IBM ServerGuide Scripting Toolkit, Linux Edition

User's Reference

Version 9.21

IBM ServerGuide Scripting Toolkit, Linux Edition

User's Reference

Version 9.21

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

Contents

Chapter 1. Introducing the ServerGuide Scripting Toolkit, Linux Edition

The ServerGuide Scripting Toolkit, Linux Edition (Linux Scripting Toolkit) enables you to tailor and build custom hardware deployment solutions. It provides hardware configuration and Linux operating system (OS) installation for IBM® System x®, BladeCenter®, and iData-Plex hardware.

The Linux Scripting Toolkit uses a console to simplify the steps in creating, customizing, and deploying hardware configurations and Network Operating System (NOS) deployments. Using the Linux Scripting Toolkit, you can create a bootable ISO image, USB key, or PXE boot image that supports the following types of deployment:

- Policy-based RAID configuration using pRAID
- Cloning of RAID configuration
- Configuration of system settings using the Advanced Settings Utility (ASU)
- Configuration of Fibre Channel Host Bus Adapters (HBAs)
- Firmware updates using 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
  - SUSE Linux Enterprise Server 11 x64 Base/SP1/SP2
  - 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
  - Red Hat Enterprise Linux 5 x64 U1/U2/U3/U4/U5/U6/U7/U8
  - Red Hat Enterprise Linux 6 32 bit U1/U2
  - Red Hat Enterprise Linux 6 x64 U1/U2
  - VMware ESX Server 3.5 U4/U5
  - VMware ESX Server 4.0/4.0u1/4.0u2/4.1/4.1u1/4.1u2
- 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
  - Red Hat Enterprise Linux 6 x64 U1/U2
- Remote Supervisor Adapter II (RSA II) and BladeCenter Management Module and Advanced Management Module remote disk scenarios

- Installation of IBM Systems Director Agent integrated with scripted NOS deployment.
- Remote deployment via Integrated Management Module (IMM).

Chapter 2. Installing the Linux Scripting Toolkit

This section explains how to install and start the Linux Scripting Toolkit on the supported operating systems.

This section describes:

Hardware and software requirements for the source server

This topic lists the hardware and software requirements for the Linux Scripting Toolkit source server.

Hardware requirements

The Linux Scripting Toolkit requires a PC-compatible computer with the following attributes to act as a source server:

- 512 MB of memory
- Sufficient disk space to store operating system files, applications, updates, and configuration files

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

Software requirements

The Linux Scripting Toolkit source server requires the following software:

- A supported operating system. The following operating systems are supported by the Linux Scripting Toolkit for use as source servers:
  - SuSE Linux Enterprise Server 11
  - SuSE Linux Enterprise Server 10 SP2 or higher
  - Red Hat Enterprise Linux 5 U2 or higher
  - Red Hat Enterprise Linux 6 U1
- Media creation software to burn created ISO images to disc
- Application sofware:
  - Net File Share
  - Firefox 2.00.14 or higher
- Linux software packages:
  - Python 2.4.2 or higher
  - Python-xml 2.4.2 or higher (for SUSE Linux)
  - NFS-utils 1.0 or higher

Installing, updating, and removing the Linux Scripting Toolkit

This section describes the process for installing, updating, and removing the Linux Scripting Toolkit, and for installing the Boot Media Creator (BoMC).

Installing the Linux Scripting Toolkit

The Linux Scripting Toolkit is available for download from http://www.ibm.com. Before installing, you must download the file ibm_utl_sgtklnx_x.xx_linux_32-64.rpm and make it accessible to the source server.

To install the Linux Scripting Toolkit package for the first time, follow these steps:

1. Download the latest version of the rpm from http://www.ibm.com/systems/support/.
2. Open a command line terminal.
3. Change the current directory to the location of the toolkit rpm file.
4. Run the following command: `rpm –ivh ibm_utl_sgtklnx_x.xx_linux_32-64.rpm`.

By default, the Linux Scripting Toolkit is installed to `/opt/ibm/sgtk`. To change the path to a different location, use the **-relocate** rpm option. For example, to relocate to `/usr/local/sgtk`:

```
rpm -ivh -thaqnrelocate /opt/ibm/sgtk=/usr/local/sgtk ibm_utl_sgtklnx_x.xx_linux_32-64.rpm
```

Updating the Linux Scripting Toolkit

To update the Linux Scripting Toolkit, follow these steps:

1. Download the latest version of the rpm from http://www.ibm.com/systems/management.
2. Open a command window.
3. Change directory to the location of the rpm file.
4. Issue the following commands:

   ```
   rpm -e ibm_utl_sgtklnx
   rpm -ivh ibm_utl_sgtklnx_x.xx_linux_32-64.rpm
   ```

5. Start the Linux Scripting Toolkit console: `./opt/ibm/sgtk/sgtklinux/sgtklinux.sh`.

Note: The `rpm -U` option is not supported when updating the Linux Scripting Toolkit. If you have already created boot images such as ISO or PXE images, these images will not be updated during the upgrade process.

Removing the Linux Scripting Toolkit

You can remove the Linux Scripting Toolkit using the following command:

```
rpm –e ibm_utl_sgtklnx
```

Starting the Linux Scripting Toolkit console

Start the Linux Scripting Toolkit console by invoking the sgtklinux script as shown:

```
./opt/ibm/sgtk/sgtklinux/sgtklinux.sh
```

The first time you use the Linux Scripting Toolkit, you enter the initial configuration wizard, which guides you through the process of acquiring the pre-installation environment, repository configuration, and network setup.

For information about valid parameters for sgtklinux.sh, use the `--help` parameter: `sgtklinux.sh --help`.

Acquiring the IBM Linux pre-installation environment

The Linux Scripting Toolkit provides a means to acquire the IBM Linux pre-installation environment in the **Boot Environment** step of the initial configuration wizard during setup. If this method is unable to download the environment, or the source server does not have access to the Internet, you can use this procedure to acquire the pre-installation environment

Before you begin

To manually acquire the pre-installation environment, you must have a workstation with access to both the Internet and the Linux Scripting Toolkit source server.

About this task

To manually acquire the pre-installation environment, use the IBM ToolsCenter Bootable Media Creator, included with the Linux Scripting Toolkit. This procedure describes the process for running the Bootable Media Creator from a workstation with access to the Internet and copying it to the source server.

Procedure

1. Copy the version of `ibm_utl_bomc` for your Linux distribution and system architecture from `/opt/ibm/sgtk/wui/bin` to a workstation with access to the IBM website.
2. On the workstation, run the Bootable Media Creator as shown here.

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

   The Bootable Media Creator acquires the pre-installation environment zip file, `ibm_utl_boot_tools-140_anyos_x86-64-full.zip` and stores it in the location indicated. In this example the location is C:\temp.
3. Copy the file to a location on the source server, for example: `/root/ibm_utl_boot_tools-140_anyos_x86-64-full.zip`.
4. Start the console using the following command:

   ```
   /opt/ibm/sgtk/sgtklinux.sh
   ```
5. At the prompt for the new boot environment file:
   a. Choose **Local** as the retrieval method.
   b. For the new boot environment file path, enter `/root/ibm_utl_boot_tools-140_anyos_x86-64-full.zip`.

Updating the IBM Linux pre-installation environment

Updating the IBM Linux pre-installation environment allows you to use pre-installation environments that have been released since the most current release of the Toolkit.

The Linux Scripting Toolkit provides a means of updating the IBM Linux pre-installation environment in the Main Menu with the **Boot Environment** option. When you select this option, the window displays the current boot environment file, which is the pre-installation environment, along with the option to update the boot environment file. You can update the file using **Download** from IBM.com, or

Local from a boot environment file stored on the server. If the Download option fails, see "Acquiring the IBM Linux pre-installation environment" on page 5 for information on how to get the boot environment file.

Note: You can only use this option to update the boot environment file to a more recent version. If you need to roll back the boot environment file, you must manually remove the newer version and replace it with an older version. By default, the boot environment files are located in the following folder:

```
/opt/ibm/sgtk/sgdeploy/sgtklinux/boot/
```

Configuring an NFS server for deployments

This section describes the process for configuring the source server for the Linux Scripting Toolkit.

About this task

To perform network deployments, you must configure the Network File System (NFS) server on the source server to work properly with the Toolkit. The NFS server allows you to share files from the source server across your network. All NFS server exports must be defined in the `/etc/exports` file. Follow this procedure to add the values required by the Toolkit to this file.

