use this keyword, the drvutils directory (copied from the <i>ServerGuide Setup and Installation CD</i>) must be located on the source tree. 2. This keyword is supported only on servers running Windows 2000.
PCIVenDevID	<p>One or more PCI VendorID\DeviceID entries.</p> <p>Separate multiple entries with commas. Entries must be in the same format as in txtsetup.oem files.</p> <p>For example:</p> <pre>PCIVenDevID = PCI\VEN_1002&DEV_5159&SUBSYS_029A1014</pre> <p>Note: This example is shown with a line break. In your file, the entire entry must be all on one line.</p>	<p>Specifies that an executable device driver is installed only when the server contains the specified device.</p>
Order Before	<p>One of the following values:</p> <ul style="list-style-type: none"> • All • Name of the device driver <p>If Order Before is set to All, the device driver will be installed before any device drivers of its type.</p> <p>Note: This keyword can be used only for Plug and Play or executable device drivers.</p>	<p>Specifies the order in which a device driver is installed. If set to All, the device driver is installed before all other device drivers of its kind. If set to a specified device driver, the device driver (for which the drvinfo.ini file pertains) is installed before the specified device driver.</p>
Supported Systems	<p>One of the following values:</p> <ul style="list-style-type: none"> • All • None • A comma-delimited list of server machine types <p>By default, this keyword is set to All.</p>	<p>Specifies the servers that the device driver supports.</p> <p>Note: The Supported Systems keyword and the Unsupported Systems keyword cannot both be assigned values in the same drvinfo.ini file.</p>
Unsupported Systems	<p>A comma-delimited list of server machine types.</p>	<p>Specifies the servers that the device driver does not support.</p> <p>Note: The Supported Systems keyword and the Unsupported Systems keyword cannot both be assigned values in the same drvinfo.ini file.</p>

Table 15. Keywords and values in drvinfo.ini files (continued)

Keyword	Value	What it does
Supported Locales	<p>One of the following values:</p> <ul style="list-style-type: none"> • All • A comma-delimited list of locales as specified by the Localization keyword in the ProdSpec.ini file <p>By default, this keyword is set to All.</p> <p>For more information about locales, see the values that the Localization keyword takes in the ProdSpec.ini file. This file is located in the i386 directory of a Windows installation CD.</p>	<p>Specifies the locales that the device driver supports.</p> <p>Note: The Supported Locales keyword and the Unsupported Locales keywords cannot both be assigned values in the same drvinfo.ini file.</p>
Unsupported Locales	<p>A comma-delimited list of locales.</p> <p>For more information about locales, see the values that the Localization keyword takes in the ProdSpec.ini file. This file is located in the i386 directory of a Windows installation CD.</p>	<p>Specifies the locales that the device driver does not support.</p> <p>Note: The Supported Locales keyword and the Unsupported Locales keywords cannot both be assigned values in the same drvinfo.ini file.</p>

drvinfo.ini examples

The following table contains examples of drvinfo.ini files.

Table 16. Examples of drvinfo.ini files

Contents of drvinfo.ini files	What it is
<pre>-----drvinfo.ini----- [Driver Information] Installation Mode = PnP Path = Parameters = Automatically Reboots = Reboot Required = PCIvenDevID = Order Before = Supported Systems = All ;Unsupported Systems = Supported Locales = -----end of drvinfo.ini-----</pre>	<p>A drvinfo.ini file for a Plug and Play device driver. It specifies the following information:</p> <ul style="list-style-type: none"> • The device driver is supported on all servers. • The device-driver installation files are located in the root of the device-driver directory.
<pre>-----drvinfo.ini----- [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 Locales = -----end of drvinfo.ini-----</pre>	<p>The drvinfo.ini file for a Plug and Play device driver. It specifies the following information:</p> <ul style="list-style-type: none"> • The device driver is supported on Machine Type 8673, Machine Type 8679, and Machine Type 8687 servers only. • The device-driver installation files are located in the win2000 subdirectory of the device-driver directory.

Table 16. Examples of drvinfo.ini files (continued)

Contents of drvinfo.ini files	What it is
<pre>-----drvinfo.ini----- [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 Locales = -----end of drvinfo.ini-----</pre>	<p>The drvinfo.ini file for a Plug and Play device driver. It specifies the following information:</p> <ul style="list-style-type: none"> • The device driver is supported on Machine Type 8673, Machine Type 8679, and Machine Type 8687 servers only. • The device driver installation files are located in the win2000 subdirectory of the device-driver directory. • The device driver is installed before any other Plug and Play device driver.
<pre>-----drvinfo.ini----- [Driver Information] Installation Mode = Executable Path = win2000\setup.exe Parameters = -Q -n Automatically Reboots = False Reboot Required = True PCIvenDevID = PCI\VEN_1002&DEV_5159&SUBSYS_029A1014 Order Before = All ;Supported Systems = Unsupported Systems = 8687 Supported Locales = ;Unsupported Locales = -----end of drvinfo.ini-----</pre>	<p>The drvinfo.ini file for an executable device driver. It specifies the following information:</p> <ul style="list-style-type: none"> • The installation file (setup.exe) is in the win2000 subdirectory of the device-driver directory. • The setup.exe file requires a -Q -n command-line parameter. • The device driver is not supported on a Machine Type 8687 server. • The device driver is installed only when the server contains the specified PCI adapter. • The device driver is installed before any other executable device drivers. • ServerGuide Scripting Toolkit will restart the server to complete the device-driver installation.

fdisk32.com

You can use the fdisk32 command to perform the following tasks:

- Create a primary or extended FAT32 partition
- Create a logical drive on an extended FAT32 partition
- Write a new master boot record
- Show the status of the hard disk drive

Syntax

The fdisk32 command uses the following syntax:

```
fdisk32 [drive] /status | drive /pri:size | drive /ext:size | drive /log:size
| drive /mbr]
```

where *drive* identifies the hard disk drive and *size* is the size of the partition or logical drive in MB.

The following table contains information about the fdisk32 optional parameters.

Table 17. fdisk32 optional parameters

Function	What it does	Syntax
Primary	Creates a primary DOS partition.	fdisk32 <i>drive</i> /pri: <i>size</i>
Extended	Creates an extended DOS partition.	fdisk32 <i>drive</i> /ext: <i>size</i>

Table 17. fdisk32 optional parameters (continued)

Function	What it does	Syntax
Logical drive	Creates a logical drive on the extended DOS partition.	fdisk32 <i>drive</i> /log: <i>size</i>
Master boot record	Writes a new master boot record.	fdisk32 <i>drive</i> /mbr
Status	Displays the status of the hard disk drive partition information.	fdisk32 [<i>drive</i>] /status where [<i>drive</i>] is an optional parameter that specifies the hard disk drive. If [<i>drive</i>] is not specified, the status of the hard disk drive partition information for all hard disk drives on the server is displayed.
Delete All	Deletes all the partitions on the specified drive.	fdisk32 <i>drive</i> /delete:all

Examples of using the fdisk32 command

The following table contains examples of using the fdisk32 command.

Table 18. Examples of using the fdisk32 command

Command	What It Does
fdisk32 1 /delete:all	Deletes all partitions on drive 1
fdisk32 1 /pri:3500	Creates a 3500 MB primary partition on drive 1
fdisk32 1 /ext:4000	Creates a 4000 MB primary partition on drive 1
fdisk32 1 /log:2000	Creates a 2000 MB logical drive in the extended partition on drive 1

findram.exe

You can use the findram command to determine the drive letter of the RAM disk image.

The findram command does not take parameters. The findram command uses the following syntax:

```
findram
```

The findram command returns a number (3-26), which corresponds to the drive letter (c-z). If a RAM disk image is not found, the findram command returns the value 27. The findram.bat file checks the value returned from the findram command and sets the RAMDSK environment variable accordingly.

Note: The findram command was tested on IBM PC DOS 7.1. The findram command will not identify the drive letter of a RAM disk image created with another version of DOS.

format32.com

You can use the format32 command to format a FAT32 disk partition.

Syntax

The format32 command uses the following syntax:

```
format32 drive: [/v:label] [/q] [/autotest]
```

where *drive* is the drive letter of the hard disk drive.

The following table contains information about the functions of the format32 command.

Table 19. format32 optional parameters

Function	What it does	Syntax
Volume	Assigns a volume label to the DOS drive	format32 drive: /v:label where <i>label</i> is the volume label and a maximum of 11 characters
Quick format	Deletes the file allocation table (FAT) and the root directory of a previously formatted hard disk but does not scan the hard disk for bad areas	format32 drive: [/q]
No prompt	Formats the partitions without prompting for input	format32 drive: [/autotest]

Example of using the format32 command

Issuing the following command does a quick format of drive C and sets the volume label of the partition to win2000 without being prompted to do so:

```
format32 /v:win2000 c: /q /autotest
```

hwdetect.exe

You can use the hwdetect command to perform the following tasks:

- Scan a server and detect information about the hardware and firmware
- Query information about a server and any PCI adapters that are installed

Syntax

The hwdetect command uses the following syntax:

```
hwdetect parameters
```

The following table contains information about the hardware scan functions of the hwdetect command.

Table 20. hwdetect: Hardware scan functions

Function	What it does	Syntax
Scan	Determines whether a system is an IBM @server or xSeries server. It returns a value of 0 if the scanned system is an IBM server.	hwdetect /s

Table 20. *hwdetect*: Hardware scan functions (continued)

Function	What it does	Syntax
Keyword format	Specifies that the hardware information is displayed in a keyword=value format.	<code>hwdetect /p [>filename]</code> where <i>filename</i> is an optional parameter that specifies the fully qualified file name.
INI file format	Specifies that the hardware information is displayed in an INI file format.	<code>hwdetect /i [>filename]</code> where <i>filename</i> is an optional parameter that specifies the fully qualified file name.
MachineType	Compares the machine type of the server to a specified machine type. It returns a value of 1 if the machine types match; it returns a value of 0 if the machine types do not match or cannot be compared.	<code>hwdetect /m:type</code> where <i>type</i> is the system type that you want to check for, for example, 8676 for an xSeries 335, Machine Type 8676 server.

