D. sigbru Utility

"**sigbru**" is Sigmet's manual/automatic backup and restore utility. This utility provides system administrators with an easy–to–use tool for creating and restoring backups from supported archive media. The supported media are:

- **DAT Tape-** for HP–UX, IRIS and Linux systems. HP SureStore DAT's are the most common and have proven to be very reliable.
- **HDD-** hard disk drive for HP–UX, IRIS and Linux systems.
- **DVD-** for Linux systems only. DVD+RW is supported for writing backups. Use only media from well–known manufacturers such as SONY, Fuji or Memorex.

Note that **sigbru** is only one component of an effective backup strategy. Proper system documentation and advance preparation are essential in assuring that when (not if) your hard disk fails, you can easily resume operation. This chapter also includes recommended procedures for a comprehensive backup strategy.

Important: SGI and HP Users should use the system backup tools supplied by these manufacturers. sigbru backups can be used in addition to these or may not be used at all.

---

sigbru is a graphical user interface which works in conjunction with Sigmet's **sigbru**sh script files to allow easy archiving without having to know any of the sigbrush command line options. **sigbrush** interfaces with gnu's tar version 1.13.11 to give the tar utility even greater flexibility and control. The output is a "tar" file that is stored either on tape or disk. The tar file can optionally be gzip compressed.

One of the important features of **sigbru** is that it allows backups of a system to be made and placed on a tape or disk drive on another computer. The advantage of this is that the archive can be centrally located and the administrator can then backup systems to this one drive (typically a tape where he/she is sitting). Note that this requires a high–speed network connection, i.e., at least a T1 connection is recommended (1 megabit per second).

Note: DVD backup and restore is supported only on a local DVD drive.

A very powerful feature of **sigbru** is the automatic archiving feature. This allows a directory on disk to be monitored so that when a specified quota of disk space is reached, **sigbru** automatically archives the contents to the archive medium. One application is for use to archive non–IRIS products that are created by format conversion through an output pipe (the normal IRIS menu archive approach does not work for these).

In this chapter:

D.1 System Configuration for sigbru

D.1.1 Authorization to login as root on a remote system

You can run **sigbru** either as root or operator. However, as operator you will have reduced privileges such as not being allowed to restore at all, or not being able to do a full system backup which requires root access to various directories.

If you want to run **sigbru** on a remote computer using an xterm (or sigterm), the operating system protection may block your login as root. You would notice that even if you provide the proper root password, you are not allowed to login.

You can circumvent this problem by going to the remote system and then moving the security file to another name, i.e., on the computer that you wish to access:

```
# cd /etc
# mv securetty securetty.orig
```

Test that you can now do a remote login as root from a another system. You only need to do this once.

D.1.2 Authorization to use a remote tape drive or remote disk drive

Note: sigbru does not support backup and restore on a remote DVD. Only a local DVD can be used. sigbru does support use of both a remote tape drive and a remote hard disk.

If the computer that you are backing-up does not have a tape drive, you can back-up to a remote computer that does have a drive. You need to set-up special authorization file for this (/etc/pam.d/rsh). First, on the remote computer (with the tape drive) backup the old file and then use your favorite editor or “vi” to edit the file:

```
# cd /etc/pam.d
# cp rsh rsh.bak
# vi rsh
```

The file will look something like below. The exact lines vary by installation:

```
#%PAM-1.0
auth       required     /lib/security/pam_rhosts_auth.so
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_pwdb.so
session    required     /lib/security/pam_pwdb.so
```

Under the first line (which is commented with #) add the line:

```
auth       sufficient   /lib/security/pam_rootok.so
```

The edited file will look something like:

```
#%PAM-1.0
auth        sufficient    /lib/security/pam_rootok.so
auth        required      /lib/security/pam_rhosts_auth.so
auth        required      /lib/security/pam_nologin.so
account     required      /lib/security/pam_pwdb.so
session     required      /lib/security/pam_pwdb.so
```

Save the modified file.

D.1.3 Archive device and media configuration for sigbru

Sigbru supports three different archive devices:

- DAT Tape
- HDD Hard disk drive
- DVD+RW

The configuration of both the drive and the media for each of these is described below:

DAT for sigbru

The most common type of DAT used on IRIS systems is the HP SureStore. This comes in several versions (e.g., version DAT 72). Make sure that you purchase tapes that are compatible with your hardware version.

In sigbru, the typical device names for DAT tapes are selected right in the menu. If this is not the correct selection for your system, you may type–in the correct device name. Check with your system manager if you are uncertain. Note that IRIS systems with tapes will also input the device name in the setup/output/archive device menu so you can check there as well.

The privileges for the device should be set as follows (in the typical Linux case of the DAT device name /dev/nst0)

```
chmod 666 /dev/nst0
```

DAT tapes for sigbru do not need to be initialized.

HDD for sigbru

You only need create a disk directory on the archive host you will be using. Note that you can make this the local computer. Having and HDD backup on your local computer makes it very convenient to restore files, but is not a good idea for a full disk backup since you want to protect against failure of the disk itself.

To make the directory and set its privileges, become root and then, for the example of a directory named /iris_data/backups, type the following:

```
# mkdir  /iris_data/backups
# chmod 666 /iris_data/backups
```

HDD directories for sigbru do not need to be initialized. Note that you can have several different backup files in this directory so it is not necessary to create a new directory every time you run sigbru.

DVD for sigbru (Linux only)

Only DVD+RW drives are supported so make sure that you have one of these (SONY makes a nice one that we use at SIGMET). First you need to determine the device name that has been assigned to the DVD by the Linux OS. To do this, as root type:

```
# cdrecord –scanbus
```

The operating system will respond with many lines that look like:

```
...
scsibus3:
        3,0,0   300) 'SAMSUNG ' 'CD-ROM SC-148C  ' 'B100' Removable CD-ROM
        3,1,0   301) 'SONY    ' 'DVD RW DRU-500A ' '2.0c' Removable CD-ROM
        3,2,0   302) *
...
```

Here we see the SONY DVD we are looking for. It's device name is /dev/scd1 which is taken from the middle of the three leading numbers, i.e., the "1" from 3,1,0.

Next set the protections as follows (continuing to use /dev/scd1 as the example):

```
# chmod 666 /dev/scd1
```

No create symbolic link

```
# ln –s /dev/scd1 /dev/dvd
```

Then create a mount point for the DVD

```
# mkdir /mnt/dvd
```

Finally, initialize the DVD (essentially formatting the DVD):

```
# init_sigbru_dvd
```

D.2 Starting sigbru

Important: sigbru is run on the machine that you want to backup / restore.

Important: You must be root to do full sigbru backup and restore operations.

Command Line Options for Starting sigbru

sigbru has several command line options summarized below:

```
–help                      Print this list.
–auto                      Start Sigbru with auto archive options.
–enabled                   Enable auto archive function.
–include <dir>             Directory included in archive.
–exclude <dir>             Directory excluded from archive.
–compress                  Enable gzip compression.
–delete                    Delete files after archive.
–archivehost <hostname>    Hostname where archive device is located.
–quota <XXX.X>             Number of GB per each archive event
–device <device>           Name of archive device.
–display                   Display name.
```

The meaning of each of these options is described in the subsequent descriptions of the **sigbru** menu fields.

Running from a Local Terminal Window (IRIS is installed)

On a local terminal window simply open a terminal as root on the system that you want to backup and then type:

```
# sigbru
```

If the system cannot find **sigbru** because the UNIX search path is not defined, then you can start **sigbru** by typing:

```
# cd /usr/sigmet/bin
# ./sigbru
```

If this does not find **sigbru**, then perhaps IRIS is not properly installed and you should use the cdrom method described below.

Running from a remote workstation (IRIS installed on target system)

From a remote machine, you can use the sigterm <hostname> command to open a terminal over the network. Then become super user and follow the “Local Terminal” procedure described above.

