RAID Software Suite for Linux

Installation Guide Rev 1.005

August 2000

Order Number: 273377-002

Contents

1 RAID Software Suite Installation for Linux

1.1 Introduction 5
1.2 Before You Begin 5
1.3 Software Requirements 6
1.4 Install the RAID Software Suite onto an Existing Linux Server 6
1.4.1 Installation 6
1.4.2 Rebuild the Kernel with I_2O Support 10
1.4.3 Activate the I_2O Subsystem to Support Kernel Modules 12
1.5 RAID Software Overview 13
1.5.1 RAID Configuration Services 13
1.5.2 RAID Configuration Services Administration 13
1.5.3 RAID Storage Console 13
1.5.4 IRVIEW 14
1.6 Using the RAID Software Suite 14
1.6.1 Using RAID Configuration Services Administration 14
1.6.2 Configure a RAID Volume in a Linux System 16
1.6.3 Monitor the Status of the RAID Subsystem 17
1.6.4 Uninstalling RAID Software Suite 18
1.7 Booting to Red Hat Linux on an I_2O RAID Volume 18
1.7.1 Method 1 - MBR on Primary IDE 18
1.7.2 Method 2 - MBR on I_2O RAID Volume 19
1.8 Technical Notes 22
1.8.1 Linux Dynamic Block Device Limitations 22
1.9 Miscellaneous Notes 22

Figures

1-1 Installation Confirmation 7
1-2 Install the Kernel Sources 8
1-3 Install Kernel Sources from Install Program 8
1-4 Bypass Installation of Kernel Source Code 9
1-5 Install Java Runtime Environment 9
1-6 Linux Kernal Configuration 10
1-7 Configuring I_2O Support in the Kernel 11
1-8 Configuring I_2O Support as Kernel Modules 11
1-9 RAID Configuration Services Administration 14
1-10 RAID Configuration Services Administration - Edit Configuration 15
1-11 RAID Configuration Services Administration - User Management 15
1-12 User Management Add User Screen 16
1-13 User Management Change Password 16
1-14 irview Screen 17

RAID Software Suite Installation for Linux 1

1.1 Introduction

This user's guide is designed to assist the Linux user in installing and operating RAID controllers using the RAID Software Suite. The end-user of this guide is assumed to have good working knowledge of the Linux operating system.

In addition to this guide, there are several other sources of information to assist you in using this product. They include the user's manual which accompanies the RAID controller, Linux-oriented newsgroups and web sites, and companies which provide their own Linux distributions.

1.2 Before You Begin

About This Distribution

The RAID Software Suite for Linux is based on the Intelligent I/O Specification (I_2O). The I_2O kernel modules were developed as an Open Source project. These modules were developed for two kernel versions: 2.2.16 (current stable versions) and 2.3.x (development version leading to 2.4 version). The modules can be obtained along with the latest kernel sources from ftp://ftp.kernel.org/pub/linux/kernel/. We suggest you obtain the very latest modules. *This distribution contains the 2.2.16 version of the* I_2O *modules.*

Over time the I_2O kernel modules will make there way into the official Linux distributions that other companies such as Red Hat incorporate into their releases.

Assumptions About Path Names

Since the user may have received the RAID Software Suite on the CD-ROM or as a download from the web, and because Linux administrators may mount volumes with any variety of names, certain assumptions were made in this document.

Path Name	Note
`/mnt/cdrom`	If the RAID Software Suite is not being installed from the CD-ROM, the `/mnt/cdrom` part of the path may be changed to the proper directory.
`(i) mylabel`	This refers to a label the administrator chooses to use throughout the document. The administrator should choose a descriptive label and use it consistently to reduce confusion or chance of error.
`/dev/hda*`	Path entries in this font are assumptions and may be changed to match what the administrator has selected for the particular system.

1.3 Software Requirements

- Linux RAID Software Suite distribution
  - — Linux Kernel I_2O Support (open source)
  - — RAID Storage Console / RAID Configuration Services
  - — Utility tools (Command Line tool: irview)
- Red Hat Linux 6.2 (installed on your machine)
- *Root* access on the Linux server
- Hard disk space: Minimum 20 MB

1.4 Install the RAID Software Suite onto an Existing Linux Server

1.4.1 Installation

1. Install the RAID Software Suite by placing the Linux RAID Software Suite CD-ROM in the CD-ROM Drive or running install.sh from the RAID Software Suite Distribution.