The following table contains information about the PCI-adapter-detection functions of the *hwdetect* command.

Table 21. *hwdetect*: PCI adapter detection functions

Function	What it does	Syntax
Vendor ID	Determines whether the server contains a PCI adapter with the specified vendor ID. It returns the number of PCI adapters that have the specified vendor ID. It returns 0 if no such adapters are found.	<code>hwdetect /vid:vendor_id [/bn:bus_number] [/dn:device_number]</code> where: <ul style="list-style-type: none"> <i>vendor_id</i> is the vendor ID in hexadecimal format. <code>[/bn:bus_number]</code> is an optional command that specifies the bus number at which the scan starts, and <i>bus_number</i> is the bus number at which you want the PCI scan to start. It can be in hexadecimal or decimal format. <code>[/dn:device_number]</code> is an optional parameter that specifies the device number at which the scan starts, and <i>device_number</i> is the device number at which you want the PCI scan to start. It can be in hexadecimal or decimal format. Note: This parameter can be used only with the <code>[/bn:bus_number]</code> parameter.
Device ID	Determines whether the server contains a PCI adapter with the specified device ID. It returns the number of PCI adapters that have the specified device ID. It returns 0 if no such adapters are found.	<code>hwdetect /did:device_id [/bn:bus_number] [/dn:device_number]</code> where: <ul style="list-style-type: none"> <i>device_id</i> is the device ID in hexadecimal format. <code>[/bn:bus_number]</code> is an optional command that specifies the bus number at which the scan starts, and <i>bus_number</i> is the bus number at which you want the PCI scan to start. It can be in hexadecimal or decimal format. <code>[/dn:device_number]</code> is an optional parameter that specifies the device number at which the scan starts, and <i>device_number</i> is the device number at which you want the PCI scan to start. It can be in hexadecimal or decimal format. Note: This parameter can be used only with the <code>[/bn:bus_number]</code> parameter.

Table 21. *hwdetect*: PCI adapter detection functions (continued)

Function	What it does	Syntax
Subvendor ID	<p>Determines whether the server contains a PCI adapter with the specified subvendor ID.</p> <p>It returns the number of PCI adapters that have the specified subvendor ID. It returns 0 if no such adapters are found.</p>	<p><code>hwdetect /svid:sub_vendor_id [/bn:bus_number] [/dn:device_number]</code></p> <p>where:</p> <ul style="list-style-type: none"> • <i>sub_vendor_id</i> is the subvendor ID in hexadecimal format. • <code>[/bn:bus_number]</code> is an optional command that specifies the bus number at which the scan starts, and <i>bus_number</i> is the bus number at which you want the PCI scan to start. It can be in hexadecimal or decimal format. • <code>[/dn:device_number]</code> is an optional parameter that specifies the device number at which the scan starts, and <i>device_number</i> is the device number at which you want the PCI scan to start. It can be in hexadecimal or decimal format. <p>Note: This parameter can be used only with the <code>[/bn:bus_number]</code> parameter.</p>
Subdevice ID	<p>Determines whether the server contains a PCI adapter with the specified subdevice ID.</p> <p>It returns the number of PCI adapters that have the specified subdevice ID. It returns 0 if no such adapters are found.</p>	<p><code>hwdetect /sdid:sub_device_id [/bn:bus_number] [/dn:device_number]</code></p> <p>where:</p> <ul style="list-style-type: none"> • <i>sub_vendor_id</i> is the subdevice ID in hexadecimal format. • <code>[/bn:bus_number]</code> is an optional command that specifies the bus number at which the scan starts, and <i>bus_number</i> is the bus number at which you want the PCI scan to start. It can be in hexadecimal or decimal format. • <code>[/dn:device_number]</code> is an optional parameter that specifies the device number at which the scan starts, and <i>device_number</i> is the device number at which you want the PCI scan to start. It can be in hexadecimal or decimal format. <p>Note: This parameter can be used only with the <code>[/bn:bus_number]</code> parameter.</p>
Add number	<p>Adds the specified integer to the return value. This is useful if you want to use multiple <i>hwdetect</i> queries to obtain a sum total of multiple PCI adapters with different PCI IDs or subsystem IDs.</p>	<p><code>hwdetect /add:number OtherParameters</code></p> <p>where:</p> <ul style="list-style-type: none"> • <i>number</i> is a positive integer. • <i>OtherParameters</i> is one or more of the following parameters: <ul style="list-style-type: none"> – <code>/vid:vendor_id</code> – <code>/did:device_id</code> – <code>/svid:sub_vendor_id</code> – <code>/sdid:sub_device_id</code>

Examples of batch files that use the hardware scan functions

The following table contains examples of batch files that use the hardware scan functions of the hwdetect command.

Table 22. Examples of batch files using the hwdetect hardware scan functions

Content of batch files	What it does
<pre>----- myscript.bat ----- hwdetect /m:8676 if errorlevel 1 echo The machine type is 8676. hwdetect /m:8669 if errorlevel 1 echo The machine type is 8669.</pre>	<p>Uses the hwdetect command to determine if the machine type of a server is either 8676 or 8669, and prints the result to the screen if a match is found.</p>
<pre>----- myscript.bat ----- hwdetect /i > hwdetect.out clini hwdetect.out /s:PCI /i:Vendor_ID.0 /e:Vendor cliniset.bat</pre>	<p>Uses the hwdetect command to return the hardware information in INI file format.</p> <p>Uses the clini command to set an environment variable (Vendor) equal to the vendor ID of the first PCI device on the bus.</p>

Examples of using the hwdetect PCI adapter detection functions

The following table contains batch files that use the PCI-adapter-detection functions of the hwdetect command.

Table 23. Examples of batch files using the PCI adapter detection functions

Contents of batch file	What it does
<pre>hwdetect /vid:0x9005 /did:0x0250 if errorlevel 1 echo There are one or more ServeRAID-6M/6i controllers</pre>	<p>Determines whether the server contains one or more IBM ServeRAID-6M/6i controllers. If it does, the code echoes the included statement.</p>
<pre>hwdetect /vid:0x9005 /did:0x0250 if errorlevel 0 set TOTAL=0 if errorlevel 1 set TOTAL=1 if errorlevel 2 set TOTAL=2 if errorlevel 3 set TOTAL=3 hwdetect /add:%TOTAL% /vid:0x1014 /did:0x01BD if errorlevel 0 set TOTAL=0 if errorlevel 1 set TOTAL=1 if errorlevel 2 set TOTAL=2 if errorlevel 3 set TOTAL=3 if errorlevel 4 set TOTAL=4 if errorlevel 5 set TOTAL=5 if errorlevel 6 set TOTAL=6 echo There are %TOTAL% IBM ServeRAID adapters in this system.</pre>	<p>Determines the total number of IBM ServeRAID-4x, IBM ServeRAID-5x, and IBM ServeRAID-6x controllers in a server.</p> <p>Note: This example assumes the following conditions:</p> <ul style="list-style-type: none"> • There are no more than three IBM ServeRAID-6x controllers. • The total of IBM ServeRAID controllers is not more than six.

Examples of hardware information returned by the hwdetect command

The following table contains examples of the hardware information returned by the hwdetect command.

Table 24. Examples of hardware information returned by the hwdetect command

Command	Hardware information displayed	Notes
hwdetect /i	<pre>[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 [PCI] Bus_Number.0 = 0 Device_Number.0 = 1 Class_Code.0 = 0 Revision.0 = 0 Header_Type.0 = 0 Vendor_ID.0 = 5333 Subvendor_ID.0 = 1014 Subdevice_ID.0 = 01C5 Bus_Number.1 = 0 Device_Number.1 = 2 Class_Code.1 = 0 Revision.1 = 0 Header_Type.1 = 0 Vendor_ID.1 = 8086 Device_ID.1 = 1229 Subvendor_ID.1 = 1014 Subdevice_ID.1 = 105C Bus_Number.n = ...</pre>	<p>All values are either in hexadecimal or decimal format.</p> <p>The following keywords return values in hexadecimal format:</p> <ul style="list-style-type: none"> Enclosure_Type Processor_Family Class_Code Revision Header_Type Vendor_ID Device_ID Subvendor_ID Subdevice_ID <p>In addition, the hexadecimal values for Enclosure_Type and Processor_Family are defined further in the SMBIOS, version 2.3.2 specification.</p> <p>The following keywords return values in decimal format:</p> <ul style="list-style-type: none"> Number_Of_Enclosures Processor_Slots Active_Processors Processor_Speed_MHz Total_Enabled_Memory_Mb Bus_Number Device_Number
hwdetect /p	<pre>System_Machine_Type = 8674 System_Model_Number = 42X System_Serial_Number = 78Z9506 ... PCI_Bus_Number.0 = 0 PCI_Device_Number.0 = 1</pre>	

hypercfg.exe

You can use the hypercfg command to perform the following tasks:

- Configure an IDE RAID controller
- List the configuration of hard disk drives attached to an IDE RAID controller and write the information to a file
- List RAID BIOS information

Syntax

The hypercfg command uses the following syntax:

`hypercfg command parameter`

The following table contains information about the hypercfg commands and parameters.