Procedure

1. Edit the file `/etc/exports`. Include the following line:

   ```
   /opt/ibm/sgtk/sgdeploy *(ro,sync,no_root_squash,no_all_squash)
   ```

   This will export the `/opt/ibm/sgtk/sgdeploy` directory for any host with read-only permissions. The base directory that you define in the `/etc/exports` file must correspond to the value in the Preferences page of the Toolkit.
2. Restart the NFS daemon.
   - For Red Hat:

     ```
     # /sbin/service nfs restart
     ```
   - For SUSE:

     ```
     # service nfsserver restart
     ```

Results

The files in the base directory are now available for use by hosts across your network.

Performing the initial configuration

This section describes how to use the Initial Configuration wizard to set the console preferences the first time you use the Toolkit.

About this task

The first time you start the Toolkit, you are presented with the Initial configuration page. You can also edit this configuration at any time by clicking **Toolkit Preferences** from the main menu.

To configure the Toolkit, follow these steps.

Procedure

1. Start the Toolkit by following these steps:
   a. Open a terminal window.
   b. Change directory to `/opt/ibm/sgtk/`.
   c. Run the following command: `./sgtklinux.sh`. This command starts the Linux Scripting Toolkit Console using the Firefox browser. If this is the first time that you have started the Toolkit, the Initial Configuration wizard opens.
2. Select the method for retrieving the boot environment file and click **Next**. You can choose to download the boot environment file or to use a locally stored version. The default is **Download**. If you select **Local**, you must supply the location of the local boot environment file (ibm_utl_boot_tools-*xxx*_anyos_x86–64–full.zip), where *xxx* is the version number.
3. Configure the current repository for tasks and workflows. Select **Create** to create a new repository. This is the default. When you create a new repository, you are prompted for a destination for the repository. This is the directory that will be exported when following the steps in "Configuring an NFS server for deployments" on page 6.

   To use an existing repository, select **Re-use an existing repository** and select the repository.
4. Set the network preferences. You must set network preferences to perform network deployments. Configure the following settings:

   Network sharing
   This setting must be enabled to perform network deployments. Enabling **Network sharing** populates the current network settings.

   By default, the **Path** field contains the same path you provided when creating the new repository. This must also be the same path used in "Configuring an NFS server for deployments" on page 6.

   NFS is the only supported protocol.

   Proxy settings
   If you connect to the network via proxy, enter your proxy settings here.

   PXE settings
   Enter the location information that the target servers will use to boot using PXE images. The default location is `/tftpboot`.
5. To apply the settings, click **Next**.
6. To complete the wizard and return to the home screen, click **Finish**.

Results

After you have completed these steps, you can begin using the Toolkit. You can change these selections at any time by selecting **Toolkit Preferences** from the main menu.

What to do next

After you have configured the Toolkit, you can configure an NFS server to share the files in the repository required to perform network deployments using the Toolkit.

Chapter 3. Preparing the Linux Scripting Toolkit

Before you begin using the Linux Scripting Toolkit to create deployments, you must acquire the latest firmware updates and determine the location of the operating system and post-installation files that are required by the deployments. This section lists the files and information you need and describes the process for acquiring them.

To create deployments using the Linux Scripting Toolkit, you must have the following:

- The location of the IBM Systems Director Agent files if the deployment includes installation of the IBM Systems Director Agent. You can download these files from the following locations:
  - The IBM Director 5.x Agent is available from http://www.ibm.com/systems/management/director/downloads.html.
  - The IBM Systems Director 6.x and 6.1.x agents are available from http://www.ibm.com/systems/management/director/downloads/agents.html.
- The location of the operating system files to be used in the deployment if the deployment includes operating system installation.
- The latest UpdateXpress System Pack (UXSP) to ensure that the operating system installation includes the most recent firmware and driver updates. "Acquiring UpdateXpress System Packs" describes the process for using the Linux Scripting Toolkit to acquire these updates.
- Adding support for new machine types. See "Extending supported systems" on page 10 for information on using the Linux Scripting Toolkit to acquire SEPs or on using the "Update system only" option.

Acquiring UpdateXpress System Packs

To ensure that the operating system files used in your deployments include the latest driver and firmware updates, you must acquire UpdateXpress System Packs (UXSPs). Use the **Updates** task to retrieve UXSPs. These updates are deployed during the **Update firmware** task.

UpdateXpress System Packs are integration-tested bundles of firmware and device driver updates for System x and BladeCenter servers. The **Updates** task helps you obtain the latest UXSPs for your systems. You can download new UXSPs from IBM.com, or if your source server is not connected to the Internet, you can acquire the updates manually and use this task to add them to your repository.

Follow these steps to complete the **Updates** task:

1. From the main menu, click **Updates**.
2. Click **Acquire new UXSPs** to begin.
3. From the **Source media** section, select a source:
   - **Acquire from IBM website**
   - **Acquire from local folder** - If you choose this option, you must provide the path to the local folder. The path is case-sensitive.
4. Click **Next.**

5. In the **Systems** section, select the systems for which you want to acquire updates from the **Available options** list and click **Add selected** to add them to the **Chosen options** list.
6. In the **Categories** section, check the **Download firmware updates for the following systems** box, and then click **Next.**
7. In the **OS** section, select the operating systems being used by the servers you selected and click **Add selected** to add them to the **Chosen options** list, then click **Next**.
8. The **Summary** section displays a summary of the chosen options. Click **Finish** to begin downloading. You can view the process using the **Running Tasks** option in the menu.
9. When the download is complete, click **Updates** to view the list of UXSPs.

Note: These versions of UpdateXpress System Pack Installer (UXSPi) are no longer included in the Linux Scripting Toolkit:

- `uxspi`*`xxx`*`.rhel3`
- `uxspi`*`xxx`*`.rhel4`
- `uxspi`*`xxx`*`.sles9`
- `uxspi`*`xxx`*`.exe`

If you are using an operating system supported by these versions of the installer, you must acquire the appropriate UpdateXpress System Pack Installer when acquiring UpdateXpress System Packs.

Extending supported systems

You can add new supported systems using the update system list only or using System Enablement Packs (SEPs).

Follow these steps to complete the Supported Systems task:

1. From the main menu, click **Supported Systems**.
2. Click **Update supported systems** to begin.
3. On the first section to update system list only, check the check box to skip this step and then click **Next**.
4. On the second section to acquire new SEPs:
   a. Enter a machine type for which to acquire SEPs. For example, 7979.
   b. From the **Source media** section, select one of these sources and then click **Next**:
      - Acquire from the IBM website
      - Acquire from local folder. If you choose this option, you must also enter the path to the local folder. The path is case-sensitive.

   The Summary section displays a summary of the chosen options.
5. Click **Finish** to begin downloading. To view the download process, on the menu, select **Running Tasks**.
6. When the download is complete, click **Supported Systems** to view the list of supported systems.

Note: If you choose to skip the two options, nothing occurs and the list of supported systems does not change.

Adding a new supported system using SEP without Internet connectivity

System Enablement Packs (SEPs) allow you to add support for hardware released after the current release of the 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 web, 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 and *machine_type* is the machine type of the system for which you want to download SEPs.
4. Copy the files from C:\temp and place them in the updates folder of the Linux Scripting Toolkit directory tree. The default location is `/opt/ibm/sgtk/sgdeploy/updates/uxsp`.

Chapter 4. Getting started

This section describes the use of tasks and workflows in the ServerGuide Scripting Toolkit, Linux Edition to create deployment images.

The Linux Scripting Toolkit creates deployment images based on workflows. A workflow is an aggregation of supported tasks. After you have created a workflow, you create an image based on a boot media profile. The files used by the deployment can be bundled locally on the deployment media or accessed over your network using a repository shared through NFS.

At a high level, the process for using the Toolkit is:

1. Create new tasks, modify existing tasks, or use the tasks provided.
2. Create a workflow.
3. Add your tasks to the workflow.
4. Select a boot media profile to deploy your workflow.
5. Create a deployment image.
6. Boot the target server using the deployment image.

When you boot the target server, the workflow is executed to perform the tasks that you included.

The Linux Scripting Toolkit Console enables you to create and modify tasks, create workflows from your task libraries, and create deployment images from your workflows. The following topics describe how to perform these tasks.

Creating tasks

The Linux Scripting Toolkit provides tasks to perform all of its supported functions. You can use these tasks as they are, or you can create new tasks. This topic describes the process for creating new tasks.

Before you begin

Before creating a new task, you should gather the information required to complete the task. This includes system settings, controller information, and available firmware updates for pre-installation tasks, the location of the operating system files and filename of the answer file for operating system deployment tasks, and the location of the IBM Systems Director Agent for post-installation tasks.

