IBM ServerGuide Scripting Toolkit, Linux Edition

Altiris Deployment Solution User's Reference

Version 9.6

IBM ServerGuide Scripting Toolkit, Linux Edition

Altiris Deployment Solution User's Reference

Version 9.6

Note:
Before using this information and the product it supports, read the information in "Notices".

Contents

Chapter 1. Introducing the IBM ServerGuide Scripting Toolkit, Linux Edition for Altiris Deployment Solution

The IBM ServerGuide Scripting Toolkit, Linux Edition for Altiris Deployment Solution (Linux Scripting Toolkit) enables you to tailor and build custom hardware deployment solutions. With it, you can configure hardware and install the Linux operating system (OS) for IBM System x, BladeCenter, and iDataPlex hardware.

When integrated with the Altiris Deployment Solution, the Linux Scripting Toolkit simplifies the steps for creating and customizing jobs to deploy hardware configurations and Network Operating System (NOS) Network Operating System (NOS)NOSdeployments. The Linux Scripting Toolkit supports the following types of deployment:

- Policy-based RAID configuration using PRAID
- Cloning of RAID configuration
- Configuration of system settings through the Advanced Settings Utility (ASU)
- Configuration of Fibre Channel Host Bus Adapters (HBAs)
- Firmware updates through the UpdateXpress System Pack Installer (UXSPi)
- UpdateXpress System Pack installation integrated with automated deployment of a Network Operating System (NOS)
- IBM Systems Director Agent installation integrated with automated deployment of a NOS
- Automated deployment of the following Network Operating Systems (NOSs):
  - SUSE Linux Enterprise Server 9 32 bit SP4
  - SUSE Linux Enterprise Server 9 x64 SP4
  - SUSE Linux Enterprise Server 10 32 bit SP1/SP2/SP3/SP4
  - SUSE Linux Enterprise Server 10 x64 SP1/SP2/SP3/SP4
  - SUSE Linux Enterprise Server 11 32 bit Base/SP1/SP2/SP3
  - SUSE Linux Enterprise Server 11 x64 Base/SP1/SP2/SP3
  - Red Hat Enterprise Linux 4 AS/ES 32 bit U6/U7/U8
  - Red Hat Enterprise Linux 4 AS/ES x64 U6/U7/U8
  - Red Hat Enterprise Linux 5 32 bit U1/U2/U3/U4/U5/U6/U7/U8/U9/U10
  - Red Hat Enterprise Linux 5 x64 U1/U2/U3/U4/U5/U6/U7/U8/U9/U10
  - Red Hat Enterprise Linux 6 32 bit U1/U2/U3/U4/U5
  - Red Hat Enterprise Linux 6 x64 U1/U2/U3/U4/U5
  - Red Hat Enterprise Linux 7 x64 Base
  - VMware ESX Server 3.5 U4/U5
  - VMware ESX Server 4.0/4.0u1/4.0u2/4.1/4.1u1/4.1u2/4.1u3/4.1u4
- Automated post-installation deployment of the following NOSs:
  - Red Hat Enterprise Linux 4 AS/ES 32 bit U9
  - Red Hat Enterprise Linux 4 AS/ES x64 U9
- Automated deployment of the following NOSs in Native UEFI mode:
  - SUSE Linux Enterprise Server 11 SP1/SP2/SP3
  - Red Hat Enterprise Linux 6 x64 U1/U2/U3/U4/U5
  - Red Hat Enterprise Linux 7 x64 Base

- Remote Supervisor Adapter II (RSA II) and BladeCenter Management Module and Advanced Management Module remote disk scenarios
- Integration with Altiris Deployment Solution
- Cloning of NOS images
- Installation of IBM Systems Director Agent integrated with scripted NOS deployment
- Post UXSP installation using the Altiris Agent
- Post IBM Systems Director Agent installation using the Altiris Agent
- Remote deployment via the Integrated Management Module (IMM)

The toolkit provides sample jobs for Altiris Deployment Solution for hardware configuration and operating system deployment using Linux PE. The Linux Scripting Toolkit supports Altiris Deployment Solution 6.9 SP1-SP5 and the Hewlett-Packard Rapid Deployment Pack (RDP) versions 3.80, 3.81, 3.82, 3.83, 6.0, and 6.2.

Chapter 2. Installing the IBM ServerGuide Scripting Toolkit, Linux Edition

This section describes how to install the Linux Scripting Toolkit. It includes information about the prerequisites, instructions for installation, and instructions for removal.

Prerequisites

The ServerGuide Scripting Toolkit, Linux Edition for Altiris Deployment Solution has three components:

1. The IBM Pre-boot environment
2. The IBM Scripting Toolkit Scripts and Utilities
3. The IBM Scripting Toolkit Altiris jobs within the Altiris Deployment Solution console

When you install or reinstall the ServerGuide Scripting Toolkit, Linux Edition for Altiris Deployment Solution, these components are updated as a group. After the update, the Altiris Deployment Solution jobs associated with the previous version of the ServerGuide Scripting Toolkit, Linux Edition for Altiris Deployment Solution no longer work.

To install and create the IBM Linux x64 PE environment, the Linux Scripting Toolkit requires the following components:

- Altiris Linux Automation Environment for Deployment Solution

  `BDCgpl_6.9.`*`xxxx`*`.frm`, where *xxxx* is the version number
- IBM Linux Pre-installation Environment boot files for x64

  `ibm_utl_boot_tools-`*`xxx`*`_anyos_x64-full.zip` where *xxx* is the version number.

The Linux Scripting Toolkit also requires that the target server on which the deployment is to be executed has at least 1 GB of memory.

Extracting the Linux Scripting Toolkit

This section describes the steps for extracting the Linux Scripting Toolkit zip file into the Altiris Deployment Server folder.

Procedure

1. Right-click the Linux Scripting Toolkit zip archive, and select **Extract All**.
2. Extract the files to the Altiris Deployment Server directory. By default this directory is `C:\Program Files\Altiris\eXpress\Deployment Server`.
3. Verify that the Linux Scripting Toolkit is extracted. If the files extracted properly, they are located in `...\Altiris\eXpress\Deployment Server\sgdeploy\sgtklinux\`

Adding the Altiris Linux x64 Automation Environment

This section describes how to add the Altiris Linux Automation Environment for Linux x64 if it is not already installed.

Before you begin

Acquire the Linux Automation Environment file. You can acquire this file from the Altiris Deployment Solution or from the Hewlett-Packard Rapid Deployment Pack.

You can get the Altiris Deployment Solution file from the Altiris web site. The file name is `BDCgpl_6.9.xxxx.frm` where *xxxx* is the version number of the file.

The Hewlett-Packard Rapid Deployment Pack version is located on the DVD at: `\ds\BDCgpl_6.9.xxxx.frm`, where *xxxx* is the version number of the file.

About this task

Procedure

There are two ways to install the Altiris Linux x64 Automation Environment:

- Manually: Use the Altiris PXE Configuration Utility to install the pre-boot environment manually.
- Automatically, as part of installation: Copy the file `BDCgpl_6.9.xxxx.frm` to `...\sgdeploy\sgtklinux\boot\`, and retain the original file name. When the **`altiris_sgtklinux.cmd`** installation script runs, it integrates the environment as part of the installation.

Adding the IBM Linux Pre-installation Environment for x64

This topic describes the process for acquiring the IBM Linux Pre-installation Environment files when the Altiris Deployment Server does not have Internet connectivity.

Before you begin

This procedure requires access to the Internet.

About this task

If the Altiris Deployment Server is connected to the Internet and can access the IBM home page at http://www.ibm.com, the installation process downloads the IBM Linux PE .zip file.If the Deployment Server does not have Internet connectivity; however, you must obtain the files first by following the instructions below.

Procedure

1. Copy the file `sgdeploy\sgtklinux\altiris\bin\win32\ibm_utl_bomc_x.xx_windows_i386.exe` from the Altiris Deployment Server to a system connected to the Internet.
2. Run the following command to acquire the pre-installation environment .zip file and save it in `C:\temp\`:

   `ibm_utl_bomc_x.xx_windows_i386.exe --function=linuxtk -l C:\temp`

   The Bootable Media Creator tool acquires the pre-installation environment .zip file `ibm_utl_boot_tools-140_anyos_x86-64_full.zip` and saves it in `C:\temp\` on the workstation.

3. Copy `C:\temp\ibm_utl_boot_tools-140_anyos_x86-64-full.zip` to the Altiris Deployment Server at `..\sgdeploy\sgtklinux\boot\`, preserving the original file name.

Adding System Enablement Packs (SEPs)

Through System Enablement Packs (SEPs), you can add support for hardware released after the current release of the ServerGuide Scripting Toolkit, Linux Edition for Altiris Deployment Solution. This section describes the process for adding SEPs to the Altiris Deployment Server.

Before you begin

If the Altiris Deployment Server is connected to the Internet and can access the IBM website, use the **`acquire_seps.cmd`** command to acquire SEPs as described here. If the Altiris Deployment Server does not have Internet connectivity, see "Adding a new supported system using SEPs without Internet connectivity" for information about using a system other than the Altiris Deployment Server to acquire SEPs.

Procedure

1. Edit the file `acquire_seps.ini` and modify the default settings for acquiring SEPs as necessary. The default location of this file is `C:\Program Files\Altiris\eXpress\Deployment Server\sgdeploy\sgtklinux\altiris\acquire_seps.ini`. The following table describes the values you can modify.

Value	Description
TK_SEP_OS	A comma-separated list of operating systems for which you can download SEPs. Valid values are: *none, all.* The default value is *none*.
TK_SEP_Arch	A comma-separated list of processor architectures for which you can download SEPs. Valid values are: *x86, x64, all*.
TK_SEP_PKG_TGT	The target directory on the local system for storing SEPs.
TK_SEPTOOL_OptionalParams	Additional parameters to be used when downloading SEPs.

2. Open a command window and change to the directory containing the **`acquire_seps.cmd`** file. By default, this file is located at `C:\Program Files\Altiris\eXpress\Deployment Server\sgdeploy\sgtklinux\altiris\acquire_seps.cmd`.
3. Enter the **`acquire_seps.cmd`** command, followed by a comma-delimited list of machine types, for example, `acquire_seps.cmd 7979,4199`. You can also acquire SEPs for all available machine types by specifying *all* instead of a machine type, for example: `acquire_seps.cmd all`.

Adding a new supported system using SEPs without Internet connectivity

With System Enablement Packs (SEPs) you can add support for hardware released after the current release of the IBM ServerGuide Scripting Toolkit, Linux Edition.

This section describes the process for adding SEPs to the Toolkit Source Server when the server does not have Internet connectivity.

Before you begin

If the Linux Scripting Toolkit Source Server is not connected to the Internet, complete the following steps to acquire SEPs.

Procedure

1. Copy the file `septool .zip` file (`win_septool`*xxx*`.zip`, where *xxx* is the version number of the tool) from the Toolkit Source Server to a system with Internet connectivity. The default location for this file is:

   `/opt/ibm/sgtk/wui/bin/windows/`
2. On the system where you copied the .zip file, extract all of the files in the archive.
3. From the directory where you extracted the .zip file, run the following command to acquire the SEP and save it in `C:\temp`:

   `septool`*xxx*`.exe acquire -l C:\temp -m` *machine_type* `-o none -a x64`

   where *xxx* is the version of the `septool` utility and *machine_type* is the machine type of the system for which you want to download SEPs.
4. Copy the files from the `C:\temp` folder and place them in the updates folder of the Linux Scripting Toolkit directory tree. The default location is `/opt/ibm/sgtk/sgdeploy/updates/uxsp`.

Integrating with Altiris Deployment Solution

This section describes the requirements and method for integrating the Linux Scripting Toolkit into your existing Altiris Deployment Solution console.

Installation

To integrate the Linux Scripting Toolkit into the Altiris Deployment Solution console, follow these steps:

1. To suppress prompts or user settings before installing the Scripting Toolkit into the Altiris Deployment Solution console, open the altiris_sgtklinux.ini file and modify the appropriate values shown in the table below.

Value	Description
Section: [altiris_SGTKLinux Build Settings]	
TK_Build_SuppressPrompts	Suppresses interactive prompts that pause the process so that user settings can be read after they are displayed. Valid values are: • Yes • No **Note:** Interactive prompts regarding required removal of jobs during upgrade or reinstallation scenarios are always displayed.
TK_Build_DisplayBuildSettings	Displays the user settings before installing the Toolkit into Altiris Deployment Solution. Valid values are: • Yes • No

2. Open a command window and change the current directory to the directory containing the altiris_sgtklinux.cmd file. The default file location is `C:\Program Files\Altiris\eXpress\Deployment Server\sgdeploy\sgtklinux\altiris\altiris_sgtklinux.cmd`.
3. At the command window enter the **altiris_sgtklinux.cmd** command. This command processes the Toolkit settings in the altiris_sgtklinux.ini file.

   This process can take between 5 and 15 minutes to complete depending on the server setup.
4. When the altiris_sgtklinux.cmd command completes, restart the server and all Altiris services to complete the integration.

After the Toolkit is installed, there is a new folder with the product name, version, and build ID under the Jobs frame in the Altiris Deployment Solution console. For example: **IBM Scripting Toolkit, Windows Edition 9.6**

Reinstalling the Linux Scripting Toolkit

This section describes the process for reinstalling the Linux Scripting Toolkit.

The ServerGuide Scripting Toolkit, Linux Edition for Altiris Deployment Solution has three components:

- The IBM Pre-boot environment
- The IBM Scripting Toolkit Scripts and Utilities
- The IBM Scripting Toolkit Altiris jobs within the Altiris Deployment Solution console

When you either install or re-install the ServerGuide Scripting Toolkit, Linux Edition for Altiris Deployment Solution, these components are updated as a group. After the update, the Altiris Deployment Solutionjobs associated with the previous version of the ServerGuide Scripting Toolkit, Linux Edition for Altiris Deployment Solution no longer works.

To re-install the Linux Scripting Toolkit, follow these steps:

1. Follow the instructions from the "Installation" on page 6 section of "Integrating with Altiris Deployment Solution" on page 6. The installer displays the following message to indicate that a previous installation has been detected:

```
The following installed items were detected:

IBM Scripting Toolkit, Linux Edition 9.6

This must be uninstalled before this process can continue. Please select
an option below:

1) Exit The Reinstaller.
2) Automatically Reinstall The Scripting Toolkit, Linux Edition.

Select an above option:
```

2. Remove the Linux Scripting Toolkit either manually or automatically by following the instructions in the appropriate section below.

Manually removing the Linux Scripting Toolkit

To manually remove the Linux Scripting Toolkit, complete these steps:

1. Select option number 1 from the prompt to exit the installation process.

2. Remove all jobs associated with the IBM® Linux Toolkit image in the Altiris Deployment Solution console.
3. Open the PXE Configuration utility from the **Tools** menu within the Altiris Deployment Solution Console.
4. Select the IBM Linux Toolkit image.
5. Verify that the column labeled **In Use By DS** does not indicate **Yes** for that image. If it does, there are additional jobs associated with that image. Find and remove them.
6. Click the **Delete** button to remove the PXE image.
7. Click **Save** to save the changes
8. Click **OK** to exit the PXE Configuration utility.

Automatically removing the Linux Scripting Toolkit

To automatically remove the Linux Scripting Toolkit, follow these steps:

1. Select option 2 from the prompt. The following message is displayed:

```
*******************************************************************************
You have chosen to reinstall the following:

IBM Scripting Toolkit, Linux Edition 9.6

You may keep the jobs associated with the currently installed
Scripting Toolkit, however, they will be moved to a folder labeled
IBM Scripting Tooklit, Linux Edition (Archived w/ver. 9.6).
*******************************************************************************

Would you like to keep those jobs?<Y/N>
```

2. Select either Y or N, depending on what you want to do with the jobs: Select **Y** to remove the current Linux PE image and move the current IBM ServerGuide Scripting Toolkit, Linux Edition jobs to a folder named `IBM ServerGuide Toolkit, Linux Edition (Archive)`. Select **N** to remove all previously installed jobs, with the exception of user-created jobs.

Chapter 3. Adding and sharing content on the Linux Scripting Toolkit source server

This section describes how to add to the source server and share content on your network with the source server.

You can add the following types of files to the source server:

UpdateXpress System Pack (UXSP) files
to the `sgdeploy/updates/usxp/` directory.

Application files, such as the IBM Systems Director Agent
to the `sgdeploy/apps/` directory.

Operating system files
to the `sgdeploy/os/` directory.

The remaining topics in this section describe the process for adding these files and for sharing them across the network.

Adding UpdateXpress System Packs (UXSPs)

To add UpdateXpress System Packs (UXSPs) to the source server, you must add them to `sgdeploy\updates\uxsp` directory on the source tree.

About this task

Follow this procedure to add UXSPs to the source server:

Procedure

1. Start the UpdateXpress System Pack Installer, ibm_utl_uxspi_x.xx_winsrvr_32-64.exe. The default location is `C:\Program Files\Altiris\eXpress\Deployment Server\sgdeploy\updates\uxsp\`.
2. Use the UpdateXpress System Pack Installer to download the updates for the machine types and operating systems you need into the source tree. Download the files to `sgdeploy/updates/uxsp`.

   Note: Do not manually create directories under `sgdeploy/updates/uxsp/`. Doing so prevents the UpdateXpress System Pack Installer from locating the update files.
3. If you are prompted to overwrite existing files, click **OK**.

What to do next

If the Altiris Deployment Server is not connected to the Internet, you can download the updates to a machine connected to the Internet, and copy them to the `sgdeploy/updates/uxsp/` directory.

Adding device drivers

To add extra device drivers, which are required for some options on the target server, into deployment media, you must first import them into the Altiris Deployment Server.

Procedure

1. Start the Toolkit Configuration Utility, **tkconfig.exe**. The default location is `C:\Program Files\Altiris\eXpress\Deployment Server\sgdeploy\sgtklinux\tkconfig\tkconfig.exe`.
2. Select **Add Operating System Device Drivers** from the task list. The wizard starts.
3. Select the machine type and OS type that you want to deploy, then click **Next**.
4. Download the device driver package from the link to a local path.
   - If the wizard displays `Don't need extra driver packages for the Machine Type and OS type`, go to step 7.
   - If the deployment scenario needs extra device drivers, the driver package download link is shown on the wizard. There are two types of device drivers, optional and mandatory. You can ignore the optional device drivers and go to step 7. You must, however, import the mandatory drivers.
5. Select the location of the package.
6. Click **Next** to import required device drivers to the Altiris Deployment Server.
7. If desired, repeat the above steps to import extra device drivers for other deployment scenarios.
8. Click **Finish** to exit the wizard.

Adding Systems Director Agent files

To include the IBM Systems Director Agent in a deployment, you must first add the files to the Altiris Deployment Server by following this process.

Procedure

1. Download the latest version of the IBM Systems Director Agent installation files from http://www.ibm.com/systems/management/director/downloads.html.
2. Unzip the files in a convenient location.
3. Start the Toolkit Configuration Utility, **tkconfig.exe**. The default location is `C:\Program Files\Altiris\eXpress\Deployment Server\sgdeploy\sgtklinux\tkconfig\tkconfig.exe`.
4. Select **Add Operating System Application Files** from the task list. The Operating System Application files wizard starts.
5. Select **IBM Director 6.x or IBM Director 6.x.x for Linux**.
6. Select **Core Services Agent (Level 1) or Common Agent (Level 2)**.
7. Browse to the location where you unzipped the Director Agent files, and click **Next** to begin copying the files.
8. When the copy is complete, exit the wizard.

What to do next

You can now include installation of the IBM Systems Director Agent in your deployments.