Table 25. hypercfg commands and parameters

Function	What it does	Syntax
Create	Creates a logical drive and assigns it a RAID level.	<code>hypercfg configuration</code> where <i>configuration</i> is one of the following commands: <ul style="list-style-type: none">• <code>/AM</code>, which assigns the logical drive RAID level-1.• <code>/ASStripeSize</code>, which assigns the logical drive RAID level-0 and stripes data across the logical drive in blocks of <i>StripeSize</i>. <i>StripeSize</i> is one of the following values: 32, 64, 128, 256, 512, 1024, 2048, 4096.
File	Creates a logical drive using configuration information stored in a file.	<code>hypercfg @FileName</code> where <i>FileName</i> is a fully qualified file name.
List	Shows the configuration for the hard disk drives attached to the IDE RAID controller. It also can write the information to a file.	<code>hypercfg /l [/f filename]</code> where <code>[/f filename]</code> is an optional parameter that specifies that the information be written to a file, and <i>filename</i> is a fully qualified file name.
BIOS	Shows the BIOS information for the IDE RAID controller.	<code>hypercfg /v</code>
Silent	Performs operations without waiting for input.	<code>hypercfg /s</code>

Examples of using the hypercfg command

The following table includes examples of using the hypercfg command.

Table 26. Examples of using the hypercfg command

Command	What it does
<code>hypercfg /AM</code>	Creates a logical drive and assigns it RAID level-1.
<code>hypercfg /AS64</code>	Creates a logical drive and assigns it RAID level-0. Stripes data across the drives in 64 KB blocks.
<code>hypercfg /l</code>	Lists information for the two physical drives attached to the IDE RAID controller.
<code>hypercfg /l /f hypercfg.cfg /s</code>	Lists the information for the two physical drives attached to the IDE RAID controller; then, without waiting for input, writes the data to the hypercfg.cfg file.
<code>hypercfg /@hypercfg.cfg</code>	Configures a logical drive using the information stored in the hypercfg.cfg file.

For more information, see the *IBM BladeCenter HS20 - LSI IDEal RAID User's Guide*. It can be downloaded from <http://www.ibm.com/pc/support/site.wss/document.do?Indocid=MIGR-52750>.

ipssend.exe

You can use the ipssend command to perform the following tasks on an IBM ServeRAID controller:

- Back up controller information
- Create logical drives
- List information about a controller, logical drive, or physical drive
- Restore a controller configuration from a file or diskette
- Reset a controller to the factory-default settings

Notes:

1. The ServerGuide Scripting Toolkit includes ipssend.exe, version 6.10.24. Upgrade ServeRAID BIOS code and firmware to version 6.10.24 before using the ipssend command.
2. Version 6.10.24 of the ipssend command supports the following ServeRAID controllers:
 - ServeRAID-4H
 - ServeRAID-4M
 - ServeRAID-4L
 - ServeRAID-4Mx
 - ServeRAID-4Lx
 - ServeRAID-5i
 - ServeRAID-6M
 - ServeRAID-6i
3. You must load the DOS-based ServeRAID device driver (ipstraspi.sys) before using ipssend. You can load ipstraspi.sys by running the loadraid.bat file, which is located in the \stkfiles\sg_stk\examples directory.

For more thorough documentation of the ipssend command, see the *IBM ServeRAID Software User's Guide*.

Syntax

The ipssend utility uses the following syntax:

`ipssend command parameters`

The following table contains information about ipssend commands and parameters.

Table 27. ipssend commands and parameters

Function	What it does	Syntax
Back up	Saves information concerning a controller configuration to a diskette or physical drive. This information includes BIOS settings, array and logical drive definitions, and cluster parameters.	<code>ipssend backup <i>controller filename</i> [noprompt]</code> where: <ul style="list-style-type: none">• <i>controller</i> is the ServeRAID controller number (1-12).• <i>filename</i> is a fully qualified file name.• <i>noprompt</i> is an optional parameter that overrides the user prompt.

Table 27. *ipssend* commands and parameters (continued)

Function	What it does	Syntax
Create	<p>Creates logical drives on either an existing or a new array. To define a new array, type the channel and SCSI ID of the physical drives.</p> <p>Note: You cannot create RAID level-00, level-10, level-1E0, and level-50 logical drives with this function.</p> <p>After the logical drive is created, a quick initialization is done. The quick initialization process initializes the first 1024 sectors of the logical drive.</p> <p>If you assign the logical drive RAID level-5 RAID level-5E and the ServeRAID controller supports auto-synchronization, the logical drive is automatically synchronized in the background. If the controller does not support the auto-synchronization feature, you can use the IPSSEND synch function to synchronize the RAID level-5 or RAID level-5E logical drives. After synchronization, the logical drives are available for storing data.</p>	<p><code>ipssend create <i>controller</i> logicaldrive <i>options</i> size <i>raidlevel</i> [<i>channel sid</i>] [noprompt]</code></p> <p>where:</p> <ul style="list-style-type: none"> • <i>controller</i> is the ServeRAID controller number (1-12). • <i>options</i> is one of the following values: <ul style="list-style-type: none"> – NEWARRAY, which specifies that you are creating a new array and logical drive – ARRAYID, which is the array ID (A-H) of an existing array • <i>size</i> is one of the following values: <ul style="list-style-type: none"> – The size, in MB, of the logical drive – MAX, which indicates that you want to use all available space in the array • <i>raidlevel</i> is the RAID level for the logical drive (0, 1, 1E, 5, or 5E). • If you specify NEWARRAY, <i>channel sid</i> is the channel number for the device (1-4), and <i>sid</i> is the SCSI ID for the device (0-15). You can specify as many as 16 physical drives. • <i>noprompt</i> is an optional parameter that overrides the user prompt.
Get configuration	<p>Lists information about a controller, logical drive, or physical drive. This information potentially includes (but is not limited to) the following items:</p> <ul style="list-style-type: none"> • Controller type • BIOS code, boot block, device driver, and firmware versions • Logical drive status, RAID level, and size • Physical drive type, SCSI ID, presence of Predictive Failure Analysis® (PFA) • Physical drive state 	<p><code>ipssend getconfig <i>controller</i> <i>options</i></code></p> <p>where:</p> <ul style="list-style-type: none"> • <i>controller</i> is the ServeRAID controller number (1-12). • <i>options</i> is one of the following values: <ul style="list-style-type: none"> – AD, which specifies the controller information. – LD, which specifies the logical drive information. – PD, which specifies the physical device information. – AL, which specifies all information. This is the default option.
Get status	<p>Shows the status of the current or most recent rebuild, synchronize, or logical drive migration. Information includes logical drive in progress (or most recently completed), remaining size, and percentage completed.</p>	<p><code>ipssend getstatus <i>controller</i></code></p> <p>where <i>controller</i> is the ServeRAID controller number (1-12).</p>
Restore	<p>Loads a configuration file from a file stored on a diskette or hard disk drive. This information overwrites the existing configuration information and BIOS settings stored on a ServeRAID controller.</p>	<p><code>ipssend restore <i>controller</i> <i>filename</i> [noprompt]</code></p> <p>where:</p> <ul style="list-style-type: none"> • <i>controller</i> is the ServeRAID controller number (1-12). • <i>filename</i> is a fully qualified file name. • <i>noprompt</i> is an optional parameter that overrides the user prompt.

Table 27. *ipssend* commands and parameters (continued)

Function	What it does	Syntax
Set configuration	Modifies the controller configuration. You can restore the configuration to the factory-default settings, or you can copy the configuration from the attached physical drives.	<p><code>ipssend setconfig controller <i>option</i> [noprompt]</code></p> <p>where:</p> <ul style="list-style-type: none"> <i>controller</i> is the ServeRAID controller number (1-12). <i>option</i> is one of the following values: <ul style="list-style-type: none"> DEFAULT, which restores the controller to the factory-default settings. All logical drives are deleted. IMPORTDRIVE, which imports the configuration from the attached physical drives. <code>noprompt</code> is an optional parameter that overrides the user prompt.
Set state	Changes the state of a physical drive.	<p><code>ipssend setstate controller channel scsi_id new_state</code></p> <p>where:</p> <ul style="list-style-type: none"> <i>controller</i> is the ServeRAID controller number (1-12). <i>channel</i> is the channel number of the physical drive (1-3). <i>scsi_id</i> is the SCSI ID number of the physical drive (0-15). <i>new_state</i> is one of the following values: <ul style="list-style-type: none"> EMP (empty) RDY (ready) HSP (hot spare) SHS (standby hot spare) DDD (defunct disk drive) DHS (defunct hot spare) RBL (rebuild) SBY (standby) ONL (online)

Examples of using ipssend

The following table contains examples of using the ipssend command.

Table 28. Examples of using the IPSSEND command

Command	What it does
<pre>ipssend create 1 logicaldrive newarray max 5 1 0 1 1 1 2 1 3 1 4 noprompt</pre> <p>Note: This example is shown with a line break after <code>max</code>. In your file, the entire command must be all on one line.</p>	<p>Uses controller 1 to create a new array with a RAID level-5 logical drive. The array uses the drives on channel 1 at SCSI ID 0, 1, 2, 3, and 4. The logical drive uses all the available space in the array.</p>
<pre>ipssend backup 1 c:\myraid.cfg</pre>	<p>Saves the configuration information for controller 1 to the <code>myraid.cfg</code> file, located in the root directory of drive C.</p>

Table 28. Examples of using the IPSSSEND command (continued)

Command	What it does
ipssend restore 1 c:\myraid.cfg	Restores the configuration information from the myraid.cfg file to controller 1.
ipssend setconfig 1 default	Restores controller 1 to the factory-default settings.