Alternatively you can rlogin or telnet to the machine that you want to backup, become root and then type:

```
# export DISPLAY=hostname:0.0
```

Here you substitute the hostname of the computer where you are sitting. You may also have to type the command “xhost +” on a terminal on your local display to allow the remote machine to display the **sigbru** menu on your screen.

If IRIS is not installed- start sigbru from the CDROM

If you haven't installed IRIS, then **sigbru** will not be installed. This might happen if you are doing a restore operation, i.e., you need to restore IRIS and **sigbru**. In this case you can start **sigbru** directly from the SIGMET IRIS Release CDROM.

Insert the IRIS Release CDROM on the system where you want to run **sigbru**. Depending on your system you may need to mount the CDROM. See the instructions in the *IRIS Installation Manual*, Section 1.2 "Mount the CD".

You can check that the CDROM is properly mounted by issuing the "df" command. This will also tell you what the mount point is (assumed here to be "/cdrom" for the linux example. Be careful to use upper or lower case as indicated by df). Once the CDROM is mounted type the following to select the correct version of **sigbru** for you workstation:

```
for SGI/IRIX   # cd /CDROM/irix/sigbru
for HP-UX      # cd /cdrom/hp-ux/sigbru
for PC/Linux   # cd /mnt/cdrom/linux/sigbru
```

Then start **sigbru** by typing:

```
# ./sigbru
```

Now that **sigbru** is running, refer to the next section which describes the various features of the **sigbru** menu.

Copying the sigbru files from a local or remote CDROM

You can copy the **sigbru** files to your system from either a local or remote CDROM. First create a directory on your local computer to hold the **sigbru** files. SIGMET recommends the following location:

```
# mkdir /root/sigbru
```

The CDROM must be mounted and the mount point (assumed here to be "/cdrom") must be properly specified. You can check both by issuing the "df" command. Now copy the files that you need from the CDROM with the IRIS Installation disk, to this directory:

- For a CDROM drive on your local system (linux example):

  ```
  # cp /cdrom/linux/sigbru/*  /root/sigbru
  ```

- For a CDROM drive on a remote system,

  ```
  # rcp nodename:/cdrom/linux/sigbru/*  /root/sigbru
  ```

To start **sigbru** type:

```
# cd /root/sigbru
# ./sigbru
```

Copying the sigbru files from another IRIS system

You can copy the **sigbru** files to your system from another system on the network. The example here uses “rcp” (remote copy), but you could use ftp or NFS (via “cp”) as well to do this. First create a directory to hold the **sigbru** files. SIGMET recommends the following location:

```
# mkdir /root/sigbru
```

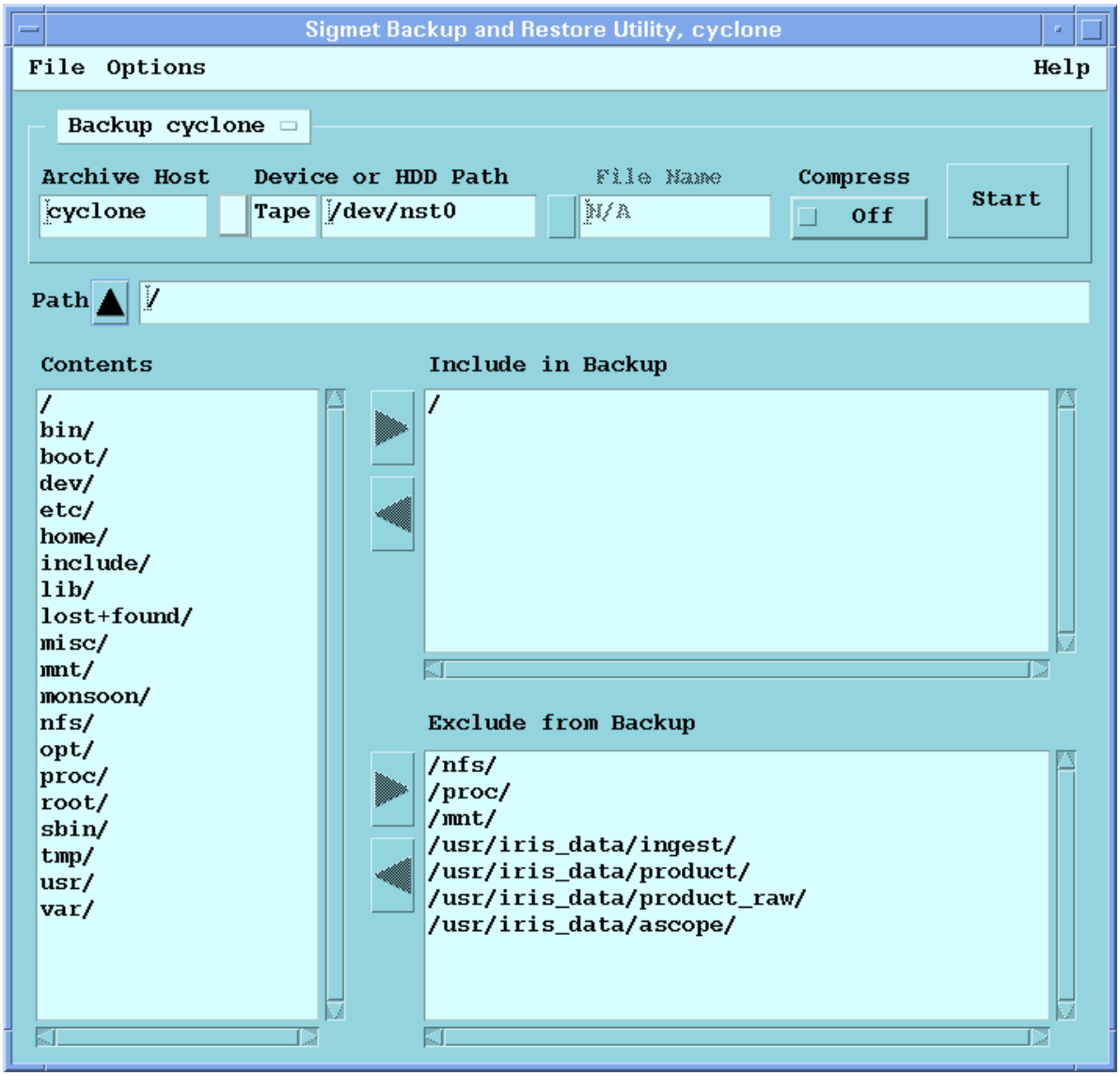
Now copy the files that you need from the remote system that has IRIS (nodename) to your local system (careful with the rename of the first file from **sigbru** to sigbru.rf):

```
# rcp nodename:/usr/sigmet/bin/app–defaults/sigbru  /root/sigbru/sigbru.rf
# rcp nodename:/usr/sigmet/bin/sigbru  /root/sigbru
# rcp nodename:/usr/sigmet/bin/sigbrush  /root/sigbru
# rcp nodename:/usr/sigmet/bin/gnufind  /root/sigbru
# rcp nodename:/usr/sigmet/bin/gnutar  /root/sigbru
# rcp nodename:/usr/sigmet/dt/icons/hour32.bm  /root/sigbru
# rcp nodename:/usr/sigmet/dt/icons/hour32m.bm  /root/sigbru
```

To start **sigbru** type:

```
# cd /root/sigbru
# ./sigbru
```

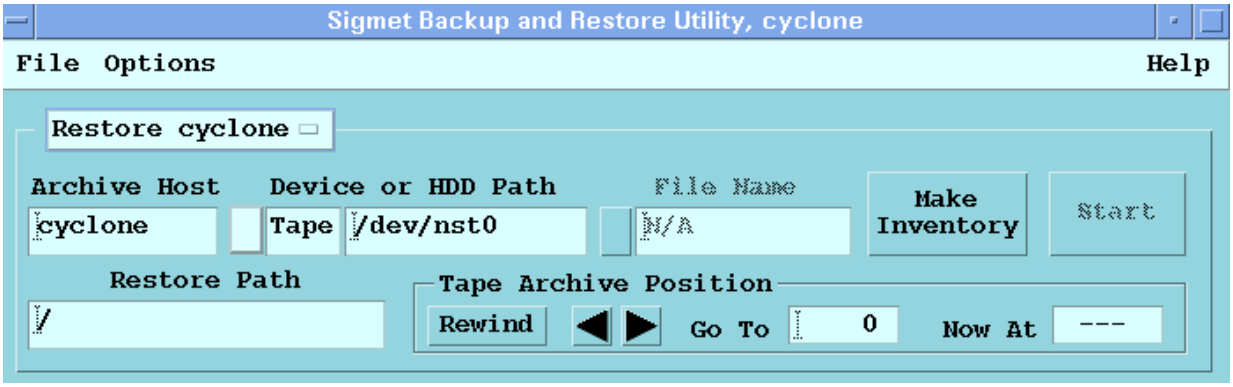
D.3 The sigbru Menu



The **sigbru** user interface, shown above, allows you to define what files are moved to/from the archive media. ***Remember,* sigbru *is run on the system where you want to backup/restore and you must be root to run sigbru.*** A tape drive or HDD can be located on another system, however a DVD must be on the same system.