About this task

You create new tasks using an existing task as a template. The Toolkit provides preconfigured tasks for the supported task types. You cannot create tasks of a type not supported by the Toolkit.

Procedure

1. Start the Linux Scripting Toolkit Console: **./opt/ibm/sgtk/sgtklinux.sh**
2. Select the type of task you want to create from the **Tasks** section of the navigation menu.
3. Click **Create**.

4. Enter the name of your new task.
5. Select the template that will be the base of your new task.
6. Click **Create**.

What to do next

The new task is displayed in the task repository. You can now select the task for editing.

Modifying tasks

You can modify tasks that you have created to customize them for your deployment. Follow these steps to modify existing user-created tasks.

About this task

The Linux Scripting Toolkit provides sample tasks for all supported task types. You cannot modify or delete these tasks. To edit a provided task, you must first create a new task using an existing task as a template. Then, you can edit the new task.

Procedure

1. Start the Linux Scripting Toolkit Console: **./opt/ibm/sgtk/sgtklinux.sh**
2. From the navigation menu, select the type of task that you want to edit.
3. Select the task you want to edit.
4. Click **Modify**.
5. Make the necessary changes.
6. Click **Apply**.

Results

The edited task is available from the task library.

Creating operating system repositories

Operating system repositories are used to control what Linux distributions are available for use in deployment workflows. This section describes the fields and controls available for the OS repositories task.

The **OS images** tab allows you to create, modify, or delete operating system repositories. You must create an operating system repository on the **OS images** tab before you can use it in a workflow.

The **OS images** tab displays information about the current repositories in the OS repositories table.

Creating an OS repository

To create an operating system repository, you must acquire the files that will be in the repository and store them in a location that is accessible to the Source Server.

When you have acquired and stored the files, follow these steps to create an OS repository:

1. On the **OS images** tab click **Create** to open the OS repository configuration window.

2. In the OS repository creation settings window, complete the following fields in the **Repository**:

   Name The name by which you will refer to this repository. This is the name that you use to include the repository in a workflow.

   Distribution
   From this dropdown list, select the Linux distribution to be included in this repository. If a distribution is not in the list, it is not supported by the Linux Scripting Toolkit.

   Source
   The source for the distribution files. Select one of the following:

   Optical disks
   Indicates that the files are on a CD or DVD.

   Network
   Indicates the network protocol to use when accessing the distribution. Valid values are nfs, ftp, or http. The default is nfs.

   ISO images
   The path to the distribution files on the source server. After inserting the location of the ISO images, click **List** to show all available ISO images.

When you have saved your selections, the **OS images** tab adds the new repository to the repositories table.

Operating system unattended files

Operating system unattended files allow you to perform unattended installation of supported Linux distributions. The **OS Installation Tasks** tab lets you manage these files.

The **OS Installation Tasks** tab lists the available unattended files for operating system installation. The Linux Scripting Toolkit includes unattended installation files for all of the supported operating systems, shown in Table 1. These files cannot be modified, but you can use them as a template for a new task using the **Create** option to create an unattended installation file for your scenarios. Table 1 lists the unattended installation files supplied with the Linux Scripting Toolkit and the operating systems to which they apply.

The Linux Scripting Toolkit provides answer files for native uEFI mode deployments of SLES 11 x64 and RHEL 6 x64. The unattended file determines whether the installation performed is a native uEFI installation or a legacy installation. The answer files for native uEFI installations are noted in *Unattended installation files supplied with the Linux Scripting Toolkit*. If you want to customize installation files for uEFI installation, you must ensure that the file contains an entry for `/boot/efi`. This entry can be commented out, but must remain visible in the file.

Table 1. Unattended installation files supplied with the Linux Scripting Toolkit

Filename	Operating system
`rhel4.ks`	Red Hat Enterprise Linux 4
`rhel5.ks`	Red Hat Enterprise Linux 5
`rhel5_xen.ks`	Red Hat Enterprise Liunx 5 with Xen

Table 1. Unattended installation files supplied with the Linux Scripting Toolkit (continued)

Filename	Operating system
rhel6.ks	Red Hat Enterprise Linux 6
rhel6_efi.ks	Red Hat Enterprise Linux 6 in native uEFI mode
sles9.xml	SUSE Linux Enterprise Server 9
sles10.xml	SUSE Linux Enterprise Server 10
sles10x64.xml	SUSE Linux Enterprise Server 10 x64
sles10_xen.xml	SUSE Linux Enterprise Server 10 with Xen
sles10x64_xen.xml	SUSE Linux Enterprise Server 10 x64 with Xen
sles11.xml	SUSE Linux Enterprise Server 11
sles11_xen.xml	SUSE Linux Enterprise Server 11 with Xen
sles11x64.xml	SUSE Linux Enterprise Server 11 x64
sles11x64_efi.xml	SUSE Linux Enterprise Server 11 x64 in uEFI mode
sles11x64_xen.xml	SUSE Linux Enterprise Server 11 x64 with Xen
sles11sp2x64_efi.xml	SUSE Linux Enterprise Server 11 SP2 x64 in uEFI mode
esx3.ks	VMware ESX 3.5
esx4.ks	VMware ESX 4

The **OS Installation Tasks** tab supports these actions:

View Opens a window displaying the contents of the selected file. You cannot modify the file from this window.

Create Opens the Create window. This window prompts you for a name for the new file and provides a list of the available files that can be used as templates when creating a new file.

Modify
Allows you to modify the contents of the selected file. Note that this option is not available for the unattended installation files supplied with the Linux Scripting Toolkit. If you want to customize these files, you must create a new file based on the supplied file using the **Create** option.

Delete Prompts for confirmation, then deletes the selected file. Note that this option is not available for the unattended installation files supplied with the Linux Scripting Toolkit. Only user-created tasks can be deleted.

Creating a workflow from tasks

After you have created and modified tasks and created repositories for operating system installation files, combine them into a workflow. Follow these steps to create a workflow.

Before you begin

Before you can create a workflow, the tasks, operating system repositories, and unattended answer files to be included in the workflow must exist. The Linux

Scripting Toolkit includes preconfigured tasks for all supported task types. You can use these to create workflows without having to create your own tasks.

Procedure

1. Start the console by entering the following command: `./opt/ibm/sgtk/sgtklinux.sh`.
2. Select **Workflows** from the main menu.
3. Click **Create** from the Workflows panel to open the **General** section for workflow creation.
4. Enter a name for the workflow you are creating.

   Note: After you have created a workflow, you can use it as the base for creating new workflows by using the **Based on a Template** option.
5. Select the level for **Log verbosity**. The verbosity options are:

   Low Logs basic execution information and provides an overview of the steps being executed.

   Medium
   Adds more detailed execution information and provides a more detailed view of the steps being executed.

   High Adds logging of the commands being executed, their output, and the exit code returned.

   Full Adds some source code trace information.
6. Click **Next** to proceed to the pre-installation section.
7. Select the types of pre-installation tasks to be run as part of this workflow, then select the task for each type from the drop down list, or check the **Skip this step and do not perform any pre-installation tasks** box to skip pre-installation.
8. Click **Next** to proceed to the operating system installation section.
9. Deselect the **Skip this step** checkbox.
10. Select the operating system repository from the list.
11. Select the answer file to use from the **OS unattended file** drop down list.
12. Click **Next** to proceed to the post installation section.
13. Select the post installation tasks to be performed as part of this workflow, or check the **Skip this step and do not perform any post-installation tasks**
14. Click **Next** to review your selections.
15. When your selections are correct, click **Finish** to save the workflow.

Results

The workflow is saved and is available from the workflow list.

What to do next

Now you can use this workflow to create boot media.

Creating bootable media from a workflow

To deploy a workflow to a target server, you must create bootable media. This topic provides the steps to create 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 deployable from a USB key.
   - **ISO** - Creates an ISO image that can be burned to a CD or DVD for deployment.
   - **PXE** - Creates a boot image that can be deployed from a network share.
6. Click **Next** to proceed to the bundling options.
7. Select a bundling option for the files.
   - **Leave files in network share** will create boot media that does not include the deployment files. This option requires network connectivity with the network share on the source server.
   - **Bundle files in the boot media** will add the files to the boot media. No network connectivity is required for this deployment.

   Note: Regardless of the option you select, operating system files are always left on the network share. This means that connectivity with the network share on the source server is always required for operating system deployment.
8. 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
9. Click **Next** to select the machine types for this deployment.
10. 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.
11. When the selections are correct, 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.