   If the installation does not automatically begin, then mount the CD-ROM:

   ```
   $ mount /dev/cdrom /mnt/cdrom
   ```

   and run the install script:

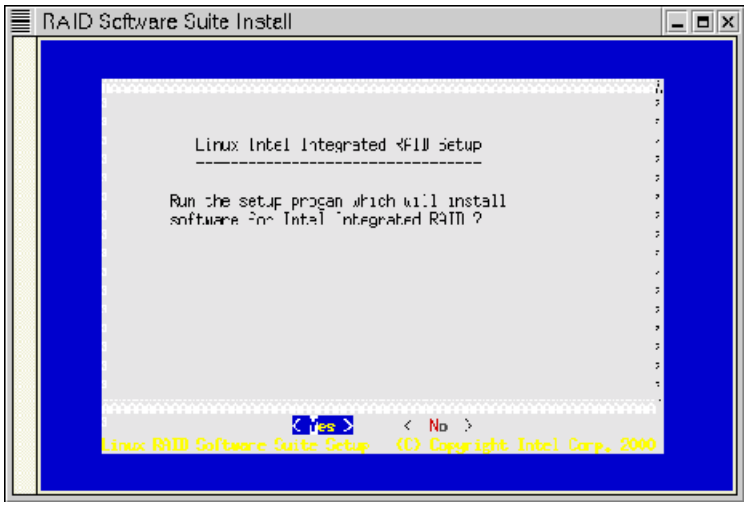
   ```
   $ /mnt/cdrom/autorun
   ```

   If X Windows is not available, then run the install script:

   ```
   $ /mnt/cdrom/install
   ```

Note: Ignore warning messages printed out on the console during installation. If the install script detects a fatal failure, it will automatically exit the program.

Figure 1-1. Installation Confirmation



2. Install the kernel sources with I_2O support (supplied with RAID Software Suite). See Figure 1-2.
   a. If you select *Yes* to install new kernel sources, the screen in Figure 1-3 appears. When you install the kernel sources supplied by the installation program, then the sources are unpacked in the `/usr/src` directory and appropriately linked to the `/usr/src/linux` directory.
   b. If you wish to install source code later than Linux 2.2.16, select *No*. The screen in Figure 1-4 appears. The kernel source code from RAID Software Suite is not installed and the remaining components of the software suite are installed.

Figure 1-2. Install the Kernel Sources

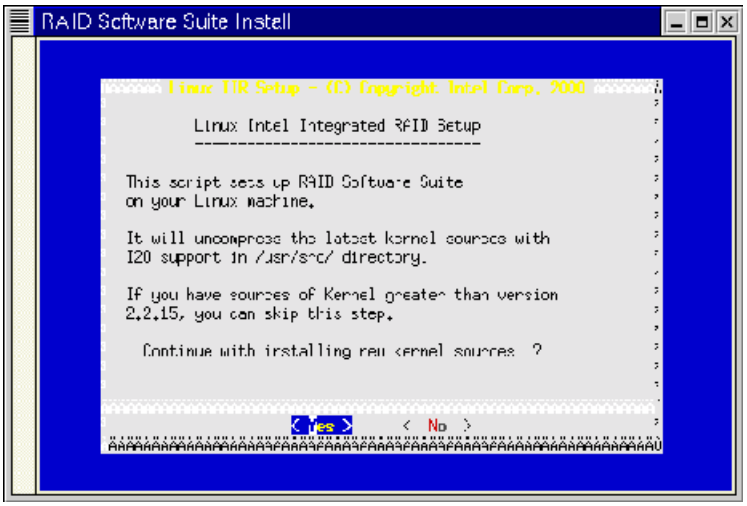


Figure 1-3. Install Kernel Sources from Install Program

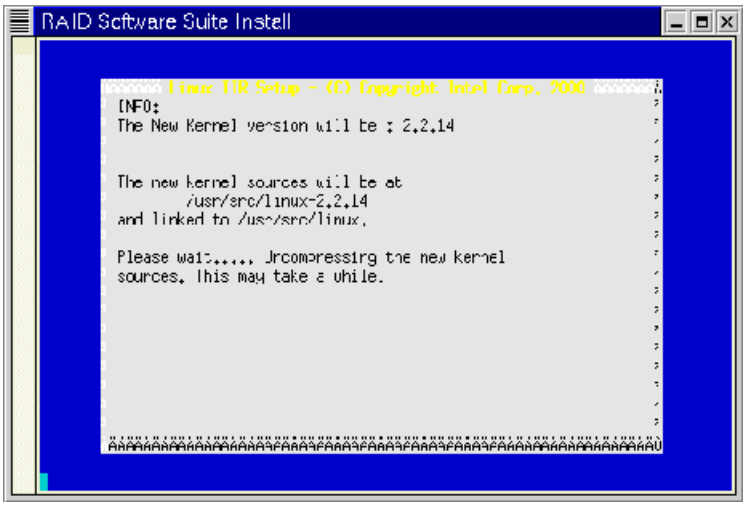
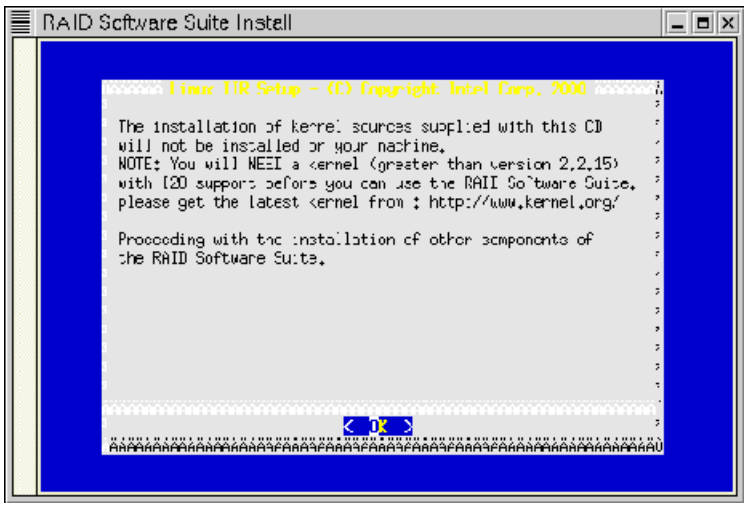