Management Processor Command-Line Interface (MPCLI)

You can use the Management Processor Command-Line Interface (MPCLI) program to configure the following IBM service processors:

- Advanced System Management processor (ASM processor)
- Advanced System Management PCI Adapter (ASM PCI adapter)
- Integrated system management processor (ISMP)
- Remote Supervisor Adapter
- Remote Supervisor Adapter II

You can download the MPCLI program and integrate it into a Windows deployment scenario. You can install the MPCLI program on the target server after the operating system is installed; then, you can run a MPCLI command to configure the management processor.

You can download the MPCLI utility and its documentation from the IBM Support Web site at <http://www.ibm.com/pc/support/>.

Note: The MPCLI installation program requires Windows 2000 Server, Service Pack 2 or later. The MPCLI installation program is not supported on Windows 2000 Advanced Server or Windows Server 2003.

The following table contains information about the MPCLI commands that you might use with the ServerGuide Scripting Toolkit.

Table 29. MPCLI commands

Function	What it does	Syntax
Unattended installation	Performs an unattended installation of MPCLI on a system running Windows. MPCLI always is installed to the <i>d</i> :\Program Files\IBM\MPCLI directory, where <i>d</i> is the drive letter of the hard disk drive.	<i>PackageName.exe /s /v/qn</i> where <i>PackageName</i> is the file name of the MPCLI installation package.
Input file	Specifies a script file that contains the settings that MPCLI applies to the service processor.	<i>mpcli inputfile ScriptFile</i> where <i>ScriptFile</i> is the fully qualified file name of a script file. The script file must contain logon information and MPCLI commands, followed by an exit or restart command. See the <i>IBM Management Processor Command-Line Interface Version 2.0 User's Guide</i> for more information.

ramdisk2.exe

You can use the ramdisk2 command to determine the drive letter of the RAM disk image. The ramdisk2 command does not take command-line parameters.

The ramdisk2 command uses the following syntax:

```
ramdisk2
```

The ramdisk2 command creates a batch file named ramdisk.bat in the current working directory. When the ramdisk.bat file is run it sets an environment variable named ramdisk to the value of the RAM drive, for example, c:.

reboot.com

You can use the reboot command to restart (reboot) a server. The reboot command does not take command-line parameters.

The reboot command uses the following syntax:

```
reboot
```

savestat.exe

You can use the savestat command to perform the following tasks:

- Store as many as five integer values to persistent storage in the server CMOS memory
- Retrieve stored values from the persistent storage in the server CMOS memory
- Resets or clears the persistent storage in the server CMOS memory

Syntax

The savestat command uses the following syntax:

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

The following table contains information about the savestat parameters.

Table 30. savestat parameters

Function	What it does	Syntax
Set	Sets a persistent data-storage location to a value. It returns 0 if the operation is successful, and it returns 255 if the operation fails.	savestat /setn=value where: <ul style="list-style-type: none">• <i>n</i> is an integer between 1 and 5.• <i>value</i> is an integer between 0 and 254.
Get	Retrieves the value stored in a persistent data-storage location. It returns the current value stored in the persistent data-storage location. It returns 255 if the operation fails.	savestat /getn where <i>n</i> is an integer between 1 and 5.
Reset	Resets all persistent data-storage locations to the default value (0). It returns 0 if the operation is successful, and it returns 255 if the operation fails.	savestat /reset

Examples of using the savestat command

The following table contains examples of using the savestat command.

Table 31. Examples of using the savestat command

Command or batch file	What it does
savestat /set2=100	Sets the second persistent data-storage location to 100.
<pre> ----- myscript.bat ----- savestat /get2 if errorlevel 101 goto LEVEL101 if errorlevel 100 goto LEVEL100 goto END :LEVEL101 echo Savestat says byte 2 contains the number 101. goto END :LEVEL100 echo Savestat says byte 2 contains the number 100. goto END :END ---- end of myscript.bat----- </pre>	Retrieves the value stored in the second persistent data-storage location, and echoes a statement to the screen if the value is either 100 or 101.
savestat /reset	Resets all persistent data storage locations to zero.

scrub3.exe

You can use the scrub3 command to securely erase data from one or more hard disk drives.

Syntax

The scrub3 command uses the following syntax:

```
scrub3 /d=drive /l=level | /w=number
```

where *drive* is one of the following values:

- ALL to erase all hard disk drives
- *n* to erase the *n*th hard disk drive, where *n* is a positive integer

The following table contains information about the scrub3 optional parameters.

Table 32. scrub3 optional parameters

Function	What it does	Syntax
Security	<p>Specifies the security level of the disposal operation.</p> <p>The following security levels are supported:</p> <ul style="list-style-type: none"> • Limited security: The master boot record and some sectors are overwritten once. • Medium security: All sectors are overwritten once. • High security: All sectors are overwritten four times. • U.S. Department of Defense-compliant security: All sectors are overwritten seven times. 	<pre>scrub3 /d=drive /l=level</pre> <p>where: <i>level</i> is one of the following values:</p> <ul style="list-style-type: none"> • 1 (limited security) • 2 (medium security) • 3 (high security) • 4 (U.S. Department of Defense-compliant security)

Table 32. *scrub3* optional parameters (continued)

Function	What it does	Syntax
Write	Specifies the number of times each sector is overwritten.	<code>scrub3 /d=drive /w=number</code> where <i>number</i> is a positive integer.

Examples of using the scrub3 command

The following table contains examples of using the scrub3 command.

Table 33. Examples of using the scrub3 command

Command	What it does
<code>scrub3 /d=all /l=1</code>	Erases data from all the hard disk drives. Because the security level is set to 1, the master boot record and some sectors are overwritten once.
<code>scrub3 /d=1 /w=2</code>	Erases data from the first drive by overwriting each sector on the drive two times.

unattend.exe

You can use the unattend command to perform the following tasks:

- Determine which system-specific device drivers need to be installed
- Dynamically add device-driver-specific information to the answer file for the unattended installation

The unattend command takes information from files (`drvinfo.ini` and `hwdetect.ini`) that are generated by running the `dscan` and `hwdetect` commands. The unattend command automatically determines which device drivers need to be installed, based on the system-hardware information and the installation mode of the device driver; then, it adds the device-driver information to the answer file.

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.

Notes:

1. If you use the unattend command, you do not need to manually add device-driver information to the answer file for the unattended installation.
2. The device drivers on the *ServerGuide Setup and Installation* CD are already configured for use with the unattend command.

Syntax

You can specify the parameters for the unattend utility from the command line, in an initialization file, or both. Information specified in the unattend initialization file is overwritten by any parameters that are issued from a command-line prompt.

The unattend command uses one of the following types of syntax:

- unattend [*INIfile*] /u:*AnswerFile* /d:*path* /h:*HwinfoFile* /i:*i386* [*Options*]
- unattend *INIfile* [/u:*AnswerFile*] [/d:*path*] [/h:*HwinfoFile*] [/i:*i386*] [*Options*]

where:

- *INIfile* is the fully qualified name of the unattend initialization file.
- *AnswerFile* is the fully qualified name of the answer file for the unattended installation.
- *path* is the fully qualified name of the device-driver directory. For text mode device drivers to be added to the answer file, either the path must contain \$oem\$\textmode or you must issue the optional /t switch.
- *HwinfoFile* is the fully qualified name of the hardware-information file. This file must be generated by the hwdetect.exe utility or formatted similarly.
- *i386* is the fully qualified name of the directory that contains the Windows installation files.

The following table contains information about the optional unattend parameters.

Table 34. Optional unattend parameters

Function	What it does	Syntax
Create	Creates a default unattend initialization file.	/c
System drive	Specifies the drive letter of the hard disk drive on which Windows is installed. By default, this is c.	/s: <i>Drive</i> where <i>Drive</i> is the drive letter of the hard disk drive on which Windows is installed.
Textmode	Specifies that only text mode device-driver information is added to the answer file for the unattended installation.	/t
Plug-and-play	Specifies that only Plug and Play device-driver information is added to the answer file for the unattended installation.	/p
Executable	Specifies that only executable device-driver information is added to the answer file for the unattended installation.	/e

Table 34. Optional unattend parameters (continued)

Function	What it does	Syntax
Verbose	Specifies the verbose level.	<i>/v:Number</i> where <i>Number</i> is an integer in the 0-5 range. 0 is quiet, 3 is the default, and 5 is the maximum.

The following table shows the return values and their meanings.

Table 35. Return values for the unattend command

Value	Meaning
0	Successful completion.
1	Syntax error.
2	Program error.
3	The destination is read-only.
4	No device-driver information files found.
5	The specified file name does not exist.
6	The specified path does not exist.

Examples of using the unattend command

The following table contains examples of using the unattend command.

Table 36. Examples of using the unattend command

Command	What it does
<pre>unattend /u:c:\unattend.txt /d:c:\w2\oem\$\1\drv /h:c:\hwdetect.ini /i:c:\i386</pre> <p>Note: This example is shown with line breaks. In your file, the entire command must be all on one line.</p>	<p>Specifies the location of following required files and directories:</p> <ul style="list-style-type: none"> • Answer file for the unattended installation • Device-driver directory • Hardware-information file • Windows installation files <p>Because the path of the device-driver directory contains \$oem\$, the text mode device-driver information is automatically added to the answer file for the unattended installation.</p>
<pre>unattend /u:c:\unattend.txt /d:c:\w2\oem\$\textmode /h:c:\hwdetect.ini /i:c:\i386 /t</pre> <p>Note: This example is shown with line breaks. In your file, the entire command must be all on one line.</p>	<p>Specifies the location of following required files and directories:</p> <ul style="list-style-type: none"> • Answer file for the unattended installation • Device-driver directory • Hardware-information file • Windows installation files <p>Because the /t parameter is issued, only the text mode device-driver information is added to the answer file for the unattended installation.</p>

Configuring an unattend initialization file

You can use an unattend initialization file to specify the parameters for the unattend command. This is especially useful for situations where the parameters might exceed 127 characters, the DOS limit for the number of characters that can be issued from a command-line prompt. An unattend installation file must be in INI file format and contain two sections: [Unattend] and [GuiRunOnce].