Chapter 5. Quick start scenarios

This section describes a set of scenarios that you can use as examples for creating your own workflows. Each scenario builds on the previous one to give examples of pre-installation, operating system installation, and post installation tasks.

This section provides examples of how to create a boot media to perform the following tasks:

- Perform default RAID configuration
- Perform default RAID configuration and install Red Hat Enterprise Linux (RHEL) 5.3.
- Perform default RAID configuration, install Red Hat Enterprise Linux (RHEL) 5.3, and install the IBM Systems Director Agent.

Configuring RAID

This topic describes how to create boot media to perform default RAID configuration on the target server. You can use this process to create boot media to perform any supported pre-installation task.

About this task

This example uses the default RAID configuration task provided by the Linux Scripting Toolkit. You can replace this RAID configuration task with any of the included RAID configuration tasks, or create your own RAID configuration task by creating a RAID configuration file and creating a job to deploy it. For information on creating RAID configuration files, see “PRAID” on page 39.

Procedure

1. Create a workflow using the default RAID configuration task:
   a. From the main menu, select **Workflows**.
   b. From the Workflows menu, select **Create**.
   c. In the **What's the name of the new workflow?** field, enter *default_raid_configuration*.
   d. In the pre-installation section of the workflow, select the **RAID** checkbox, and select the **Default** task from the drop down list.
   e. Click **Next** to proceed through the wizard.
   f. Select the **Skip this step...** checkboxes for the **OS install** and **Post-install** sections.
   g. Review your selections and click **Finish** when you have completed the wizard.

   The workflow is created and available in the **Workflows** list.
2. Create a Bootable Media Profile to deploy the new workflow:
   a. From the main menu, select **Bootable Media Profiles**.
   b. From the Boot Media Creation menu, select **Create**.
   c. In the **What's the name of the new Boot Media Profile?** field, enter *usb_local_default_raid_configuration*.
   d. From the **Boot method** drop-down list, select **USB**.

e. Click **Next**.

f. From the **Source medias** menu, select **Bundle files in the boot media** and click **Next**. This selection places all of the files necessary for this deployment on the boot media.

Note: Operating system files are not bundled on the bootable media, regardless of this setting.

g. From the **Target system IP settings** menu, select **Configure network using a DHCP server** and click **Next**.

h. Click **Next** to continue through the **Select the machine models** panel. This panel is used for firmware update tasks and operating system installations.

i. Review your selections and, when they are correct, click **Create Boot Media**.

j. When prompted, enter the path to the USB key you want to use to hold your deployment image.

The boot media is created and ready for deployment.

Note: When the boot media is a USB key that has not previously been formatted by the Linux Scripting Toolkit, the Toolkit formats the key and adds the necessary files. All other information on the key will be lost.

3. To complete the deployment, start the target system using the boot media.

Adding an operating system installation

This example builds on the example of creating a local USB deployment of the default RAID configuration by adding installation of Red Hat Enterprise Linux 5.3 x64 to your deployment, and performing the deployment over the network.

Before you begin

This task requires you to have created an operating system repository and unattended answer file for Red Hat Enterprise Linux 5.3 x64, as described previously. This example uses the name *rhel_53_x64* and the default answer file for Red Hat Enterprise Linux 5.3 x64 provided by the Linux Scripting Toolkit.

About this task

Because operating system installation files are not bundled on the boot media, this example takes advantage of the need for connectivity with the network share to place the PXE boot image on the network share as well. This process allows you to boot multiple servers from the same network share, making it easier to perform deployments for geographically distributed systems.

Procedure

1. Create a workflow using the default RAID configuration task and an operating system repository and answer file for Red Hat Enterprise Linux 5.3 x64:
   a. From the main menu, select **Workflows**.
   b. From the Workflows menu, select **Create**.
   c. In the **What's the name of the new workflow?** field, enter *default_raid_rhel5*.
   d. In the pre-installation section of the workflow, select the **RAID** checkbox, and select the **Default** task from the drop down list.
   e. Click **Next** to proceed to the **OS install** section.

f. Select the operating system repository for Red Hat Enterprise Linux 5.3 x64 from the **Operating System repositories** drop down list.
g. Select *rhel5* from the **OS unattended files** drop down list and click **Next**
h. Select the **Skip this step...** checkbox for the **Post-install** section.
i. Review your selections and click **Finish** when you have completed the wizard.

The workflow is created and available in the **Workflows** list.

2. Create a Bootable Media Profile to deploy the new workflow:
   a. From the main menu, select **Bootable Media Profile**.
   b. From the Bootable Media Profile menu, select **Create**.
   c. In the **What's the name of the new Boot Media Profile?** enter *network_default_raid_rhel53x64*.
   d. From the **Boot method** drop-down menu, select **PXE**.
   e. Click **Next**.
   f. From the **Source medias** menu, select **Leave files in network share** and click **Next**. This option places the PXE boot image on the network share used by the source server.
   g. From the **Target system IP settings** menu, select **Configure network using a DHCP server** and click **Next**.
   h. From the **Select the machine models** panel, select the system models for deployment and select the check box to check the updates repository for the UXSPs for the selected servers.
   i. Click **Next** to check the repository for the necessary UXSPs. If any are missing, acquire them.
   j. Review your selections and, when they are correct, click **Create Boot Media**.
   k. When prompted, enter the path to place the generated files to be used to boot from PXE.

   The boot media is created and ready for deployment.
3. To complete the deployment, start the target system using the boot media.

Results

When the target system boots from the media:

1. The RAID configuration runs:
   - If RAID is not already configured, a new RAID array is created and the system is rebooted. When the reboot occurs, ensure that the system returns to the boot media, either by specifying it in the boot order or using F12 to set it in the boot menu.
   - If RAID is already configured, the RAID configuration task is skipped.
2. The operating system installation task runs.

Adding installation of the IBM Systems Director Agent

This example builds on the example of creating a network deployment of default RAID configuration and Red Hat Enterprise Linux 5.3 x64 by adding installation of the IBM Systems Director Agent to your network deployment.

Before you begin

This task requires you to have created an operating system repository and unattended answer file for Red Hat Enterprise Linux 5.3 x64, as described previously. This example uses the default answer file for Red Hat Enterprise Linux 5.3 x64 provided by the Linux Scripting Toolkit. This example also requires you to have the IBM Systems Director Agent files available to the source server. You can download the IBM Systems Director from http://www.ibm.com/systems/management/director/downloads/.

Note: This download requires registration with IBM.com.

About this task

Because operating system installation files are not bundled on the boot media, this example takes advantage of the need for connectivity with the network share to place the PXE boot image on the network share as well. This allows you to boot multiple servers from the same network share, making it easier to perform deployments for geographically distributed systems.

This example builds on the example of creating a network bundled deployment media for default RAID configuration and installation of Red Hat Enterprise Linux 5.3 x64, adding post installation of the IBM Systems Director Agent. The deployment media generated from this example is used to start the Linux Toolkit processes and the files used for configuration and deployments are located on the network.

Procedure

1. Create a workflow using the default RAID configuration task, an operating system repository and answer file for Red Hat Enterprise Linux 5.3 x64, and a post-install task to install the IBM Systems Director Agent:
   a. From the main menu, select **Workflows**.
   b. From the Workflows menu, select **Create**.
   c. In the **What's the name of the new workflow?** field, enter *default_raid_rhel5*.
   d. In the pre-installation section of the workflow, select the **RAID** check box, and select the **Default** task from the drop-down list.
   e. Click **Next** to proceed to the **OS install** section.
   f. From the **Operating System repositories** drop-down list, select the operating system repository for Red Hat Enterprise Linux 5.3 x64.
   g. From the **OS unattended files** drop-down list, select *rhel5* and click **Next**.
   h. Select the checkbox for **Install IBM Director** and select the correct task from the drop down list.
   i. Review your selections and, when you have completed the wizard, click **Finish**.

   The workflow is created and available in the **Workflows** list.
2. Create a Bootable Media Profile to deploy the new workflow:
   a. From the main menu, select **Bootable Media Profile** from the main menu.
   b. From the Bootable Media Profile menu, select **Create**.
   c. In the **What's the name of the new Boot Media Profile?** enter *network_default_raid_rhel53x64_diragent*.
   d. From the **Boot method** drop-down list, select **ISO**.

e. Click **Next**.

f. From the **Source medias** menu, select **Leave files in network share** and click **Next**. This will place the PXE boot image on the network share used by the source server.

g. From the **Target system IP settings** menu, select **Configure network using a DHCP server** and click **Next**.

h. From the **Select the machine models** panel, select the system models for deployment and select the check box to check the updates repository for the UXSPs for the selected servers.

i. Click **Next** to check the repository for the necessary UXSPs. If any are missing, acquire them.

j. Review your selections and, when they are correct, click **Create Boot Media**.

k. When prompted, enter the path to the directory where the CD/DVD ISO image will be created.