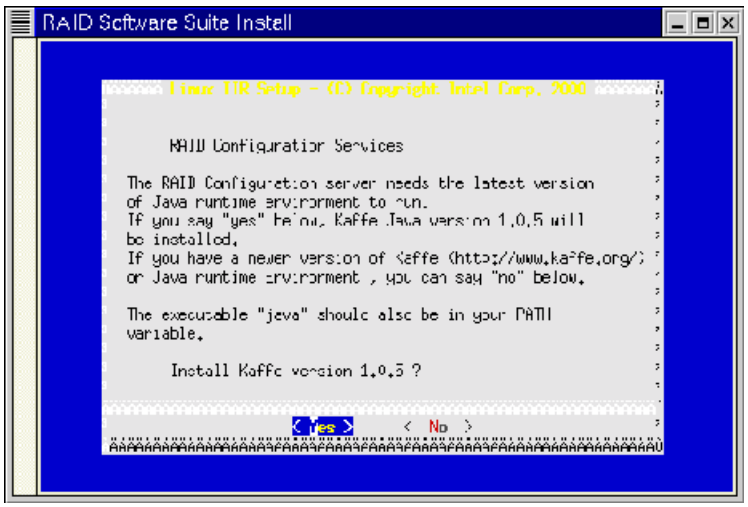
Figure 1-4. Bypass Installation of Kernel Source Code



3. Install Kaffe Java version 1.0.5 or later. Check the version of Kaffe installed on your machine, with the command "rpm -q kaffe". If your version of Kaffe Java is later than 1.0.5, then select *No*, otherwise select *Yes*. See Figure 1-5.

Note: to use the RAID Configuration Service, you must have "Java Runtime Environment" installed on your machine. Kaffe is an open-source Java initiative, which is also supplied with Red Hat Linux.

Figure 1-5. Install Java Runtime Environment



4. Complete the installation by clicking on *OK* in the screens.
5. Proceed to Section 1.4.2. to rebuild the kernel with I_2O support.

1.4.2 Rebuild the Kernel with I_2O Support

The Linux RAID Software Suite installation program will install the latest kernel source with I_2O support in the `/usr/src/linux` directory. You must rebuild the kernel with I_2O support. I_2O support can be either statically linked with the kernel or it can be built as kernel modules.

Note: The kernel can be built with many options and should be customized for your particular machine.

1. Type the following at the command line:

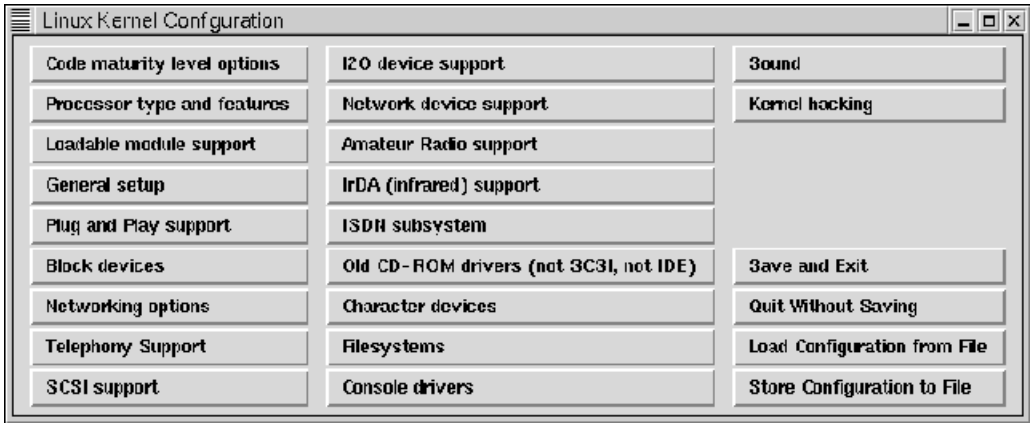
   ```
   $ cd /usr/src/linux
   ```

   then

   ```
   $ make xconfig
   ```

The screen in Figure 1-6 appears.

Figure 1-6. Linux Kernal Configuration



2. Configure I_2O support. Click on the I_2O support button on the Kernel Configuration screen. See Figure 1-6.

   a. There are two ways to configure I_2O support, stactic support in the kernel or as kernel modules. Select one of the following methods:

      Method A: *Statically* build I_2O support into the new kernel. Select "y" to the relevant I_2O choices. See Figure 1-7.

      Method B: Build the I_2O support as kernel modules. Select "m" to the relevant I_2O choices. See Figure 1-8.