To get the automatic archiving features of sigbru, use "sigbru –auto"

The figure below shows the appearance of the top part of the sigbru menu for the case of restore operation.



The various menu fields are described below:

Title Bar

This identifies the network node name on which **sigbru** is being run. In the example, the node name is “cyclone”. This is the hostname of the system that will be backed–up or restored.

File

The only option is “Exit”. This is how **sigbru** is normally exited.

Options

This is used to manually start or view the status log window. The status log contains useful messages about the backup and reports any errors or problems.

Backup/Restore <host name>

Use this to select whether you are doing a backup or restore. The hostname of the computer is indicated as a reminder of what system is being used. The choice of Backup/Restore changes some of the other menu options. In the description of the various fields, both the backup and restore functions will be clearly indicated.

Archive Host (Backup and Restore)

Specify the network hostname where the archive tape or disk file is located. The tape or disk drive can be on another machine on the network. Note that the slower your network, the longer it will take to make the remote backup to another machine. SIGMET recommends a 1 MBit/sec line (T1) as a minimum.

Note that in the case of a DVD, the archive host is fixed to be your local workstation.

Device or HDD (Hard Disk Drive) Path (Backup Case)

First use the button to select the kind of media (DVD, HDD or Tape). For the case of “Tape” you will be prompted to specify the type of UNIX that you are running HP–UX, IRIX, Linux). **sigbru** then automatically fills–in a device file name which

would be typical for your system and in most cases will work fine as is. However, depending on the specific configuration of your system, you may have to type in a different device file name. Check with your system administrator.

The example shown at the beginning of this section is the Linux device file name for a tape drive (the n specifies a non-rewind on close). If you select HDD or DVD, then you would type-in the filename that you want to use for the archive.

Note on Tape Drive Device Name: SIGMET recommends using the device name corresponding to "no rewind on opening". For Linux systems, for example /dev/nst0. Use of a non-rewind tape device driver permits multiple archive or backup files to be placed on the same tape for both archive and restore functions.

If you store your archive in a DVD or HDD file, then you should use the following naming convention:

- *.gz for compressed files
- *.tar for uncompressed files

where the you would substitute a file name of your choosing for the *. For example, if you want to store a compressed backup in a directory called /tmp and you want to name it "back01" then you would type-in:

```
/tmp/back01.gz
```

For more information on backup devices and media please see section D.1.3.

Device or HDD (Hard Disk Drive) Path (Restore Case)

In the case of doing a restore from an HDD or DVD, the button next to filename is activated. Click this to select among the archive tar files files in the specified directory. Then click the "Inventory" button to get a full listing of what is stored in the backup file.

gzip Compress (Backup only)

Click the button in for compression. Compression uses less space on the disk or tape archive, or, if you are using a remote networked tape or disk, compression will allow faster network transmission. The disadvantage of compression is that it does slow the archive process. For this reason, compression is recommended only for the following two applications:

- When the backup would not otherwise fit on the tape or disk archive. In this case, your only choice is to use compression.
- When you are using a remote tape or disk and network speed is the limiting factor. In this case it does not matter that the compression slows the archive process since the network transmission step is the limiting factor.

Make Inventory (Restore only)

Push the “Update” button to get an inventory of the files on the tape. In the case of multiple archive files on the same tape, the tape positioning features of the restore menu can be used to select the archive file.

When you perform an inventory, the Status display will show all of the files that are in the the archive record. At the top of the list, is the date and time at which the archive record was written. This is generally useful, but especially useful for tapes on which there are multiple archive records.

Restore Path (Restore Only)

This specifies the starting path for the restore. In most cases this should always be set to “/” (the default). However, if you want to place files in a temporary directory (for example) so that you do not overwrite existing files, then you can type–in the directory name. For example, if you wanted to restore all files in the /usr/sigmet directory to /tmp/usr/sigmet, then you would enter /tmp in the Restore Path.

Path and Contents (Backup and Restore)

This field shows the current UNIX file path. Note that “/” is at the top of the UNIX file tree. Click the up arrow to go up a level. The files and sub-directories (indicated by trailing “/”) in the path are shown in the left hand column labeled “Contents”. You can double click on a sub-directory to go down a level. This will be reflected in the “Path”.

Include and Exclude from Backup (Backup Only)

You can select files to include in your backup by highlighting one or more files or directories in the “Contents” list and then clicking the right arrow to put them in the “Include” list. You can highlight multiple files by click–dragging the mouse over consecutive files, or by holding the “Shift” key and clicking. If you select a directory to include, all files in that directory and in any subdirectories will be included in the backup. The directory structure will be preserved.

If you change your mind, you can highlight files and directories in the “Include” column and then click the left arrow to remove them from the list.

Similarly you can select files or directories to exclude from the backup, e.g., nfs directories exported from other machines.

Include in Restore (Restore Only)

In the Restore mode, first select the tape or disk file that has the backup archive and then click in “Contents” to highlight the files or directories that you want to restore. Use the right arrow to move them in to the “Include in Restore” list.

Tape Archive Position Features (Restore Only)

In some cases a single tape may contain multiple archive records from **sigbru** backups or archives written sequentially. This can result from either manual or automatic archive operations that were performed with a "non-rewind" tape device.

In restore mode, there is a tape positioning section to allow the user to control the archive record at which the tape is positioned. This is shown below:

- **Rewind** This is the first button you should click to select an archive file. The **Go To** and **Now At** fields will both show "1" when the rewind is complete.
- <– (back) and –> (forward) arrow buttons decrement/increment the 1 the **Go To** archive record request field.
- **Go To** Shows the current archive record request. Type–in a value or use the arrow keys to change this. You can change this "on-the-fly" even while a search is being made. If you did not first do a rewind, then the field shows only the relative position from where you started and the **Now At** field shows ——.
- **Now At** Shows at what archive record the drive is positioned at. When it matches the **Go To** field then the search is complete. Before a rewind, it always shows —— since it is uncertain where the tape is positioned after startup.

When you reach the archive record number that you want, click the **Make Inventory** button to see what the archive contains. Check the archive record date and time at the beginning of the Status display after you do the **Make Inventory**.

Note on tape drive Device Name: SIGMET recommends always using the device name corresponding to "no rewind on opening". This is required for proper operation of the tape positioning features (e.g., for Linux /dev/nst0).

D.4 Making System Backups for Linux Computers

Important: HP and SGI users should use the system backup/restore tools provided by these manufacturers. Optionally after you have restored using these procedures, you can then restore files from sigbru backups.

Important: The device file permissions for the tape drive must be properly set. As root type the following:

For PC Linux Systems	**# chmod 666 /dev/st0 or nst0**
For HP-UX Systems	**# chmod 666 /dev/rmt/0m**
For SGI IRIX Systems	**# chmod 666 /dev/rmt/tps1d3**

While **sigbru** is useful in backing–up individual files and directories, it is best used for making full system backups in case you have a disk failure. Note that these system backups can also be used to restore individual files and directories so we will use the system backup as an example.