Information specified in the unattend initialization file is overwritten by any parameters that are issued from a command-line prompt.

The following table contains information about the keywords that can be used in the [Unattend] section of the unattend initialization file. This section specifies each of the parameters that the unattend command needs to add device-driver information to the answer file for the unattended installation.

Table 37. Variables used in the [Unattend] section of the unattend initialization file

Keyword	Values	What it does
UnattendTxt	The fully qualified path to the answer file for the unattended installation. This parameter is required.	Specifies the location of the answer file for the unattended installation.
Drivers Path	The fully qualified path to the device-drivers directory. Note: For text mode device drivers to be added to the answer file, one of the following conditions must be true: <ul style="list-style-type: none"> • The path of the device-drivers directory contains \$OEM\$/textmode. • The Textmode keyword is set to True. This parameter is required.	Specifies the location of the device-drivers directory.
HWDetectIni	The fully qualified path to the hardware-information file. This file either must be generated by the hwdetect.exe command or be formatted similarly. This parameter is required.	Specifies the location of the hardware-information file.
I386 Path	The fully qualified path of the i386 directory that contains the Windows installation files. This parameter is required.	Specifies the location of the Windows installation files.
System Drive	The drive letter of the hard disk drive where Windows is installed. By default, this is set to c:.	Specifies the drive letter of the hard disk drive where Windows is installed.
Textmode	<ul style="list-style-type: none"> • True • False 	Specifies that only text mode device-driver information is added to the answer file for the unattended installation.
PnP	<ul style="list-style-type: none"> • True • False 	Specifies that only Plug and Play device-driver information is added to the answer file for the unattended installation.

Table 37. Variables used in the [Unattend] section of the unattend initialization file (continued)

Keyword	Values	What it does
Executable	<ul style="list-style-type: none"> • True • False 	Specifies that only executable device-driver information is added to the answer file for the unattended installation.
Verbose Level	An integer in the 0-5 range. 0 is quiet, 3 is the default, and 5 is the maximum.	Specifies the verbose level.

The following table contains information about the keywords that can be used in the [GuiRunOnce] section of the unattend initialization file. This section specifies commands that are run after the operating system is installed. You can specify whether the commands are run before or after the executable device drivers are installed, because installing executable device drivers might require the server to be restarted several times.

Table 38. Variables used in the [GuiRunOnce] section of the unattend initialization file

Keyword	Values	What it does
Before Drivers	A comma-delimited list of text strings. These text strings must be variables used in the <i>Prefix_Command</i> keyword.	Specifies the commands that are run before the executable device drivers are installed.
After Drivers	A comma-delimited list of text strings. These text strings must be variables used in the <i>Prefix_Command</i> keyword.	Specifies the commands that are run after the executable device drivers are installed.
Prefix_Command where <i>Prefix</i> is a text string.	Commands. Notes: <ol style="list-style-type: none"> 1. Every variable used in a <i>Prefix_Command</i> keyword must be a value for either the Before Drivers or After Drivers keyword. 2. The unattend initialization file must assign a value to either <i>Prefix_Supported_Systems</i> or <i>Prefix_Unsupported_Systems</i> for every variable used in a <i>Prefix_Command</i> keyword. 	Specifies a command to run, either before or after the device drivers are installed. Notes: <ol style="list-style-type: none"> 1. The commands cannot restart (reboot) the server. 2. Be sure to run any interactive commands after the device drivers are installed.
Prefix_Supported_Systems where <i>Prefix</i> is a text string.	All, None, or a comma-delimited list of server machine types. Note: The <i>Prefix_Supported_Systems</i> keyword and the <i>Prefix_Unsupported_Systems</i> keywords cannot both be assigned values in the same initialization file.	Specifies the servers on which the command is run.
Prefix_Unsupported_Systems where <i>Prefix</i> is a text string.	All, None, or a comma-delimited list of server machine types. Note: The <i>Prefix_Supported_Systems</i> keyword and the <i>Prefix_Unsupported_Systems</i> keywords cannot both be assigned values in the same initialization file.	Specifies the servers on which the command is <i>not</i> run.

Examples of unattend initialization files

The following table contains examples of unattend initialization files.

Table 39. Examples of unattend initialization files

Content of unattend initialization file	What it does
<pre>-----unattend.ini----- [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 = -----end of unattend.ini-----</pre>	<p>Adds information concerning the following device drivers to the c:\unattend.txt file:</p> <ul style="list-style-type: none"> • Plug and Play device drivers and executable device drivers that are located in the c:\w2\%oem%\\$1\drv directory • Text mode device drivers that are located in the c:\w2\%oem%\textmode directory
<pre>-----unattend.ini----- [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 = -----end of unattend.ini-----</pre>	<p>Adds information concerning the text mode device drivers located in the c:\w2\%oem%\textmode directory to the c:\unattend.txt file.</p>
<pre>-----unattend.ini----- [Unattend] UnattendTxt = C:\unattend.txt Drivers Path = C:\w2\%oem%\\$1\drv HWDetectIni = C:\hwdetect.ini I386 Path = C:\i386 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 -----end of unattend.ini-----</pre>	<p>Adds information concerning the following device drivers to the c:\unattend.txt file:</p> <ul style="list-style-type: none"> • Plug and Play device drivers and executable device drivers that are located in the c:\w2\%oem%\\$1\drv directory • Text mode device drivers that are located in the c:\w2\%oem%\textmode directory <p>Specifies some additional user-added commands.</p>

Appendix A. Working with device drivers

The set of device drivers from the *ServerGuide Setup and Installation CD* are configured for use with the unattend command. To add or update device drivers, or to remove device drivers from the set, you must run a ServerGuide Scripting Toolkit process.

Adding or updating a device driver

Complete the following steps to add a device driver to or update a device driver in the set of device drivers from the *ServerGuide Setup and Installation CD*:

1. Obtain the new device driver. You can get device drivers from the IBM Support Web site at <http://www.ibm.com/pc/support/>, the latest version of the IBM UpdateXpress CD, or the other vendor manufacturer.
2. Complete one of the following actions:

If adding a device driver	Create a directory for the device driver below the <code>\\$oem\$\\$1\drv</code> directory. Note: Limit the length of directory names and paths when creating the new device-driver directory; DOS has a path limit of 127 characters.
If updating a device driver	Delete all files and subdirectories from the device-driver directory. Do not delete the empty device-driver directory.

3. Copy the device-driver files to the new or newly-emptied directory.

Note: Be sure to copy all files and maintain the directory structure.

4. From a command prompt on the source system, type the following command and press Enter:

```
\stkfiles\sg_stk\utils\dscan32.exe driverpath /s
```

where *driverpath* is the fully qualified name of a directory that contains device-driver directories, for example, `c:\src_tree\w2000drv\oem\$1\drv`. Issuing this command scans the device-driver set, determines the installation mode for the new or updated device driver, and creates a `drvinfo.ini` file for the new or updated device driver in the device-driver directory.

5. If necessary, edit the generated `drvinfo.ini` file and add device-driver-specific information, such as supported systems and command-line parameters that are necessary for executable device drivers. For more information, see “`drvinfo.ini` files” on page 50.
6. (If adding or updating a text mode device driver only) From a command prompt, type the following command and press Enter:

```
\stkfiles\sg_stk\utils\dscan32.exe driverpath /t
```

where *driverpath* is the fully qualified name of a directory that contains device-driver directories, for example, `c:\src_tree\w2000drv\oem\$1\drv`. Issuing this command deletes and then recreates both the textmode directory and the master `txtsetup.oem` file.

Removing a device driver

Complete the following steps to remove a device driver from the set of device drivers from the *ServerGuide Setup and Installation* CD:

1. Delete the device-driver directory and all files and subdirectories that it contains.
2. (Text mode device driver only) From a command prompt, type the following command and press Enter:

```
\stkfiles\sg_stk\utils\dscan32.exe %oem%\$1\drv /t
```

Issuing this command runs the dscan command against the device-driver set, recreates the %oem%\textmode directory, and recreates the master txtsetup.oem file.

Manually adding device-driver information to the answer file

The ServerGuide Scripting Toolkit installation process uses the unattend command to dynamically add server-specific device driver information to the answer file for an unattended installation. The device drivers on the *ServerGuide Setup and Installation* CD are already configured for use with the unattend command.

However, if you decide not to use the unattend command along with the configured device drivers from the *ServerGuide Setup and Installation* CD, you must manually add the device-driver information to the answer file before performing an unattended installation of Windows.

This section contains information about adding the following types of device drivers to the answer file:

- Text mode
- Custom hardware abstraction layers (HAL)
- Plug and Play
- Executable

Text mode device drivers

Complete the following steps to manually add text mode device drivers to the answer file:

1. In the [MassStorageDrivers] section, add the description of the device driver. Use the nomenclature specified by Microsoft for an answer file.

The following example shows the description of an Adaptec SCSI controller in the [MassStorageDrivers] section of an answer file.

```
[MassStorageDrivers]
"LSI Logic PCI SCSI/FC MPI Driver (Server 2003 32-bit)" = "OEM"
"IDE CD-ROM (ATAPI 1.2)/PCI IDE Controller" = "RETAIL"
```

2. Add the device-driver file names to the [OemBootFiles] section.

The following example shows the device-driver file names for an Adaptec SCSI controller in the [OemBootFiles] section of an answer file.

```
[OemBootFiles]
sympm.sys
sympm.inf
mpi2k332.cattxtsetup.oem
```

Custom HAL device drivers

Complete the following steps to manually add custom HAL device drivers to the answer file:

1. In the [Unattended] section, set ComputerType to the name of the device driver. Use the nomenclature specified by Microsoft for an answer file.