      To be able to use the RAID controller, the I_2O PCI support and I_2O Block OSM components must be selected. Other components are optional.

Figure 1-7. Configuring I_2O Support in the Kernel

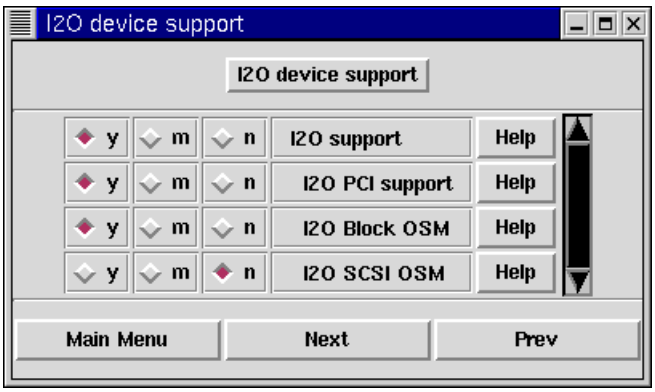
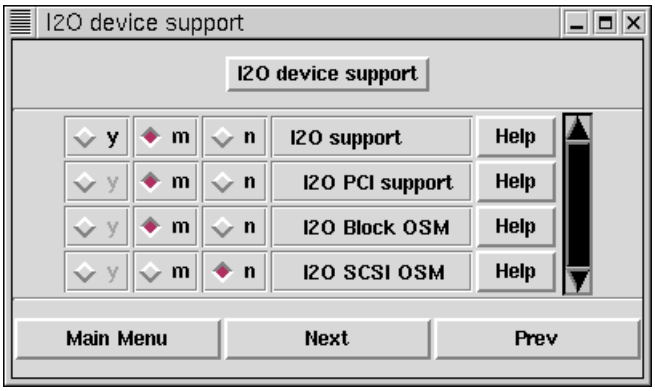


Figure 1-8. Configuring I_2O Support as Kernel Modules



Note: Ensure that "n" is selected for "I20 SCSI OSM." This option is not supported by Intel Integrated RAID controllers.

3. At this point, it is necessary to configure additional options in the "Linux Kernel Configuration" tool.

- If you are currently booting from a SCSI disk, ensure that support for your SCSI controller is built into the new kernel. Select "SCSI Support", then "SCSI low level drivers" to choose your SCSI card.
- Select "File Systems" and choose "y" for "Microsoft Joliet CDROM extensions".
- If "Symmetric multi-processing support" is selected under "Processor type and feature", select "Enhanced Real Time Clock Support" from the Character devices" option in the main menu.

Note: Before proceeding to step 4, make any other necessary changes to the kernel based on your server's hardware configuration.

4. Once you have completely configured the kernel, save your configuration and build the kernel and modules as appropriate.

```
$ make dep
$ make clean
$ make bzImage
$ cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz-mylabel
$ make modules
$ make modules_install
```

Additional reading for configuring and rebuilding the Linux kernel refer to /usr/src/linux/README or Linux Kernel Configuration HOWTO at http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html

5. Make relevant changes to `/etc/lilo.conf` per the following example:

   Before example:

   ```
   image=/boot/vmlinuz-2.2.12-20
   label=linux
   initrd=/boot/initrd-2.2.12-20.img
   read-only
   root=/dev/hda2
   ```

   After example:

   ```
   image=/boot/vmlinuz-mylabel
   label=mylabel
   initrd=/boot/initrd-2.2.12-20.img
   read-only
   root=/dev/hda2
   ```

6. Run `/sbin/lilo`. See Linux Loader (LILO) HOWTO at http://www.linuxdoc.org/HOWTO/mini/LILO.html for more information.
7. Reboot your machine and load the new kernel.
8. If you built I_2O support with kernel modules, go to Section 1.4.3 to activate the I_2O subsystem. If you built I_2O support statically into the kernel, go to Section 1.5 to begin the RAID Software Suite.

1.4.3 Activate the I_2O Subsystem to Support Kernel Modules

If you built the I_2O support statically into the kernel, then the RAID controller card will be automatically detected at bootup.

If you built the kernel with I_2O support as kernel modules, you must load the kernel modules.

1. Run the following commands:

   ```
   $ modprobe i2o_block
   ```

   then

   ```
   $ modprobe i2o_config
   ```

2. Confirm that the modules were loaded properly using the `lsmod` command.

   Type the following on the command line: `$ lsmod`

   The following output appears:

   ```
   Module Size Used by
   ```

```
i2o_config 6576 0
i2o_block 23476 1
i2o_core 36016 0 [i2o_config i2o_block]
i2o_pci 2368 1 [i2o_core]
```

For more information on Linux Kernel modules read /usr/src/linux/Documentation/modules.txt

1.5 RAID Software Overview

1.5.1 RAID Configuration Services

The RAID Configuration Services allow you to configure your RAID volumes, disks and arrays. You must start RAID Configuration Services before configuring your RAID volume. Type the following at the command line:

```
$ /opt/iir/bin/iird {start | stop | restart}
```

Access the RAID Configuration Services Administration through the shortcut located off of the system of your GNOME/KDE start menu.