Adding operating system installation files

To include operating system files in your deployment image, you must make the files accessible to the deployment server. If the operating system files you want to use in your deployments cannot be accessed by the deployment server, you can copy the files to the server by following the steps provided in this topic.

Before you begin

If the operating system files you want to use in your deployments are already in a location accessible to the deployment server, you do not have to add them to the deployment server itself.

Note: The Linux Scripting Toolkit supports only Service Pack 4 for SuSE Linux Enterprise Server (SLES) 9. To add Service Pack 4 to the source server, you must first add all CDs of base SLES 9. After adding these files, you can add the service pack by following the same procedure. When you add the service pack, you must select the same target directory you used for the base.

About this task

If the operating system files are not already in a location accessible to the deployment server, add them to the server by following these steps.

Procedure

1. Start the Toolkit Configuration Utility, **tkconfig.exe**. The default location is `C:\Program Files\Altiris\eXpress\Deployment Server\sgdeploy\sgtklinux\tkconfig\tkconfig.exe`.
2. Select **Add Operating System Application Files** from the task list. The Operating System Application files wizard starts.
3. Select the operating system you want to add from the top panel.
4. Select the version of the operating system to import from the lower panel.
5. Select the location of the files.
6. Click **Next** to copy the files to the source server.
7. For each operating system that you want to add, repeat steps 3 through 6.

Sharing operating system files on your network

The operating system files you are using for your deployments must be shared over the network to be accessible to the target server during deployment. Follow these steps to share the files.

About this task

To complete a deployment, the operating system files used for the deployment must be shared with the target system by using one of the following protocols:

- FTP
- HTTP
- NFS

To share the files from the Altiris Deployment Server, you must configure the server to share the `sgdeploy/` directory.

The operating system installation jobs provided by the Linux Scripting Toolkit use the FTP protocol. Follow the steps below to configure the Altiris Deployment Server running Windows Internet Information Server (IIS) to share the `sgdeploy/` directory.

Note: Consult the IIS documentation for more options and security considerations.

Procedure

1. Start the Internet Information Services Manager.
2. Select **Default FTP Site**.
3. Right click the target server, and select **New** > **Virtual Directory...**.
4. Set the alias name to **`sgdeploy`**.
5. Set the home directory to `C:\Program Files\Altiris\Deployment Server\`.
6. Allow Read access.
7. Start the FTP Site Service.

What to do next

You can now use the operating system files in the `sgdeploy` directory in your deployments.

Chapter 4. Quick start deployment scenarios

This section contains basic information about deployment scenarios to help you begin using the Linux Scripting Toolkit with Altiris Deployment Solution as quickly as possible.

The Linux Scripting Toolkit provides a collection of sample jobs for use by the Altiris Deployment Solution. The following topics provide additional information about the most commonly used jobs, such as those that configure Fibre, RAID, and system settings.

For more information about the sample jobs provided by the Scripting Toolkit, see "Sample job definitions" on page 29

The Linux Scripting Toolkit for use with Altiris Deployment Solution provides two types of jobs:

- "Modular deployment tools" on page 14, which are designed to be used individually to complete common deployment tasks, such as configuring pre-installation hardware and installing an operating system
- "Deployment solutions" on page 21, which are designed to consolidate multiple deployment elements in a single Altiris job, such as scripted deployment and image-based cloning

For more information about tailoring deployments to your needs, see Chapter 5, "Customizing deployment scenarios," on page 23.

Validating the Altiris environment

After installing the Linux Scripting Toolkit for use with Altiris Deployment Solution, you should validate that the environment is operating properly. This topic describes the process for doing so, using the sample job provided.

Before you begin

Complete the installation method appropriate for your environment as described in Chapter 2, "Installing the IBM ServerGuide Scripting Toolkit, Linux Edition," on page 3.

About this task

When the installation steps are complete:

Procedure

1. Open the Altiris Deployment Solution console.
2. Run the job **Linux Pre-Boot Test**, located in the Pre-Boot OS Connectivity Test folder, in the **IBM Scripting Toolkit, Linux Edition 9.6** folder. The job boots into the Linux PE x64 environment on the target server. After connectivity is established, it performs a **`wait`** task.

Results

If the task is successful, you can then right-click on the computer icon and select **Power Option** > **Shutdown**. Execution of any subsequent job will use Wake on LAN to wake the system.

Modular deployment tools

The sample jobs provided in the **Modular Deployment Tools** folder are designed to be used individually to complete common deployment tasks. This section describes how and in what order to use the modular deployment tools.

The modular deployment tools provided by the Linux Scripting Toolkit are each designed to complete one step in the deployment process. The following sections will provide information on using the sample jobs for:

- Pre-installation hardware configuration
- Operating system installation
- Post-installation configuration

Pre-installation tasks

This section provides information about the tasks available before you install an operating system using the modular tools.

Before installing an operating system using the modular tools, you can perform the following tasks:

- Replicate system settings
- Configure RAID
- Replicate RAID
- Configure Fibre HBAs
- Update firmware using UpdateXpress System Packs
- Server disk analysis
- Server data disposal

The following sections describe how to perform these tasks using the modular tools.

Replicate system settings

Before installing an operating system, you can capture system settings from a compatible server to a file and deploy them to your new server by using the sample jobs provided by theLinux Scripting Toolkit and the Advanced Settings Utility.

The Toolkit provides sample jobs to capture and deploy settings using ASU. These jobs are located in the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **System Settings Replication** folder.

Capture system settings

The **Capture System Settings** sample job captures the system settings of the server and stores them in a file named `ASU_Capture` By default all captured System Settings are stored in `sgdeploy/sgtklinux/ini/asu/`*`PlatformID`* where *PlatformID* is the machine type of the captured system.

Deploy system settings

The **Deploy System Settings** sample job deploys the system settings from the `ASU_Capture.ini` file located in the specified directory.

You can customize the following variables for these jobs from the Altiris Console. The values shown are the defaults.

- TK_ASU_File=sgdeploy/sgtklinux/ini/asu/\${BIOS_Build_Level:0:2}/asu_capture
- TK_ASU_Mode=save

RAID configuration

Before installing an operating system, you can configure RAID on the target server by using the Linux Scripting Toolkit and the PRAID utility.

Linux Scripting Toolkit provides sample jobs to perform a number of RAID configurations. These jobs are located in the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **RAID Configuration** folder, and are named according to the RAID configuration they will create. For example, **Configure RAID [RAID 0]**, and **Configure RAID [RAID 1 + Hot Spare]**.

To perform RAID configuration using the modular tools:

1. Start the Altiris Deployment Solution console.
2. Navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **RAID Configuration** folder.
3. Run the appropriate job against the target system to create the RAID configuration you want.

RAID replication

If you have an existing RAID configuration that you want to replicate to other systems, you can use the sample jobs **RAID Capture** and **RAID configuration** to capture and deploy the settings to a new target server.

The sample jobs are located in the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **RAID Replication** folder in the Altiris Deployment Solution console.

To capture an existing RAID configuration, follow these steps:

1. Start the Altiris Deployment Solution console.
2. Navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **RAID Replication** folder.
3. Run the **Capture RAID configuration** job against the system whose RAID configuration you want to replicate.

   You can customize the following variables for these jobs from the Altiris Deployment Solution console. The values shown are the defaults:

   - **TK_pRAID_File**=`sgdeploy/sgtklinux/ini/praid/`*`Machine_Type`*`/raid_capture.ini` where *Machine_Type* is the machine type of the system being captured.
   - **TK_pRAID_Flags**=

When you have captured the RAID configuration, you can deploy it using the **Deploy RAID configuration** job in the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **RAID Replication** folder.

Configure a Fibre HBA

You can use this pre-installation task to configure your Fibre Channel Host Bus Adapter (HBA) to boot from your Storage Area Network (SAN).

Before you begin

In order to boot from the SAN, it must be configured with a logical drive mapped to a Logical Unit Number (LUN). The HBA you are configuring must have permission to access the logical drive and the LUN.

About this task

Note that the default settings configure the first drive mapped to LUN 0 as a boot device (for QLogic Fibre HBAs only). For information about Emulex Fibre HBAs, see "Known problems and limitations" on page 88. To avoid errors, ensure that your SAN configuration includes at least one logical drive mapped to LUN 0.

Procedure

1. Start the Altiris Deployment Solution console
2. Navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **Fibre Configuration** folder.
3. Run the **Configure Fibre HBA for Boot** job against the target system. You can customize the following variables for these jobs from the Altiris Deployment Solution console. The default values are shown.
   - TK_FIBRE_COUNT="1"
   - TK_FIBRE_1_HBA_ID="0"
   - TK_FIBRE_1_BOOT_DISABLE="No"
   - TK_FIBRE_1_BOOT_PRIM="0 0 0"
   - TK_FIBRE_1_BOOT_ALT1=""
   - TK_FIBRE_1_BOOT_ALT2=""
   - TK_FIBRE_1_BOOT_ALT3=""

Update Firmware

You can use the **Update Firmware (UXSP)** job to update all of the firmware on a target server from the IBM Preboot environment.

About this task

To update the firmware on a server using the **Update Firmware (UXSP)** job, follow these steps:

Procedure

1. Acquire UpdateXpress System Packs (UXSPs) for SUSE Linux Enterprise Server 10 for all of the systems to be updated.
2. Start the Altiris Deployment Solution console.

3. Navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **Update Firmware (UXSP)** folder.
4. Run the **Update System Firmware** job against the target system. You can customize the following variables for these jobs from the Altiris Deployment Solution console. The default values are shown.
   - TK_UXSP_UpdateXpressSystemPacks="/sgdeploy/updates/uxsp"
   - TK_UXSP_ApplyLatest="No"
   - TK_UXSP_UXSPIUpdateFlags="update --unattended --firmware"

Server disk analysis

This sample job captures the disk information for all disks attached to a given server. You can use this job after performing RAID configuration to ensure that when you install a supported operating system, you install to disk `/dev/sda` on the target server.

Before you begin

You must perform RAID configuration before running this job, as the disk information collected during analysis is changed during RAID configuration.

About this task

To capture disk information, follow these steps:

Procedure

1. Start the Altiris Deployment Solution console.
2. Navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **Server Disk Analysis** folder.
3. Run the **Capture Disk Data** job against the target system. You can customize the following variables for these jobs from the Altiris Deployment Solution console. The default values are shown.
   - TK_DISK_File=/mnt/ds/Temp/\$ID/disk_info.txt

Server data disposal

The Linux Scripting Toolkit provides sample jobs to destroy data from drives and optionally to delete or reset RAID configuration. These jobs are located in the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **Server Data Disposal** folder.

The following topics describe the process for using these sample jobs.

Server data disposal with RAID reset:

Use this job to remove data from the drives and reset the RAID configuration of the target system.

Procedure

1. Start the Altiris Deployment Solution console.
2. Navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **Server Data Disposal** folder.

3. Run the **Server Data Disposal and Reset RAID** job against the target system. You can customize the following variables for these jobs from the Altiris Deployment Solution console. The default values are shown.
   - TK_Wipe_Disk=1
   - TK_Wipe_Repeat_Number=00
   - TK_Wipe_Level=quick

Server data disposal without RAID reset:

Use this job to remove data from the drives without resetting the RAID configuration of the target system.

Procedure

1. Start the Altiris Deployment Solution console.
2. Navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 1 - Pre-installation Hardware Configuration** > **Server Data Disposal** folder.
3. Run the **Server Data Disposal non-RAID** job against the target system. You can customize the following variables for these jobs from the Altiris Deployment Solution console. The default values are shown.
   - TK_Wipe_Disk=1
   - TK_Wipe_Repeat_Number=00
   - TK_Wipe_Level=quick

Operating system installation

After you have configured RAID on the target server, you can use the jobs provided by the Linux Scripting Toolkit to install a supported operating system. This section describes that process.

The Linux Scripting Toolkit provides two types of jobs for deploying operating systems:

- Operating system cloning
- Scripted operating system installation

The following sections describe how to perform each type of installation.

Operating system cloning

The Linux Scripting Toolkit provides sample jobs to capture and deploy supported Linux operating systems.

Before you begin

In order to run the **Capture Linux Image** sample job, the Altiris Agent must first be installed on the donor system. The scripted operating system installation jobs provided by the Linux Scripting Toolkit install the Altiris Agentautomatically. If you are cloning an operating system that was not installed using the Toolkit, you must install the Altiris Agent manually.

About this task

To capture a supported operating system from one server and deploy it to another, follow these steps:

Procedure

1. Start the Altiris Deployment Solution console.
2. Navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 2 - Operating System Installation** > **Operating System Imaging** folder.
3. Open the **Capture Linux Image** job.
4. Select the **Create Disk Image** task and click **Modify**.
5. Change the path and filename for the captured image and click **Finish** to save your changes.
6. Open the **Deploy Linux Image** job.
7. Select the **Distribute Disk Image** task and click **Modify**.
8. Change the path and filename to match the ones you entered for the captured image and click **Finish** to save your changes.
9. Run the **Capture Linux Image** job against the donor server.
10. When the capture job is complete, run the **Deploy Linux Image** job against the target server.

Scripting operating system installation

If you do not want to deploy a clone of an existing operating system installation, the Linux Scripting Toolkit provides sample jobs to deploy each of the supported operating systems. Use the following procedure to install a supported operating system using the sample jobs provided:

Before you begin

Before you can install an operating system, the operating system installation files must be available to the source server. See "Adding operating system installation files" on page 11 for more information.

Procedure

1. Start the Altiris Deployment Solution console.
2. Navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 2 - Operating System Installation** > **Scripted Operating System Installation** folder.
3. Select the job that corresponds to the operating system you want to install and run it against the target server.

Scripted operating system installation in uEFI

Linux Scripting Toolkit provides sample jobs to deploy supported operating systems in uEFI mode. This topic describes the process for performing these deployments.

Before you begin

Before you can install an operating system, the operating system installation files must be available to the source server. See "Adding operating system installation files" on page 11 for more information.

About this task

Use the following procedure to install a supported operating system in uEFI mode using these sample jobs:

- **Combined Deployment Solutions** > **Scripted Deployments:**

  - **Configure RAID + Install RHEL 6 x64 in uEFI**
  - **Configure RAID + Install SLES 11 x64 in uEFI**
- **Modular Deployment Tools** > **Scripted Operating System Installation:**
  - **Install RHEL 6 x64 in uEFI**
  - **Install SLES 11 x64 in uEFI**

Procedure

1. Start the Altiris Deployment Solution console.
2. Navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 2 - Operating System Installation** > **Scripted Operating System Installation** folder.
3. Select the job that corresponds to the operating system you want to install and run it against the target server.

   Note: Only SLES 11 x64 and RHEL x64 are supported for native uEFI installation. The *TK_DISTRO_CFG* job variable determines whether the installation is a native uEFI installation or a legacy installation. The Linux Scripting Toolkit provides two sample answer files for native uEFI installations:
   - `sles11x64_efi.xml` for native uEFI installation of SLES 11 x64.
   - `rhel6x64_efi.xml` for native uEFI installation of RHEL 6 x64.

   If you want to customize other answer files for native uEFI installation, ensure that the file contains an entry for `/boot/efi`. This entry can be commented out, but must remain visible in the file.

Post installation tasks

When you have completed RAID configuration and operating system installation, you can perform post installation tasks. This section describes the methods for installing the IBM Systems Director Agent and applying UpdateXpress System Pack (UXSP) updates.

The Linux Scripting Toolkit provides sample jobs for installing the IBM Systems Director Agent and applying UpdateXpress System Pack updates to a target system on which RAID has been configured and a supported operating system installed. Note that to complete these jobs, the Altiris Agent must be installed on the target system. The Altiris Agent is installed automatically by the operating system installation tasks provided by the Linux Scripting Toolkit. If the operating system on the target system was not installed using these jobs, you might have to install the Altiris Agentmanually. See “Operating system installation” on page 18 for more information about using the modular deployment tools to install operating systems.

The following sections describe the process for using the modular deployment tools to install the IBMDirector Agent and apply UXSP updates to a target system.

Installing the IBM Systems Director Agent

After you have configured RAID and installed a supported operating system , use the Linux Scripting Toolkit to install the IBM Systems Director Agent.

Procedure

1. Ensure that the IBM Systems Director Agent source files have been added to the source tree on the source server. For more information, see “Adding Systems Director Agent files” on page 10.

2. Open the Altiris Deployment Solution console and navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 3 - Post-Installation Configuration** > **Application Installation** folder.
3. Run the **IBM Director Agent for Linux nstall** job against the target system.

Installing UpdateXpress System Packupdates

You can ensure that your target server has current firmware fixes by using the Linux Scripting Toolkit to install UpdateXpress System Pack (UXSP) updates.

Before you begin

Before you can install UXSPs, you must have added the latest updates for your target server machine type to the source server. See "Adding UpdateXpress System Packs (UXSPs)" on page 9 for more information.

About this task

Follow these steps to apply UXSPs to your target system:

Procedure

1. Open the Altiris Deployment Solution console.
2. Navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Modular Deployment Tools** > **Step 3 - Post-Installation Configuration** > **IBM UpdateXpress System Packs** folder.
3. Run the **Run UpdateXpress System Pack Installer** job against the target server.

Deployment solutions

The sample jobs provided in the **Combined Deployment Solutions** folder are designed to consolidate multiple deployment elements in a single Altiris job. This section describes how to use these jobs to perform complete deployments.

The deployment solutions provided by the Linux Scripting Toolkit are designed to perform a combination of deployment steps in a single job. The following sections will provide information on using the sample jobs for:

- Image based deployment (image cloning)
- Scripted deployment

Image based deployment

The Linux Scripting Toolkit provides a set of tools to capture and deploy an existing system configuration, including RAID configuration, operating system, and installed applications and updates. These jobs require the Altiris Agent to be installed on the donor system. The Altiris Agent is installed automatically as part of the Scripting Toolkit operating system installation jobs. If the donor system operating system was not installed using the Scripting Toolkit, you might need to install the Altiris Agent manually.

Follow these steps to capture and deploy a system configuration:

1. Open the Altiris Deployment Solution console and navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Combined Deployment Solutions** > **Image-based Deployment** folder.
2. Run the **Capture RAID Configuration and Linux Image** job against the donor system.

3. When that job completes, run the **Deploy RAID Configuration and Linux Image** against the target system.

Scripted deployment

The scripted deployment sample jobs provided by the ServerGuide Scripting Toolkit are designed to integrate all of the steps in deploying a system configuration in a single Altiris job. To use a scripted deployment job to configure RAID, install an operating system, install the IBM Director Agent, and install UpdateXpress System Pack updates on the target system, follow these steps:

1. Ensure that the necessary operating system, Director Agent, and UXSP files are properly included in the source tree. See Chapter 3, "Adding and sharing content on the Linux Scripting Toolkit source server," on page 9 for more information.
2. Open the Altiris Deployment Solution console and navigate to the **IBM Scripting Toolkit, Linux Edition 9.6** > **Combined Deployment Solutions** > **Scripted Deployment** folder.
3. In this folder, select the job to deploy the operating system of your choice. Modify the following job variables:

   TK_DirAgent_DirectorAgent=dirlin
   To install the IBM Systems Director Agent. Uncomment this variable and assign the folder under `/sgdeploy/apps/` where the IBM Systems Director Agent files are located.

   TK_UXSP_UpdateXpressSystemPacks=uxsp
   To install UpdateXpress System Pack updates.

   See Chapter 5, "Customizing deployment scenarios," on page 23 for more information on configuring job variables.
4. Run the job against the target server.

Chapter 5. Customizing deployment scenarios

This section provides information on customizing deployment scenarios.