What are the Steps in Making a System ping Backup

There are two steps in backing up your system:

- 1. Make a **sigbru** backup from "/", i.e., all of the files on your system.
- 2. Document the disk partition information using fdisk and df commands (described in Section D.5)

You need both of these to recover from the worst case- a disk crash.

When should I Backup

Certainly after you have completed and tested an IRIS upgrade or installed an IRIS patch you should make a backup. Also, if you have made configuration changes you may want to backup the system, or at least the /usr/sigmet/config directory. Routine backups should be made at least every other month (6 times per year), depending on the changes that are made on your system. Development systems may require daily backups.

What should go into a system backup?

The idea behind a system backup is that all of the program files for your computer should be restored. Therefore the system backup should be made of the entire UNIX tree starting with "/".

What should NOT go into a system backup?

Large data files should not be included. These simply take–up space and don't add anything. Also, NFS (Network File Sharing) directories should not be backed up since these are on other computers. The "df" command can be used to display NFS directories. The large data files that are part of IRIS that should be excluded are:

- /usr/iris_data/ingest
- /usr/iris_data/product
- /usr/iris_data/product_raw
- /usr/iris_data/ascope (if used)

Note that if you have clutter maps or special product or ingest files that are tagged with a "keep" bit, then you should archive these using the standard IRIS archive features. Ingest files will first have to be converted to RAW products. Alternatively you use the IRIS menus to delete all of the ingest and product files except for the ones that you want to save on the archive.

There are two other directories that should not be included in the backup, since they might cause your system to hang when they are restored. These are:

- /proc
- /mnt

On some system, excluding /mnt will automatically exclude a CDROM mounted at /mnt/cdrom. However, for some systems the CDROM may have be mounted at a point such as

- /cdrom

Check the df command to see if there is a CDROM on your system and make sure it is either unmounted or excluded from the backup. The example menu in Section D.3 shows the exclusion of these directories.

Note: Even though you exclude directories, when you do a restore, the directory entries will be restored but, the contents will be empty. This means that you do not have to recreate excluded directories when you do a restore. However, you will have to recreate any subdirectories beneath them.

The step–by–step backup procedure is provided below.

1. Archive any special IRIS data files or delete the other data files

Here you can convert ingest clutter maps to RAW products and then save them on tape or disk using the standard IRIS archive features. Also any other special products such as a RAIN1 clutter map or "kept" products can be archived as well. In this case you would exclude the four IRIS data directories from the archive.

The alternative is to use the IRIS menus to delete all of the ingest and product files that you do not need and then not exclude these directories, i.e., the special files would be included on the archive.

2. Stop IRIS before making the backup

Do a qiris and a qant on the machine to be backed–up. Check that no IRIS processes are running by doing a ps_iris and "kill <process ID> " any remaining processes. Note that you might have do a "kill –9 <process ID> " .

3. Ready the Archive

Put in fresh archive media. Make sure it is labelled with at least the text "Backup" and the hostname and date, and that a tape is not write–protected or, if you are archiving to a HDD, make sure that the directory exists and that there is enough space for the archive. You can estimate the size of the "/" backup by doing a "df –h" command for Linux or SGI systems or a "df –k" command for HP systems. Exclude any NFS directories or CDROM's from your size estimate.

4. Run sigbru on the machine that is to be backed–up

- Start **sigbru** on the machine where the backup is to be made (of course you can do this over the network as described in Section D.2). ***REMEMBER, YOU MUST BE ROOT TO RUN SIGBRU.***
- Select the Backup <hostname> option.
- Select your archive host and device name- typically a tape drive either on the local host or on the network. For best speed, use the local tape drive if one exists.
- Select compression off unless the you are doing a networked backup or the backup would exceed your backup medium size..
- Click the Up Arrow on "Path" to select "/" . Highlight "/" in the contents and click the right arrow to "Include in Backup".
- Exclude NFS directories:
  Scroll down the contents list and find any "nfs" directories (check the "df" list). Note that NFS directories may not always be preceded by the text "nfs". SIGMET does not generally use NFS in its applications. Highlight them and click the right arrow to "Exclude from Backup". It is not recommended to exclude anything in /usr/sigmet.
- Optional: Exclude the large data directories in /usr/iris_data/, i.e., the subdirectories ingest, product, product_raw and ascope. Refer to discussion in Step 1.
- Exclude /proc, /mnt
- Exclude the CDROM (if necessary):
  Use df to identify any CDROM and its associated mount point (e.g., /cdrom).

Note that if the CDROM is mounted at /mnt/cdrom. then excluding /mnt is sufficient.

- Click “Start” to start the backup.

The cursor will change to an hour glass shape and the only button that can be accessed is the “Cancel” button (in case you change your mind). Note that if you cancel, you do not have a valid backup and will need to start over. You can re-use the same tape and it will rewind automatically when you restart the backup again.

The status log will pop–up automatically and show the files that are being saved. At the end of backup, the log will say “Backup Complete” and the cursor will change back to a pointer shape.

5. Write Protect, Label and Store your Backups

When you remove the tape, slide the write protection tab on the tape to be in the protect position to avoid accidental over-writing. You should save several previous version (e.g., the last 6 months) of backups. This provides some assurance that if a backup fails for some reason and you need to restore, you will have more than one. Old tapes can then be re-labelled and used for new backups.

It is a very good idea to store the backups in a location different than the computer. Then if the entire building is destroyed, hopefully the backup will be safe. At SIGMET, our backups are stored in a safety deposit box in a bank vault as well as on site.

D.5 Documenting Your Linux Disk Partitions

Important: It is critical that you make a hardcopy print–out of your disk partition information using the fdisk and df commands. Without this, recovery of your disk in the event of failure will be more difficult.

An important part of your backup documentation is to document your disk partitions. This does not have to be done every time that you do a system backup, but it should be done when your system is installed or when you change the disk partitions. The results must be recorded ***on hardcopy.***

D.5.1 Running df

First run the "df –h" command (as root). This will show the disks that are mounted and any CDROM or NFS directories. The –h option displays the disk usage in bytes ((G for giga and M for mega):

```
# df –h
```

```
[root@cyclone operator]# df –h
Filesystem            Size  Used Avail Use% Mounted on
/dev/hda2             3.1G  1.8G  1.1G  61% /
/dev/hda3             577M  490M   57M  89% /usr/iris_data
/dev/hdb1              13G  9.6G  3.3G  75% /usr/images
/dev/hdb2             4.4G  889M  3.5G  20% /mnt/hdb2
haze–gw:/usr/sigmet   2.4G  1.7G  492M  78% /nfs/haze/usr/sigmet
```

In the example above there are two hard disks, /dev/hda and /dev/hdb. There is also an NFS directory mounted at /nfs/haze/usr/sigmet which points to the /usr/sigmet directory on a different computer called "haze–gw". The "used" column tells you how many bytes (G for giga and M for mega) are actually used. This information is useful in determining if your backup will fit on a tape. For example, if our backup is to be of "/" (/dev/hda2), then this will require us to store 1.8 GB on tape which is easily done for most tape backups.

You can store the results of df to a file by using a standard editor such as vi and pasting the output into the file. Alternatively you can automatically put the results in a file by typing:

```
# df –h > /root/filename.lis
```

The resulting file will be stored in the root's home directory. The file can viewed by typing the command

```
# cat /root/filename.lis
```

The file can be printed using the command:

```
# lpr /root/filename.lis
```

If there is no printer on your system, the file can be sent to another computer (using rcp) for printing. As a last resort, record the results by hand. Store your hardcopy in a safe place.