This program may be called from any of the `init` scripts at boot-up and shutdown time. However, the RAID Configuration Services are only required to run when you are configuring the RAID subsystem with RAID Storage Console. Once the RAID subsystem configuration is completed, you may wish to stop RAID Configuration Services to free system resources.

1.5.2 RAID Configuration Services Administration

The RAID Configuration Services Administration utility is used to configure parameters of the RAID Configuration Services. This X Windows program provides the ability to manage user access, TCP/IP port settings, and remote access of the RAID Configuration Services.

An alternative to using the RAID Configuration Services Administration utility is to manually edit the `/opt/iir/iirserver/etc/config.iir` text file and restart the RAID Configuration Services.

More information on RAID Configuration Services Administration can be found in Section 1.5.1.

1.5.3 RAID Storage Console

The RAID Storage Console is an HTML-based configuration utility that is used to configure the RAID subsystem. The RAID Storage Console can be accessed locally or remotely.

Local

Access the local RAID Storage Console through the shortcut located off of the *system* branch of your GNOME/KDE start menu.

Remote

The RAID Storage Console can also be accessed remotely. If you want to access it remotely, use a web-browser and point it to:

http://hostname:960/

The RAID storage console is under restricted access. For authorized access, use the following initial username/password pair: `root/root`

Warning: Change the default password and /or username immediately after the installation. Failing to do so is a security risk. Refer to Section 1.6.1 for instructions.

See the RAID Controller User's Manual for a complete description of the RAID Storage Console utility and its usage.

1.5.4 IRVIEW

IRVIEW is a command line utility that provides real-time status of the RAID subsystem.

1.6 Using the RAID Software Suite

1.6.1 Using RAID Configuration Services Administration

1. Start the RAID Configuration Services Administration program by running:

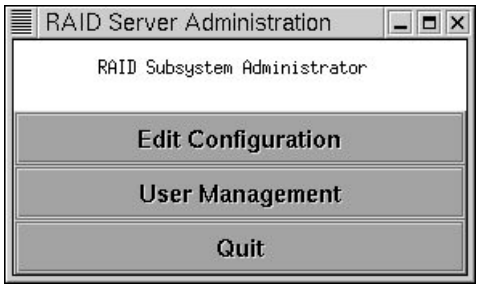
   ```
   $ /opt/iir/bin/iradmin
   ```

   or

   Access the RAID Configuration Services Administration through the shortcut located off of the system branch of your GNOME/KDE start menu.

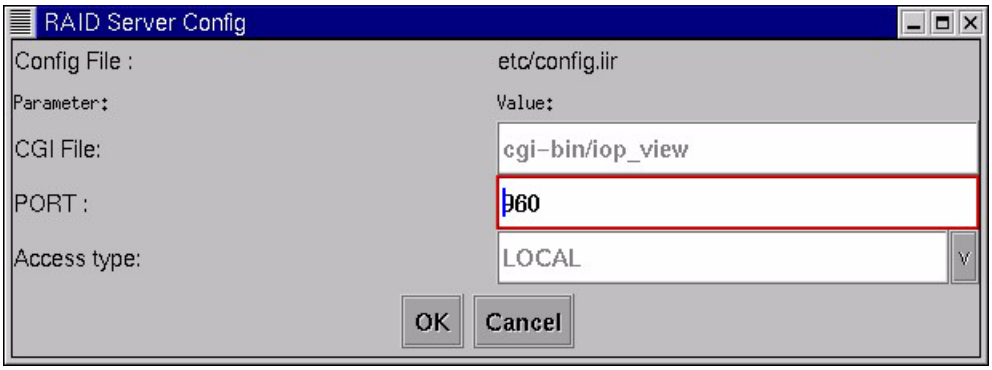
   The screen in Figure 1-9 appears.

Figure 1-9. RAID Configuration Services Administration



2. Modify the port and remote access settings of the RAID Configuration Services. Click the button labeled *Edit Configuration.* See Figure 1-10.

Figure 1-10. RAID Configuration Services Administration - Edit Configuration



The following parameters are available:

CGI File: This parameter indicates the interface to the I/O processor and should not be changed.

PORT: This is the TCP/IP port used to access the CGI file. Default value is 960.

Access type: This parameter indicates whether or not RAID Storage Console can be accessed from a remote server. The default value of LOCAL. For remote access, change the value to REMOTE.

After any change to the RAID Configuration Server, restart the server with the following command:

```
/opt/iir/bin/iird restart
```

3. To manage user access of the RAID Configuration Services, click on the *User Management* button on the screen in Figure 1-9. The screen in Figure 1-11 appears.