You can customize your deployment scenarios in the following ways:

- Customize source server settings
- Add PRAID policy files
- Add ASU files
- Customize Fibre HBA boot configuration
- Customize your Linux installation
- Add installation of the IBM Systems Director Agent to your deployment
- Add installation of UpdateXpress System Packs (UXSPs) to your deployment

Customizing sample jobs

This section provides information about customizing the sample Altiris jobs provided with the ServerGuide Scripting Toolkit, Linux Editionthat perform functions such as configuring PRAID, ASU, and Fibre settings.

Each sample job provided within the Altiris Deployment Solution console that contains customizable settings begins with a task that sets all variables for the job. This task is labeled **Customize Job Variables.** The variables that you can modify are contained within a block that is similar to the following:

```
[User_Customizable_Variables]

Variables

[End_User_Customizable_Variables]
```

You cannot modify variables outside this block. When you modify a job, it is a good practice to create a copy of the job and modify the copy, rather than modify the original job.

Table 1 provides a list of the variables in the sample Altiris jobs that you can customize. Note that all values for variables must be enclosed in quotation marks.

Table 1. Customizable variables in Altiris sample jobs

Variable	Description
[PRAID Settings]	
TK_pRAID_File	Specifies the pRAID policy file to be used for the job. **Default:**: Varies per job, see next section.
TK_pRAID_Flags	Specifies any additional flags to be passed to the pRAID utility. **Default:** `-r`
[Fibre Settings]	

Table 1. Customizable variables in Altiris sample jobs (continued)

Variable	Description
TK_FIBRE_COUNT	Specifies the number of HBA ports to configure. Valid values are 1–*n*, where *n* is the number of HBA ports available. This variable affects the use of the following variables: • TK_FIBRE_N_HBA_ID • TK_FIBRE_N_BOOT_DISABLE • TK_FIBRE_N_BOOT_PRIM • TK_FIBRE_N_BOOT_ALT1 • TK_FIBRE_N_BOOT_ALT2 • TK_FIBRE_N_BOOT_ALT3 Where *N* is the HBA number to be configured. **Note:** You must complete one of each of these variables for every HBA port you configure. So if `TK_FIBRE_COUNT=2`, you must complete one set of these variables for the first port and one for the second.

Table 1. Customizable variables in Altiris sample jobs (continued)

Variable	Description

TK_FIBRE_*N*_HBA_ID	Identifies the Qlogic/Emulex HBA to be configured, where *N* is the HBA number to be configured.
	
	Valid values are:
	
	hba_instance
	
	The instance number of an HBA port. Valid values are integers from 0 to *n-1*, where *n* is the number of HBAs in the system.
	
	For example, to configure HBA instance 0:`TK_FIBRE_1_HBA_ID=0`
	
	hba_wwpn
	
	The World Wide Port Name of an HBA port, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxxxx*.
	
	For example, to configure HBA: 90-87-AA-BB-65-34-BB-E0:
	
	`TK_FIBRE_1_HBA_ID=`
	`90-87-AA-BB-65-34-BB-E0`
	
	Default: `0`

	Identifies the Brocade HBA to be configured, where *N* is the HBA number to be configured.
	
	Valid values are:
	
	hba_instance
	
	The instance number of an HBA port. Valid format is *N/P*, where *N* is the adapter number from 1 to *N*, and *P* is the port number from 0 to *p-1*.
	
	For example, to configure HBA instance 0: `TK_FIBRE_1_HBA_ID=1/0`
	
	hba_wwpn
	
	The World Wide Port Name of an HBA port, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxxxx*.
	
	For example, to configure HBA: 90-87-AA-BB-65-34-BB-E0:
	
	`TK_FIBRE_1_HBA_ID=`
	`90-87-AA-BB-65-34-BB-E0`
	
	Default: `0`

Table 1. Customizable variables in Altiris sample jobs (continued)

Variable	Description
TK_FIBRE_*N*_BOOT_DISABLE	Disable the selected current boot device settings on the specified HBA port, where *N* is the HBA number to be configured. Valid values are **No** Does not clear or disable any boot settings. **All** Disables the primary and all alternate boot settings – Prim, Alt1, Alt2, and Alt3. **Prim** Disables only the primary boot setting. **Alt1** Disables the Alternative 1 boot setting. **Alt2** Disables the Alternative 2 boot setting. **Alt3** Disables the Alternative 3 boot setting. Default: `No`

TK_FIBRE_*N*_BOOT_PRIM = *target_wwnn target_wwpn lun_id*	Defines the primary boot target settings, where *N* is the HBA number to be configured, and:
	• *target_wwnn* is the World Wide Node Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*.
	• *target_wwpn* is the World Wide Port Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*.
	• *lun_id* is the Logical Unit Number of a device.
	
	Default: `0 0 0`
	
	Example:
	`TK_FIBRE_1_BOOT_PRIM=`
	`BB-CC-AA-BB-65-34-BB-F1`
	`BB-CC-AA-BB-FF-34-BB-F1 9`

TK_FIBRE_*N*_BOOT_ALT1 = *target_wwnn target_wwpn lun_id*	Configures the operating system to use the indicated target as the first alternate boot device, where *N* is the HBA number to be configured, and
	• *target_wwnn* is the World Wide Node Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*.
	• *target_wwpn* is the World Wide Port Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*.
	• *lun_id* is the Logical Unit Number of a device.
	
	Default: blank
	
	Example:
	`TK_FIBRE_1_BOOT_ALT1=`
	`BB-CC-AA-BB-65-34-BB-FD`
	`BB-CC-AA-BB-FF-40-BB-F1 5`

Table 1. Customizable variables in Altiris sample jobs (continued)

Variable	Description

TK_FIBRE_*N*_BOOT_ALT2 = *target_wwnn target_wwpn lun_id*	Configures the operating system to use the indicated target as the second alternate boot device, where *N* is the HBA number to be configured, and
	• *target_wwnn* is the World Wide Node Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx*or *xxxxxxxxxxxxxx*.
	• *target_wwpn* is the World Wide Port Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*.
	• *lun_id* is the Logical Unit Number of a device.
	Default: blank
	Example:
	`TK_FIBRE_1_BOOT_ALT2=`
	`BB-CC-AA-BB-65-34-BB-FD`
	`BB-CC-AA-BB-FF-40-BB-F1 5`

TK_FIBRE_*N*_BOOT_ALT3 = *target_wwnn target_wwpn lun_id*	Configures the operating system to use the indicated target as the third alternate boot device, where *N* is the HBA number to be configured, and
	• *target_wwnn* is the World Wide Node Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*.
	• *target_wwpn* is the World Wide Port Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*.
	• *lun_id* is the Logical Unit Number of a device.
	Default: blank
	Example:
	`TK_FIBRE_1_BOOT_ALT3=`
	`BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5`

[ASU Settings]	
TK_ASU_File	Specifies the file to be used for a Capture System Settings or Deploy System Settings operation. If you are capturing system settings, they will be written to a file with this name. If you are deploying system settings, they will be read from this file. **Default:** `/sgdeploy/sgtklinux/ini/asu/$(BIOS_Build_Level0:2)/asu_capture`
TK_ASU_Mode	Specifies the ASU mode being used. This value is used with the **asu** command and the value of the *TK_ASU_File* variable to create a complete ASU command. Valid values are: `save`, `replicate`, and `batch`.
TK_ASU_Flags	Specifies additional flags for the current mode. The default value is **–group-bios**.
[NOS Installation Settings]	
TK_DISTRO_TYPE	Specifies the Linux distribution to be installed. Valid values are: `RHEL`, `SLES`, and `VMWARE`

Table 1. Customizable variables in Altiris sample jobs (continued)

Variable	Description
TK_DISTRO_VERSION	Specifies the version of the operating system indicated by TK_DISTRO_TYPE. Valid values correspond to the supported versions of the supported operating systems. **Note:** The version number must be a valid and supported version of the Linux distribution specified by TK_DISTRO_TYPE.
TK_DISTRO_PATH	Specifies the fully qualified URL for the location of the network share where the operating system files are located. Supported protocols are FTP, HTTP, and NFS. **Default:** ftp://$TK_Altiris_IP/sgdeploy/os/osfolder
TK_DISTRO_CFG	Specifies the unattended answer file to be used for the operating system installation. This file must be formatted as a `kickstart` or `autoyast` file, depending on the distribution being deployed. **Default:** ./sgdeploy/sgtklinux/altiris/answerfiles/answerfile
TK_DISTRO_KPARAM	Specifies additional kernel parameters to be used for the deployment, such as VNC parameters or a static IP address. **Default:** none
TK_DISTRO_NIC_PORT	Specifies a Network Interface Card (NIC) to be used for the installation, if the NIC is different than the one used for the pre-installation execution environment (PXE) boot. If no NIC is specified, the NIC used for PXE boot will be used for the installation, by ng the **`bootif`** method. **Note:** This variable is not valid for VMware ESX Server. To indicate the network device for VMware ESX Server, edit the network section of the VMware kickstart file.
[Director Agent Settings]	
TK_DirAgent_DirectorAgent	Specifies the location of the IBM Director Agent application files on the source server. Valid values are: • dirlin • A user-supplied directory in the `sgdeploy/apps` directory **Default:**`dirlin`
[UXSP Settings]	
TK_UXSP_UpdateXpressSystemPacks	Specifies the location of the UpdateXpress System Packs on the source server. Valid values are: • uxsp • a user-specified directory in the `sgdeploy/updates` directory **Default:** uxsp

Sample job definitions

This section provides descriptions of the sample Altiris jobs provided by the Linux Scripting Toolkit, including a description of the variables that can be customized for each job.

All of the sample jobs except imaging jobs without RAID configuration and the Linux Pre-Boot test add logging information to the ToolkitProcess.log file.

Customizing Fibre HBA boot configuration

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

By default, the ServerGuide Scripting Toolkit will configure the first HBA on the system to boot from the first available SAN target (for QLogic Fibre HBAs only).

Note: For information about Emulex Fibre HBAs, see "Known problems and limitations" on page 88)

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

Note: Some examples in the table are broken into multiple lines for formatting reasons. When you use these settings, however, you must include all the information for each variable on a single line.

Table 2. Fibre HBA boot configuration variables

Variable	Description
TK_FIBRE_COUNT	Specifies the number of HBA ports to configure. Valid values are 1–*n*, where *n* is the number of available HBA ports. This variable affects the use of the following variables: • TK_FIBRE_*N*_HBA_ID • TK_FIBRE_*N*_BOOT_DISABLE • TK_FIBRE_*N*_BOOT_PRIM • TK_FIBRE_*N*_BOOT_ALT1 • TK_FIBRE_*N*_BOOT_ALT2 • TK_FIBRE_*N*_BOOT_ALT3 Where *N* is the HBA number to be configured. **Note:** You must complete one of each of these variables for every HBA port you configure. So if `TK_FIBRE_COUNT=2`, you must complete one set of these variables for the first port and one for the second.

Table 2. Fibre HBA boot configuration variables (continued)

Variable	Description

TK_FIBRE_*N*_HBA_ID	Identifies the Qlogic/Emulex HBA to be configured, where *N* is the HBA number to be configured.
	
	Valid values are:
	
	hba_instance
	
	The instance number of an HBA port. Valid values are integers from 0 to *n-1*, where *n* is the number of HBAs in the system.
	
	For example, to configure HBA instance 0:`TK_FIBRE_1_HBA_ID=0`
	
	hba_wwpn
	
	The World Wide Port Name of an HBA port, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxxxx*.
	
	For example, to configure HBA: 90-87-AA-BB-65-34-BB-E0:
	
	`TK_FIBRE_1_HBA_ID=`
	`90-87-AA-BB-65-34-BB-E0`
	
	Default: `0`

	Identifies the Brocade HBA to be configured, where *N* is the HBA number to be configured.
	
	Valid values are:
	
	hba_instance
	
	The instance number of an HBA port. The format is *N/P*, where *N* is the adapter number from 1 to *N*, and *P* is the port number from 0 to *p-1*.
	
	For example, to configure HBA instance 0:
	`TK_FIBRE_1_HBA_ID=1/0`
	
	hba_wwpn
	
	The World Wide Port Name of an HBA port, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxxxx*.
	
	For example, to configure HBA: 90-87-AA-BB-65-34-BB-E0:
	
	`TK_FIBRE_1_HBA_ID=`
	`90-87-AA-BB-65-34-BB-E0`
	
	Default: `0`

Table 2. Fibre HBA boot configuration variables (continued)

Variable	Description
TK_FIBRE_*N*_BOOT_DISABLE	Disable the selected current boot device settings on the specified HBA port, where *N* is the HBA number to be configured. Valid values are **No** Does not clear or disable any boot settings. **All** Disables the primary and all alternate boot settings - Prim, Alt1, Alt2, and Alt3. **Prim** Disables only the primary boot setting. **Alt1** Disables the Alternative 1 boot setting. **Alt2** Disables the Alternative 2 boot setting. **Alt3** Disables the Alternative 3 boot setting. Default: `No`

TK_FIBRE_*N*_BOOT_PRIM = *target_wwnn target_wwpn lun_id*	Defines the primary boot target settings, where *N* is the HBA number to be configured, and:
	• *target_wwnn* is the World Wide Node Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*.
	• *target_wwpn* is the World Wide Port Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*.
	• *lun_id* is the Logical Unit Number of a device.
	
	Default: 0 0 0
	
	Example:
	`TK_FIBRE_1_BOOT_PRIM=`
	`BB-CC-AA-BB-65-34-BB-F1 BB-CC-AA-BB-FF-34-BB-F1 9`

TK_FIBRE_*N*_BOOT_ALT1 = *target_wwnn target_wwpn lun_id*	Configures the operating system to use the indicated target as the first alternate boot device, where *N* is the HBA number to be configured, and
	• *target_wwnn* is the World Wide Node Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*.
	• *target_wwpn* is the World Wide Port Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*.
	• *lun_id* is the Logical Unit Number of a device.
	
	Default: blank
	
	Example:
	`TK_FIBRE_1_BOOT_ALT1=`
	`BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5`

Table 2. Fibre HBA boot configuration variables (continued)

Variable	Description
TK_FIBRE_*N*_BOOT_ALT2 = *target_wwnn target_wwpn lun_id*	Configures the operating system to use the indicated target as the second alternate boot device, where *N* is the HBA number to be configured, and • *target_wwnn* is the World Wide Node Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*. • *target_wwpn* is the World Wide Port Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*. • *lun_id* is the Logical Unit Number of a device. Default: blank Example: `TK_FIBRE_1_BOOT_ALT2= BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5`
TK_FIBRE_*N*_BOOT_ALT3 = *target_wwnn target_wwpn lun_id*	Configures the operating system to use the indicated target as the third alternate boot device, where *N* is the HBA number to be configured, and • *target_wwnn* is the World Wide Node Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*. • *target_wwpn* is the World Wide Port Name of a device, in the format *xx-xx-xx-xx-xx-xx-xx-xx* or *xxxxxxxxxxxxxx*. • *lun_id* is the Logical Unit Number of a device. Default: blank Example: `TK_FIBRE_1_BOOT_ALT3= BB-CC-AA-BB-65-34-BB-FD BB-CC-AA-BB-FF-40-BB-F1 5`

Capture and Deploy RAID Configuration and Linux image

The IBM ServerGuide Scripting Toolkit, Linux Edition provides sample Altiris jobs to perform operating system imaging tasks, such as capturing a copy of an installed operating system and deploying that image to a target server. You might need to configure the target server before you deploy the image.

Note: Todeploy the captured RAID configuration and Linux image, the donor and target systems must be configured identically in order.

The sample jobs capture a RAID Configuration along with the operating system (OS) image, so that you can use both to redeploy the image. To run the RAID configuration and Linux image sample capture job, you must first install the Altiris agent on the donor system. The Toolkit's scripted OS installation jobs automatically install the Altiris agent.

The sample capture job captures the RAID configuration of the server and then captures an OS image from the target server and then stores both on the Altiris server.

Each sample deployment job deploys the captured RAID configuration and OS image to the target server. The sample job configures RAID, calls a reboot, and then deploys the image.

You can customize the following variable for these jobs within the Altiris Console: TK_pRAID_File="/sgdeploy/sgtklinux/ini/praid/\$Machine_Type/raid_capture.ini"

Configure RAID and install Linux OS

This sample job performs a RAID configuration, restarts the system, and then performs a scripted operating system installation. You must set up the operating system being deployed in the Linux Scripting Toolkit source tree on the Altiris Agent. See “Adding operating system installation files” on page 11 for details.

The following variables may be customized for this job within the Altiris Deployment Solution console. The values shown are the defaults:

- TK_pRAID_File="/sgdeploy/sgtklinux/ini/praid/default_raid.ini"
- TK_pRAID_Flags="-r"
- TK_Altiris_IP
- TK_DISTRO_TYPE
- TK_DISTRO_VERSION
- TK_DISTRO_PATH=ftp://$TK_Altiris_IP/sgdeploy/os/rhel5x64
- TK_DISTRO_CFG
- TK_DirAgent_DirectorAgent=dirlin
- TK_UXSP_UpdateXpressSystemPacks=uxsp
- TK_DISTRO_KPARAM
- TK_DISTRO_NIC_PORT="link"

Configuring RAID with a policy file

You can configure RAID controllers by using standalone jobs that only configure the RAID controllers and then stop, and jobs integrated with scripted and image-based deployment. The jobs that specify a RAID level in them use a pre-configured policy file to create a RAID array of that type on the target machine. The other jobs either deploy a previously captured RAID configuration or create the default RAID configuration if no policy file is specified.

To use a user-defined policy file, follow these steps:

1. Create the policy file in the `sgdeploy/sgtklinux/ini/praid` directory.
2. Copy a pre-existing Scripting Toolkit job with RAID configuration.
3. Modify the TK_pRAID_File variable in the copied job to point to your policy file, for example: `TK_pRAID_File="/sgdeploy/sgtklinux/ini/praid/UserPolicy1.ini"`.

Capture and deploy RAID configuration

TheLinux Scripting Toolkit provides sample Altiris jobs to capture and deploy a RAID configuration, allowing the RAID configuration of a single server to provide a configuration basis for any number of servers. The sample capture job captures the RAID configuration of the server and stores it in a file named `RAID_Configuration.ini`.

By default, all captured RAID configurations are stored in the following directory on the Deployment Server:

AltirisPath\sgdeploy\sgtklinux\ini\praid*machinetype*

where:

- *AltirisPath* is the fully qualified path to the Altiris Deployment Solution directory, for example: C:\Program Files\Altiris\eXpress\Deployment Server.
- *machinetype* is the machine type of the captured system.

The deployment sample job deploys the captured RAID configuration from the RAID_Configuration.ini file located in the previously-mentioned directory by default. You can customize the following variables for these jobs within the Altiris Console:

- TK_pRAID_File=/sgdeploy/sgtklinux/ini/praid/*machinetype*/raid_capture.ini
- TK_pRAID_Flags="-r"

Server data disposal

The IBM ServerGuide Scripting Toolkit, Linux Edition provides sample jobs to perform disk tasks, such as resetting the RAID configuration or wiping the disk. You might need to configure the disk on the target server before you deploy s new operating system on it.

These sample jobs perform a server data disposal. The first job performs a disposal and resets the RAID Configuration. The second job performs only a server data disposal.

The following variables may be customized for these jobs within the Altiris Console:

Variable	Values
TK_Wipe_Level	• "quick" - performs a quick wipe of the disks. • "dod" - performs a multipass wipe of the disks that conforms to DOD standards.
TK_Wipe_Repeat_Number	*number* – indicates the number of passes to complete.
TK_Wipe_Disk	*disk_number* – indicates the number of the disk to wipe. By default, "1" is the first disk in the system.

Note: Depending on the size of the drives involved and the wipe level, these jobs can take up to several hours to complete.

Linux and VMware unattended scripted installation

These sample jobs perform scripted installation of the supported Linux distributions. You must set up the distribution to be installed on the Altiris server before you run these jobs. See “Adding operating system installation files” on page 11 for more information.