The following example shows the [Unattended] description section of an answer file prepared to install Windows 2000 on an xSeries 440 server:

```
[Unattended]
ComputerType = "IBM eServer xSeries 440/445 (Windows 2000 HAL)", OEM
```

2. Add the device-driver file names to the [OemBootFiles] section.

The following example shows the device-driver file names in the [OemBootFiles] section of an answer file prepared to install Windows 2000 on an xSeries 440 server:

```
[OemBootFiles]
halx44n.dll
halx44n.cat
halx44n.inf
txtsetup.oem
```

Plug and Play device drivers

Complete the following steps to manually add Plug and Play device drivers to the unattend.txt file:

1. In the [Unattended] section, set OemFilesPath to the fully qualified path of the \$oem\$ directory.
2. Set OemPnPDriversPath to a semicolon-delimited list of the directories that contain the INF files, as in the following example:

```
[Unattended]
OemFilesPath = C:\oem_path\$oem$
OemPnPDriversPath = drv\ACT;drv\ASF;drv\BC;drv\E1;drv\E2;drv\E3
```

Executable device drivers

Executable device drivers usually require command-line parameters to run in unattended mode. You must specify these device drivers and their command-line parameters in either the [GUIRunOnce] section of the answer file or in the cmdlines.txt file. Consult the Microsoft documentation for a full description of these two methods of running executable programs.

The following example illustrates how you can 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 runs.

The Microsoft unattended installation process automatically copies the device-driver files from the \$oem\$ directory structure to the root of the target server. The commands in the [GUIRunOnce] section must include paths to the location where Windows copies the device-driver files. For example, if the setup.exe file is located in C:\wininst\\$oem\$\\$1\drv\mydriver\, then the path in the command must be specified as C:\drv\mydriver\setup.exe.

Appendix B. Further automating the deployment process

This appendix contains information about additional ways you can automate the deployment process by:

- Incorporating BIOS code and firmware updates in the deployment scenarios
- Incorporating the MPCLI program and MPCLI commands in the deployment scenarios for Windows 2000 Server
- Dynamically updating the answer file for an unattended installation of Windows

Incorporating BIOS code and firmware updates in the deployment process

You can further automate the deployment process by incorporating BIOS code and firmware updates in the deployment scenarios. If you incorporate BIOS code and firmware updates in the deployment scenarios, you do not need to use the UpdateXpress CD during the deployment process.

Complete the following steps to incorporate BIOS code and firmware updates in a deployment scenario:

1. Obtain the latest BIOS code and firmware updates. They can be downloaded from the IBM Support Web site at <http://www.ibm.com/pc/support/> or extracted from an IBM UpdateXpress CD.

Complete the following steps to obtain BIOS code and firmware updates from an IBM UpdateXpress CD:

- a. On a system running Windows, open the index.htm file. This file is in the root directory of the UpdateXpress CD.

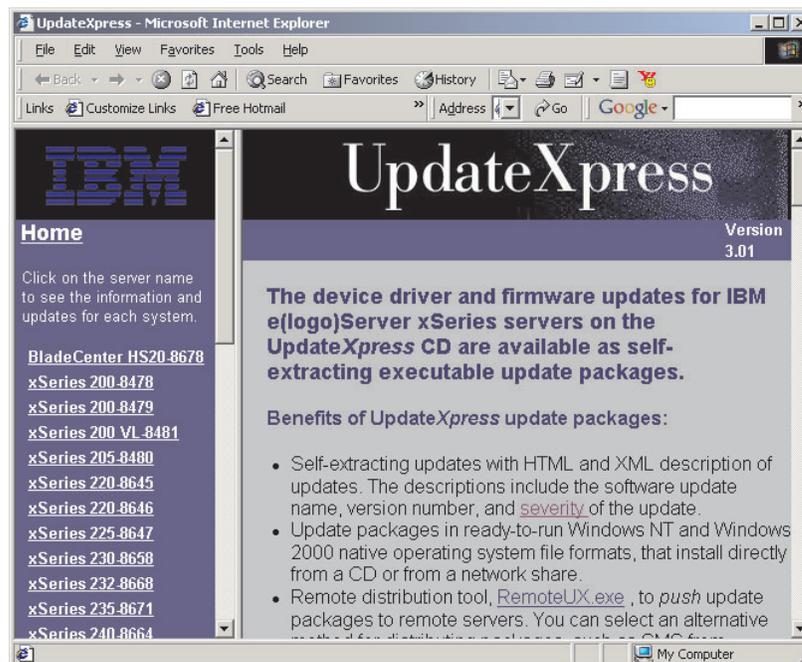


Figure 15. "UpdateXpress" window

- b. In the left pane, click the server for which you want to obtain BIOS code and firmware updates. The updates are displayed in the right pane.
- c. Select the updates and click **Download Now**. The “File Download” window opens.
- d. Click **Run this program from the current location** and click **OK**. The files are extracted, and the “IBM License Agreement” window opens.
- e. Click **Accept**. The “xSeries Firmware Update” window opens.

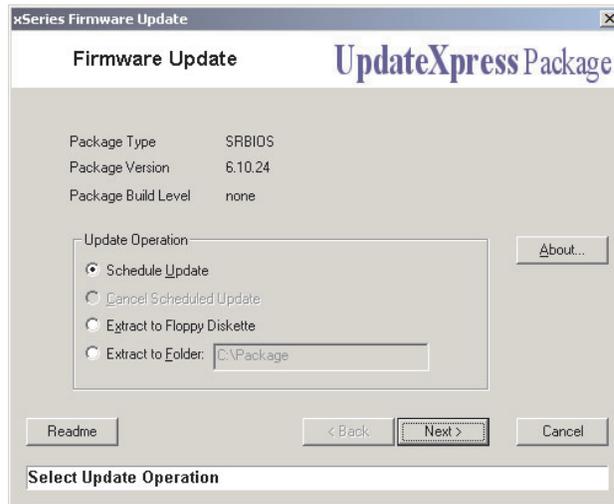


Figure 16. “xSeries Firmware Update” window

- f. Click **Extract to folder**, and type the fully qualified directory name.
 - g. Click **Next** twice; then, click **Finish** after the files are successfully extracted.
 - h. Repeat steps 1b through 1g for each IBM server for which you want to obtain BIOS code and firmware updates.
2. Add the BIOS code and firmware updates to the source tree:
 - a. Add a updates directory to the source tree.
 - b. Add subdirectories for each IBM server for which you have BIOS code and firmware updates. Use the machine type of the IBM server as the name of the subdirectory, for example, 8671 for the xSeries 235 server.
 - c. To the server-specific directories, add subdirectories for each update that you want to deploy, for example, BIOS code, and the applicable service processor update.

The updates portion of a source tree containing BIOS code and firmware updates for the xSeries 235 and xSeries 345 server might have the following structure:

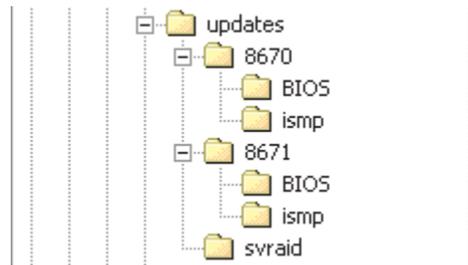


Figure 17. Updates directory of source tree displayed in Windows Explorer

- d. Add a svraid directory that contains ServeRAID BIOS code and firmware updates.
 - e. Copy the entire contents of the updates packages, including readme.txt files, into the applicable directory in the source tree.
3. Customize the usrvars.bat file for the deployment scenario:
- a. Open the usrvars.bat file in an ASCII text editor.
 - b. Modify the [Toolkit_Updates] section so that it contains the following commands:

```
SET DO_UPDATES=YES
SET UPDATES_PATH=UpdatesDirectory
```

where *UpdatesDirectory* is the directory that you created in 2a on page 80.

- c. Set DO_BIOS_UPDATES equal to YES if the deployment scenario includes BIOS code and firmware updates. By default, this variable is set to NO.
 - d. Set DO_ISMP_UPDATES equal to YES if the deployment scenario includes ISMP updates. By default, this variable is set to NO.
 - e. Set DO_SERVERAID_UPDATES equal to YES if the deployment scenario includes ServeRAID updates. By default, this variable is set to NO.
4. Configure the fwupdate.bat file:

- a. Determine the command that is needed to run the updates. This information is typically provided in the Unattended Mode section of the readme.txt file that comes with the update.

The following table contains examples of commands that must be issued to perform unattended installations of certain IBM updates.

For an IBM BIOS code update	<code>flash2.exe /u</code>
For an ISMP firmware update	<code>flash2.exe /u /e</code>
For a ServeRAID BIOS code and firmware update	<code>flashman.exe /cd:Files /autoexpress</code>

where *Files* is the fully qualified name of the directory that contains the update files.

- b. Make sure the fwupdate.bat file contains the necessary commands to call each update program. Also, make sure that any added or edited commands do not contain parameters or switches that force the server to restart. (The fwupdate.bat file is located in the \stkfiles\sg_stk\examples directory of the ServerGuide Scripting Toolkit.)

Notes:

- 1) Run BIOS code updates before any other updates.
- 2) If a server contains an ISMP and an optional service processor, update the firmware for the optional service processor (Advanced System Management PCI Adapter, Remote Supervisor Adapter, or Remote Supervisor II) before you update the firmware for the ISMP.
- 3) Many updates contain a flash2.exe file, which is usually specific to each update.
- 4) The /autoxpress flag indicates an unattended ServeRAID update. This flag is not documented in the readme.txt for this update.