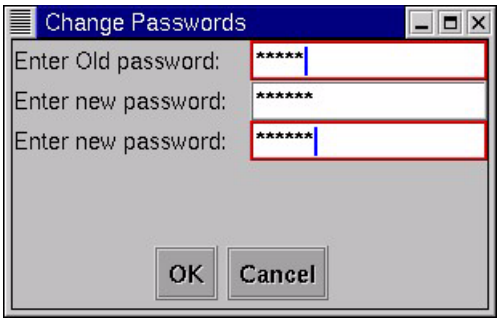
Figure 1-11. RAID Configuration Services Administration - User Management

To add a new user, enter a new user name in the text field and click the button labeled *Add*. Enter a new password at the prompt. See Figure 1-12.

Figure 1-12. User Management Add User Screen

To change the password of an existing user, enter an existing username and click on *Change*. Complete the password prompts as directed. See Figure 1-13.

Figure 1-13. User Management Change Password



User Management changes do not require a restart of the RAID Configuration Services to take effect.

1.6.2 Configure a RAID Volume in a Linux System

Warning: See Section 1.8.1, “Linux Dynamic Block Device Limitations” on page 1-22. Understanding the information in this section is critical to completing the following procedures.

1. To create RAID volumes, click on the RAID Storage Console icon in the system branch of your Gnome or KDE start menu or use a web browser and point to: http://localhost:960/.
2. Use fdisk to create partitions on the RAID volumes using the following command:

   `$ fdisk /dev/i2o/hd` [`a` : first RAID volume - `p` : sixteenth volume created]

   Note: All I_2O devices created are represented as "device files" on a Linux system. These device files are: `/dev/i2o/hda` to `/dev/i2o/hdp` for a total of 16 volumes that can be created. Individual partitions can be accesses (after creation using fdisk) as `/dev/i2o/hda1`, `/dev/i2o/hda2` … `/dev/i2o/hdp4`

3. Make Linux native filesystem (ext2) using the following command:

   `$ mke2fs /dev/i2o/hd` [a-p][1-15: the partition number]

Note: Use `mkfs` to create any type of filesystem that is supported by your Linux system/kernel. Ignore any ‘Invalid argument passed to ext2...’ error messages that may appear on the screen while in text mode.

4. Mount the partition on a directory (create one if necessary):

   ```
   $ mkdir /mnt/i2o
   $ mount /dev/i2o/hd [a-f][1-15] /mnt/i2o
   ```

   Check the RAID volume as a mounted directory with the following command:

   ```
   $ df -k
   ```

   The following is a sample of the output that appears:

```
Filesystem     1k-blocks  Used     Available  Use%   Mounted on
/dev/hda2      2912980    2356260  408748     85%    /
/dev/i2o/hda1  13116500   20       12450188   0%     /mnt/i2o
[root@localhost /root]#
```

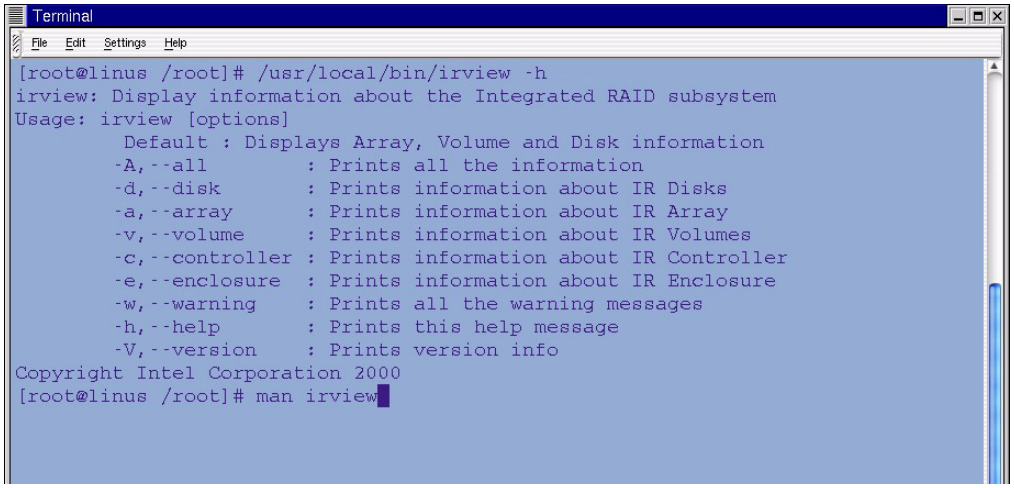
1.6.3 Monitor the Status of the RAID Subsystem

Monitor the status of the RAID subsystem by using the command line utility called `irview`. To invoke this utility, type `/usr/local/bin/irview` on the command line. To get more information on irview, invoke the corresponding man page (`man irview`).

Note: If not in XWindows while running irview, ignore any non-fatal error messages that may appear, such as ‘i2o_core:post_waitreply...’

A sample irview help screen is shown in Figure 1-14.

Figure 1-14. irview Help Screen



1.6.4 Uninstalling RAID Software Suite

To uninstall the RAID Software Suite run the following:

```
/opt/iir/bin/uninstall
```

Note: The uninstall program will not remove the kernel sources or installed modules.