You can customize the following variables for these jobs:

- TK_Altiris_IP
- TK_DISTRO_TYPE

- TK_DISTRO_VERSION
- TK_DISTRO_PATH
- TK_DISTRO_CFG
- TK_DirAgent_DirectorAgent
- TK_UXSP_UpdateXpressSystemPacks

Capture and deploy a Linux image

The Linux Scripting Toolkit provides sample jobs to perform operating system (OS) imaging tasks by using Altiris, such as capturing a copy of an installed OS (an OS image) and deploying the image to a target server. You might need to perform RAID configuration on the target server before you can deploy the OS image to it. To run the **Capture Linux Image** sample job, the Altiris Agent must first be installed on the donor system. The Scripting Toolkit scripted OS installation jobs automatically install the Altiris Agent.

Note: The **Capture Linux Image** job is not supported for VMware ESX Server.

To modify the file name and location of the captured image, perform the following steps:

1. Double-click the **Capture Linux Image** job.
2. Select the **Create Disk Image** task, and click **Modify**.
3. Change the path and file name for the captured image.
4. Click **Finish** to save it.
5. Double-click the **Deploy Linux Image** job.
6. Select the **Distribute Disk Image** task, and click **Modify**.
7. Change the path and file name to the same values used above for the captured image.
8. Click **Finish** to save the changes.

Note:

- When you deploy a target server by using the Image Install job, the image will include the Altiris Agent, because the donor server is required to have the Altiris Agent installed.
- Because the Linux imaging sample jobs use a sector-based imaging technique, the donor and target servers must have identical hardware.
- Altiris RapiDeploy, which is used for the cloning jobs, does not support cloning of multiple disks in a single pass. To work around this issue, you can place all of the Linux partitions on a single disk during installation by modifying the sample kickstart files to include the **–ondisk** parameter in the partition configuration *part* lines. Refer to the operating system documentation for additional information.
- To specify the disk to be cloned, you must add the **-d** parameter, specifying the disk number, to the command line options for the Capture Linux Image and Deploy Linux Image jobs. For example, to clone the operating system image on disk two, you would add: `-d2` to the command line options for the job.
- If either the capture or deploy job fails, refer to the `rdeploy log` file for more information. By default, the log file location is *$AltirisInstalledPath*`\RDeploy\Linux\x64\rdlog\` where *$AltirisInstalledPath* is the fully qualified path to the Altiris Deployment

Solution directory. For example, `C:\Program Files\Altiris\eXpress\Deployment Server\RDeploy\Linux\x64\rdlog\`.

Linux Pre-Boot Test

This sample job performs a basic task to test connectivity between the target server and the Altiris Deployment Server. The job boots into the IBM Linux PE x64 environment on the target server.

Chapter 6. Supported hardware and software

This section lists the operating systems, adapters, and RAID controllers supported by the Linux Scripting ToolkitServerGuide Scripting Toolkit, Linux Edition, as well as systems that support BIOS and firmware updates by using the Advanced Settings Utility (ASU)ASU.

The Linux Scripting Toolkit supports the deployment of Linux operating systems on IBM System x and BladeCenter servers. In general, the Linux Scripting Toolkit provides support for IBM ServerProven and third-party adapters in the following categories:

- Ethernet
- Fibre Channel
- IDE and IDE RAID
- SAS and SAS RAID
- SATA and SATA RAID
- SCSI and SCSI RAID, including Ultra-SCSI

This section provides information about specific hardware and software support for deployment scenarios, including:

- Supported operating system and server combinations
- RAID and Fibre channel HBA support by the server
- Network device driver support by the server
- Limitations of support for applicable servers

The most current support information is provided on the ServerGuide Scripting Toolkit web page. See IBM deployment resources on the Internet for information.

Operating system support

This section lists operating system deployment and server combinations that are supported by the Linux Scripting ToolkitServerGuide Scripting Toolkit, Linux Edition.

You can use the Linux Scripting Toolkit to deploy supported Linux distributions to any IBM System x, BladeCenter, or iDataPlex server that supports that distribution. To determine what distribution and server combinations are supported, see IBM ServerProven.

The Linux Scripting Toolkit supports these Linux distributions:

- SUSE Linux Enterprise Server 9 32 bit SP4
- SUSE Linux Enterprise Server 9 x64 SP4
- SUSE Linux Enterprise Server 10 32 bit SP1/SP2/SP3/SP4
- SUSE Linux Enterprise Server 10 x64 SP1/SP2/SP3/SP4
- SUSE Linux Enterprise Server 11 32 bit Base/SP1/SP2/SP3
- SUSE Linux Enterprise Server 11 x64 Base/SP1/SP2/SP3
- Red Hat Enterprise Linux 4 AS/ES 32 bit U6/U7/U8
- Red Hat Enterprise Linux 4 AS/ES x64 U6/U7/U8

- Red Hat Enterprise Linux 5 32 bit U1/U2/U3/U4/U5/U6/U7/U8/U9/U10
- Red Hat Enterprise Linux 5 x64 U1/U2/U3/U4/U5/U6/U7/U8/U9/U10
- Red Hat Enterprise Linux 6 32 bit U1/U2/U3/U4/U5
- Red Hat Enterprise Linux 6 x64 U1/U2/U3/U4/U5
- Red Hat Enterprise Linux 7 x64 Base
- VMware ESX Server 3.5 U4/U5
- VMware ESX Server 4.0/4.0u1/4.0u2/4.1/4.1u1/4.1u2/4.1u3/4.1u4

RAID controller support

You can use the Linux Scripting Toolkit to configure any RAID controller supported by the IBM System x, BladeCenter, iDataPlex, or PureFlex server in which it is installed. For information about supported RAID controller and server combinations, see Storage Controllers and IBM ServerProven.

Fibre Channel HBA support

You can use the Linux Scripting Toolkit to configure any Fibre Channel (FC) HBA supported by the IBM System x, BladeCenter, iDataPlex, or PureFlex server in which it is installed.

For information about supported FC HBA and server combinations, see Shared Storage Adapters and IBM ServerProven.

Chapter 7. Linux Scripting ToolkitServerGuide Scripting Toolkit, Linux Edition utilities and tools

This section contains information about the utilities that are included in the Linux Scripting Toolkit and the tools that are shipped with it. For each utility there is a description of parameters, along with examples.

It also briefly describes the tools shipped with the Scripting Toolkit and provides instructions for using them with the Linux Scripting Toolkit, as well as describes where to get more information.

altiris_sgtklinux.cmd

Purpose

This script builds the IBM specific Linux Pre-installation Environment for Altiris Deployment Solution and integrates the Scripting Toolkit into Altiris Deployment Solution.

Note:

- This script must be executed from the directory in which it resides.
- The Altiris Deployment Solution and Linux Scripting Toolkit must be installed in a path that does not contain parentheses.

altiris_sgtklinux.cmd

Parameters

There are no parameters.

Sample

```
Altiris_SGTKLinux
```

acquire_seps.cmd

This script acquires the SEPs for the specified machine types given and copies them to a location where Scripting Toolkit can use them during deployments.

acquire_seps.cmd

Parameters

machine_type
: Specifies the machine type for which to acquire SEPs. You can specify multiple machine types by separating them with commas.

All
: Acquires all available SEPs.

Sample

```
acquire_seps.cmd 7979

acquire_seps.cmd 7979,4199

acquire_seps.cmd all
```

Linux Scripting ToolkitServerGuide Scripting Toolkit, Linux Edition utilities

This section contains information about the utilities that are included in the Linux Scripting Toolkit. It provides a description of the parameters for each utility, with examples. These utilities are included in the Linux Scripting Toolkit directory structure. The default location is `C:\Program Files\Altiris\eXpress\Deployment Server\sgdeploy\sgtklinux\tk\bin`.

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

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

Note: Some of the information in this documentation is shown on multiple lines due to formatting constraints; however, you must enter all parameters for a utility on a single command line.

HWDETECT

The HWDETECT utility performs basic hardware detection functions that are typically performed by using SMBIOS and a PCI scan. This utility contains options that can be used to either dump all of the hardware information to an output file or can be used to query hardware information and return values that set the *errorlevel* environment variable or the return code, for example `$?`.

HWDETECT includes basic hardware scan functions and more complex PCI device detection options. The basic hardware scan functions can only be used one at a time. The PCI device detection functions, however, can be combined or used more than once on the same command line to produce a query based on multiple restrictions.

Usage:

`hwdetect [-s|-i|-p|--m=`*`machinetype`*`] -f=`*`filename`*

Parameter	Description	Example

`-s`	Determines if the target server is an IBM System x, xSeries, or BladeCenter server. The return values are:
	• **0** for an IBM system
	• **1** for a non-IBM system
		`./hwdetect -s`
		`if [ $? –eq 1 ]; then`
		`  echo "Perform non-IBM equipment specific steps here."`
		
		`else`
		`  echo "Perform IBM equipment specific steps here."`
		`fi`

Parameter	Description	Example
-i	Dumps all available information about the system hardware to the screen in a .ini file format. You can use the **-f** parameter to send this information to a file. A return code of zero indicates success. All other return codes indicate an error.	./hwdetect -i
-f=*filename*	Directs the output to the indicated file. This parameter can be used in conjunction with the **-i** or **-p** parameters. A return code of 254 indicates that HWDETECT was unable to open the specified file.	./hwdetect -i –-f=hwdetect.out cat hwdetect.out | grep "Bus_Number.21 = 41"

--m=*machinetype*	Compares the machine type of the current system to the specified machine type.
	
	Return codes:
	• **0** indicates that the machine types do not match.
	• **1** indicates a match.
		./hwdetect –-m=8676
		if [$? –eq 8676]; then
		  echo "It is an IBM system."
		else
		  echo "It is not an IBM system."

You can also use **HWDETECT** to inventory PCI devices on the target system.

Usage:

hwdetect [--vid=*vendor_id*|--did=*device_id*|--svid=*sub-vendor_id*|
--sdid=*sub-device_id*|--bn=*bus_number*|--dn=*device_number*|--add=*number*]

Parameter	Description	Example
--vid=*vendor_id*	Searches for PCI devices with the indicated hexadecimal vendor ID.	./hwdetect --vid=40 echo "Found $? matches..."
--did=*device_id*	Searches for PCI devices with the indicated hexadecimal device ID.	./hwdetect --did=41 echo "Found $? matches..."
--svid=*sub-vendor_id*	Searches for PCI devices with the indicated hexadecimal sub-vendor ID.	./hwdetect --svid=42 echo "Found $? matches..."
--sdid=*sub-device_id*	Searches for PCI devices with the indicated hexadecimal sub-device ID.	./hwdetect --sdid=43 echo "Found $? matches..."
--bn=*bus_number*	Starts the search at the indicated decimal bus number.	./hwdetect --bn=44 echo "Found $? matches..."
--dn=*device_number*	Starts the search at the indicated decimal device number.	./hwdetect --dn=45 echo "Found $? matches..."
--add=*number*	Adds the specified decimal value to the return value before exiting.	./hwdetect --vid=46 --add=1 echo "Found $? - 1 matches..."

The following example shows a hwdetect.out file created by the –i flag:

```
[System]
Machine_Type=8674
Model_Number=42X
Serial_Number=78Z9506
Product_Name=eserver xSeries 330
BIOS_version=1.04
BIOS_Build_Level=EME112A
BIOS_DATE=06/28/2002
```

```
BIOS_Manufacturer=IBM
BIOS_Language=US
Number_Of_Enclosures=1
Enclosure_Type.0=23
Processor_Slots=2
Active_Processors=1
Processor_Family.0=17
Processor_Speed_MHz.0=1400
Processor_X64 = TRUE
Total_Enabled_Memory_Mb=256
ROM_Diagnostics_Build_Level=EME112A
ISMP_Build_Level=BR8T30A
RSA_Build_Level=GEE834A
System_UUID = 8030E01060F010B010605090D0A020F0
Blade_Chassis_UUID = 0F020A0D0900F00F020A0D0900F00F02
Blade_Slot = 02

[PCI]
Total_Number_Devices=10
Bus_Number.0=0
Device_Number.0=1
Function_Number.0=0
Class_Code.0=0000
Revision.0=0
Header_Type.0=0
Vendor_ID.0=5333
Device_ID.0=8A22
Subvendor_ID.0=1014
Subdevice_ID.0=01C5
Bus_Number.1=0
Device_Number.1=2
Function_Number.1=0
Class_Code.1=0000
Revision.1=0
Header_Type.1=0
Vendor_ID.1=8086
Device_ID.1=1229
Subvendor_ID.1=1014
Subdevice_ID.1=105C
```

Using the –p flag produces the same output with the exception that the section names are tacked onto the beginning of each keyword:

```
System_Machine_Type = 8674
System_Model_Number = 42X
System_Serial_Number = 78Z9506
...
PCI_Bus_Number.0 = 0
PCI_Device_Number.0 = 1
...
```

Notes:

1. The BIOS_DATE value is listed in *mm/dd/yyyy* format.
2. The Enclosure_Type.0=23 is based on SMBIOS 2.3 spec. 23 = Main chassis.
3. There is an entry for Processor_Family and Processor_Speed_MHz for each microprocessor in the server.
4. The ROM_Diagnostics_Build_Level is empty for servers that do not support ROM diagnostics.
5. PCI devices are listed in the order that they are scanned.
6. PCI devices are listed in the *Value.n* format, where *Value* is the variable name and *n* is the nth PCI device scanned.

7. The header_type field is not available for versions of HWDETECT running on Windows 32- or 64-bit operating systems.
8. The vendor, device, subvendor, and subdevice values are in hexadecimal notation.

SAVESTAT

The Savestat utility enables you to store and retrieve up to 20 values to persistent storage. The utility is designed to identify where you left off in an installation script even when a system reboot is required. This utility is designed to return values that set the *?* environment variable so that you can branch in a script (.sh) file based on the result of the utility's execution.

The Savestat utility uses the persistent storage capability of the **ASU** command. Therefore the following files must be available for the script to work:

- ASU package (`ibm_utl_asu_asut69*_linux_x86-64.tgz`)
- `savestat.sh script`
- `savestat.def`

Usage

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

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

Parameter	Description	Usage
`-setn=value`	Saves an integer value, *value*, to the *n*th location in persistent-storage memory, where *n* is an integer from 1-21. Return codes: • `0` if successful • `1` if not successful	`./savestat.sh -setn=value` Where: • *n* is an integer from 1–21 • *value* is an integer from 0–254
`-getn`	Retrieves a value currently set in the *n*th location in persistent-storage memory. Return codes: • The value stored at the location specified by *n*, if successful. • `255` if not successful.	`./savestat.sh -getn` Where *n* is the location of a previously-stored value.
`-signature`	Verifies that the persistent storage contains the **savestat** signature. Return codes: • `0` if storage contains the signature • `1` if storage does not contain the signature	`./savestat.sh -signature`
`-validate`	Verifies that the system is supported by **savestat**. Return codes: • `0` if the system is supported • `1` if the system is not supported	`./savestat.sh -validate`

Parameter	Description	Usage
-q	Invokes the quiet mode, which suppresses prompting. This parameter is optional and can be used with any other **savestat** parameter.	./savestat.sh -q -set1=100

Note: The help for **savestat.sh** indicates that the **–reset** parameter is supported. **savestat.sh** does not currently support the **–reset** parameter. To reset all of the storage locations to zero, use the **savestat.sh –set** command as shown here:

```
savestat.sh --set1=0 --set2=0 --set3=0 --set4=0 --set5=0 --set6=0 --set7=0
--set8=0 --set9=0 --set10=0 --set11=0 --set12=0 --set13=0 --set14=0
--set15=0 --set16=0 --set17=0 --set18=0 --set19=0 --set20=0 --set21=0
```

Examples

The following examples illustrate how to use the Savestat utility.

Example	Description
./savestat.sh -set2=100	Stores the value 100 in the second persistent-storage memory location.

./savestat.sh -get2
 	
if [$? –eq 100]; then
  echo "The value 100 was found successfully."
else
  echo "The value 100 was not found."
fi
	Retrieves the value of the second persistent-storage memory location and branches in the script file according to the value returned.

PRAID

PRAID is a scriptable utility that offers a single user interface for both configuring and replicating all RAID controllers supported by the Linux Scripting ToolkitServerGuide Scripting Toolkit, Linux Edition.

PRAID has three modes of operation:

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

Deploy mode

Used in Deploy mode, PRAID offers the following features:

- Configures all RAID controllers in a server with a single call to the program.
- Automatically resets all RAID controllers to factory-default settings before configuring.
- Uses customizable logic to decide which configuration (policy) is applied to a server based on system hardware. The logic can involve:
  - Machine type of the server
  - Serial number of the server
  - Number of drives connected to the RAID controller

  - RAID controller type
  - Controller number (order) of the RAID controller
- Can be highly customized for specific RAID configurations or generalized to handle many different RAID configurations.
- Provides a default or AUTO mode for automatically creating arrays and logical drives by using default settings. This mode requires no knowledge of the number, size, or location of the drives connected to the RAID controllers.
- Automatically applies default values for any RAID configuration parameters that you do not supply. You supply only the parameters that you want to change.
- Default values for each configuration parameter are equivalent to the default settings of the ServeRAID Manager express configuration method where applicable.
- Allows up to 50 policies for configuring RAID controllers to be specified in a single policies file.

Note:

When using PRAID in Deploy mode, the **-r** parameter is required.

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

For example, `praid -f:policiy.ini -r -y`.

Deploy Mode Examples

```
PRAID -r -d -y
```

This example is useful for unattended scripted installations.

- Configures all RAID controllers in the system using default settings.
- Does not prompt you before setting controllers to factory-default settings.
- Synchronizes drives without prompting, when required.

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

- Configures the RAID controllers in the system by using the policies file: `policies.ini`
- Sets the verbose mode to maximum.
- Returns an error code if there are no matching policies for one or more controllers.

Capture mode

Used in Capture mode, PRAID offers the following features:

- Captures the RAID configurations of all supported controllers to a text file, the policies file, with a common format.
- Captured RAID configurations can be immediately used with PRAID in deploy mode to easily replicate the RAID configuration to many servers.
- Allows customizable logic when saving the captured parameters to determine when each captured configuration must be deployed.
- Saves useful information about each captured configuration, including the system machine type, date, and time when the configuration was captured.

- Allows you to edit any RAID configurations that you capture before deploying them to other systems.

Capture Mode Examples

```
PRAID -c -f:policies.ini
```

Captures the configuration of all RAID controllers into the `policies.ini` file.