Go to one of the deployment scenarios described in Chapter 3, “Using the ServerGuide Scripting Toolkit to deploy Windows”, on page 15 and Chapter 4, “Using the ServerGuide Scripting Toolkit to deploy Linux”, on page 23.

Incorporating MPCLI in the deployment process

You can use the Management Processor Command-Line Interface (MPCLI) program to configure IBM service processors. You can configure the deployment scenario to install the MPCLI program on the target server after the operating system is installed, and then run an MPCLI script file to configure the service processors.

Note: You can include MPCLI only in the deployment scenarios that install Windows 2000 Server with an integrated Service Pack 2 or later.

Complete the following steps to incorporate MPCLI in the deployment process:

1. Add a mpcli directory to the source tree.
2. Download the MPCLI Windows installation package from the IBM Support Web site at <http://www.ibm.com/pc/support/> and save it to the directory that you created in step 1.
3. Create an MPCLI script file. See the MPCLI documentation for information about MPCLI commands and syntax.

For example, a script designed to assign a host name and IP address to a Remote Supervisor Adapter II might include the following text:

```
logonip -hostname 192.168.0.10 -userid USERID -password PASSWORD
setip -interface 1 -hostname myrsaII
exit
```

4. Add the MPCLI script file to the mpcli directory that you created in step 1.
5. To copy the MPCLI program to the target server hard disk drive, add the following commands to the \stkfiles\sg_stk\examples\windows\instos.bat file:

```
mkdir %NOSDRV%\mpcli
copy %TKDRV%\mpcli\*. * %NOSDRV%\mpcli
```

6. Modify the [GUIRunOnce] section of the answer file for the unattended installation so that it includes the following text:

```
Command0="cmd.exe /c d:\mpcli\PackageName /s /v/qn"  
Command1="cmd.exe /c d:\Program Files\IBM\MPCLI\bin\mpcli inputfile Script"
```

where *d* is the drive letter of the hard disk drive, *PackageName* is the name of the MPCLI installation package, and *Script* is the fully qualified name of the MPCLI script file that you added to the source tree in step 4 on page 82.

This command installs the MPCLI program and runs the MPCLI script on the target server after the operating system is installed.

Dynamically updating the answer file for an unattended installation

You can use ServerGuide Scripting Toolkit commands to dynamically assign user information in the answer file for an unattended installation of Windows.

The following code example uses the `clini` and `hwdetect` commands to determine the server serial number, then assigns varying values to the `ComputerName` and `ProductID` keywords in the [UserData] section of the answer file.

```
@Echo off  
HWDetect /I > HWDetect.ini  
CLIni HWDetect.ini /S:System /I:Serial_Number /E:SerialNumber  
Call CLIniSet.bat  
If %SerialNumber%==78Z3210 goto Comp1  
If %SerialNumber%==78Z9507 goto Comp2  
Echo System not supported.  
Goto Finish  
:Comp1  
CLIni Unattend.txt /S:UserData /I:ComputerName /V:"Computer1"  
CLIni Unattend.txt /S:UserData /I:ProductID /V:11111-11111-11111-11111-11111  
Goto Finish  
:Comp2  
CLIni Unattend.txt /S:UserData /I:ComputerName /V:"Computer2"  
CLIni Unattend.txt /S:UserData /I:ProductID /V:22222-22222-22222-22222-22222  
Goto Finish  
:Finish
```

Add this code to the CUSTUNAT section of the `instos.bat` file. This file is located in the `\stkfiles\sg_stk\examples\windows` directory.

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.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 Integrated Technology 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 publication 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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. 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 publication 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 applicable 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

© COPYRIGHT INTERNATIONAL BUSINESS MACHINES CORPORATION, 2003.
All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights —
Use, duplication or disclosure is subject to restrictions set forth in GSA ADP
Schedule Contract with IBM Corp.

Trademarks

The following terms are trademarks of International Business Machines Corporation
in the United States, other countries, or both:

BladeCenter	ServeRAID
e-business logo	ServerGuide
@server	ServerProven
IBM	Update <i>Xpress</i>
IntelliStation	xSeries
Predictive Failure Analysis	

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

Other company, product, or service names may be trademarks or service marks of
others.

Index

A

- Advanced System Management PCI Adapter, configuring 65
- Advanced System Management processor, configuring 65
- altboot.exe
 - parameters 41
 - return values 42
 - syntax 41
 - tasks performed 41
 - using, examples of 42
- alternative parameters, syntax conventions 41
- answer file
 - dynamically adding information 68
 - dynamically updating 79
 - example of dynamically updating 83
 - manually adding device-driver information 76
- section
 - GUIRunOnce 77
 - MassStorageDrivers 76
 - OemBootFiles 76, 77
 - Unattended 77
 - UserData 83
- automating the deployment process 79

B

- BIOS code, obtaining 79
- boot sector information
 - backing up 41
 - restoring 41
- BOOTMEDIA_WRITEPROTECTED variable, setting 9, 18

C

- cfg1030.exe
 - commands and parameters 43
 - syntax 43
 - tasks performed 42
 - using, examples of 44
- CLIENT_NAME variable, setting 14
- clini.exe
 - optional parameters 45
 - return values 45
 - syntax 45
 - tasks performed 44
 - using, examples of 46
- clini32.exe
 - optional parameters 45
 - return values 45
 - syntax 45
 - tasks performed 44
 - using, examples of 46
- cliniset.bat file, definition 44
- CMOS memory
 - clearing persistent storage 66

- CMOS memory (*continued*)
 - retrieving values from 66
 - storing values in 66
- command-line syntax, conventions 41
- commands, syntax conventions 41
- custom HAL device drivers
 - automatically adding information to the answer file 68
 - manually adding information to the answer file 77
- CustomizeNetworkVariables.bat 14

D

- data CD nested directory depth limit, deployment scenarios
 - using a DOS-startable CD 18
 - using a DOS-startable diskette and data CD 16
- data, erasing 67
- default values, syntax conventions 41
- deployment scenarios 15
 - assumed ServerGuide Scripting Toolkit installation location 15
 - BIOS and firmware updates, incorporating 79
 - further automating 79
 - Linux, using a DOS-startable diskette and network share 24
 - MPCLI, incorporating 79
 - using a DOS-startable CD 17
 - using a DOS-startable CD, data CD nested directory depth limit 18
 - using a DOS-startable diskette and data CD 16
 - using a DOS-startable diskette and data CD, data CD nested directory depth limit 16
 - using a DOS-startable diskette and network share 19
 - using a Remote Supervisor Adapter II and network share 20
- detecting
 - firmware information 56
 - hardware information 56
 - PCI-adapter information 56
- device drivers
 - adding or updating 75
 - custom HAL
 - automatically adding information to the answer file 68
 - manually adding information to the answer file 77
 - determining which to install 68
 - dynamically adding information to the answer file 68
- executable
 - automatically adding information to the answer file 68
 - manually adding information to the answer file 77
- installation mode, determining 47
- obtaining 75

- device drivers *(continued)*
 - Plug and Play
 - automatically adding information to the answer file 68
 - manually adding information to the answer file 77
 - removing 76
 - scanning 47
 - text mode
 - automatically adding information to the answer file 68
 - manually adding information to the answer file 76
- device-driver directory, definition 47
- diskette drive, bypassing 41
- disposal scenarios
 - creating the Microsoft ADS server disposal diskette 27
 - creating the server disposal diskette 37
- DO_BIOS_UPDATES variable, setting 81
- DO_CLEAN_FIRST variable, setting 11
- DO_ISMP_UPDATES variable, setting 81
- DO_RAID_CFGS variable, setting 11
- DO_SERVERAID_UPDATES variable, setting 81
- DO_UPDATES variable, setting 81
- documentation
 - IBM BladeCenter HS20 - LSI IDEal RAID User's Guide 61
 - IBM ServeRAID Software User's Guide 62
- DOS command-line prompt limit 71
- DOS path limit 5, 44
- DOS write-protect error 41
- drvinfo.ini file
 - creating 47
 - definition 47
 - editing 75
 - examples of 53
 - keywords and values 51
 - modifying 50
 - purpose of 50
- drvutils directory 47
- dscan.exe
 - optional parameters 48
 - return values 49
 - syntax 48
 - tasks performed 47
 - using, examples of 49
- dscan32.exe
 - optional parameters 48
 - return values 49
 - syntax 48
 - tasks performed 47
 - using, examples of 49

E

- environment variable, setting 44
- erasing data securely 67
- examples
 - altboot commands 42
 - cfg1030 commands 44

- examples *(continued)*
 - clini commands 46
 - drvinfo.ini files 53
 - dscan commands 49
 - fdisk32 commands 55
 - format32 command 56
 - hwdetect commands 59
 - hypercfg commands 61
 - ipssend commands 64
 - kickstart file installation statements 23
 - savestat commands 67
 - scrub3 commands 68
 - unattend commands 70
 - unattend initialization files 73
- executable device drivers
 - automatically adding information to the answer file 68
 - manually adding information to the answer file 77

F

- factory-default settings, restoring to 62
- FAT32 disk, formatting 56
- FAT32 partition, creating 54
- fdisk32.com
 - optional parameters 54
 - replacing 3
 - syntax 54
 - tasks performed 54
 - using, examples of 55
- findram.exe
 - return values 55
 - syntax 55
 - tasks 55
- firmware information, detecting 56
- firmware updates, obtaining 79
- flash2.exe file 82
- format32.com
 - optional parameters 56
 - replacing 3
 - syntax 56
 - tasks performed 56
 - using, example of 56
- fwupdate.bat file, configuring 81