1.7 Booting to Red Hat Linux on an I_2O RAID Volume

There are two methods available for booting Red Hat Linux from an I_2O RAID Controller. The first method installs Red Hat Linux on an IDE or SCSI hard drive and loads LILO (the Linux Loader) from the Master Boot Record (MBR) on that original drive (***/dev/hda***), which then loads the Linux kernel which is located on a disk or volume connected to the I_2O RAID Controller. The second method installs LILO onto the MBR of the I_2O boot device (`/dev/i2o/hda`) and allows the system to boot directly from a disk or volume connected to the I_2O RAID Controller.

Important: You must have built I_2O support statically into the kernel for either of the following methods (Section 1.7.1 or Section 1.7.2) to work. See Section 1.4.2 and Section 1.4.3 for more information.

Warning: See Section 1.8.1, "Linux Dynamic Block Device Limitations" on page 1-22. Understanding the information in this section is critical to completing the following procedures.

1.7.1 Method 1 - MBR on Primary IDE

Method 1 assumes that you have installed Red Hat Linux 6.2 on the primary IDE drive (***/dev/hda***). Although an IDE drive is assumed for these instructions, Method 1 could be used when Red Hat Linux is installed on a SCSI hard drive. For example, the boot device file might change depending on the installation.

1. Complete the Red Hat installation and start the RAID Software Suite installation as described in Section 1.4.1, "Installation" on page 1-6.
2. Install the I_2O kernel as described in Section 1.4.2, "Rebuild the Kernel with I2O Support" on page 1-10. You MUST statically build I_2O support into the new kernel.
3. Complete the RAID Software Suite installation and reboot the system with I2O support.
4. Using RAID Storage Console, create a primary I_2O volume that is large enough to accommodate your complete Red Hat installation.
5. Create a `/dev/i2o/hda1` partition on the primary I_2O volume. Ensure that the partition is equal to or greater than the partition ***/dev/hda1***. This procedure assumes that your root partition is installed on ***/dev/hda1***. If your volume is > 8 GB, you must ensure that the partition fits within the first 1024 cylinders of the disk.
6. A modified version of LILO is provided with the RAID Software Suite. Copy it to `/sbin` and make it executable as follows:

   ```
   $ cp /mnt/cdrom/iir_install/lilo/lilo-i2o /sbin/lilo-i2o
   $ chmod 500 /sbin/lilo-i2o
   ```

7. Copy the contents of the root partition on the IDE drive to the new root partition:

   ```
   $ dd if=/dev/hda1 of=/dev/i2o/hda1
   ```

   Note: This operation may take several hours to complete.

8. Run a file system check on the new root partition and answer yes (y) to all changes:

   ```
   $ fsck /dev/i2o/hdal
   ```

9. Mount the root partition on a file system mount point:

   ```
   $ mount /dev/i2o/hda1 /mnt/i2o -t ext2
   ```

10. If your /boot was on a different partition than the root partition, then copy /boot to this new root partition:

    ```
    $ cp -R /boot /mnt/i2o
    ```

11. Edit `/etc/lilo.conf` and add the following boot entry:

```
image=/mnt/i2o/boot/vmlinuz-mylabel
label=boot-i2o
root=/dev/i2o/hda1
read-only
```

12. Edit `/mnt/i2o/etc/fstab`. Change the following line:

```
/dev/hda1 / ext2 defaults 1 1
```

to read:

```
/dev/i2o/hda1 / ext2 defaults 1 1
```

The above line will cause the kernel to mount `/dev/i2o/hda1` as the root file system.

13. Run LILO with I_2O support using this command:

```
$ /sbin/lilo-i2o
```

14. Reboot

1.7.2 Method 2 - MBR on I_2O RAID Volume

Method 2 assumes that you have installed Red Hat Linux 6.2 on the primary IDE drive (***/dev/hda***). Copy the Linux kernel, LILO, and the MBR to a RAID volume that is connected to the I_2O RAID controller. To boot from the RAID volume, a modified version of the LILO binary is required. This modified version ignores the existence of IDE drives on the system.

Note: This boot method differs from the way that Linux and LILO are designed to function. The LILO boot loader provided in the 2.2.16 kernel has a limitation that does not allow the system to boot from a non-IDE drive if an IDE drive exists. The modified LILO bypasses this requirement by ignoring the fact that there maybe devices other than the I_2O device on the system. However if you have other IDE devices on your system, you may experience problems with those devices when using method 2.

1. Complete the Red Hat installation and start the RAID Software Suite installation as described in Section 1.4.1, “Installation” on page 1-6.
2. Install the I_2O kernel as described in Section 1.4.2, “Rebuild the Kernel with I2O Support” on page 1-10. You MUST statically build I_2O support into the new kernel.
3. Complete the RAID Software Suite installation and reboot the system with I_2O support.
4. Create a RAID volume that is large enough to accommodate your complete Red Hat installation. Be sure the new volume contains enough space for the boot, swap and other partitions.
5. Create a `/dev/i2o/hda1` partition on the RAID volume. Ensure that this partition is equal to or greater than the partition ***/dev/hdal***. This procedure assumes that your root partition is installed on ***/dev/hda1***. Be sure to leave enough free space on the RAID volume for swap and other partitions. If your volume is > 8 GB, you must ensure that the partition fits within the first 1024 cylinders of the disk.