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

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

Restore-defaults mode

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

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

Restore-defaults mode example

```
PRAID -r -v:0 -y
```

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

Environment requirements

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

Table 3. Supported RAID adapter information

Adapter	Controller type	Utility
ServeRAID 7t	ServeRAID-7t	arcconf
ServeRAID 8i	ServeRAID-8i	
ServeRAID 8k	ServeRAID-8k	
ServeRAID 8k l	ServeRAID-8k-l	
ServeRAID 8s	ServeRAID-8s	
ServeRAID B5015	ServeRAID-B5015	brcli
LSI SAS 1078 IR	LSI-SAS-1078-IR	cfggen
LSI SAS (1064/1064E/1068/1078)	LSI-SAS-RAID	
LSI SCSI (1020/1030)	LSI-SCSI-RAID	
ServeRAID BR10i	ServeRAID-BR10i	
ServeRAID BR10il	ServeRAID-BR10il	
ServeRAID 7e SATA	ServeRAID-7e-SATA	hrconf
ServeRAID 7e SCSI	ServeRAID-7e-SCSI	
ServeRAID 8e SAS	ServeRAID-8e-SAS	
ServeRAID 8e SATA	ServeRAID-8e-SATA	
ServeRAID 6M	ServeRAID-6M	ipssend

Table 3. Supported RAID adapter information (continued)

Adapter	Controller type	Utility
LSI MegaRAID 8480	LSI-MegaRAID-8480	storcli
ServeRAID C105	ServeRAID-C105	
ServeRAID C100	ServeRAID-M100	
ServeRAID C100 R5	ServeRAID-M100-R5	
ServeRAID M1xxx Series	ServeRAID-M1xxx	
ServeRAID M1xxx Series R5	ServeRAID-M1xxx_R5	
ServeRAID M5014	ServeRAID-M5014	
ServeRAID M5014 R6/R60	ServeRAID-M5014-R6-R60	
ServeRAID M5015	ServeRAID-M5015	
ServeRAID M5015 R6/R60	ServeRAID-M5015-R6-R60	
ServeRAID M5025	ServeRAID-M5025	
ServeRAID-M5025-R6-R60	ServeRAID M5025 R6/R60	
ServeRAID M51xx Series	ServeRAID-M51xx	
ServeRAID M51xx Series R5	ServeRAID-M51xx_R5	
ServeRAID M51xx Series R5/R6	ServeRAID-M51xx_R5_R6	
ServeRAID M51xx Series R6	ServeRAID-M51xx_R6	
ServeRAID MR10i	ServeRAID-MR10i	
ServeRAID MR10ie	ServeRAID-MR10ie	
ServeRAID MR10il	ServeRAID-MR10il	
ServeRAID MR10is	ServeRAID-MR10is	
ServeRAID MR10k	ServeRAID-MR10k	
ServeRAID MR10M	ServeRAID-MR10M	
ServeRAID M5210	ServeRAID M5210	
ServeRAID M5210 R5	ServeRAID M5210 R5	
ServeRAID M1215	ServeRAID-M1215	
ServeRAID M1215 R5	ServeRAID-M1215-R5	
ServeRAID M1210e	ServeRAID-M1210e	
ServeRAID H1110/H1135	SAS2004	sas2ircu

Usage

Each of the modes supported by PRAID (Deploy mode, Capture mode, and Restore-defaults mode) requires a specific syntax, but they all share some common parameters, which are described in the Table 4 on page 48 table that follows.

The sections that follow Table 1 describe each mode and provide a description and usage information for each.

Table 4. PRAID parameters common to multiple modes

Parameter	Description	Usage

`-r:n`
 	
Restore-defaults mode
	Restores the RAID controller with the controller number specified by *n* to factory-default settings and then returns immediately. No RAID configuration is done if you use this parameter.
	
	If no value is specified for the controller number, all RAID controllers are reset to factory-default settings.
	
	Used alone, the parameter provides Restore-defaults mode. You must use this parameter in conjunction with Deploy mode parameters to reset controllers to the factory default settings before deploying a new configuration.
		`praid -r`
		
		Restores all controllers to factory-default settings.
		`praid -r:3`
		
		Restores controller three to factory-default settings. No other controllers are affected.
		`PRAID -f:policies.ini -r -v:5 -e1`
		
		Configures the RAID controllers in the system using the policies file `policies.ini`, sets the verbose mode to maximum, and returns an error code if there were no matching policies for any controllers.

`-f:policies_file`
 	
Specifies the policy file
	The policy file name. This parameter is required for the capture and deploy modes, unless the **-d** parameter is used.
	
	In deploy mode, this points to the policies that you would like PRAID to use when configuring the RAID controllers. You cannot use this parameter with the **-d** parameter.
	
	In capture mode, this points to the file where you would like the captured configurations to be written. If the file does not exist, PRAID will create it. If the file does exist, PRAID appends to the end of it.
	
	The **-f** parameter is valid in deploy and capture modes.
		`praid -f:myfile.ini`
		
		Uses the policies file, `myfile.ini`, to configure all RAID controllers.
		`praid -c -f:myfile.ini`
		
		Captures the RAID configuration of all controllers to the policy file, `myfile.ini`.

`-y` Suppresses prompting	This parameter suppresses the confirmation prompt. If you select the **-y** parameter, PRAID does not prompt you before resetting controllers to factory-default settings. PRAID always resets all controllers to factory-default settings before configuring them. If you do not supply this parameter, PRAID will pause to warn you before resetting the RAID controllers to factory-default settings. The **-y** parameter is valid in deploy and restore-defaults modes.This parameter is optional.	`praid -f:myfile.ini -y` Uses the policies in `myfile.ini` to configure the RAID controllers and does not prompt before resetting all controllers to factory-default settings.

Table 4. PRAID parameters common to multiple modes (continued)

Parameter	Description	Usage
`-e2` Error code 2 if no supported controllers found	Returns an error code of 2 if there were no supported RAID controllers found in the system. By default, PRAID does not return an error if no controllers are found in the system. This parameter is valid in all modes. This parameter is optional.	`praid -c -f:myfile.ini -e2` Captures the RAID configuration of all RAID controllers to the `myfile.ini` file and returns an error if no controllers are found in the system.
`-e3` Error code 3 if no supported drives found	Returns an error code of 3 if at least one controller was found with no drives attached. By default, PRAID does not return an error if no drives are attached to a RAID controller. This parameter is valid in any mode. This parameter is optional.	`praid -d -e3` Configures all RAID controllers with default settings and returns an error if one or more controllers has no drives attached.
`-v:n` Verbose level	Sets the verbosity level, where *n* is: • 0 - quiet • 3 - default • 5 - maximum This parameter is valid in any mode. This parameter is optional.	`praid -d -v:5` Configures all RAID controllers with default settings, and sets the verbose level to maximum.

Deploy mode

The syntax for Deploy mode is:

```
PRAID -f:policies -r -d -p:path -e1 -e2 -e3
-v:n -y -b
```

The parameters unique to Deploy mode are described below.

Table 5. PRAID Deploy mode parameters

Parameter	Description	Usage
`-d` Configure with defaults	Configures all controllers in the system by using default settings instead of using a policies file. The default settings used are the same as the default settings for the policies file. You cannot use this parameter with the **-f** parameter. See "Default RAID levels" on page 63 for the default values that are assigned for each RAID controller based on the number of drives attached to the controller. This parameter is required unless the **-f** parameter is specified.	`praid -d -r` Configures all RAID controllers in the system using default settings.
`-e1` Error if no policy found	Returns an error code of 1 if one or more controllers are not configured due to the fact that there was no policy found to configure them. This parameter is optional.	`praid  -f:policy.ini -r -e1` Configures all RAID controllers using the policies file, `policy.ini`, and returns an error if no matching policy was found.

Capture mode

The syntax for Capture mode is:

```
PRAID -c[:p] -f:policies -e2 -e3 -v:n
```

The parameters unique to Capture mode are described below.

Table 6. Capture mode parameters

Parameter	Description	Usage
`-c[:p]` Capture mode	Indicates capture mode. The `:p` portion is optional. If you do not include the optional portion, `:p` will assume the default value: `t,d`. You can use `:p` to provide a list of parameters describing the **`AppliesTo`** parameter that is created when capturing the parameters to a policy. See "AppliesTo.*n*" on page 55. `:p` is a list containing any of the following: • t – use the type of the RAID controller in the `AppliesTo.1` entry for the policy. • c – use the controller number (scan order relative to all other RAID controllers in the system) in the `AppliesTo.1` entry for the policy. • d – use the number of drives connected to the RAID controller in the `AppliesTo.1` entry for the policy. **Note:** You must specify the name of the policies file by using the **`-f`** parameter when using the **`-c`** parameter. If the file exists, the policy or policies created are appended to the end of the file. If the file does not exist, a new file is created. If there are multiple RAID controllers in the system, their configurations are placed in the file in scan order.	`praid -c:m,t -f:myfile.ini` Captures the configuration of all RAID controllers to the `myfile.ini` file by using the machine type of the server and the RAID controller type as the `AppliesTo.1` entry.

Restore-defaults mode

The syntax for Restore-defaults mode is:

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

Usage examples

The usage examples provide typical ways that PRAID can be used to configure the RAID adapter.

Deploy mode examples

```
PRAID -r -d -y
```

This example is useful for unattended scripted installations.

- Configures all RAID controllers in the system using default settings.
- Does not prompt you before setting controllers to factory-default settings.
- Synchronizes drives without prompting, when required.

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

- Configures the RAID controllers in the system by using the policies file: `policies.ini`
- Sets the verbose mode to maximum.
- Returns an error code if there are no matching policies for one or more controllers.

Capture mode examples

```
PRAID -c -f:policies.ini
```

Captures the configuration of all RAID controllers into the `policies.ini` file.

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

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

Restore-defaults mode examples

```
PRAID -r -v:0 -y
```

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

Return codes

The return codes for PRAID are listed and explained.

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

Policies file

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

You can create a policies file through the following methods:

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

The policies file is an ASCII text file that is organized in a .ini file format. Each .ini file section name indicates the start of a new policy for configuring RAID controllers.

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

Some examples of legal section names are: [Policy.1], [Policy.mypolicy], and [Policy.My-RAID5-config]. Each section in the policies file represents a single policy for configuring RAID controllers. You can have up to 50 policies in a single policies file.

How PRAID selects a policy: Each section in the policies file represents a single policy for configuring the RAID controllers. In configure mode, each RAID controller is configured by a single policy, but a single policy can be used to configure multiple controllers. Each policy in a policies file contains one or more AppliesTo.*n* entries, where *n* is the number of the **AppliesTo** parameter within the policy.

This entry is required in each section, so every section must contain an AppliesTo.1 entry. See "Policies file parameters" for a full description of the AppliesTo.*n* entry.

These entries are followed by a list of hardware parameters, including machine type, number of drives connected to the RAID controller, and scan order, which are evaluated against the current system hardware. If all of the hardware parameters of an AppliesTo.*n* entry match the hardware being evaluated, this policy is used to configure the hardware. For each policy in the policies file, the AppliesTo.*n* entries for that policy are evaluated in order starting with AppliesTo.1.

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

Policies file parameters: This section describes the parameters used in the policies file. The Policy.*name* header and AppliesTo.1 entry are the only parameters required. All values are case-insensitive.

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

In addition to this reference, the ServerGuide Scripting Toolkit also provides the following sample policies files that you can either use directly or use as the basis

for your customized policies files. Arrays are created by using drives that have the same size in MB, which is the default. Each set of drives of the same size are combined into a single array.

Table 7. Sample policies files

File name	Description
default_raid.ini	Creates an AUTO array using drives that have the same size in MB.
RAID0.ini	Creates a single RAID-0 array using all available drives.
RAID1-1.ini	Creates a RAID-1 array using the first two drives, and a RAID-1 array using the second two drives. A single logical drive is created using all available space on each array.
RAID1-5.ini	Creates a RAID-1 array using the first two drives, and a RAID-1 array using all remaining drives. A single logical drive is created using all available space on each array.
RAID1.ini	Creates a single RAID-1 array using the first two drives.
RAID1HSP.ini	Creates a single RAID-1 array using the first two drives and a single hot-spare drive using the third drive.
RAID5.ini	Creates a single RAID-5 array using all available drives.
RAID5HSP.ini	Creates a single RAID-5 array with a single hot-spare drive using all available drives.
RAID10.ini	Creates a single RAID-10 array using all available drives.
RAID50.ini	Creates a single RAID-50 array using all available drives.
RAID60.ini	Creates a single RAID-60 array using all available drives.
RAID6.ini	Creates a single RAID-6 array using all available drives.
RAID6HSP.ini	Creates a single RAID-5 array with a single hot-spare drive using all available drives.
template.ini	Provides a policies file template that contains all parameters, with details about each parameter.

Table 8. Policy file parameters

Keyword	Required?	Default	Description
`Policy.`*`name`*	Yes	None	This header designates the start of a new policy. See "Policy.*name*" on page 55 for additional information.
`AppliesTo.`*`n`*	Yes	None	Use this parameter to describe when the current policy should be chosen to configure the RAID controllers. See "AppliesTo.*n*" on page 55 for additional information.
`ReadAhead`	No	• `ADAPTIVE` (for ServeRAID 6M) • `ON` (for ServeRAID-7t 8i, 8k, and 8k-l)	Specifies the read ahead setting that should be applied to the RAID controller. See "ReadAhead" on page 56 for additional information.
`RebuildRate`	No	`HIGH`	Specifies the rebuild rate that should be applied to the RAID controller. See "RebuildRate" on page 56 for additional information.

Table 8. Policy file parameters (continued)

Keyword	Required?	Default	Description
`StripeSize`	No	• `8` (for ServeRAID 6M) • `64` (for ServeRAID-7t, 8i, 8k, 8k-l,)	Specifies the stripe-unit size in KB that the controller should use for its arrays. See "StripeSize" on page 57 for additional information.
`Array_Mode`	No	`AUTO`	Defines the array-creation policy to use when selecting physical disk drives to include in an array. See "Array_Mode" on page 57 for additional information.
`Array_Defaults`	No	• `0%:1` for ServeRAID-8e-SATA and 8e-SAS, LSI-SCSI-RAID when at least 3 drives are available • `0%:1` for ServeRAID-6M, when one or more arrays has 4 or more physical drives • `0%:0` for all other cases	Defines the default values to use for the variance and number of hot-spare drives when `AUTO` is specified for `Array_Mode`. See "Array_Defaults" on page 57 for additional information.
`Array.`*`letter`*	No	None	Specifies how many arrays are created and the physical drives that you would like in each array. See "Array.*letter*" on page 58 for additional information.
`Hotspares`	No	None	Defines a list of specific physical drives to designate as hot-spare drives. See "Hotspares" on page 58 for additional information.
`Logical_Mode`	No	`AUTO`	Defines the logical-drive creation policy to use when creating logical drives. See "Logical_Mode" on page 59 for additional information.
`Logical_Defaults`	No	`FILL:AUTO:AUTO`	Defines the default logical drive settings that should be used when creating logical drives. See "Logical_Defaults" on page 59 for additional information.
`Logical.`*`num`*	No	None	Specifies the number of logical drives that are created and the parameters for each logical drive. See "Logical.*num*" on page 60 for additional information.

Policy.name:
Description

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

```
[Policy.RAID-5-Hotspare]
```

AppliesTo.n:

Description

Use this parameter to describe when the current policy is chosen to configure the RAID controllers. You can define up to 20 `AppliesTo.`*n* entries per policy. You must have an `AppliesTo.1` entry for each policy; the **AppliesTo.n** parameter is the only required parameter of a policy.

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

- m:*mtype*, where *mtype* is the four digit machine type of an IBM eServer™ or xSeries server.
- s:*serial*, where *serial* is the serial number of an IBM eServer or xSeries server.
- c:*contn*, where *contn* is the controller number (scan order) of the RAID controller with respect to all other RAID controllers in the system.

  The number assigned to a particular controller is dependent on the controller's physical PCI slot and the order in which the system scans its PCI slots.
- t:*ctype*, where *ctype* is the type of the controller. The type is not case-sensitive, and it must be one of the controller types listed in the table of RAID adapters supported by PRAID.
- d:*drives*, where *drives* is an integer value that specifies the number of drives connected to the controller. Only drives in a **Ready** state after resetting the controller to factory-default settings are counted.
- **ALL** Indicates that this policy must be used for all RAID controllers. This parameter is useful when you declare a default policy that is not covered by any of the other policies.

Examples

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

```
AppliesTo.1 = m:8865,t:ServeRAID-7t
AppliesTo.2 = c:1,d:15,s:87R478U
```

Example using the **ALL** parameter:

```
AppliesTo.1 = ALL
```

ReadAhead:
Description

The **ReadAhead** parameter specifies the read ahead setting that must be applied to the RAID controller. If this parameter is not applicable for a RAID controller, it is ignored. See “Supported settings for RAID controllers” on page 60 for the list of ReadAhead settings supported by PRAID for each RAID controller. Possible settings are:

- `Adaptive`
- `On`
- `Off`

Example

```
ReadAhead = On
```

RebuildRate:

Description

The **RebuildRate** parameter specifies the rebuild rate that is applied to the RAID controller. If this parameter is not applicable for a RAID controller, it is ignored. See "Supported settings for RAID controllers" on page 60 for the list of RebuildRate settings supported by PRAID for each RAID controller.

- `High`
- `Medium`
- `Low`

Example

```
RebuildRate = High
```

StripeSize:
Description

The **StripeSize** parameter specifies the stripe-unit size in KB that the controller uses for its arrays. If this parameter is not applicable for a RAID controller, it is ignored. See "Supported settings for RAID controllers" on page 60 for the list of StripeSize settings supported by PRAID for each RAID controller. Possible values are any stripe size supported by the controller.

Example

```
StripeSize = 32
```

Array_Mode:
Description

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

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

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

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

Example

```
Array_mode = CUSTOM
```

Array_Defaults:

Description

The **Array_Defaults** parameter defines the default values to use for the variance and number of hot-spare drives when AUTO is specified for **Array_Mode**. The **Array_Defaults**parameter is not valid if the**Array_Mode** parameter is set to CUSTOM.

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

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

- 0% - Combine only drives with equal size in MB into a single array.
- 5% - Combine all drives within 5% size in MB into a single array.
- 10% - Combine all drives within 10% size in MB into a single array.
- 100% - Combine all drives, regardless of size in MB, into a single array.

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

Example

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

Array.letter:
Description

The **Array.letter** parameter specifies how many arrays are created and the physical drives to include in each array. You can specify the physical drives through any of the following methods:

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

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

Examples

Example using channel number and SCSI ID:

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

Example using integer values:

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

Hotspares:

Description

The **Hotspares** parameter defines a list of physical drives to use as hot-spare drives. You can specify the physical drives by using any one of these methods:

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

Examples

Example using channel number and SCSI ID:

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

Example using integer value:

```
Hotspares = 12, 13
```

Logical_Mode:
Description

The **Logical_Mode** parameter defines the policy to use when creating logical drives. Possible values are:

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

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

You can adjust these default values through the **Logical_Defaults** parameter.

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

Example

```
Logical_Mode = CUSTOM
```

Logical_Defaults:
Description

The **Logical_Defaults** parameter defines the default logical drive settings to be used when creating logical drives. This parameter is only valid when `AUTO` is specified for `Logical_Mode`. Values for this parameter are expressed in the format: *`size:raidlevel:writecmode`*, where:

Size specifies the size of each logical drive. One logical drive is created on each array using the given size. *Size* can take any of the following formats:

- A positive integer – specifies the size in MB.
- A percentage – specifies that a percentage of the total space must be used.
- FILL – indicates that all available space on the array must be used.

Raidlevel specifies the RAID level for the logical drive. See "Supported settings for RAID controllers" for the list of RAID level settings supported by PRAID for each controller.

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

Valid values are:

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

Example

```
Logical_Defaults = 50%:5EE:AUTO
```

Logical.num:
Description

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

Values for this parameter are expressed in the format:
array:size:raidlevel:writecmode where *array* specifies the array letter, and *size*, *raidlevel*, and *writecmode* are as described in "Logical_Defaults" on page 59.

Example

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

Supported settings for RAID controllers: The supported settings for RAID controllers are provided in the following table.

In some cases, the list of supported settings when using PRAID might differ from the supported settings of the RAID controller. These known cases are indicated in the table. For a list of supported settings for each RAID controller when using PRAID, refer to the topic Table 9.

Table 9. Supported settings for each RAID controller when using PRAID. The default settings are underlined.

RAID adapters	Read policy	Write policy	RAID Levels[1]	Stripe Size (KB)
ServeRAID-B5015	• ON • OFF	[n/a]	R1, R5	4, 8, 16, 32, 64, 128, 256, 512, 1024
LSI-IDEal-RAID	[n/a]	[n/a]	R0, R1	32, 64, 128, 256, 512, 1024, 2048, 4096

Table 9. Supported settings for each RAID controller when using PRAID (continued). The default settings are underlined.