The boot media is created and ready for deployment.

3. To complete the deployment, start the target system using the boot media.

Chapter 6. Customizing deployments

This section gives you information on how to customize the tools used by the Linux Scripting Toolkit to perform the tasks included in a deployment.

The Linux Scripting Toolkit allows you to customize these types of jobs:

- RAID
- Fibre Channel
- Advanced Settings Utility
- Firmware update

Customizing RAID configuration

Before installing the operating system, you must configure RAID for the target system. This pre-installation task performs any default RAID or custom raid that you want. Use this task to view, create, delete, and modify raid policy files.

The Linux Scripting Toolkit provides sample RAID policy files to perform common RAID configurations. You can view these policy files on the RAID tab of the Pre-installation tasks. Use these samples as a base to create new policy files if they are required. For more information on creating policy files, see “PRAID” on page 39.

Customizing Fibre Channel configuration

Before you install the operating system, you can configure a Fibre Host Bus Adapter to boot from Storage Area Network (SAN). After configuration, it is possible to install an operating system to the SAN. The Toolkit provides a sample fibre policy file which can be used for fibre configuration deployment.

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

By default, the Linux Scripting Toolkit will configure the first HBA on the system to boot from the first available SAN target (for QLogic Fibre HBAs only. See “Known problems and limitations” on page 76 for limitations concerning Emulex Fibre HBAs). The BIOS configures the first disk drive that it finds that is also a LUN 0 as a boot device. The Linux Scripting Toolkit uses the following variables to configure Fibre HBAs.

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

Table 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 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 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. 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 2. Fibre HBA boot configuration variables (continued)

Variable	Description
TK_FIBRE_*N*_BOOT_DISABLE	Disables 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`

For more configuration options, please refer to “SCLI” on page 65.

Customizing the Advanced Settings Utility

Before installing the operating system, you can configure system settings for the target system using the Advanced Settings Utility (ASU).

The Linux Scripting Toolkit provides a sample ASU settings file that can be used to deploy system settings. The sample ASU file can be used to load the default settings on the target system. Toolkit uses the ASU **batch** command to configure the system settings on the target system. Please refer to “Advanced Settings Utility” on page 65 for more information on the settings and configuration file.

Customizing firmware updates

The Update firmware task is used to update the firmware on the target system with UpdateXpress System Packs (UXSPs). The Linux Scripting Toolkit provides a sample configuration file for the UpdateXpress System Pack Installer (uxspi*xxx*, where *xxx* is the version of the installer).

The following table describes the settings available in the sample configuration file.

Setting	Description
TK_UXSP_UpdateXpressSystemPacks	Specifies the location where the UXSPs are copied. Value: `/sgdeploy/updates/uxsp`
TK_UXSP_ApplyLatest	Specifies whether the UXSPi should apply the latest updates to the target system if no UXSPs are found for that system. Setting this variable to *yes* will force the installer to apply the latest updates for the system if no UXSPs are found for it. Valid values: Yes, No Default: No
TK_UXSP_UXSPIUpdateFlags	Specifies user provided command line arguments for processing by the UpdateXpress System Pack Installer in Update mode. To provide command line arguments to be processed by UXSPI, set this variable to the command line arguments. See “UpdateXpress System Pack Installer” on page 68 for a list of command line arguments to use with UXSPI in Update mode. Default: update –unattended –firmware

For more information, please see “UpdateXpress System Pack Installer” on page 68.

Chapter 7. Supported hardware and software

This section lists the operating systems, adapters, and RAID controllers supported by the Linux Scripting Toolkit, as well as systems that support BIOS and firmware updates using the ASU.

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

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

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

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

The most up-to-date support information is contained in the `readme.htm` file. You can download the latest version of `readme.htm` file from the ServerGuide Scripting Toolkit Web page. See IBM deployment resources on the World Wide Web for information.

Operating system support

This section lists operating system deployment/server combinations supported by the Linux Scripting Toolkit.

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/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
- SUSE Linux Enterprise Server 11 x64 Base/SP1/SP2
- 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
- Red Hat Enterprise Linux 5 x64 U1/U2/U3/U4/U5/U6/U7/U8
- Red Hat Enterprise Linux 6 32 bit U1/U2
- Red Hat Enterprise Linux 6 x64 U1/U2
- VMware ESX Server 3.5 U4/U5
- VMware ESX Server 4.0/4.0u1/4.0u2/4.1/4.1u1/4.1u2

RAID controller support

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

Fibre Channel HBA support

This section lists the Fibre Channel adapters supported by the Linux Scripting Toolkit.

The Linux Scripting Toolkit provides full support for the following adapters. For information on supported server/adapter combinations, please see IBM ServerProven.

IBM HBAs:

- 2 GB IBM SMB 2–Gbps Fibre Channel HBA (13N1873)
- 2 GB IBM HS20 Fibre Channel Expansion Card (13N2203)
- 2 GB DS4000® HBA (24P0960)
- 2 GB SFF Fibre Channel Expansion card (26K4841)
- 4 GB DS4000 PCI-X Single Port HBA (39M5894)
- 4 GB DS4000 FC PCI-X Dual Port HBA (39M5895
- IBM HBA SAS controller (44W1853)
- 2 GB IBM HS20 Fibre Channel Expansion Card (48P7061)

Qlogic HBAs:

- 4 GB Standard Fibre Channel Expansion Card for IBM eServer™ BladeCenter (26R0884)
- 4 GB SFF Fibre Channel Expansion Card for IBM eServer BladeCenter (26R0890)
- 4 GB Single-Port PCIe HBA for IBM System x (39R6525)
- 4 GB Dual-Port PCIe HBA for IBM System x (39R6527)
- Ethernet and 4 GB Fibre Channel Expansion Card for IBM BladeCenter (39Y9306)
- iSCSI Single-port PCIe HBA for IBM System x (39Y6146)
- iSCSI Dual-port PCIe HBA for IBM System x (42C1770)
- 4 GB Fibre Channel Expansion Card for IBM BladeCenter (41Y8527)
- 8 GB Fibre Channel Singleport HBA (42D0501)
- 8 GB Fibre Channel Dualport HBA (42D0510)
- 8 GB Ethernet and Fibre Channel Expansion Card CFFh (44X1940)
- 8 GB Fibre Channel Expansion Card CIOv (44X1945)
- 8 GB Fibre Channel Dualport HBA (69Y1938)

Emulex HBAs:

- 4 GB Fibre Channel Single-port PCIe HBA for IBM System x (42C2069)
- 4 GB Fibre Channel Single-port PCIe HBA for IBM System x (42C2071)
- 8 GB Fibre Channel Single- Port Expansion Card (42D0485)
- 8 GB Fibre Channel Single/ Dual- Port HBA (42D0494)
- 8GB Fibre Channel Dual- Port Expansion Card (42D0494)
- 8 GB Fibre Channel Dual-port Expansion Card (46M6140)
- 8Gb Fibre Channel Mezz card (95Y2375)
- 16Gb Fibre Channel Single-port for IBM System x (81Y1655)
- 16Gb Fibre Channel Dual-port for IBM System x (81Y1662)

Brocade adapters:

- 10 GB 2-port Converged Network Adapter for IBM BladeCenter (81Y1650)
- 8 GB Single Port Fibre Channel Adapter (46M6049)
- 8 GB Dual Port Fibre Channel Adapter (46M6050)
- 4 GB Fibre Channel Single-port HBA for IBM System x (59Y1987)
- 4 GB Fibre Channel Dual-port HBA for IBM System x (59Y1993)
- 16Gb Fibre Channel Single-port for IBM System x (81Y1668)
- 16Gb Fibre Channel Dual-port for IBM System x (81Y1675)
- 16Gb Fibre Channel Dual-port Mezz card (88Y6370)

Note: Support of some Fibre Channel HBA cards, particularly the QLogic 39Y9306, by some models of the BladeCenter chassis are subject to limitations and can require the installation of additional supporting equipment. Refer to the IBM Red Paper *Implementing the QLogic Intelligent Pass-thru module for IBM BladeCenter* at http://www.ibm.com/redbooks for additional compatibility information.