D.5.2 Running fdisk

The fdisk command displays the partition table information that will be used during the restore. Sizes are shown in 512-byte blocks. Convert to MBytes by dividing the block count by 2000. An example is shown below for a system that has two disks (hda and hdb). The example command is issued for /dev/hda. Your system will probably not look like the example.

```
# fdisk –l /dev/hda
```

```
[root@cyclone operator]# fdisk –l /dev/hda

Disk /dev/hda: 255 heads, 63 sectors, 524 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1         6     48163+   6  FAT16
/dev/hda2             7       424   3357585   83  Linux
/dev/hda3           425       500    610470   83  Linux
/dev/hda4           501       524    192780   82  Linux swap
```

To store the results in a file named /root/fdisk.lis, use the command

```
# fdisk –l > /root/fdisk.lis
```

Use “cat” to view the file and “lpr” to print it as described in the previous section for the df command.

An example of a df listing is shown below. This is useful in showing you how much disk space is used to assist in sizing the required backup medium (i.e., will your backup fit on the tape). The –h option displays disk usage in gigabytes and megabytes which is a more convenient.

You should make a hardcopy of the fdisk and df information The easiest way to do this is to grab the text into a file and then print the file. If you do not have a printer you could send it by email to somebody who does or, as a last resort, copy it by hand.

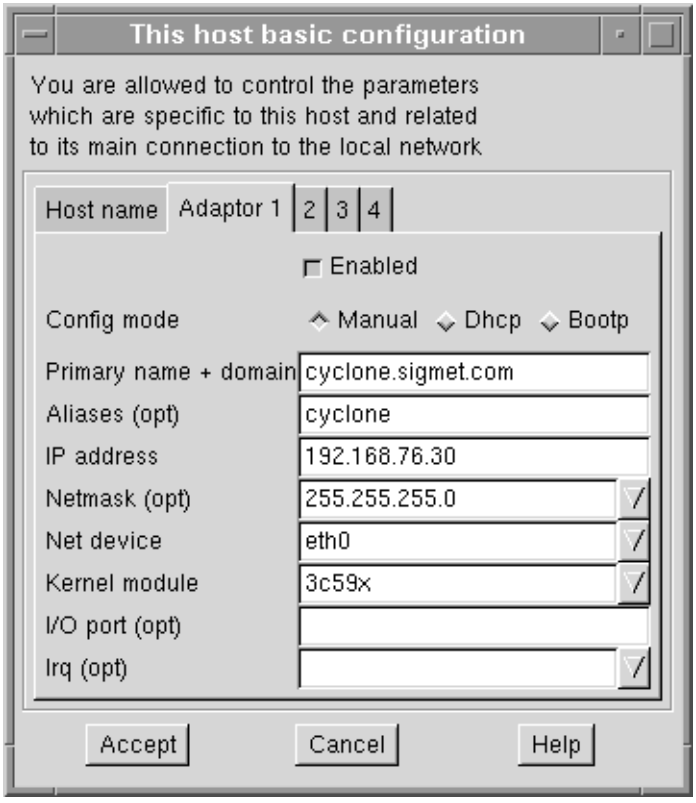
D.6 Documenting Your Basic Network Configuration

Important: You may need basic networking support to restore your system. For example, you may need to use a tape drive on another system to restore files over the network. It is critical that you make a hardcopy of your basic network configuration information so that you can get your network back up after a disk failure. Without this, recovery of your disk will be more difficult.

This is done on Linux systems by running the netconf utility. As root type the command:

```
# netconf
```

When the netconf utility screen appears (in either text or X–Window mode), select "Basic Host Information" for your network adapter (usually adapter 1). Record ALL of the information by hand (including the button positions) or use "xv" to make a print-out of the menu. An example is shown below. The information for your system will be be different. After you have copied the information, exit netconf without activating any changes.



Note: the kernel module for RxNet7 customers is smc9194

D.7 Selected File Restore Functions

Disk files can become corrupted accidently by users or by malfunction of the disk. The worst case is when the entire disk fails and a new disk must be installed. In all these cases, having a backup archive makes it much easier to resume operation since the software does not have to be installed from scratch and then completely reconfigured.

There are two basic restore scenarios:

- Restoration of selected files and/or directories.
- Full disk recovery

This section discusses the recovery of selected files or directories. It is assumed that your computer is basically working and that the disk does not have to be repartitioned and the operating system re–installed. Section D.8 discusses the Linux recovery procedures in the event that the disk fails and you have to reinstall the operating system.

In this example, we shall assume that the **/usr/sigmet/config** directory is to be restored. Perhaps some of the IRIS configuration files (e.g., TASK Configuration, overlays, etc.) have been inadvertently changed by an inexperienced operator. Restoring the entire directory can sometimes be easier than trying to figure out what the operator has done.

1. Locate backup

Identify the backup (tape or disk file) that you want to use. Here is where proper labelling and storage are important. It is also useful to keep a log of the backups identifying any system changes that are in the backup.

If you are using a tape, make sure that the tape is write–protected and insert the tape into the drive. Note that the drive can be on another system.

2. Stop all IRIS processes

Since you will be restoring the /usr/sigmet/config directory which has IRIS configuration files, you need to stop ALL IRIS processes. Do a qiris and qant to do this. Check by doing a ps_iris and manually do a “kill <process ID>”. You may have to be root to kill some processes or do a “kill –9 <process ID>”. Recheck with ps_iris that all processes have indeed been stopped (nothing reported back).

3. Start sigbru as described in Section D.2

Start **sigbru** from a terminal by typing “sigbru” as root, or start it from the CDROM is described in Section D.2. Note that if you do a ps_iris now, you will see sigbru, but this is OK. ***REMEMBER, YOU MUST BE ROOT TO RUN SIGBRU.***

4. Select Restore in the sigbru menu

5. Select Archive Host, Archive Device Name and Make Inventory

During the"Make Inventory" process, the status window will show a growing list of files that are on the archive. After the inventory is completed, you will see the top level directory appear in the "Contents". The message "Retrieving Dirs and Files Complete" will be displayed in the status log.

6. Select the Retrieve Path (default is /)

Here you almost always want to use "/". One reason not to would be if you wanted to restore some files and then selectively copy them to another directory. For example you might want to restore an old version of /usr/sigmet/config to another directory (e.g., /tmp/usr/sigmet/config) so that you could later select some config files to copy into the /usr/sigmet/config directory.

7. Select the files or directory to restore

In this example (restoring /usr/sigmet/config) you would select directories until "/usr/sigmet" appears in the Path and then highlight "config/". Click the right arrow to "Include in Restore".

8. Click the "Start" button

The cursor will change to an hour glass shape and the status log will pop-up.. You can monitor the progress of the restore in the status log which will show the files that have been restored. Any errors during the restore will pop–up in a separate error window. When completed the status log will show "Restore Complete".

The Cancel button can be used to stop the restore before completion.

D.8 Linux Disk Restore Functions

D.8.1 Disk Restore Overview

Eventually it will happen- your hard disk will fail. Repairing the hardware failure is as easy as putting in a new disk, but now you must format the disk and reload and configure all of the software. Installing and configuring the operating system and IRIS can be very time consuming unless you have prepared in advance with a proper backup strategy.

Here are the things that you need in order to restore the system using sigbru:

- Your Red Hat distribution CDROM. The version must match the version that was backed–up by sigbru.
- A hardcopy of the disk partition information that you collected (see Section D.5)
- The IRIS release cdrom (this has **sigbru** on it).
- Your **sigbru** system backup (from "/") of the failed disk.

Here we will assume the worst case– that your disk has crashed and you need a new one. You will need to get a disk that is the same size or larger than the one that failed.

The basic steps are as follows:

- **Step 1:** Use the Red Hat Linux CDROM to get a basic version of the operating system installed on a "mini-root" partition, i.e., a small disk partition to hold a temporary version of the operating system.
- **Step 2:** Create a large "main" partition and restore the files from your **sigbru** backup of "/" to the main partition.
- **Step 3**: Swap the system to boot from the main partition. The mini-root partition will be kept for possible future use.