RAID adapters	Read policy	Write policy	RAID Levels[1]	Stripe Size (KB)
LSI-MegaRAID-8480	[n/a]	[n/a]	R0, R1, R10, R5, R50	4, 8, 16, 32, 64, 128
LSI-SAS-1078-IR	[n/a]	[n/a]	R0, R1	[n/a]
LSI-SAS-RAID	[n/a]	[n/a]	R0, R1, R1E	[n/a]
LSI-SCSI-RAID	[n/a]	[n/a]	R1	[n/a]
ServeRAID-7t	• ON • OFF • AUTO	• ON • OFF	RVOLUME, R0, R1, R10, R5	16, 32, 64
ServeRAID-8i	• ON • OFF • AUTO	• ON • OFF	RVOLUME, R0, R1, R10, R1E, R5, R50, R5EE, R6, R60	16, 32, 64, 128, 256, 512, 1024
ServeRAID-8k	• ON • OFF • AUTO	• ON • OFF	RVOLUME, R0, R1, R10, R1E, R5, R6	16, 32, 64, 128,256, 512, 1024
ServeRAID-8k-l	• ON • OFF • AUTO	• ON • OFF	RVOLUME, R0, R1, R10	16, 32, 64, 128, 256, 512, 1024
ServeRAID-8s	• ON • OFF • AUTO	• ON • OFF	RVOLUME, R0, R1, R10, R1E, R5, R50, R6	16, 32, 64, 128, 256, 512, 1024
ServeRAID-BR10ie	[n/a]	[n/a]	R0, R1, R1E	[n/a]
ServeRAID-BR10il	[n/a]	[n/a]	R0, R1, R1E	[n/a]
ServeRAID-M1015	[n/a]	[n/a]	R0, R1, R10	8, 16, 32, 64
ServeRAID-M1015–R5	[n/a]	[n/a]	R0, R1, R10, R5, R50	8, 16, 32, 64
ServeRAID-M1xxx	[n/a]	[n/a]	R0, R1, R10	8, 16, 32, 64
ServeRAID-M1xxx_R5	[n/a]	[n/a]	R0, R1, R10, R5, R50	8, 16, 32, 64
ServeRAID-M5014	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50	8, 16, 32, 64, 128
ServeRAID-M5014-R6-R60	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64, 128
ServeRAID-M5015	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50	8, 16, 32, 64, 128
ServeRAID-M5015-R6-R60	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64, 128
ServeRAID-M5025	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50	8, 16, 32, 64, 128, 256, 512, 1024

Table 9. Supported settings for each RAID controller when using PRAID (continued). The default settings are underlined.

RAID adapters	Read policy	Write policy	RAID Levels[1]	Stripe Size (KB)
ServeRAID-M5025-R6-R60	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64, 128, 256, 512, 1024
ServeRAID-M5xxx	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64, 128, 256, 512, 1024
ServeRAID-M51xx	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10	8, 16, 32, 64, 128, 256, 512, 1024
ServeRAID-M51xx_R5	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50	8, 16, 32, 64, 128, 256, 512, 1024
ServeRAID-M51xx_R6	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R6, R60	8, 16, 32, 64, 128, 256, 512, 1024
ServeRAID-M51xx_R5_R6	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64, 128, 256, 512, 1024
ServeRAID-M5210	• ON • OFF • AUTO	• ON • OFF	R0, R1,R10	8,16,32,64
ServeRAID-M5210-R5	• ON • OFF • AUTO	• ON • OFF	R0, R1,R10,R5,R50,R6,R60	8,16,32,64
ServeRAID-M1215	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10	64
ServeRAID-M1215-R5	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50	64
ServeRAID-M1210e	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10	64
ServeRAID-MR10i	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64, 128
ServeRAID-MR10il	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64, 128

Table 9. Supported settings for each RAID controller when using PRAID (continued). The default settings are underlined.

RAID adapters	Read policy	Write policy	RAID Levels[1]	Stripe Size (KB)
ServeRAID-MR10is	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64, 128, 256, 512, 1024
ServeRAID-MR10k	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50, R6, R60	16, 32, 64, 128, 256, 512, 1024
ServeRAID-MR10M	• ON • OFF • AUTO	• ON • OFF	R0, R1, R10, R5, R50, R6, R60	8, 16, 32, 64, 128
ServeRAID-C100	[n/a]	[n/a]	R0, R1, R10	64
ServeRAID-C100-R5	[n/a]	[n/a]	R0, R1, R10, R5	64
ServeRAID-C105	[n/a]	[n/a]	R0, R1,R10	64
SAS2004	[n/a]	[n/a]	R0, R1, R10, R1E	[n/a]

1. RAID levels 5E and 5EE support only one logical drive per array.

Default RAID levels are described in "Default RAID levels."

Default RAID levels: The default RAID level that is applied to a logical drive depends on the number of drives in the array and the controller type. These default values are designed to match the default values of the express configuration method in ServeRAID Manager where applicable. The following table shows the default RAID values that PRAID uses when `AUTO` is specified for *raidlevel*.

Table 10. Default RAID levels

	Drives in array				
Controller	**1**	**2**	**3**	**4**	**5 or more**
ServeRAID-B5015	[n/a]	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare
LSI-IDEal-RAID	[n/a]	RAID 1	[n/a]	[n/a]	[n/a]
LSI-MegaRAID-8480	RAID 0	RAID 0	RAID 0	RAID 0+Hotspare	RAID 0+Hotspare
LSI-SAS-1078-IR	[n/a]	RAID 1	RAID 1+Hotspare	RAID 1+Hotspare	RAID 1+Hotspare
LSI-SAS-RAID	[n/a]	RAID 1	RAID 1E+Hotspare	RAID 1E+Hotspare	RAID 1E+Hotspare
LSI-SCSI-RAID	[n/a]	RAID 1	RAID 1+Hotspare	RAID 1+Hotspare	RAID 1+Hotspare
ServeRAID-7t	RAID 0	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare
ServeRAID-8i	VOLUME	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare
ServeRAID-8k	VOLUME	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare
ServeRAID-8k-l	VOLUME	RAID 1	RAID 1+Hotspare	RAID 10	RAID 10+Hotspare
ServeRAID-8s	VOLUME	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare

Table 10. Default RAID levels (continued)

	Drives in array				
Controller	**1**	**2**	**3**	**4**	**5 or more**
ServeRAID-BR10ie	[n/a]	RAID 1	RAID IE	RAID IE + Hotspare	RAID IE + Hotspare
ServeRAID-BR10il	[n/a]	RAID 1	RAID IE	RAID IE + Hotspare	RAID IE + Hotspare
ServeRAID-M1015	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1015–R5	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1xxx	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1xxx_R5	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5014	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5014-R6-R60	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5015	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5015-R6-R60	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5025	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5025-R6-R60	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M5xxx	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M51xx	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M51xx_R5	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M51xx_R6	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M51xx_R5_R6	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID M5210	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID M5210 R5	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1215	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1215-R5	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-M1210e	RAID 0	RAID 0	RAID 0	RAID 0 + Hotspare	RAID 0 + Hotspare
ServeRAID-MR10i	RAID 0	RAID 0	RAID 0	RAID 0+Hotspare	RAID 0+Hotspare
ServeRAID-MR10il	RAID 0	RAID 0	RAID 0	RAID 0+Hotspare	RAID 0+Hotspare
ServeRAID-MR10is	RAID 0	RAID 0	RAID 0	RAID 0+Hotspare	RAID 0+Hotspare
ServeRAID-MR10k	RAID 0	RAID 0	RAID 0	RAID 0+Hotspare	RAID 0+Hotspare
ServeRAID-MR10M	RAID 0	RAID 0	RAID 0	RAID 0+Hotspare	RAID 0+Hotspare
ServeRAID-C100	RAID 0	RAID 0	RAID 0	RAID 0+Hotspare	RAID 0+Hotspare
ServeRAID-C100-R5	RAID 0	RAID 0	RAID 0	RAID 0+Hotspare	RAID 0+Hotspare
ServeRAID-C105	RAID 0	RAID 0	RAID 0	RAID 0+Hotspare	RAID 0+Hotspare
SAS2004	[n/a]	RAID 0	RAID 0	RAID 10	RAID 10

INVRAID

Use the INVRAID program to dump all of the RAID controller configuration information to an output file. For information about the RAID controllers that are supported by INVRAID, refer to the section Table 3 on page 46.

Environment requirements

INVRAID works by parsing the output of other RAID configuration utilities. To accomplish this, the utilities used by INVRAID must be located in the system search path.

Usage

```
invraid [-I | -P] -F
```

Table 11. INVRAID parameters

Parameter	Description
`-I`	Displays information about all host adapters in the system in an .ini file format.
`-P`	Dumps information about all host adapters in a system in a keyword=value format.
`-F :`*`filename`*	Directs the output of `invraid` to the specified file.

Return values

Table 12 lists the values returned by INVRAID.

Table 12. Values returned by INVRAID

Return Value	Description
0	Success
1	Syntax Error
2	Program Error

Examples

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

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

Returns:

```
[System]
Machine_Type = 7233
Serial_Number = 23A0075
Total_Number_Of_Controllers = 2

[RAIDController.1]
Model = LSI-SAS-1078-IR
BIOSVersion = 6.22.00.00
FirmwareVersion = 1.25.82.00
DriverVersion =
RebuildRate = HIGH
StripeSize =
ReadAhead = ADAPTIVE
PCI = 4:0:0:1000:0062:FFFF:FFFF
```

```
[RAIDController.1.Array]
Total_Number_Of_Arrays = 1
ID.1 = A
Members.1 = 1,2

[RAIDController.1.Hotspares]
Total_Number_Of_Hotspares = 0

[RAIDController.1.Logical]
Total_Number_Of_Logicals = 1

Array.1 = A
Size.1 = 139236
Raid_Level.1 = 1
WriteCache.1 = AUTO
State.1 = Okay (OKY)
Derived_State.1 = GOOD

[RAIDController.1.Physical]
Total_Number_Of_Physicals = 4

Channel.1 = 1
ID.1 = 0
Size.1 = 140013
Type.1 = SAS
Serial_Number.1 = 3NM2SQED0000980322JB
State.1 = Online (ONL)
Derived_State.1 = GOOD

Channel.2 = 1
ID.2 = 1
Size.2 = 140013
Type.2 = SAS
Serial_Number.2 = 3NM223CV0000974732Y9
State.2 = Online (ONL)
Derived_State.2 = GOOD

Channel.3 = 1
ID.3 = 2
Size.3 = 140013
Type.3 = SAS
Serial_Number.3 = 3NM2000900009746H8BY
State.3 = Ready (RDY)
Derived_State.3 = GOOD

Channel.4 = 1
ID.4 = 3
Size.4 = 140013
Type.4 = SAS
Serial_Number.4 = 3NM23J1J00009746XNSB
State.4 = Ready (RDY)
Derived_State.4 = GOOD

[RAIDController.2]
Model = ServeRAID-MR10M
BIOSVersion = 2.02.00
FirmwareVersion = 1.40.12-0551
DriverVersion =
PCI = 30:0:0:1000:0060:1014:0379

[RAIDController.2.Array]
Total_Number_Of_Arrays = 0

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

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


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

Using the -p parameter returns the same information, but the section title from the properties file is shown for each value:

```
invraid -p -f:myfile.ini
```

Returns:

```
System_Machine_Type = 7233
System_Serial_Number = 23A0075
RAIDController.1.Model = LSI-SAS-1078-IR
RAIDController.1.BIOSVersion = 6.22.00.00
RAIDController.1.FirmwareVersion = 1.25.82.00
RAIDController.1.DriverVersion =
RAIDController.1.RebuildRate = HIGH
RAIDController.1.StripeSize =
RAIDController.1.ReadAhead = ADAPTIVE
RAIDController.1.PCI = 4:0:0:1000:0062:FFFF:FFFF

RAIDController.1.Array.ID.1 = A
RAIDController.1.Array.Members.1 = 1,2

RAIDController.1.Logical.Array.1 = A
RAIDController.1.Logical.Size.1 = 139236
RAIDController.1.Logical.Raid_Level.1 = 1
RAIDController.1.Logical.WriteCache.1 = AUTO
RAIDController.1.Logical.State.1 = Okay (OKY)
RAIDController.1.Logical.Derived_State.1 = GOOD

RAIDController.1.Physical.Channel.1 = 1
RAIDController.1.Physical.ID.1 = 0
RAIDController.1.Physical.Size.1 = 140013
RAIDController.1.Physical.Type.1 = SAS
RAIDController.1.Physical.Serial_Number.1 = 3NM2SQED0000980322JB
RAIDController.1.Physical.State.1 = Online (ONL)
RAIDController.1.Physical.Derived_State.1 = GOOD

RAIDController.1.Physical.Channel.2 = 1
RAIDController.1.Physical.ID.2 = 1
RAIDController.1.Physical.Size.2 = 140013
RAIDController.1.Physical.Type.2 = SAS
RAIDController.1.Physical.Serial_Number.2 = 3NM223CV0000974732Y9
RAIDController.1.Physical.State.2 = Online (ONL)
RAIDController.1.Physical.Derived_State.2 = GOOD

RAIDController.1.Physical.Channel.3 = 1
RAIDController.1.Physical.ID.3 = 2
RAIDController.1.Physical.Size.3 = 140013
RAIDController.1.Physical.Type.3 = SAS
RAIDController.1.Physical.Serial_Number.3 = 3NM2000900009746H8BY
RAIDController.1.Physical.State.3 = Ready (RDY)
RAIDController.1.Physical.Derived_State.3 = GOOD

RAIDController.1.Physical.Channel.4 = 1
RAIDController.1.Physical.ID.4 = 3
RAIDController.1.Physical.Size.4 = 140013
RAIDController.1.Physical.Type.4 = SAS
RAIDController.1.Physical.Serial_Number.4 = 3NM23J1J00009746XNSB
RAIDController.1.Physical.State.4 = Ready (RDY)
RAIDController.1.Physical.Derived_State.4 = GOOD
```

```
RAIDController.2.Model = ServeRAID-MR10M
RAIDController.2.BIOSVersion = 2.02.00
RAIDController.2.FirmwareVersion = 1.40.12-0551
RAIDController.2.DriverVersion =
RAIDController.2.PCI = 30:0:0:1000:0060:1014:0379
```

VALRAID

VALRAID is a utility program that can be used to validate policy files against inventory files generated by the INVRAID utility.

The VALRAID utility has two modes of operation:

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

Simulation mode

Used in simulation mode, VALRAID simulates the effect that a policy file has on a RAID configuration if it is applied by using the PRAID utility. This capability can be used when creating PRAID policy files. The policy files can be tested without running PRAID on the target system.

Check mode

Used in check mode, VALRAID determines if the policy file specified matches the RAID configuration represented in the inventory file. Use this capability in operating system deployment scripts to bypass the RAID configuration step if the controller is already configured with the required RAID configuration. With this process, you can avoid restarting the system before installing the operating system. To indicate that the policy file does not match the configuration represented by the inventory file, VALRAID sets the return code to 20.

Usage

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

The simulation mode syntax is:

```
valraid -ini:input_inventory_file -inp:input_policy_file -outi:output_inventory_file
-outp:output_policy_file -raid:inifiles
```

The check mode syntax is:

```
valraid -c -ini:input_inventory_file -inp:input_policy_file -raid:inifiles
```

Table 13. VALRAID parameters

Parameter	Description	Example
`-ini:input_inventory_file`	Specifies the input inventory file. Generate the inventory file by running INVRAID against a target system.	`valraid -ini:myfile.inv -inp:policy.ini -outi:newfile.inv -outp:newpolicy.ini -raid:/inifiles`

Table 13. VALRAID parameters (continued)

Parameter	Description	Example
`-inp:`*`input_policy_file`*	Specifies the input policy file.	`valraid -ini:myfile.inv -inp:policy.ini -outi:newfile.inv -outp:newpolicy.ini -raid:/inifiles`
`-outi:`*`output_inventory_file`*	Specifies the file name for the output inventory file. This inventory file represents the RAID configuration that would result from using the PRAID utility to apply *input_policy_file* to the system described in *input_inventory_file*. This option is valid only for simulation mode.	`valraid -ini:myfile.inv -inp:policy.ini -outi:newfile.inv -outp:newpolicy.ini -raid:/inifiles`
`-outp:`*`output_policy_file`*	Specifies the file name for the output policy file. This file can be applied to a target system by using the PRAID utility. This option is valid only for simulation mode.	`valraid -ini:myfile.inv -inp:policy.ini -outi:newfile.inv -outp:newpolicy.ini -raid:/inifiles`
`-raid:`*`inifiles`*	Specifies the directory that contains the RAID .ini files.	`valraid -ini:myfile.inv -inp:policy.ini -outi:newfile.inv -outp:newpolicy.ini -raid:/inifiles`
`-c`	Specifies check mode. Check mode compares the configuration from *input_inventory_file* to the configuration represented in *input_policy_file*. The default is simulation mode.	`valraid -c -ini:myfile.inv -inp:policy.ini -raid:/inifiles`

Return codes

VALRAID uses the following return codes:

- 0 – The execution was successful.
- 1 – An error occurred while parsing the input policy file.
- 2 – An error occurred while parsing the input inventory file.
- 3 – The controller is not supported.
- 4 – The RAID level is not supported.
- 5 – The Stripesize is not supported.
- 6 – The number of arrays is not supported.
- 7 – The number of drives in array is not supported.
- 8 – The number of logical volumes in array is not supported.
- 9 – There are not enough drives to create a hotspare.

- 10 – There are not enough drives of the same size.
- 11 – An error occurred while opening the input policy file.
- 12 – An error occurred while opening the input inventory file.
- 13 – An error occurred while opening the output inventory file.
- 14 – An error occurred while writing to the output inventory file.
- 15 – An error occurred while opening the output policy file.
- 16 – An error occurred while writing the output policy file.
- 17 – Partial drive sizing is not supported.
- 18 – Command line syntax error
- 19 – No policy match
- 20 – Controller not configured, does not match policy file

Tools included with the Linux Scripting ToolkitServerGuide Scripting Toolkit, Linux Edition

The Linux Scripting Toolkit includes several IBM system configuration tools that make the Toolkit more functional. This section describes the additional tools provided by this release of the Linux Scripting Toolkit:

- Advanced Settings Utility
- QAUCLI
- UpdateXpress System Pack Installer

Advanced Settings Utility

For convenience, the Linux Scripting ToolkitServerGuide Scripting Toolkit, Linux Edition includes the Advanced Settings Utility (ASU). You can use the ASU to modify firmware settings from the command line on multiple operating system platforms.

The Linux Scripting Toolkit uses a subset of the ASU functions to capture and deploy firmware settings as part of your scripted deployments.

Usage

This section describes the ASU functions used by the Linux Scripting Toolkit.

Table 14. ASU functions in Linux Scripting Toolkit

Command	Description
`asu show all`	Displays and captures BIOS settings. You can use redirection to store this output in a file as shown here: `asu.exe show bios > bios_settings.ini`

Table 14. ASU functions in Linux Scripting Toolkit (continued)

Command	Description
`asu save filename`	Applies CMOS settings from a file. ASU looks for the file name specified by *filename* and reads the contents. If the contents are valid CMOS settings, they are applied, one line at a time, to the server. The following example applies the settings captured above: `asu save bios_settings.ini` **Note:** Only settings captured from an identical model can be replicated, due to difference in BIOS settings and valid values between models.
`asu set IMM.HostIPAddress IP address`	Sets the external IP address in the Integrated Management Module (IMM) to the specified address. This setting is part of the IMM group.
`asu set IMM.LandOverUsb enabled|disabled –kcs`	Enables or disables the IMM LAN over USB interface. **Note:** When you enable or disable this setting, you must use the KCS interface to ensure that the **asu** command completes correctly and returns status.

QAUCLI

You can use the QAUCLI utility to configure Fibre Host Bus Adapters (HBAs). A 32–bit version of this utility comes with the Linux Scripting ToolkitServerGuide Scripting Toolkit, Linux Edition. You can download this utility from QLogic at http://www.qlogic.com.

Usage

Table 15. QAUCLI usage

Command	Description
`qaucli -pr fc –e (view | ?)`	Shows the current boot device information on all HBAs

Table 15. QAUCLI usage (continued)

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

Examples

The following examples illustrate how to use the QAUCLI utility.

Note: Some of these examples are broken across multiple lines due to formatting constraints. When using QAUCLI, you must enter all of the parameters on a single line.

Example	Description
`qaucli -pr fc -e view`	Displays the current boot device information on all HBAs.

`qaucli -pr fc –e`
`E0-FF-EE-DE-CD-34-56-30`
`E0-00-ED-DE-CD-34-56-30`
`E0-10-ED-DE-CD-34-56-30 1 prim`
	Configures HBA E0-FF-EE-DE-CD-34-56-30 E0-00-ED-DE-CD-34-56-30 E0-10-ED-DE-CD-34-56-30 to boot from the primary target.

Example	Description
`qaucli -pr fc –e E0-FF-EE-DE-CD-34-56-30 view`	Displays the current boot setting information for HBA port E0-FF-EE-DE-CD-34-56-30.
`qaucli -pr fc –e E0-FF-EE-DE-CD-34-56-30 disable prim`	Clears the selected boot device setting on HBA port E0-FF-EE-DE-CD-34-56-30.
`qaucli -pr fc –l E0-FF-EE-DE-CD-34-56-30`	Displays information about the LUNs attached to HBA port E0-FF-EE-DE-CD-34-56-30.

SEPTool941

For convenience, the Linux Scripting ToolkitServerGuide Scripting Toolkit, Linux Edition, includes the SEPTool.

The SEPTool downloads System Enablement Packs (SEPs) from the IBM web site. With it, you can acquire additional System Enablement Packs (SEPs) that are needed.

Usage

This section describes the SEPTool941 functions.

Table 16. SEPTool functions in Linux Scripting Toolkit

Command	Description
SEPTool941 acquire	Downloads SEPs from the IBM website.

UpdateXpress System Pack Installer

For convenience, the Linux Scripting ToolkitServerGuide Scripting Toolkit, Linux Edition includes the UpdateXpress System Pack Installer (UXSPi)UXSPi to help you acquire updates to include in your deployment scenarios. The UpdateXpress System Pack Installer is located at `...sgdeploy\updates\uxsp`.

The UpdateXpress System Pack Installer can perform these functions:

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

For more information about running the UpdateXpress System Pack Installer, change directory to the UXSPi directory and run the UXSPi executable command shown below:

```
./ibm_utl_uxspi_x.xx_anyos_x86-64.bin -update -help
```

Usage

The Linux Scripting Toolkit uses the UXSPi in the update mode to acquire and deploy device drivers and firmware as part of Linux Scripting Toolkit deployments. This section details the command-line options for the uxspi -update mode.

Table 17. UXSPi update mode options

Option	Description
–firmware	The firmware option forces UXSPi to install only firmware updates.
–drivers	The driver option forces UXSPi to install only driver updates.
-f *update_ids*, –force=*update_ids*	Specifies that UXSPi use the unattendedForcedInstallCommandLine field in the update XML rather than the unattendedInstallCommand field.
-s *update_ids* –select=*update_ids*	The select option deploys the specified set of updates to the target system even if the system version is newer than the version in the update package. Use this option to roll back firmware and driver levels where necessary.
-l *update_xml_path*, –local=*update_xml_path*	Specifies the file name of a local UXSP XML file or the path to search for one.
-n, –new	Selects all updates that are newer than the current system versions or not currently installed on the system.
-e *update_ids*, –exclude=*update_ids*	Excludes the specified update IDs. You can provide multiple IDs in a comma-separated list.
-i *update_ids*, –include= *update_ids*	Includes the updates specified in the list of update IDs. You can provide multiple IDs in a comma-separated list.
–ignore-undected=*update_ids*	Specifies not to apply the indicated update IDs. You can provide multiple IDs in a comma-separated list.
-L, –latest	The default behavior of UXSPi is to apply the latest UXSPi update pack found in the UXSPi directory. This option forces UXSPi to install the latest updates whether they are from an update pack, are individual updates, or are a combination of the two.
–remote=*remote_address*	Runs the update command on the remote server specified by *remote_address*.
–remote-user=*remote_user*	Specifies the remote user ID to use when connecting to a remote system specified with –remote.
–remote-password=*password*	Sets the password for the user ID specified by –remote-user.
--remote-dir=*directory*	Specifies the staging or working directory on the remote system.
–noinventory	Causes UXSPi to gather only the machine type and operating system information without performing an inventory of existing updates.
–nouxsp	Prevents UXSPs from being deployed.
-r, –report	Displays a summary report of updates used in the compare step.

Example

The following example can be used to specify an UpdateXpress System Pack XML file named `uxsp.xml` located in the same directory as the UXSPi executable.

```
./ibm_utl_uxspi_x.xx_anyos_x86-64.bin update –l uxsp.xml
```

Chapter 8. Incorporating the Scripting Toolkitwith your existing process

To incorporate the Scripting Toolkit procedures into an existing deployment process, use the HWDETECT utility to determine if the combined process is being executed on Scripting Toolkit supported hardware. You can then add appropriate branches in the batch files to use either the existing process or the Scripting Toolkit process.

For example, you might use the `-s` option of HWDETECT to determine if the current system is an IBM server:

```
hwdetect -s
if errorlevel 1 goto NONIBM
if errorlevel 0 goto IBM

:NONIBM
rem Perform non-IBM equipment specific processing here.

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

Chapter 9. IBM ServerProven compatibility

The IBM ServerProven web site provides valuable information about selected products for compatibility with IBM System x, BladeCenter, and xSeries servers. For current information about compatibility with operating systems, configuration, and hardware options, see the IBM ServerProven web site.

Chapter 10. Hints and tips

This section contains information about known problems and limitations, best practices, and hints and tips for using the Linux Scripting ToolkitServerGuide Scripting Toolkit, Linux Edition.

Performing PXE deployments by using the Linux Scripting Toolkit

To perform a PXE deployment by using the Linux Scripting Toolkit, you must first configure the TFTP server on the source server and update the Toolkit Preferences page with the IP address of the TFTP server.

When you use the Linux Scripting Toolkit to create PXE image deployments based on the provided Boot Media Profiles, the files are placed in the `/tftpboot` directory. For example, to apply a PXE deployment image created from a Boot Media Profile called **`PXE_test`**, you must follow these steps:

1. In the Linux Scripting Toolkit, select **Create Boot Media** . When the process is complete, the following directory structure is created in the `/tftpboot` directory:

   /tftpboot/
   /tftpboot/lnxtoolkit
   /tftpboot/lnxtoolkit/pxelinux.cfg
   /tftpboot/lnxtoolkit/pxelinux.cfg/PXE_test
   /tftpboot/lnxtoolkit/PXE_test
   /tftpboot/lnxtoolkit/PXE_test/tc.zip
   /tftpboot/lnxtoolkit/PXE_test/img2a
   /tftpboot/lnxtoolkit/PXE_test/tcrootfs
   /tftpboot/lnxtoolkit/PXE_test/img3a
   /tftpboot/lnxtoolkit/bsb1.lss
   /tftpboot/lnxtoolkit/pxelinux.0
   /tftpboot/lnxtoolkit/bsb.msg
   /tftpboot/pxelinux.cfg

2. Examine the contents of the configuration file. As a rule, no changes should be required. In the following example, the configuration file is `/tftpboot/lnxtoolkit/PXE_test`. The contents will be similar to the following:

   ```
   prompt 0
   default toolscenter
   timeout 100
   label toolscenter
   display bsb.msg
   kernel /PXE_test/img2a
   append initrd=/PXE_test/img3a vga=0x317 root=/dev/ram0 rw ramdisk_size=100000
   tftp_server=192.168.0.1 tftp_tcrootfs=/lnxtoolkit/PXE_test/tcrootfs
   tftp_tczip=/lnxtoolkit/PXE_test/tc.zip debug_level=1
   silent_boot=no boot_src=4 tftp_blksize=1420 media_boot=no
   ```

3. Copy the contents of the configuration file to the default file:

   ```
   cp /tftpboot/lnxtoolkit/pxelinux.cfg/PXE_test /tftpboot/lnxtoolkit/pxelinux.cfg/default
   ```

4. Using the IP address of your server, ensure that the DHCP configuration contains a block similar to this example:

```
if substring(option vendor-class-identifier, 0,9) = "PXEClient" {
  filename "lnxtoolkit/pxelinux.0";      # file to be served
  next-server 192.168.0.1;               # This server's ipaddress
}
```

After you have completed these steps, any system within the DHCP server network can start this generated PXE image.

Performing a PXE deployment to a specific device

To perform a PXE deployment to a specific target, you must have the MAC address of the target. Using that address, follow these steps:

1. In the Linux Scripting Toolkit, select **Create Boot Media** . When the process is complete, the following directory structure is created in the `/tftpboot` directory:

   ```
   /tftpboot/
   /tftpboot/lnxtoolkit
   /tftpboot/lnxtoolkit/pxelinux.cfg
   /tftpboot/lnxtoolkit/pxelinux.cfg/PXE_test
   /tftpboot/lnxtoolkit/PXE_test
   /tftpboot/lnxtoolkit/PXE_test/tc.zip
   /tftpboot/lnxtoolkit/PXE_test/img2a
   /tftpboot/lnxtoolkit/PXE_test/tcrootfs
   /tftpboot/lnxtoolkit/PXE_test/img3a
   /tftpboot/lnxtoolkit/bsb1.lss
   /tftpboot/lnxtoolkit/pxelinux.0
   /tftpboot/lnxtoolkit/bsb.msg
   /tftpboot/pxelinux.cfg
   ```

2. Examine the contents of the configuration file. As a rule, no changes should be required. In the following example, the configuration file is `/tftpboot/lnxtoolkit/PXE_test`. The contents will be similar to the following:

   ```
   prompt 0
   default toolscenter
   timeout 100
   label toolscenter
   display bsb.msg
   kernel /PXE_test/img2a
   append initrd=/PXE_test/img3a vga=0x317 root=/dev/ram0 rw ramdisk_size=100000
   tftp_server=192.168.0.1 tftp_tcrootfs=/lnxtoolkit/PXE_test/tcrootfs
   tftp_tczip=/lnxtoolkit/PXE_test/tc.zip debug_level=1
   silent_boot=no boot_src=4 tftp_blksize=1420 media_boot=no
   ```

3. Change to the `/tftpboot/lnxtoolkit/pxelinux.cfg/` directory and create a symbolic link using the MAC address of the target system to point to the bootable media configuration file:

   ```
   ln -s PXE_test 01-00-14-5e-b5-4a-7e
   ```

4. Ensure that the rest of the DHCP configuration contains a block similar to this example:

   ```
   host mymachine {
       hardware ethernet 00:14:5e:b5:4a:7e;
       option domain-name-servers 192.168.0.1;   # DNS server
       fixed-address 192.168.0.2;                # Target system IP
       filename "lnxtoolkit/pxelinux.0";         # file to be served
       next-server 192.168.0.1;                  # This server's IP
   }
   ```

Installing network operating systems using HTTP

To install network operating systems using HTTP, Linux Scripting Toolkitrequires that you enable directory browsing on the web server. If enabling browsing presents a security risk, you can provide the information needed to perform installations by adding and setting the following environment variables in the Altiris jobs:

```
TK_DISTRO_HTTP_KERNEL_PKG
TK_DISTRO_HTTP_RELEASE_PKG
TK_DISTRO_HTTP_ISOLINUX
TK_DISTRO_HTTP_INITRD
TK_DISTRO_HTTP_VMLINUX
```

This example shows the values for these variables for a Red Hat Enterprise Linux 5, Update 4 x64 installation:

```
TK_DISTRO_HTTP_KERNEL_PKG=Server/kernel-2.6.18-160.el5.x86_64.rpm
TK_DISTRO_HTTP_RELEASE_PKG=Server/redhat-release-5Server-5.4.0.2.x86_64.rpm
TK_DISTRO_HTTP_ISOLINUX=isolinux/isolinux.cfg
TK_DISTRO_HTTP_INITRD=isolinux/initrd.img
TK_DISTRO_HTTP_VMLINUX=isolinux/vmlinuz
```

Improving PXE boot transfer speed

To speed the TFTP transfer of the preboot environment, increase the packet size for the TFTP server of Altiris. To increase the packet size, complete the following steps:

1. Open the PXE configuration utility.
2. Click the **Multicast** tab.
3. Change the packet size to 1420 (the default packet size is 768).

If the system is uEFI-based, see "Disabling uEFI PXE to decrease network boot time" for more ways to increase performance.

Disabling uEFI PXE to decrease network boot time

To improve the time it takes to start the network for uEFI-based systems, follow these steps:

1. Start the system.
2. Press **F1** to display the menu options.
3. Navigate to **System Settings** > **Network** > **PXE Configuration**.
4. Select Port **%MAC1%**.
5. Select **Enable PXE**, and press **Enter**.
6. Select **Legacy Support**, and press **Enter**.
7. Select **Save Changes**, and press **ESC**.
8. Select Port **%MAC2%**.
9. Select **Enable PXE**, and press **Enter**.
10. Select **Legacy Support**, and press **Enter**.
11. Select **Save Changes**, and press **ESC**.

Performing clone and deployment of SUSE Linux Enterprise Server 11

To perform a clone of SUSE Linux Enterprise Server 11 with Altiris, you must use Altiris Deployment Solution 6.9 SP3. To deploy a cloned image of SUSE Linux Enterprise Server 11 from Altiris, refer to Altiris Knowledgebase article 48984 to enable the necessary support.

After enablement, complete the following steps to deploy the SUSE Linux Enterprise Server 11 cloned image:

1. On the Altiris console, navigate to **Deploy Linux TK**.
2. In the Distribute Disk Image Task of the Deploy Linux TK job, clear the option **Automatically perform configuration task after completing this image task.**

   Note: If you do not clear the check box, the network cannot be configured.

Using RapiDeploy with SUSE Linux Enterprise Server 10 and VMware

RapiDeploy provides native imaging support for EXT2 and EXT3 file systems. By default, SUSE Linux Enterprise Server 10 (SLES 10) uses the reiserfs file system when installing. To clone and redeploy a SLES 10 image using the Toolkit jobs, you must perform the original deployment using an EXT3 file system.

To change the file system prior to installation with the Toolkit, edit the sles10.xml AutoYast file provided by the Toolkit.

Replace this partitioning section of the AutoYast file:

```
<partitioning config:type="list">
 <drive>
  <initialize config:type="boolean">true</initialize>
  <use>all</use>
 </drive>
</partitioning>
```

with the following partitioning section:

```
<partitioning config:type="list">
 <drive>
  <initialize config:type="boolean">true</initialize>
  <use>all</use>
  <partitions config:type="list">
   <partition>
    <filesystem config:type="symbol">ext3</filesystem>
    <format config:type="boolean">true</format>
    <mount>/boot</mount>
    <size>100mb</size>
   </partition>
   <partition>
    <filesystem config:type="symbol">swap</filesystem>
    <format config:type="boolean">true</format>
    <mount>swap</mount>
    <size>auto</size>
   </partition>
   <partition>
    <filesystem config:type="symbol">ext3</filesystem>
    <format config:type="boolean">true</format>
    <mount>/</mount>
    <size>auto</size>
```

```
   </partition>
  </partitions>
 </drive>
</partitioning>
```

Linux X server considerations

There are special considerations for using Linux X server with a Remote Supervisor Adapter II (RSA II) port. If you are using this configuration, consider the following items:

If the Remote Supervisor Adapter II-EXA is installed on a server that is running either the Red Hat Enterprise Linux (RHEL) or SUSE Linux operating system, make sure that the Linux operating system is selected in the Remote Supervisor Adapter II settings in the server BIOS. To set Linux as the operating system in the server BIOS, complete the following steps:

1. Either start or restart the server.
2. When prompted, press **F1** to display the configuration menu.
3. Click **Advanced Setup** > **ASM Settings**.
4. In the **OS USB** field, select Linux.
5. Select **Save Values and Reboot ASM**.

Note: If you run an automated X Windows system configuration utility, repeat these configuration changes.

Install the operating system in text mode. Set the color depth to 16–bit and the screen resolution to 1024 x 768.

If SUSE Linux or Red Hat Enterprise Linux is already installed and configured to run in text mode, and will never use the X Window system, no additional configuration is required for the RSA II-EXA to function correctly.

The Remote Supervisor Adapter II-EXA requires a Video Electronics Standard Association (VESA) device driver. The VESA video device driver enables the remote control screen and the local screen to display the same information (clone mode).

When using power management, the video output might not return correctly from some power saving states. To correct this problem, use the **xset** command to disable DPMS: `xset -dpms`

For more information, see *IBM Remote Supervisor Adapter II-EXA Technical Update for Linux Operating Systems* available from ftp://ftp.software.ibm.com/systems/support/system_x_pdf/88p9275.pdf.

Special considerations for BladeCenter Blades and Linux X server configuration

After installing BladeCenter blade and Linux X servers, do not change the monitor configuration or any other graphical settings. If you must change the graphics settings, enter the following command to start the configuration utility:

```
sax2 -m 0=fbdev
```

When using power management, the video output might not return correctly from some power saving states. To correct this problem, use the **xset** command to disable DPMS as shown: `xset -dpms`

Booting from a USB key

To boot from a USB key, the key must be configured for IBM ServerGuide Scripting Toolkit, Linux Edition deployment. For more information about configuring a USB key for deployments, see "Creating bootable media from a workflow" on page 87.

BIOS settings for booting from a USB key are system-specific. Refer to the documentation for your systems for the correct BIOS settings and procedures to boot from USB keys.

Some systems support booting from USB keys by pressing **F12** during startup. This method is the recommended one to use the Linux Scripting Toolkit to deploy from a USB key. uEFI-based systems only support booting from a USB key using **F12**.

IPv6 compliance

Beginning in version 2.20, the Linux Scripting Toolkit provides support for locating operating system files on IPv6 network shares. To use operating system files located in a share on an IPv6 network, you must update the variable **TK_DISTRO_PATH** in the appropriate Altiris job.

Performing network based installations of SLES11 SP1 using static IPv6 addresses

When performing a network-based installation of SLES11 SP1 in a static IPv6 environment, you must use either **ipv6=1**, which accepts both IPv4 and IPv6 addresses, or **ipv6only=1**, which accepts only IPv6 addresses, as a boot parameter.

For example, to configure static IPv6 addresses for an IPv6-only network, use these boot parameters:

```
ipv6only=1  netdevice=eth0  hostip=2000::2dae:2390/64
```

During the **Perform installation** stage of the deployment, the system will restart. You might receive the error message: `Download (curl) error ... /media.1/media` after the system starts. To avoid this problem, follow these steps:

1. Navigate to another console that has a command prompt.
2. Configure the network by issuing this command:

   ```
   ifconfig eth0 inet6 add 2000::2dae:2390/64
   ```
3. Return to the console with the error.
4. Click **Retry** to continue the installation.