Chapter 8. Linux Scripting Toolkit 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.

For each included tool there is a brief description of the tool and instructions on using it with the Linux Scripting Toolkit, as well as pointers on where to get more information on the tool and its use.

Linux Scripting Toolkit utilities

This section contains information about the utilities that are included in the Linux Scripting Toolkit. For each utility there is a description of the parameters with examples. These utilities can be found in `/opt/ibm/sgtk/wui/.data/sgdeploy/sgtklinux/tk/bin` on the source server.

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

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

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

HWDETECT

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

HWDETECT has basic hardware scan functions and more complex PCI device detection options. The basic hardware scan functions can only be used singularly. The PCI device detection functions can be used with each other to produce a query based on multiple restrictions. You can only use the hwdetect utility basic hardware scan functions one at a time. The PCI-device detection functions can be combined or used more than once on the same command line.

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`

`-i`	Dumps all available information about the system hardware to the screen in an ini file format. You can use the **-f** parameter to output 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..."`

Below is an example of the 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 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 allows you to store and retrieve up to twenty values to persistent storage. The utility is designed to help you remember 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.

Savestat uses the persistent storage capability of **ASU**. 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.

Parameter	Description	Usage
-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
-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 savestat utility usage.

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 Toolkit.

PRAID has three modes of operation:

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

Deploy mode

Used in Deploy mode, PRAID offers the following features:

- Configures all RAID controllers in a server with a single call to the program.
- Automatically resets all RAID controllers to factory-default settings before configuring.
- Uses customizable logic to decide which configuration (policy) is applied to a server based on system hardware. The logic can involve:
  - Machine type of the server
  - Serial number of the server
  - Number of drives connected to the RAID controller
  - RAID controller type
  - Controller number (order) of the RAID controller
- Can be highly customized for specific RAID configurations or generalized to handle many different RAID configurations.
- Provides a default or AUTO mode for automatically creating arrays and logical drives 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`.

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.

Restore-defaults mode

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

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

Environment requirements

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

Table 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	megacli
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 H1110/H1135	SAS2004	sas2ircu

Usage

Each of the modes supported by PRAID requires a specific syntax, but they all share some common parameters, described in Table 4 on page 43.

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 capture mode and for deploy mode 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	Suppresses the confirmation prompt. This parameter is optional. 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 myfile.ini, 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.EXE -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	Configure all controllers in the system 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 58 for the default values that will be 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.EXE -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` that is created when capturing the parameters to a policy. See "AppliesTo.*n*" on page 50. :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 using the -f parameter when using the -c parameter. The policy or policies created are appended to the end of the file if the file exists. 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 myfile.ini 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.EXE -r:n -e2 -v:n -y
```

Usage examples

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 before setting controllers to factory-default settings.
- Performs drive synchronization without prompting, when required.

```
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.

- Returns an error code if there were 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 file: `policies.ini`.

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

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

Restore-defaults mode examples

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

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

Return codes

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

Policies file

When used in configure mode, the policies file directs how PRAID configures the RAID controllers in a system using keywords and values that 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.

A policies file can be created using any of the following methods:

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

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

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

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

How PRAID selects a policy: Each section in the policies file represents a single policy for configuring the RAID controllers. In configure mode, each RAID controller is configured using a single policy, but a single policy can be used to configure multiple controllers. Each policy in a policies file contains one or more *AppliesTo.n* entries, where *n* is the number of the AppliesTo parameter within the policy. This entry is required in each section, so every section must contain at least an AppliesTo.1 entry. See "Policies file parameters" for a full description of the AppliesTo.n entry.

These entries are followed by a list of hardware parameters including machine type, number of drives connected to the RAID controller, and scan order, that are evaluated against the current system hardware. If all of the hardware parameters of an AppliesTo.n entry match the hardware being evaluated, this policy is used 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 their 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 two example policies files that you can modify for your own use.

- RAID1-5.ini creates a RAID-1 array using the first two drives, and a RAID-5 array using the remaining drives. Valid for ServeRAID-6M and 8i.
- RAID5HSP.ini creates a single RAID-5 array with a single hot-spare drive using all available drives. Valid for ServeRAID-6M and 8i.
- template.ini provides a policies file template containing all parameters with details about each parameter.
- syntax.txt provides a syntax specification for the polices file.

Table 7. 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 50 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 50 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 51 for additional information.
`RebuildRate`	No	HIGH	Specifies the rebuild rate that should be applied to the RAID controller. See "RebuildRate" on page 51 for additional information.
`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 51 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 51 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 52 for additional information.
`Array.letter`	No	None	Lets you specify exactly how many arrays are created and the exact physical drives that you would like in each array. See "Array.*letter*" on page 52 for additional information.
`Hotspares`	No	None	Defines a list of specific physical drives to designate as hot-spare drives. See "Hotspares" on page 53 for additional information.
`Logical_Mode`	No	AUTO	Defines the logical-drive creation policy to use when creating logical drives. See "Logical_Mode" on page 53 for additional information.

Table 7. Policy file parameters (continued)

Keyword	Required?	Default	Description
`Logical_Defaults`	No	FILL:AUTO:AUTO	Defines the default logical drive settings that should be used when creating logical drives. See "Logical_Defaults" on page 54 for additional information.
`Logical.num`	No	None	Lets you specify how many logical drives are created and the specific parameters for each logical drive. See "Logical.*num*" on page 54 for additional information.

Policy.name:
Description

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

Examples

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

AppliesTo.n:
Description

Use this parameter to describe when the current policy 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, and AppliesTo.n is the only required parameter of a policy.

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

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

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

Examples

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

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

Example using the ALL parameter:

```
AppliesTo.1 = ALL
```

ReadAhead:
Description

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 55 for the list of ReadAhead settings supported by PRAID for each RAID controller. Possible settings are:

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

Examples

```
ReadAhead = On
```

RebuildRate:
Description

Specifies the rebuild rate that 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 55 for the list of RebuildRate settings supported by PRAID for each RAID controller.

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

Examples

```
RebuildRate = High
```

StripeSize:
Description

Specifies the stripe-unit size in KB that the controller 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 55 for the list of StripeSize settings supported by PRAID for each RAID controller. Possible values are any stripe size supported by the controller.

Examples

```
StripeSize = 32
```

Array_Mode:
Description

Defines the array-creation policy to use when selecting physical disk drives to include in an array. Possible values are:

Auto Creates arrays using drives that have the same size in MB. This is the default. Each set of drives 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 exact 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.

Examples

```
Array_mode = CUSTOM
```

Array_Defaults:
Description

Defines the default values to use for the variance and number of hot-spare drives when `AUTO` is specified for `Array_Mode`. This parameter is not valid if `Array_Mode` is set to `CUSTOM`.

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

variance specifies the percentage variance to use when selecting drives to add to the array. This parameter is useful when you are using drives that 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.

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

Examples

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

Array.letter:
Description

Allows you to specify exactly how many arrays are created and the exact physical drives that you would like in each array. You can specify the physical drives using any of the following methods:

- The channel number and SCSI ID (for SCSI) or bus number and target ID (for SATA/SAS) of each drive. The channel number or bus number is always 1-based. The SCSI ID or target ID is always 0-based.
- A list of integer values indicating that the *n*th drive should be included in the array.

- The keyword ALL to indicate that all remaining drives attached to the controller that are not specified in previous arrays 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

Defines a list of specific physical drives to designate as hot-spare drives. You can specify the physical drives using any one of these methods:

- The channel number and SCSI ID (for SCSI) or bus number and target ID (for SATA/SAS) of each drive. The channel number or bus number is always 1-based. The SCSI ID or target ID is always 0-based.
- A list of integer values indicating that the *n*th drive 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

Defines the logical-drive creation 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 using the Logical_Defaults parameter.

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

Examples

```
Logical_Mode = CUSTOM
```

Logical_Defaults:
Description

Defines the default logical drive settings that must 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 be in any of the following formats:

- A positive integer – specifies the size in MB.
- A percentage – specifies that a percentage of the total space 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" on page 55 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" on page 55 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.

Examples

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

Logical.num:
Description

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

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

Examples

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

Supported settings for RAID controllers: Table 8 lists the supported settings for each RAID controller when using PRAID.

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

Table 8. Supported settings for each RAID controller when using PRAID. Bold settings are defaults.