6. Create a `/dev/i2o/hda2` partition for swap space on the RAID volume.
7. A modified version of LILO is provided with the RAID Software Suite. Copy it to `/sbin` and make it executable as follows:

   ```
   $ cp /mnt/cdrom/iir_install/lilo/lilo-i2o-hack /sbin/lilo-i2o-hack
   $ chmod 500 /sbin/lilo-i2o-hack
   ```

8. Copy the contents of the root partition on the IDE drive to the new root partition:

   ```
   $ dd if=/dev/hda1 of=/dev/i2o/hda1
   ```

   Note: This operation may take several hours to complete.

9. Run a file system check on the new root partition:

   ```
   $ fsck /dev/i2o/hdal
   ```

10. Mount the root partition on a file system mount point:

    ```
    $ mount /dev/i2o/hda1 /mnt/i2o -t ext2
    ```

11. If your /boot was on a different partition than the root partition, then copy /boot to this new root partition:

    ```
    $ cp -R /boot /mnt/i2o
    ```

12. Edit the `/mnt/i2o/etc/lilo.conf` file by modifying the `boot`, `map`, `install` and default lines at the top of the file to read as follows:

    ```
    boot=/dev/i2o/hda
    map=/mnt/i2o/boot/map
    install=/mnt/i2o/boot/boot.b
    default=mylabel
    prompt
    timeout=20
    ```

13. Add the following lines to the `/mnt/i2o/etc/lilo.conf` file.

    ```
    image=/mnt/i2o/boot/vmlinuz-mylabel
    label=mylabel
    root=/dev/i2o/hda1
    read-only
    ```

14. Edit `/mnt/i2o/etc/fstab.` Change the following line:

    ```
    /dev/hda1 / ext2 defaults 1 1
    ```

    to read:

    ```
    /dev/i2o/hda1 / ext2 defaults 1 1
    ```

    The above line will cause the kernel to mount `/dev/i2o/hda1` as the root file system.

15. Change your swap drive from the IDE drive partition (assumed to be ***/dev/hda5***) to the RAID volume by changing the following line in `/mnt/i2o/etc/fstab:`

    ```
    /dev/hda5 swap swap defaults 0 0
    ```

    to read:

    ```
    /dev/i2o/hda2 swap swap defaults 0 0
    ```

16. Run the modified LILO binary with the LILO configuration on the root RAID volume:

    ```
    $ /sbin/lilo-i2o-hack -C /mnt/i2o/etc/lilo.conf
    ```

17. Shutdown the system:

    ```
    $ shutdown -h now
    ```

18. Remove the IDE drive and restart the system.

19. Once the system has powered up, clean up `/etc/lilo.conf` by changing the map point reference of the `map`, `install` and `image` entries as indicated below. This will make future updates of the kernel on the I_2O device easier.

    ```
    boot=/dev/i2o/hda
    map=/boot/map
    install=/boot/boot.b
    prompt
    timeout=20

    image=/boot/vmlinuz-mylabel
    label=mylabel
    root=/dev/i2o/hda1
    read-only
    ```

In the future, if you edit `lilo.conf`, run LILO w/I_2O support (lilo-i2o) instead of the modified LILO (i.e lilo-i2o-hack).

1.8 Technical Notes

1.8.1 Linux Dynamic Block Device Limitations

The RAID volumes created using the I_2O Linux drivers are seen by Linux as "Block Devices". The RAID Software Suite gives a powerful new feature to create new block devices while a Linux system is up and running via the use of the Storage Console. The new RAID volume created is detected as /dev/i2o/hd* and can be immediately used as a normal block storage device. This block device was dynamically created while the system was up and running and is associated to a particular device file (/dev/i2o/hd*) and a Major/Minor number, however, Linux does not associate a physical storage device with the device file (or rather the Major/Minor) number across reboots.

The implication of this limitation is that when a reboot occurs after a user has dynamically created a dynamic block device, the device files associated with the physical storage device *changes* once the system comes back online. If there were entries in the /etc/fstab and /etc/mtab associated with these dynamic block devices, then they will no longer be valid and the system will either incorrectly mount these devices or it may not mount them at all. Your system may not boot if the root file system was also on a dynamic block device.

Workarounds

- Create all the RAID volumes you require and reboot before you designate persistent mount points for them. Take this approach especially if you are planning to mount the

root file system on a RAID volume. Avoid creating additional RAID volumes after designating one of the existing volumes as the root file system.

- Remove any existing entries in /etc/fstab pertaining to I_2O devices. Create a filesystem on the new RAID volume and reboot. When the system has rebooted, examine the I_2O block devices to determine the mapping of the physical I_2O block devices to /dev/i2o/hd* device files. Mount the I_2O block devices onto the desired mount points and enter the corresponding entries in the /etc/fstab config file. If you built the I_2O support as kernel modules, then a reboot is not necessary. Simply unload and reload the I_2O modules.

1.9 Miscellaneous Notes

- The `/opt/iir` directory requires a minimum of read, write, and execute privileges for `root`.