Enabling Linux Scripting Toolkit PXE images to work with other PXE images

The Linux Scripting Toolkit uses a customized `pxelinux.0` file rather than the default file that comes with syslinux. If you already have a PXE server in your network and want to use PXE images generated by the Linux Scripting Toolkit with other PXE images, you must implement a PXE chain. To implement a PXE chain, perform these steps:

1. Download syslinux 3.72 or higher from http://www.kernel.org/pub/linux/utils/boot/syslinux/.
2. Copy the file `core/pxelinux.0` from the `syslinux` directory structure to your `tftproot` directory.
3. Extract the file `pxechain.com` from a PXE image created with the Linux Scripting Toolkit. The `pxechain.com` file is located in `/tftpboot/lnxtoolkit/`*`image_name`*`/tc.zip`. For example, if you have created a PXE image called `PXE_test` using the Linux Scripting Toolkit, you can extract `pxechain.com` by using the following command:

   ```
   unzip /tftpboot/lnxtoolkit/PXE_test/tc.zip
   ```
4. Copy the `pxechain.com` file to your `tftproot` directory.
5. Copy `/tftpboot/lnxtoolkit/pxelinux.0` to your `tftproot` directory.
6. Create a subdirectory of your `tftproot` directory called `ibm`.
7. Copy the PXE files created by the Linux Scripting Toolkit into the `tftproot/ibm` directory.
8. Create a subdirectory of `tftproot` called `pxelinux.cfg`.
9. Create the file `tftproot/pxelinux.cfg/default`. This sample `default` file includes the PXE image created by the Linux Scripting Toolkit. You can add other existing PXE images as shown:

   ```
   prompt 0
   default ibmchain
   timeout 100
   label ibmchain
   kernel pxechain.com
   append ::ibm/pxelinux.0
   label your_other_pxe
   kernel pxechain.com
   append ::your_other_pxe/pxelinux.0
   ```

When you have completed these steps, the `tftproot` file structure looks like this:

```
|-- ibm
|   |-- img2a
|   |-- img3a
|   |-- pxelinux.0      <- IBM's modified pxelinux.0
|   |-- pxelinux.cfg
|   |   `-- PXE_test     <- The default file for Linux Scripting Toolkit created PXE Image
|   |-- tc.zip
|   `-- tcrootfs
|-- your_other_pxe
|   |-- vmlinux
|   |-- initrd.gz
|   |-- pxelinux.0      <- your_other_pxe's pxelinux.0
|   `-- pxelinux.cfg
|     `-- default       <- your_other_pxe's default
|-- pxechain.com        <- pxechain.com from tc.zip (Step 1)
|-- pxelinux.0          <- pxelinux.0 from syslinux 3.72 (or later)
`-- pxelinux.cfg
    `-- default         <- default file for pxechain.com
```

Creating bootable media from a workflow

To deploy a workflow to a target server, you must create bootable media. This topic provides the steps for creating a deployment image on boot media.

Before you begin

Before you can create boot media, you must have created a workflow to be deployed on the boot media.

Procedure

1. From the main menu, select **Bootable Media Profiles**.
2. Click **Create** to create a boot media profile.
3. Enter a name for the profile.
4. From the drop-down menu, select a workflow to be deployed on the boot media.
5. From the drop-down menu, select a boot method. Supported methods are:
   - **USB** - Creates a boot image that is deployed from a USB key.
   - **ISO** - Creates an ISO image to be burned to a CD or DVD for deployment.
   - **PXE** - Creates a boot image to be deployed from a network share.
6. Click **Next** to select TCP/IP configuration options for the target server. If you want to use a static network configuration for the target server, enter the configuration information on this panel.
7. Click **Next** to select the machine types for this deployment.
8. Click **Next**. If all of the required UXSPs and SEPs are available or you did not elect to validate their availability, you can review your selections.
9. When you are satisfied with your selections, click **Create Boot Media** to begin creating the media. When prompted, provide the path information for the media you selected.

Results

The boot media you selected is created.

What to do next

To begin the deployment, start your target server from this media.

Known problems and limitations

This section provides information and alternative solutions for known problems and limitations of the Linux Scripting Toolkit.

Integration script `altiris_sgtklinux.cmd` fails with error code 16

When running the **`altiris_sgtklinux.cmd`** command, it fails with the following error:

```
[Altiris_PerformUpgradeSteps.cmd]
Running
"C:\Program Files (x86)\Altiris\eXpress\Deployment Server\pxe\pxeconfig.exe"
-delete "IBM Linux Toolkit 1.00"
"C:\Program Files (x86)\Altiris\eXpress\Deployment Server\pxe\pxeconfig.exe"
-delete "IBM Linux Toolkit 1.00" returned a non-zero error code
     ReturnCode = 16
    The Altiris_PerformUpgradeSteps.cmd script was aborted.
```

To correct this problem, restart the Windows operating system on which the Altiris Deployment Server and the PXE server are running.

Integration script `altiris_sgtklinux.cmd` fails with error code 10

When running the **`altiris_sgtklinux.cmd`** command, it fails with the following error:

```
[Altiris_CreatePXEImage.cmd]
Running
"E:\Program Files\Altiris\eXpress\Deployment Server\pxe\pxeconfig"
-create -linux "IBM Linux Toolkit 1.00" -x64 -defaultLinux -oem "ibm"
-save -notmanaged
     "C:\Program Files\Altiris\eXpress\Deployment Server\pxe\pxeconfig"
-create -linux "IBM Linux Toolkit 1.00" -x64 -defaultLinux -oem "ibm"
-save -notmanaged returned a non-zero error code
     ReturnCode = 10
    The Altiris_CreatePXEImage.cmd script was aborted.
```

This problem can occur when the Altiris Linux x64 Automation Environment is not properly installed by the **altiris_sgtklinux.cmd** script.

To correct this problem, manually install the Altiris Linux x64 Automation Environment (BDCgpl_6.9.*xxxx*.frm) then try running the script again.

To manually install the Altiris Linux x64 Automation Environment:

1. Start the PXE Configuration Utility from the Altiris Console.
2. Click **Boot** > **New**.
3. Select the **Linux** radio button, and click **Add pre-boot**.
4. Select the row **Linux X64** from the table.
5. Enter the path to the BDCgpl_6.9.*xxxx*.frm file.
6. Click **Next**, and follow the on-screen instructions.

Operating system installation halts after reboot when using LSI SAS RAID controller

Some combinations of LSI SAS RAID controllers and operating systems might experience a system halt after rebooting during an operating system installation. The affected operating systems are:

- SLES 10
- SLES 11
- RHEL 5
- VMware 4

in combination with one of these RAID controllers:

- LSI-SAS-1078-IR
- LSI-SAS-(1064,1068)
- ServeRAID-BR10i
- ServeRAID-BR10ie

This problem occurs when the server has a drive that is not part of a RAID array and is not configured as a hot spare. The problem is caused by the ordering of Linux mptsas devices.

The following example depicts the problem. A system has four drives with two configured in a RAID 1 array, one configured as a hot spare, and one outside the array. The BIOS sees the drive outside the array, /dev/sda, as HDD1. The RAID array, /dev/sdb, is treated as HDD0. The operating system installation puts the boot files on /dev/sda, the drive outside the array, but after the reboot, the installation looks to HDD0 for the boot files.

To work around this problem, use one of these options:

- Do not configure RAID.
- Change the RAID configuration so that all drives are included in a RAID array.
- Remove the drive outside the RAID array from the controller.
- Modify the boot order of the system to point to the drive outside the array instead of the array.

UpdateXpress System Pack Installer returns errors when supported hardware is not present

Deployment tasks that include installation of UpdateXpress System Packs (UXSPs) will return errors if the hardware supported by the UXSPs is not present in the target system. These errors can be safely ignored.

Unattended Linux installation requests network device

When performing unattended Linux operating system installations, the process might pause to ask which network device to use if there are multiple devices available. To avoid this problem, you can add a kernel parameter to specify the desired network device during the workflow creation process.

In the **OS install** section of the workflow, a field is provided for optional kernel parameters.

The kernel parameter varies by operating system:

- For Red Hat and VMware: `ksdevice=`*`eth`*, where *eth* is the network device to use. For example eth0, eth1, and so on.
- For SUSE Linux: `netdevice=`*`eth`* where *eth* is the network device to use. For example eth0, eth1, and so on.

Unattended file not found during installation of SLES on uEFI systems

When using Linux Scripting Toolkit to install SLES on a uEFI based system, the installation task might be unable to find the answer file, causing the installation to attempt to continue in manual mode.

To resolve this issue, perform these steps:

1. Edit the workflow for your installation.
2. In the **OS install** section of the workflow, add `brokenmodules=usb_storage` to the optional kernel parameters.
3. Save the workflow.
4. Create bootable media from the workflow, and perform the installation.
5. After the installation is complete, edit the file `/etc/modules.d/blacklist`. It is recommended that you make a copy of this file before editing it.
6. Remove the line `blacklist usb_storage`.

This limitation affects the following systems:

- System x3400 M2, types 7836 and 7837
- System x3500 M2, type 7839
- System x3550 M2, types 7946 and 4198
- System x3650 M2, types 7947 and 4199
- System x iDataplex dx360 M2 types 7321, 7323 and 6380

- BladeCenter HS22, types 7870 and 1936

Altiris clone of image fails on SLES 11

Altiris RapiDeploy does not support the cloning of SLES 11 for capturing or deploying images, which causes this job to fail. To avoid this problem, use the **-raw** parameter for the **Capture Linux Image**and **Deploy Linux Image** jobs.

You must add **-raw** to the **Additional Parameters line** of both the **Capture Linux Image** and **Deploy Linux Image** jobs. When using this parameter, the operation will take significantly longer. You must also ensure that enough free space is available on the target location to save the images.

VMware 4 installations using Altiris fail

The *computername* variable used in VMware installations must contain only letters, digits, and hyphens. If any other characters are detected, the installation will halt.

By default, the Altiris Console supplies the *compname* variable used by the console as the *computername* variable for VMware installations. Therefore to successfully use Altiris to install VMware, you must ensure that the *compname* variable used by the Altiris console is allowable as the VMware *computername* variable.

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

The ServeRAID BR10i adapter is not supported on the iDataPlex dx360 M2 with 12 Bay Storage Chassis, machine type 7321.

RAID configuration fails for LSI SATA RAID

When performing RAID configuration to configure an LSI 1064/1064e SATA controller, you might receive error code 7 or 11. This error is caused when the cfggen utility is unable to remove or create a configuration on SATA drives larger than 250 GB.

To avoid this problem, remove any logical volumes, including RAID arrays, on the adapters by using the Ctrl + C menu on system POST prior to using Linux Scripting Toolkit.

Incorrect association of OS unattended files for SUSE Linux Enterprise Server x64

During the OS Install step in the workflow creation process, the operating system repositories for SLES 10x64 and SLES 11x64 are associated with the 32–bit versions of the unattended files by default. This can cause the installation to fail or the operating system to installed without the correct packages.

To avoid this potential problem, you must manually associate the correct operating system unattended files with the operating system repositories when creating a workflow to install SLES 10 x64 or SLES 11 x64. The correct file associations are shown below.

Operating System	Unattended File Name
SUSE Linux Enterprise Server 10 x64	sles10x64.xml

Operating System	Unattended File Name
SUSE Linux Enterprise Server 10 x64 with Xen	sles10x64_xen.xml
SUSE Linux Enterprise Server 11 x64	sles11x64.xml
SUSE Linux Enterprise Server 11 x64 with Xen	sles11x64_xen.xml

Default Fibre Configurations not supported on Emulex Fibre HBAs

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

All values are case sensitive. You must ensure that the configured values are identical to the adapter values with regard to case.

ASU configuration fails for Load Defaults

When performing ASU configuration to load the system defaults, you might receive an error code of 45. This error is caused when the ASU utility is unable to load defaults for the `ISCSI.InitiatorName` setting. This limitation affects the following systems:

- System x3200 M3, types 7327 and 7328
- System x3250 M3, types 4251, 4252, and 4261
- System x3400 M2, types 7836 and 7837
- System x3500 M2, type 7839
- System x3550 M2, types 7946 and 4198
- System x3650 M2, types 7947 and 4199
- System x iDataplex dx360 M2 types 7321, 7323 and 6380
- BladeCenter HS22, types 7870 and 1936

To avoid this problem, create a new `asu.ini` file with these contents:

```
loaddefault uEFI
loaddefault SYSTEM_PROD_DATA
loaddefault BootOrder
  loaddefault IMM
```

VMware ESX 4 installation requires a minimum of 4 GB of memory

When performing an installation of VMware ESX 4, ensure that the target system has a minimum of 4 GB of memory.

PXE will not bind to port 4011 on a Windows 2008 server

When Altiris Deployment Server is installed on a Windows 2008 server, you might encounter this problem with PXE-E55 on the target system. Instructions for avoiding the problem are included in this document:Altiris Deployment Solution from Symantec 6.9 SP2 Release Notes

Altiris clone of image only supports Ext2/Ext3 file system

Altiris RapiDeploy only supports the cloning of the ext2/ext3 file system, which means that the job "Capture Linux Image" or "Capture RAID Configuration and Linux Image" may fail for a Linux operating system with a non-ext2/ext3 file system.

Altiris deployment of clone image fails on SLES 10.x and 11.x

When using the Toolkit for Altiris to run either "Deploy Linux Image" or "Deploy RAID Configuration and Linux Image" to deploy SUSE Linux Enterprise Server 10.x or11.x, you may see an error message similar to the following:

```
Waiting for device  /dev/disk/by-id/xxxxxxxx to appear:
    ..................................... not found
    --exiting to  /bin/sh
```

To avoid the problem, before running "Capture Linux Image" or "Capture RAID Configuration and Linux Image" to capture the SUSE Linux Enterprise Server 10.x or 11.x, you need to boot into the system that you want to capture and perform the following steps:

1. Call up **YaST2** > **System** > **Partitioner**.
2. Type `yast bootloader` (or `yast2 bootloader` for the graphical version).
3. For every boot menu entry, change the root device to match what was put into `/etc/fstab` by YaST Partitioner.
4. Reboot the system to see if it boots properly. If so, the system should be ready for cloning.

If you would rather make the changes manually instead of by using YaST, follow these steps:

1. From the root directory, open the files `/etc/fstab`and `/etc/zipl.conf` in a text editor.
2. Replace all entries, such as `root=/dev/disk/by-id/[partition]` with `root=/dev/disk/by-path/[partition]`. If you are unsure about the existing partitions, use the **`ls -l`** command to view the directory entries in `/dev/disk/by-path`.
3. After making the changes to `/etc/fstab` and `/etc/zipl.conf`, run the commands **`mkinitrd`** and **`zipl`**. If you do not, the system will not boot.

For more details, refer to the Novell Knowledgebase document 3580082 athttp://www.novell.com/support/kb/doc.php?id=3580082.

VMWARE ESX requires that NUMA system memory be balanced

VMWare installations may fail to load the VMkernel when Non-Uniform Memory Access (NUMA) is enabled and each processor does not have memory in its adjoining memory banks.

VMware ESX Server 4.1 installation hangs at "Starting vmkernel initialization"

When installing VMware ESX Server 4.1 on a system with a MAX5 memory expansion module, the installation might hang on this screen. This issue can occur on the following systems:

- BladeCenter HX5, type 7872
- System x3690 X5, types 7148, 7149

- System x3850 X5, type 7145

To avoid this problem, add the kernel parameter **allowInterleaveNUMAnodes=TRUE** to the kickstart file and the Altiris job.

To update the kickstart file, follow these steps:

1. Navigate to `..sgdeploy\sgtklinux\altiris\answerfiles\` and make a copy of the existing `esx4.ks` kickstart file.
2. Edit `esx4.ks` to modify the line:

   ```
   bootloader --location=mbr
   ```

   to be:

   ```
   bootloader --location=mbr --append="allowInterleavedNUMAnodes=TRUE"
   ```

To modify the Altiris job, follow these steps:

1. In the **Customize Job Variables** task of the Altiris job, click **Modify** to edit the task.
2. Edit the line:

   ```
   #TK_DISTRO_KPARAM=
   ```

   by uncommenting it and adding the kernel parameter **allowInterleavedNUMAnodes=TRUE** so that the line is now:

   ```
   TK_DISTRO=KPARAM="allowInterleavedNUMAnodes=TRUE"
   ```
3. Save your changes.

uEFI operating system installations do not boot from hard drive

During native uEFI operating system installations, the target system might fail to boot from the hard drive after Linux Scripting Toolkit processes are complete. This can occur if the target system does not automatically boot the .efi file (`bootx64.efi` for RHEL6 or `elilo.efi` for SLES11) from the drive.

The solution to this problem is dependent on the operating system. Consult the operating system information for instructions about adding a new boot option entry for the .efi file.

For example, to correct this problem on most IBM systems, you can create a new boot entry for the .efi file and continue the installation using that option. Follow these steps to create a new boot entry for the .efi file:

1. Power on the system, and, press **F1** to enter setup.
2. Select **Boot Manager**.
3. Select **Add Boot Option**.
4. Select the boot entry that includes string "*.efi"
5. Enter the description as `OS_Install` and select **Commit Changes**.

Follow these steps to continue the installation:

- Power on the system, and press **F1** to enter setup.
- Select **Boot Manager**.
- Select **Boot from File**.
- Select the GUID Partition Tables (GPT) System Partition with the name `OS_Install`.
- Select **EFI**.

- Select **Boot**.
- Select **efi file**.

Note: If the installation completes and the system does not boot to the operating system, go to the **Start Options** section of the setup menu and select the boot entry for the operating system

Appendix A. 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 System x 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 Portal at http://www.ibm.com/support.
- Go to the IBM Support website 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 System x 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 access publications through the IBM Publications Center at http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi.

Getting help and information from the World Wide Web

The IBM web site has up-to-date information about IBM System x and IntelliStation products, services, and support.

- IBM System x information: http://www.ibm.com/systems/x/
- IBM IntelliStation: http://www.ibm.com/systems/intellistation/pro/index.html
- Service information for IBM products, including support options: http://www.ibm.com/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 the IT Services web page at http://www.ibm.com/services/. For support telephone numbers outside of the U.S. and Canada, go to http://www.ibm.com/planetwide/. In the U.S. and Canada, call 1-800-IBM-SERV (1-800-426-7378).

Hardware service and support

You can receive hardware service through IBM Services or through your IBM reseller, if your reseller is authorized by IBM to provide warranty service. For support telephone numbers outside of the U.S. and Canada, seehttp://www.ibm.com/planetwide/. 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 B. Notices

This book contains the following notices designed to highlight key information:

- **Note:** These notices provide important tips, guidance, or advice.
- **Important:** These notices provide information or advice that might help you avoid inconvenient or difficult situations.
- **Attention:** These notices indicate possible damage to programs, devices, or data. An attention notice is placed just before the instruction or situation in which damage could occur.

Edition notice

Trademarks

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

BladeCenter	IntelliStation
e-business logo	ServeRAID
eServer	ServerGuide
IBM	ServerProven
IBM (logo)	xSeries
TotalStorage	System x

Intel and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

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

UNIX is a trademark of The Open Group in the United States, other countries, or both.

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

Important notes

When referring to processor storage, real and virtual storage, or channel volume, KB stands for approximately 1000 bytes, MB stands for approximately 1,000,000 bytes, and GB stands for approximately 1,000,000,000 bytes.

When referring to hard disk drive capacity or communications volume, MB stands for 1,000,000 bytes, and GB stands for 1,000,000,000 bytes. Total user-accessible capacity might vary depending on operating environments.

IBM makes no representation or warranties regarding non-IBM products and services that areIBM ServerProven, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. These products are offered and warranted solely by third parties.

IBM makes no representations or warranties with respect to non-IBM products. Support (if any) for the non-IBM products is provided by the third party, not IBM.

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

Index

U

V

W

Printed in USA