H

- hard disk drive
 - show status of 54
 - starting (booting) from 41
- hardware information, detecting 56
- Holdit.exe 47
- hot-spare drive, creating 42
- hwdetect.exe
 - hardware information returned 60
 - hardware scan functions 56
 - PCI-adaptor-detection functions 57
 - syntax 56
 - tasks performed 56
 - using, examples of 59
- hwdetect.ini 83

hypercfg.exe
 commands and parameters 61
 syntax 61
 tasks performed 60
 using, examples of 61

I

IBM BladeCenter HS20 - LSI IDEal RAID User's Guide 61
IBM ServeRAID controller, configuring 62
IBM ServeRAID Software User's Guide 62
IBM UpdateXpress
 BIOS code and firmware updates, obtaining 79
IBM Web sites
 ServerGuide x
 ServerProven x
 Support x
 Systems Management Software Download/Electronic Support x
 UpdateXpress x
 xSeries Systems Management x, xi
IDE RAID controller 25
 configuration, listing 60
 configuring 60
 configuring in usrvars.bat 12
IDE_CONFIGURE variable, setting 12
IDE_STRIPE_SIZE1 variable, setting 12
illustration
 source tree 7
 source tree with updates directory 81
INI files, modifying 44
INITRD_FILE variable, setting 10
integrated SCSI controller with RAID capabilities, configuring 12, 42
Integrated system management processor, configuring 65
ipraspi.sys 62
ipssend.exe
 ipraspi.sys 62
 syntax 62
 tasks performed 62
 upgrading BIOS and firmware code 62
 using, examples of 64

K

KERNEL_FILE variable, setting 10
kickstart file
 enabling DHCP for use of non-local kickstart file 11
 examples of installation statements 23
 preparing 23
 sample file location 23
 section, Installation Media Configuration 23
KS_DEVICE variable, setting 11
KS_FILENAME variable, setting 10
KS_LOCATION variable, setting 11

L

LOADLIN_PATH variable, setting 11
logical drive, creating 42, 62

M

MakeADSDisposalToolkitDisk.bat 28
MakeADSToolkitHardwareConfig.bat 30
MakeDisposalToolkitDisk.bat 37
MakeLinuxNetworkInstallToolkitDisk.bat 25
MakeWindowsCDInstallToolkitDisk.bat 16, 17
MakeWindowsNetworkInstallToolkitDisk.bat 19, 21
Management Processor Command-Line Interface
 commands 65
 configuring service processors 82
 downloading 65, 82
 incorporating in deployment scenarios 79
 installing on target server 82
 prerequisites 65
 script file, creating 82
 tasks performed 65
master boot record, writing 54
master txtsetup.oem file, creating 47
MPCLI
 commands 65
 configuring service processors 82
 downloading 65, 82
 incorporating in deployment scenarios 79
 installing on target server 82
 prerequisites 65
 script file, creating 82
 tasks performed 65

N

netvars.bat 21
 customizing 13, 14
 section
 Bootable_Media_Information 13
 Source_System_Information 14
 stored user name and password 14, 19
 variable
 CLIENT_NAME 14
 PWD 14
 SERVER_GATEWAY 14
 SERVER_IP 11, 14
 SERVER_NAME 14
 SERVER_SHARE 14
 SERVER_SUBNETMASK 14
 TARGET_IP 13
 TARGET_NAME 13
network.ini file 14
NOSDRV variable, setting 10
NOSTYPE variable, setting 9

O

optional commands, syntax conventions 41
OS_PATH variable, setting 9, 10

P

- PARTITION_SIZE variable, setting 10
- PCI-adapter information, detecting 56
- Plug and Play device drivers
 - automatically adding information to the answer file 68
 - manually adding information to the answer file 77
- protocol.ini file 14
- PWD variable, setting 14

R

- RAID BIOS information, listing 60
- RAID configuration, backing up 42, 62
- RAID controller information, generating 42
- RAID level, assigning 42
- RAM disk image, determining drive letter of 55, 66
- RAMDISK_SIZE variable, setting 11
- ramdisk2.exe
 - syntax 66
 - tasks performed 66
- Reboot.exe 47
- Remote Supervisor Adapter II, configuring 65
- Remote Supervisor Adapter, configuring 65
- repeatable parameters, syntax conventions 41
- restarting (rebooting) a server 66
- return values
 - altboot.exe 42
 - clini.exe 45
 - clini32.exe 45
 - dscan.exe 49
 - dscan32.exe 49
 - findram.exe 55
- root directory of ServerGuide Scripting Toolkit 5
- RSA_INSTALL variable, setting 9, 21
- RUN_ALTBOOT variable, setting 9

S

- SaveRSABootDisk.bat 21
- savestat.exe
 - parameters 66
 - syntax 66
 - tasks performed 66
 - using, examples of 67
- scanning a server 56
- SCRUB_DRIVE variable, setting 38
- SCRUB_SECURITY_LEVEL variable, setting 38
- scrub3.exe
 - optional parameters 67
 - syntax 67
 - tasks performed 67
 - using, examples of 68
- SERVER_GATEWAY variable, setting 14
- SERVER_IP variable, setting 11, 14
- SERVER_NAME variable, setting 14
- SERVER_SHARE variable, setting 14
- SERVER_SUBNETMASK variable, setting 14
- ServerRAID controller, usrvars.bat variable 11
- service processors, configuring 65

- SET_UPDATES_PATH variable, setting 81
- SGDD_SRC_DRV_PATH variable, setting 10
- SGDD_SRC_TEXTMODE_PATH variable, setting 10
- SGDD_TARGET_PATH variable, setting 10
- source tree 5
 - BIOS code and firmware updates 80
 - configuration 6
 - configure directory as read-only 19, 20
 - contents 5
 - illustration 7
 - location
 - HTTP-based installation 6
 - shared network directory 5
 - mpcli directory, adding 82
 - updates directory (illustration) 81
- stkfiles directory 5
- syntax
 - altboot.exe 41
 - cfg1030.exe 43
 - fdisk32.com 54
 - hypercfg.exe 61
 - ipssend.exe 62
- syntax conventions 41

T

- TARGET_IP variable, setting 13
- TARGET_NAME variable, setting 13
- terminology
 - device-drivers directory 47
 - drvinfo.ini file 47
 - master txtsetup.oem file 47
 - text mode directory 47
- text mode device drivers
 - automatically adding information to the answer file 68
 - manually adding information to the answer file 76
- text mode directory, creating 47
- TKDIR variable, setting 9
- trademarks 88

U

- unattend initialization file
 - configuring 71
 - examples of 73
 - format of 71
 - GuiRunOnce section keywords 72
 - Unattend section keywords 71
- UNATTEND_DRV variable, setting 9
- UNATTEND_FILENAME variable, setting 10
- UNATTEND_PATH variable, setting 10
- unattend.exe
 - optional parameters 69
 - syntax 69
 - tasks performed 68
 - using, examples of 70
- unattend.txt file 10, 83
 - dynamically updating 79
 - example of dynamically updating 83
 - keyword, ProductID 15

- unattend.txt file (*continued*)
 - manually adding device-driver information 76
 - preparing 15
 - section
 - GUIRunOnce 77
 - MassStorageDrivers 76
 - OemBootFiles 76, 77
 - Unattended 77
 - UserData section 15, 83
- UpdateXpress 79
- usrvars.bat 38
 - BIOS code and firmware updates, adding 81
 - configuring IDE RAID controllers 12
 - configuring integrated SCSI controllers with RAID capabilities 12
 - configuring ServerRAID controllers 11
 - customizing 8
 - section
 - General_Toolkit 8, 9
 - Linux_Install 10
 - OS_Install 8
 - RAID_Configuration 8, 11
 - RAID_Configuration_CFG1030 12
 - RAID_Configuration_HYPERCFG 12
 - RAID_Configuration_IPSSSEND 11
 - Toolkit_Updates 8, 9, 81
 - Windows_Install 9
 - variable
 - BOOTMEDIA_WRITEPROTECTED 9, 18
 - DO_BIOS_UPDATES 81
 - DO_CLEAN_FIRST 11
 - DO_ISMP_UPDATES 81
 - DO_RAID_CFGS 11
 - DO_SERVERAID_UPDATES 81
 - DO_UPDATES 81
 - IDE_CONFIGURE 12
 - IDE_STRIPESIZE1 12
 - INITRD_FILE 10
 - KERNEL_FILE 10
 - KS_DEVICE 11
 - KS_FILENAME 10
 - KS_LOCATION 11
 - LOADLIN_PATH 11
 - NOSDRV 10
 - NOSTYPE 9
 - OS_PATH 9, 10
 - PARTITION_SIZE 10
 - RAMDISK_SIZE 11
 - RSA_INSTALL 9, 21
 - RUN_ALTBOOT 9
 - SCRUB_DRIVE 38
 - SCRUB_SECURITY_LEVEL 38
 - SET_UPDATES_PATH 81
 - SGDD_SRC_DRV_PATH 10
 - SGDD_SRC_TEXTMODE_PATH 10
 - SGDD_TARGET_PATH 10
 - TKDIR 9
 - UNATTEND_DRV 9
 - UNATTEND_FILENAME 10
 - UNATTEND_PATH 10

V

variables, syntax conventions 41

W

Web sites

- IBM ServerGuide x
- IBM ServerProven x
- IBM Support x
- IBM Systems Management Software
 - Download/Electronic Support x
- IBM UpdateXpress x
- IBM xSeries Systems Management x, xi



Part Number: 90P2888

Printed in U.S.A.

SC90-P288-80



(1P) P/N: 90P2888