Step 1 gets your system up and running on the mini-root partition. Step 2 restores to the main partition all of the files on the backup. This recovers any customization that you performed for IRIS and Linux. Step 3 gets the system booting off the large partition.

Important: The mini-root partition may already be installed on your disk. You can check this by typing "miniroot" (the standard SIGMET name) at the LILO boot prompt and see if your system boots. If you do not get a LILO boot prompt or the system does not boot then you must install the mini-root. If the system boots the miniroot, then skip to Step 2 in Section D.8.3.

D.8.2 Step 1: Basic Linux Installation into a Mini-Root Partition

What you need

You will need to have your Red Hat LINUX CD and the the hardcopy listing that you made of the basic system configuration, i.e.,

- The file system information from df: **see Section D.5.1**
- The disk partition information from fdisk: **see Section D.5.2**
- The basic network configuration information from netconf: **see Section D.6**
- Basic knowledge of the use of the vi editor.

For this procedure, you will need to be able to use the "vi" editor. If you need some help with this, you can refer to the *UNIX for Dummies* book that SIGMET provides with its systems, you can type "man vi" and read the on–line documentation or, better yet, find someone who knows how to use vi and ask that person to help you.

Installing Linux

The restore procedure described here assumes that your computer can boot from CDROM. Turn on the computer and immediately insert the Linux release CDROM into the drive so that the computer boots off the CD. At this point, we shall install the basic Linux operation system.

Important: See the *IRIS Installation Manual, Appendix A* for detailed instructions on installing Linux from CDROM. The installation procedure described there contains important information.

Follow the usual steps, selecting "Text" style installation menus and a "Custom" installation. The exceptions to the installation procedure for installing the mini-root partition are:

- For the disk partitioning step, delete all partitions and then create a 200 MB Linux partition with the mount point "/". This will be the "mini-root" partition. Also, create a 128 MB Linux swap partition (or use the size of the swap partition that is documented in your fdisk hardcopy, see **Section D.5.2**).
- When you are prompted for what packages to install, select only "Networked Workstation". This allows you to set up networking on your system in case you need to use the network for the restore operation. When prompted for the network node name, use the node name that is documented in your hardcopy of the basic network information (see **Section D.6**).

The installation will not take very long since there is not much to install. When it is done, the computer will reboot automatically. Be sure to remove the CDROM before the reboot so that the system will boot from the mini-root partition.

After reboot from the mini-root partition, the only post-installation step that you need do is install the "k–shell" which is used by the **sigbru** restore utility. Follow the *IRIS Installation Manual* steps for installing "pdksh..." using the the RPM post installation step.

If you are on an RxNet7 and plan to use a local tape drive for the restore, you can do a shutdown (# shutdown –h now) and connect the SCSI tape drive by daisy-chaining it on the CDROM. Be sure that the tape and CDROM both have unique SCSI addresses set via switches on the back. Also be sure that the last SCSI device on the chain has a terminator or else the system will not work reliably (or perhaps at all).

Setting-Up Basic Networking (Optional)

Note that if you have a local CDROM and a local tape for the restore, then you do not need to do this step. If you need to do this, run netconf as root and provide the basic host information from your hardcopy documentation (see **Section D.6)**. Reboot the system. You will need to edit two files to allow you to access the other machines on the network that you might have to use (e.g., the one with the tape drive or the copy of sigbru).

`/etc/hosts`

This file has the list of node names and IP addresses. Be sure to check that your "alias" is specified- the short version of your node name without the domain name. After you have done this, test with the "ping" command, i.e. type:

`# ping` *`nodename`*

where *nodename* is the name of the computer(s) that you configured in /etc/hosts.

The next step is to configure the file:

`/etc/hosts.equiv`

This file is a list of computers who are authorized to use your system.

You can look on other systems on the network to see examples of these. Remember, you probably only need to have entries for one or two other systems.

You must also make sure that the corresponding /etc/hosts and /etc/hosts.equiv files on the remote computer that you will access include the computer that you will need to restore. For example, if you are going to use a tape drive, cdrom or copy files from a node called "cyclone", make sure that the /etc/hosts and /etc/hosts.equiv files on cyclone contain entries for the target local computer that will be restored. Since you are restoring to a computer that was originally on the network, there is a good chance that these files are already configured on the other network computers.

D.8.3 Step 2: Restore the sigbru backup to main partition

Creating and mounting the main partition

First we need to create the main partition. This only need be done once. In the mini-root run fdisk as root:

```
# fdisk /dev/hda
```

In fdisk issue the "p" command to view the partitions. You should see three partitions as shown in the example below:

```
Disk /dev/hda: 66 heads, 63 sectors, 1018 cylinders
Units = cylinders of 4158 * 512 bytes

   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1        99    205789+  83  Linux
/dev/hda2           100      1018   1910601    5  Extended
/dev/hda5           100       163    133024+  82  Linux swap
```

In the example, /dev/hda1 is the "mini" partition. /dev/hda2 is an extended partition. This is available to add logical partitions, or can be removed and this space repartitioned. At this point, you should use the features of fdisk to repartition the disk according to the hardcopy documentation that you obtained during your backup procedure (see **Section D.5.2). However, you must not remove or change the mini-root partition or your system will not be bootable.**

Note: If your main partition had already been created (e.g., you are restoring a system that was already configured with main and mini-root partitions) then you can skip the next creation step and start the procedure with the "mkfs" (make file system) step below.

A simple thing to do that will serve most systems is to simply create a logical partition within the extended partition. Do this with the "n" command. Make the partition start after the swap area and use the entire disk, e.g., start at 164 and end at 1018. Do not specify a mount point. When you are done, use the "w" command to write the partition information. The "p" command will then show (for example):

```
   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1        99    205789+  83  Linux
/dev/hda2           100      1018   1910601    5  Extended
/dev/hda5           100       163    133024+  82  Linux swap
/dev/hda6           164      1018   1777513+  83  Linux
```

The "main" partition is /dev/hda6.

.At this point you must reboot the system by issuing the command:

```
# reboot
```

After the reboot you must make the file system on the main partition (/dev/hda*N* would be /dev/hda6 in the example):

```
# mkfs /dev/hdaN
```

Finally create the mount point and mount the file system:

```
# mkdir /mnt/hdaN
# mount /dev/hdaN /mnt/hdaN
```

Restoring the sigbru backup to the main partition

For this step you will need two things:

- A copy of **sigbru** programs. These can be found on the IRIS release CDROM. For Linux systems, these are in /mnt/cdrom/linux/sigbru. The CDROM can be local or, if you do not have a CDROM on the local system, you can copy the files from another IRIS computer on the network.
- Your **sigbru** system backup tape which will be placed on either a local or remote tape drive.

There are several restore scenarios corresponding to the different combinations of the above (i.e., local/networked IRIS release CDROM or local/networked **sigbru** backup tape). The easiest and fastest case is that of restoring from a local CDROM and a local tape drive.

In the case of using either a networked CDROM or tape drive, you will have to set up some minimal networking on the target machine that is being restored. Networking needs to be installed as part of Step 1 (the Linux installation).

All of the procedures below assume that the **sigbru** backup is the full disk image (from "/" with /proc and /mnt excluded). We will restore the backup to the main partition mounted at /mnt/hdaN where N is the number of the partition (viewed via df).

If the main partition is not mounted then mount it with (see previous step):

```
# mount /dev/hdaN /mnt/hdaN
```

Local Tape and Local CDROM

Insert the system backup tape (write-protected) into the local tape drive and the SIGMET IRIS release CDROM into the local CDROM drive (this has the **sigbru** command utility on it). Next perform the following steps as root:

```
# cd /mnt/hdaN     (hdaN is the number of the main partition from df)
# /mnt/cdrom/linux/sigbru/sigbrush –extract –device /dev/st0
```

Here the mount point of the CDROM is assumed to be /mnt/cdrom.