Controller	Rebuild Rate	Read Ahead	Stripe Size	RAID Levels[1]	Write-cache Mode
ServeRAID-6m	• HIGH • MEDIUM • LOW	• **ADAPTIVE** • ON • OFF	• **8** • 16 • 32 • 64	• 0 • 1 • 1E • 10 • 5 • 50 • 5EE • AUTO	• ON • OFF • AUTO
ServeRAID-7t	[n/a]	• **ON** • OFF	• 16 • 32 • **64**	• 0 • 1 • 5 • 10 • VOLUME • AUTO	• ON • OFF • AUTO
ServeRAID-8i	[n/a]	• **ON** • OFF	• 16 • 32 • **64** • 128 • 256 • 512	• 0 • 1 • 1E • 10 • 5 • 50 • 5EE • VOLUME	• ON • OFF • AUTO
ServeRAID-8k	[n/a]	• **ON** • OFF	• 16 • 32 • **64** • 128 • 256 • 512 • 1024	• 0 • 1 • 1E • 10 • 5 • 6 • VOLUME	• ON • OFF • AUTO

Table 8. Supported settings for each RAID controller when using PRAID (continued). Bold settings are defaults.

Controller	Rebuild Rate	Read Ahead	Stripe Size	RAID Levels[1]	Write-cache Mode
ServeRAID-8k-l	[n/a]	• **ON** • OFF	• 16 • 32 • **64** • 128 • 256 • 512 • 1024	• 0 • 1 • 10 • VOLUME • AUTO	[n/a]
ServeRAID-8s	[n/a]	[n/a]	• 16 • 32 • **64** • 128 • 256 • 512 • 1024	• 0 • 1 • 1E • 10 • 5 • 50 • 6 • 60 • VOLUME	[n/a]
LSI-SCSI-RAID (1020/1030)	[n/a]	[n/a]	• **8** • 16 • 32 • 64	• 0 • 1 • 1E[2]	[n/a]
• LSI-SAS-RAID (1064/1064E/ 1068) • IBM SAS HBA • IBM 3 Gb SAS HBA v2 • ServeRAID BR10i • ServeRAID BR10il	[n/a]	[n/a]		• 0 • 1 • 1E	[n/a]

Table 8. Supported settings for each RAID controller when using PRAID (continued). Bold settings are defaults.

Controller	Rebuild Rate	Read Ahead	Stripe Size	RAID Levels[1]	Write-cache Mode
• ServeRAID-MR10i • ServeRAID-MR10ie • ServeRAID-MR10il • ServeRAID-MR10is • ServeRAID-MR10k • ServeRAID-MR10M • ServeRAID-MR1015 • ServeRAID-MR1015–R5	• HIGH • MEDIUM • **LOW**	• **ADAPTIVE** • ON • OFF	• 8 • 16 • 32 • **64** • 128	• 0 • 1 • 10 • 5 • 50 • 6 • 60	• ON • OFF • AUTO
LSI SAS 1078	[n/a]	[n/a]	[n/a]	• 0 • 1	[n/a]
LSI MegaRAID 8480	[n/a]	[n/a]	• 8 • 16 • 32 • 64 • 128	• 0 • 1 • 5 • 10 • AUTO	[n/a]
ServeRAID-M5014	• HIGH • MEDIUM • LOW	• **ADAPTIVE** • ON • OFF	• 8 • 16 • 32 • 64 • 128	• 0 • 1 • 5 • 10 • 50 • AUTO	• ON • OFF • AUTO
ServeRAID-M5015	• HIGH • MEDIUM • LOW	• ON • OFF	• 8 • 16 • 32 • 64 • 128	• 0 • 1 • 5 • 10 • 50 • AUTO	• ON • OFF • AUTO

Table 8. Supported settings for each RAID controller when using PRAID (continued). Bold settings are defaults.

Controller	Rebuild Rate	Read Ahead	Stripe Size	RAID Levels[1]	Write-cache Mode
ServeRAID-M5025	• HIGH • MEDIUM • LOW	• **ADAPTIVE** • ON • OFF	• 8 • 16 • 32 • 64 • 128	• 0 • 1 • 5 • 6 • 10 • 50 • 60 • AUTO	• ON
ServeRAID-B5015	[n/a]	[n/a]	• 4 • 8 • 16 • 32 • 64 • 128 • 256 • 512 • 1024	• 1 • 5 • AUTO	• ON

1. RAID levels 5E and 5EE support only one logical drive per array.
2. RAID level 1E is supported for the LSI 1030 only on the xSeries model 336.

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 9. Default RAID levels

	Drives in Array				
Controller	**1**	**2**	**3**	**4**	**5 or more**
ServeRAID-6M	RAID 0	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+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 9. Default RAID levels (continued)

	Drives in Array				
Controller	**1**	**2**	**3**	**4**	**5 or more**
LSI-SCSI-RAID (1020/1030)	[n/a]	RAID 1	RAID 5	RAID 1+Hotspare	RAID 1+Hotspare
• LSI-SAS-RAID (1064/1064E/1068) • IBM SAS HBA • IBM 3Gb SAS HBA v2 • ServeRAID BR10i • ServeRAID BR10il	[n/a]	RAID 1	RAID 1+Hotspare	RAID IME+Hotspare	RAID IME+Hotspare
• ServeRAID-MR10i • ServeRAID-MR10ie • ServeRAID-MR10il • ServeRAID-MR10is • ServeRAID-MR10k • ServeRAID-MR10M • ServeRAID-MR1015	RAID 0	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare
LSI-SAS-1078	[n/a]	RAID 1	2RAID 1+1Hotspare	2RAID 1+1Hotspare	2RAID 1+1Hotspare
LSI MegaRAID 8480	RAID 0	RAID 1	RAID 5	RAID 5+Hotspare	RAID 5+Hotspare

INVRAID

Use this program to dump all of the RAID controller configuration information to an output file. Supported RAID controllers by INVRAID are listed in the table, Table 3 on page 41.

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 10. 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 11 lists the values returned by INVRAID.

Table 11. 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 a file in INI file format with the name `myraid.ini`, use the -I parameter as shown here:

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

Returns:

```
[System]
Machine_Type = 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.

VALRAID 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 using the PRAID uitlity. 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. This process lets you avoid restarting the system before installing the operating system. VALRAID sets the return code to 20 to indicate that the policy file does not match the configuration represented by the inventory file.

Usage

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

The simulation mode syntax is:

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

The check mode syntax is:

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

Table 12. 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
-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 filename for the output inventory file. This is an inventory file representing 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 filename for the output policy file. This file can be applied to a target system 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

Table 12. VALRAID parameters (continued)

Parameter	Description	Example
`-raid:inifiles`	Specifies the directory that contains the RAID ini files. The default is `/opt/ibm/sgtk/sgdeploy/sgtklinux/.data/valraid`	`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 – Success
- 1 – Error parsing input policy file
- 2 – Error parsing input inventory file
- 3 – Controller is not supported
- 4 – Raid level is not supported
- 5 – Stripesize is not supported
- 6 – Number of arrays not supported
- 7 – Number of drives in array not supported
- 8 – Number of logical volumes in array is not supported
- 9 – Not enough drives to create hotspare
- 10 – Not enough drives of the same size
- 11 – Error opening input policy file
- 12 – Error opening input inventory file
- 13 – Error opening output inventory file
- 14 – Error writing to output inventory file
- 15 – Error opening output policy file
- 16 – Error writing output policy file
- 17 – Partial drive sizing 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 Toolkit

The Linux Scripting Toolkit includes several additional tools to make the Toolkit more efficient. This section describes the additional tools provided by this release of the Linux Scripting Toolkit:

- Advanced Settings Utility
- SCLI

- UpdateXpress System Pack Installer

Advanced Settings Utility

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

The Linux Scripting Toolkit uses a subset of the ASU function 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 13. ASU functions in Linux Scripting Toolkit

Command	Description
asu show all	Is used to display and capture BIOS settings. You can use redirection to store this output in a file as shown here: `asu.exe show bios > bios_settings.ini`
asu save *filename*	Is used to apply CMOS settings from a file. ASU looks for the filename 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. This 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.

SCLI

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

Usage

Table 14. SCLI usage

Command	Description
scli –e (view | ?)	Shows the current boot device information on all HBAs
scli -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.
scli -e (*hba_instance* | *hba_wwpn*) (view | ?)	Shows the current boot device information for the specified HBA port.
scli -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.
scli -e (*hba_instance* | *hba_wwpn*) disable [prim | alt1 | alt2 | alt3]	Clears the selected boot device settings on the indicated HBA port.
scli -l (*hba_instance* | *hba_wwpn*)	Displays information about the LUNs attached to the specified HBA port.

Examples

The following examples illustrate scli utility usage.

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

Example	Description
scli -e view	Displays the current boot device information on all HBAs.

Example	Description

`scli –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.

`scli –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.
`scli –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.
`scli –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.

LINLPCFG

Use the LINLPCFG utility that comes with Linux Scripting Toolkit to configure Fibre Host Bus Adapters (HBAs). You can download this utility from Emulex at http://www.emulex.com.

Usage

Table 15. LINLPCFG usage

Command	Description

`linlpcfg help`
`linlpcfg ?`
`linlpcfg help` *`command`*
`linlpcfg ?` *`command`*
	Displays the syntax for LINLPCFG commands.

`linlpcfg listwwn`	Lists all adapters installed in the system with the following information: • WWN • WWPN • WWNN
`linlpcfg listwwn`	Lists all adapters installed in the system with the following information: • adapter number • IEEE address (from the manufacturer) • functional firmware level • adapter type • any possible mailbox errors
`linlpcfg readbootdevice n=`*`adapter_number`*	Displays the following information about the currently selected boot device: • WWN • LUN • topology in use
`linlpcfg enableboot n=`*`adapter_number`* `i=`*`index`*	Enables the BootBIOS for the specified adapter number. Index (*i*) is the index number given by the `listboot` command.
`linlpcfg disableboot n=`*`adapter_number`* `i=`*`index`*	Disables the BootBIOS for the specified adapter number. Index (*i*) is the index number given by the `listboot` command.

Table 15. LINLPCFG usage (continued)

Command	Description
`linlpcfg setbootdevice n=adapter_number w0=wwpn_word_0 w1=wwpn_word_1 l=decimal_id_of_lun t=topology`	Sets the boot device to the device specified by the adapter number, WWPN words, LUN ID, and topology. Enter this command on a single line.
`linlpcfg readaltboot n=adapter_number`	Displays the WWPN and LUN numbers of all possible alternate boot devices. You can have up to seven alternate boot devices.
`linlpcfg setaltboot n=adapter_number i=index w0=wwpn_word_0 w1=wwpn_word_1 l=decimal_id_of_lun`	Sets an alternate boot device. You can have up to seven alternate boot devices, specified by indices from 1 to 7.

UpdateXpress System Pack Installer

For convenience, the Linux Scripting Toolkit includes the UXSPi to help you acquire updates for inclusion 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/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 on running the UpdateXpress System Pack Installer, change directory to the UXSPi directory and run the UXSPi executable 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 16. 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.

Table 16. UXSPi update mode options (continued)

Option	Description
-l *update_xml_path*, –local=*update_xml_path*	Specifies the filename 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 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	Do not deploy UXSPs.
-r, –report	Displays a summary report of updates used in the compare step.

Examples

The following example can be used to specify an UpdateXpress System PackXML 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 9. Hints and tips

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

Performing PXE deployments using the Linux Scripting Toolkit

To perform a PXE deployment 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. Select **Create Boot Media** in the Linux Scripting Toolkit. 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. Check the contents of the configuration file. As a rule, no changes should be required. In this example the configuration file is `/tftpboot/lnxtoolkit/PXE_test`. The contents will be similar to this example:

   ```
   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. Select **Create Boot Media** in the Linux Scripting Toolkit. 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. Check the contents of the configuration file. As a rule, no changes should be required. In this example the configuration file is `/tftpboot/lnxtoolkit/PXE_test`. The contents will be similar to this example:

   ```
   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 that points 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
   }
   ```

Disabling uEFI PXE to decrease network boot time

To improve the time it takes to start the network for uEFI-based systems, complete the following 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**.

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 Linux 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. 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 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 USB mouse support from the operating system. If you install a supported Linux operating system on your server using a USB mouse, the installation process automatically establishes USB mouse support; no further action is required. If you install a supported Linux operating system using a PS/2 mouse, you must modify the Linux files after installation to add USB mouse support. Follow the instructions in this document to add USB mouse support

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 the *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 ServerGuide Scripting Toolkit, Linux Edition deployment. For more information on configuring a USB key for deployments, see “Creating bootable media from a workflow” on page 17.

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 IPv6 networks. IPv6 implementation includes:

- Support for IPv6 stateless and stateful address configuration in the pre-installation environment.
- Support for network-based installation of operating systems using FTP and HTTP servers.
- Support for creating a remote operating system repository using an IPv6 address as an **OS images** task.
- Support for IPv6 networks in the Create Boot Media Profile wizard.

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
```

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, complete the following 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 using 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` 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
```

IBM ServerProven compatibility

The IBM ServerProven website provides valuable information on selected products for compatibility with IBM System x, BladeCenter, and xSeries servers. Check the following link for up-to-date compatibility with operating systems, configuration, and hardware options http://www.ibm.com/servers/eserver/serverproven/compat/us/.

Known problems and limitations

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

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 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 not configured as a hot spare. The problem is caused by the ordering of Linux mptsas devices.

For example, consider a system that 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 hardware supported by the UXSPs is not present in the target

system. These errors can be safely ignored.

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

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

Missing files in USB key network deployment

You might receive errors due to missing files when using a USB key as a boot method for network Linux Scripting Toolkit deployments when the key was used previously for local deployments.

To perform network installations with a key that has been used for local installations, manually remove the `sgdeploy` directory from the key before creating the boot media using the Linux Scripting Toolkit.

Unattended Linux installation requests network device

When performing unattended Linux operating system installs, 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.

RAID configuration fails for ServeRAID-8E SATA

RAID configuration is not supported for the ServeRAID-8E SATA disk controller. Before you can install an operating system, you must create one or more simple volumes using the BIOS-based utility.

This limitation affects the following systems:

- System x3400, types 7973 and 7974
- System x3550, types 1913 and 7978

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

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 250GB.

To avoid this problem, remove any logical volumes including RAID arrays on the adapters 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 filename
SUSE Linux Enterprise Server 10 x64	sles10x64.xml
SUSE Linux Enterprise Server 10 x64 with Xen	sles10x64_xen.xml
SUSE Linux Enterprise Server 11 x64	sles11x64.xml

Operating System	Unattended filename
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 4GB of memory

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

RAID configuration fails for disks in JBOD

When performing RAID configuration for ServeRAID M-series controllers, any disks in state JBOD will not be used.

To avoid this problem, change the state of the disks from JBOD to unconfigured-good using the **Ctrl+H** menu during System POST prior to using theLinux Scripting Toolkit.

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. For more information on this problem, see RETAIN tip H1974.

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** during the Workflow Creation and OS installation task phases.

This deployment requires a new kickstart file. Create the new file by following these steps:

1. Create a new OS installation task based on the `esx4.ks` kickstart file.
2. Modify the new task to add the necessary kernel parameter:
   a. Modify the line:

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

      to be:

      ```
      bootloader --location=mbr --append="allowInterleavedNUMAnodes=TRUE"
      ```
3. In the OS installation section of the workflow, a field is provided for optional kernel parameters. Add the following value to this field:

   ```
   allowInterleavedNUMAnodes=TRUE
   ```

uEFI operating system installations not booting 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. Please consult the operating system information for instructions on 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 efi file and continue the installation using that option. Follow the steps below 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 which includes string "*.efi"
5. Input the description as `OS_Install` and select **Commit Changes**.

Follow the steps below 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 Web site.
- 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 order publications through the IBM Publications Ordering System at http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi.

Getting help and information from the World Wide Web

On the World Wide Web, the IBM website has up-to-date information about IBM System x and IntelliStation products, services, and support. The address for IBM System x information is http://www.ibm.com/systems/x/. The address for IBM IntelliStation information is http://www.ibm.com/systems/intellistation/pro/index.html.

You can find service information for your IBM products, including supported options, at http://www.ibm.com/supporthttp://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 http://www.ibm.com/services/, or go to http://www.ibm.com/planetwide/ for support telephone numbers. In the U.S. and Canada, call 1-800-IBM-SERV (1-800-426-7378).

Hardware service and support

You can receive hardware service through IBM Services or through your IBM reseller, if your reseller is authorized by IBM to provide warranty service. Go to http://www.ibm.com/planetwide/ for support telephone numbers, or in the U.S. and Canada, call 1-800-IBM-SERV (1-800-426-7378).

In the U.S. and Canada, hardware service and support is available 24 hours a day, 7 days a week. In the U.K., these services are available Monday through Friday, from 9 a.m. to 6 p.m.

Appendix 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 are ServerProven, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. These products are offered and warranted solely by third parties.

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

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

Printed in USA