Remote Tape and Local CDROM

Insert the system backup tape (write–protected) into the remote tape drive and the SIGMET IRIS release CDROM into the local CDROM drive (this has the **sigbru** command utility on it). The command is similar except that the network nodename of the workstation with the tape drive is specified.

```
# cd /mnt/hdaN     (hdaN is the number of the main partition from df)
# /mnt/cdrom/linux/sigbru/sigbrush –extract –device /dev/st0 –node nodename
```

Local or Remote Tape and Remote CDROM

Getting the sigbru *Program Over the Network*

If you have a remote CDROM, then you will need to copy the **sigbru** files over the network to a directory on your local machine. The example procedure assumes that the "rcp" command is working (remote copy). ftp could also be used. The procedure for doing this as follows.

First create a special **sigbru** directory (/root/sigbru) on the local machine to hold the **sigbru** files. On the local computer that is to be restored type (as root):

```
# cd /root
# mkdir sigbru
```

The next step is to copy the **sigbru** files to the directory /root/sigbru. The minimal required files are called sigbrush and gnutar and are stored on the CDROM under the directory for your platform (e.g., /cdrom/linux/sigbru). They can also be found in the /usr/sigmet/bin directory of an IRIS system on the network. You can obtain them from either place.

Getting sigbru *from a remote CDROM*

Insert the IRIS CDROM into the remote machine and mount it (see the *IRIS Installation Manual* Section 1.2 on "Mount the CD"). On Linux systems this is usually accomplished by typing (as root on the remote computer with CDROM):

```
# mount /dev/cdrom
```

Next, copy the two files from the remote CDROM to the local computer's /root/**sigbru** directory. On the local computer (where you want to store the files in /root/sigbru) type:

```
# rcp nodename:/cdrom/linux/sigbru/sigbrush /root/sigbru
# rcp nodename:/cdrom/linux/sigbru/gnutar /root/sigbru
```

Nodename is the name of the computer with the CDROM. The local computer now has the **sigbru** files stored in a directory called /root/sigbru.

Getting sigbru *from a remote computer with IRIS installed*

Identify the remote computer with the installed IRIS system. The two **sigbru** files that you need to copy are stored in the /usr/sigmet/bin directory. Copy them to the /root/**sigbru** directory on the local computer by typing (on the local computer):

```
# rcp nodename:/usr/sigmet/bin/sigbrush /root/sigbru
# rcp nodename:/usr/sigmet/bin/gnutar /root/sigbru
```

Nodename is the name of the remote computer with IRIS installed. The local computer now has the **sigbru** files stored in a directory called /root/sigbru.

Now follow the steps below to do the restore from tape to the main partition.

- **For a local tape drive type:**

```
# cd /mnt/hdaN
# /root/sigbru/sigbrush –extract –device /dev/st0
```

- **For a remote tape drive type:**

```
# cd /mnt/hdaN
# /root/sigbru/sigbrush –extract –device /dev/st0 –node nodename
```

Nodename is the name of the remote network computer with the tape drive. /dev/hda*N* is the device name of the main partition from df.

We are now ready to configure Linux to boot from either the main partition or the mini-root partition.

D.8.4 Step 3: Configuring to boot from the main or mini partitions

We have restored the backup to the main partition. Now we must configure the system to boot from the main partition. For possible future use, we will keep the mini-root partition since we might need it in the future restore operations.

First, if it is not already booted, boot your computer. At this point it will be booted in the mini-root partition since we have not activated the main partition. Also, if it is not mounted, mount the main partition with:

```
# mount /dev/hdaN /mnt/hdaN
```

Modify /mnt/hda*N*/etc/fstab on the main partition

For this step, you need to know what your disk partitions are. These were just configured, but to refresh your memory you can use fdisk, i.e. type,

```
# fdisk –l /dev/hda
```

The disk partition information should look something like:

```
   Device Boot    Start       End    Blocks   Id  System
/dev/hda1   *         1        99    205789+  83  Linux
/dev/hda2           100      1018   1910601    5  Extended
/dev/hda5           100       163    133024+  82  Linux swap
/dev/hda6           164      1018   1777513+  83  Linux
```

In this example, hda1 is the mini-root partition, hda2 is an extended partition that contains two logical partitions, i.e., the swap space in hda5 and the main partition in hda6. This is the information that we need to edit fstab in the main partition.

For this step you will need to be able to use the “vi” editor. Start the editor on the fstab file by typing:

```
# vi /mnt/hda6/etc/fstab
```

The file should look something like (”/” is incorrectly pointing to the min-root):

```
/dev/hda1     /              ext2    defaults         1 1
/dev/cdrom    /mnt/cdrom     iso9660 noauto,owner,ro  0 0
/dev/hda5     swap           swap    defaults         0 0
/dev/fd0      /mnt/floppy    ext2    noauto,owner     0 0
none          /proc          proc    defaults         0 0
none          /dev/pts       devpts  gid=5,mode=620   0 0
```

“/” is currently pointing to /dev/hda1. Change this to /dev/hdaN where N corresponds to your main partition. Also check that the swap partition is pointing to the correct disk partition. In the example, the swap partition from fdisk is /dev/hda5 so the fstab entry for swap is OK. However for this example, the entry for /dev/hda1 pointing to “/” needs to be changed to /dev/hda6. After you have finished editing, save your results and proceed to the next step.

Modify /etc/lilo.conf file on the mini-root partition and run lilo

The file /etc/lilo.conf on the mini-root partition will be used to configure the LILO boot loader. We use the mini-root version since we do not yet fully trust our restore and the mini-root is fully tested.

First, we need to document the lilo.conf file on the main partition. Do this by typing the command:

```
# cat /mnt/hda6/etc/lilo.conf
```

This will show the lilo.conf that was used on your old disk. You want to record by hand the lines corresponding to the "image" that has the "label=linux". The lines that you want to record will look something like:

```
image=/boot/vmlinuz-2.2.12-20
        label=linux
        initrd=/boot/initrd-2.2.12-20.img
        read-only
        root=/dev/hda1
        vga=773
        append="mem=128M"
```

No changes will be made to this file. Now we use this information to edit /etc/lilo.conf in the mini-root. As in the previous step, you will need to be able to use the vi editor to do this.

```
# vi /etc/lilo.conf
```

The mini-root lilo.conf file will look something like:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux

image=/boot/vmlinuz-2.2.12-20
        label=linux
        initrd=/boot/initrd-2.2.12-20.img
        read-only
        root=/dev/hda1
```

You only need to make one change to these lines, i.e.,

- Change the "label=linux" to "label=miniroot"

Now after these lines, type-in the lines that you recorded and make the following changes to them (hdaN refers to the main partition):

- Change "image=/boot..." to "image=/mnt/hdaN/boot..."
- Change (if necessary) "label=linux"

- Change “initrd=/boot...” to “initrd=/mnt/hdaN/boot...”
- Change “root=/dev/hda1” to “root=/dev/hdaN”

All other lines should stay the same since. When you are done, the /etc/lilo.conf file (mini-root) should look something like:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux

image=/boot/vmlinuz-2.2.12-20
        label=miniroot
        initrd=/boot/initrd-2.2.12-20.img
        read-only
        root=/dev/hda1

image=/mnt/hda6/boot/vmlinuz-2.2.12-20
        label=linux
        initrd=/mnt/hda6/boot/initrd-2.2.12-20.img
        read-only
        root=/dev/hda6
        vga=773
        append=”mem=128M”
```

Carefully check your entries and save the file. Now run lilo by typing:

```
# lilo -v
```

Carefully check that lilo runs without errors. Errors are most likely due to typo’s and should be repaired by re-editing the /etc/lilo.conf file. Re-run lilo until it is error free.

At this point we have two bootable partitions that can be selected at boot time at the “LILO:” prompt. The main partition can be booted by:

- Typing “linux”.
- Simply hitting the ENTER key.
- Waiting for the timeout (50 seconds in the example).

The mini-root partition can be booted by typing “miniroot” at the LILO prompt. Perform the following tests to verify that you can boot either partition:

- Issue the “reboot” command and type “miniroot <Enter>” at the LILO prompt to verify the mini-root boots OK.
- Issue the “reboot” command and type “linux <enter>” at the LILO prompt to verify that the main partition boots OK.

Proceed to the next step.

Modify the /etc/lilo.conf file on the main partition and rerun lilo

Note: If you restored a backup of the main partition for a system that was already configured for the min-root, then your /mnt/hda6/etc/lilo.conf file may not require any modification. Check it by going through the procedure below.

We need to configure the /etc/lilo.conf file on the main partition so that if lilo is run here, it will properly install the boot record for both the main and the mini–root partitions. To do this, first reboot the system in the main partition and at the LILO prompt type linux, i.e.,

```
LILO boot: linux
```

After reboot, manually mount the mini-root partition, i.e. (with N corresponding to the mini-root partition),

```
# mount /dev/hdaN /mnt/hdaN
```

If this does not work, you may have to first create the mount point, i.e.,

```
# mkdir /mnt/hdaN
```

Now look at the lilo.conf file in the mini-root by typing:

```
# cat /mnt/hdaN/etc/lilo.conf
```

This will look something like the example on the previous page. Copy by hand all the lines corresponding to the mini-root label, i.e., following the example:

```
image=/boot/vmlinuz–2.2.12–20
        label=miniroot
        initrd=/boot/initrd–2.2.12–20.img
        read–only
        root=/dev/hda1
```

Use vi to edit the /etc/lilo.conf file (main partition). Add the lines that you copied for the miniroot, with the following changes (here N refers to the mini–root, 1 in the example):

- Change “image=/boot...” to “image=/mnt/hda*N*/boot...”
- Change “initrd=/boot...” to “initrd=/mnt/hda*N*/boot...”
- The miniroot “root” line should already point to the proper partition for the mini-root.
- Change (if necessary) the linux “root” line to point to the main partition.

When you are done the /etc/lilo.conf file (main partition) should look something like:

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
timeout=50
default=linux

image=/boot/vmlinuz-2.2.12-20
        label=linux
        initrd=/boot/initrd-2.2.12-20.img
        read-only
        root=/dev/hda6
        vga=773
        append="mem=128M"

image=/mnt/hda1/boot/vmlinuz-2.2.12-20
        label=miniroot
        initrd=/mnt/hda1/boot/initrd-2.2.12-20.img
        read-only
        root=/dev/hda1
```

In this example, hda1 is the miniroot partition and hda6 is the main "linux" partition. Save your results and then run lilo:

```
# lilo -v
```

Carefully check that lilo runs without errors. Errors are most likely due to typo's and should be repaired by re-editing the /etc/lilo.conf file. Re-run lilo until it is error free.

Reboot both the "miniroot" and main "linux" partitions to test them, then proceed to the next (final) step.

D.9 Test IRIS and Backup Your Restored System

At this point you should test IRIS in its full operational mode. All should be as it was before. Note that if you separately archived ingest clutter maps (as RAW products), RAIN1 clutter maps or special “kept” data files, you should restore them now.

After IRIS has been tested, you should then do a backup of your system. Only the main partition need be backed–up. Follow the procedure in Section D.4 and subsequent sections to record the df, fdisk and netconf information as part of your backup. You should of course maintain this backup over the years.

D.10 Disk Crash After Mini-Root is Installed

When your next disk failure occurs (hopefully 10 years from now), if the disk hardware is OK and the miniroot is still intact, you will be able to restore your disk more easily- just boot the mini–root and start your restore procedure at Step 2 (Section D.8.3). Since you faithfully carried-out the backup prescribed in preceding section, your backup will be exactly the main partition that you want to restore.

Important: If you use an existing miniroot to restore your main partition from a backup tape, be sure to go through all of the steps of running lilo in both the mini–root and then the main partition. Failure to do so might cause your system to become un-bootable, in which case you would have to re-install the mini-root. Do not skip steps in the procedure, although you will have less work to do since all the files in the main partition should be configured properly already.

D.11 sigbru –auto: Auto Archive Features

A powerful feature of **sigbru** is the auto archive feature. This allows **sigbru** to monitor a disk directory so that when a specified "quota" of files is placed there, **sigbru** automatically:

- Archives the files to tape (or perhaps disk).
- Optionally deletes the files in the directory that is being monitored.

Note that if the optional delete is enabled, then **sigbru** will continue to monitor the directory and write sequential archives to the tape. Otherwise, the auto archive is to disabled, i.e., it is "write–once".

An ideal application of this is as follows:

- Use the product output menu to send files in .gif format to a directory.
- Use **sigbru** –auto to monitor the directory, and when a selected size is reached, archive the files to tape.
- **sigbru** can then delete the disk files so that the disk does not fill–up.

To be on the safe side it is always best for IRIS to send disk files to a directory that is on a separate disk partition, i.e., not / or /usr, since filling the disk space in these will cause the system to fail. The /usr/iris_data directory is also not a good choice since filling this will cause IRIS to fail. We recommend that if you do this, you create a separate disk partition for the directory.

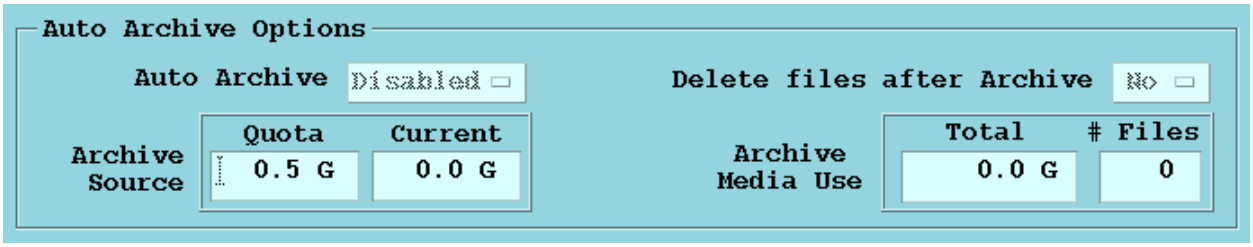
Note: Only tape drives support multiple automatic archive files. The tape must use a non–rewind driver (e.g., /dev/nst0 for linux).

WARNING: If you inadvertently specify a tape device driver that performs rewind on opening, then attempt to do multiple automatic archives to tape, then the you will always overwrite the last archive, i.e., when you are done only the most recent archive will be on the tape.

To invoke the auto archive features of sigbru, start by typing as root:

```
# sigbru -auto
```

At the bottom of the **sigbru** menu (in "Backup" mode), you will see the following:



The menu features are described below:

Auto Archive Enable/Disable

This defaults to disabled when you start **sigbru** unless you specify **`-enable`** at startup. This enables the automatic polling of the "Included" directories that are to be archived.

Archive Source: Quota and Current, sigbru polling

Type–in the quota in GB. When Auto Archive is enabled, **sigbru** polls the included directories and calculates the total size of all files that are more than one minute old. This is updated every 5 minutes and the result is displayed in the field called **Current**. When the **Current** size exceeds the **Quota** that you specified, **sigbru** performs an archive operation for all included files that are more than one minute old.

Note that the max size of a DVD archive quota is 4 GB — the size of a typical DVD.

Archive Media Use: Total and Record

These are display-only text fields are valid only during automatic archive operation. They are not valid during manual archive operation when the operator clicks the **Start** button.

- **Total** shows the total size of all files that have been written to the archive medium. The total is uncompressed. If you use compression, then the actual amount written will be less.
- **Record #** shows the number of archive records that have been written to the tape, i.e., whenever the **Quota** is exceeded.

Delete Files After Archive

WARNING: This is a potentially dangerous command, since you might end up deleting key system files if you are not careful to specify the Included files correctly.

The **Delete Files after Archive** feature is designed for automatic maintenance of a disk directory. See the example at the beginning of Section D.11

This defaults to **No** when **sigbru** is started unless –delete is specified at startup. It can only be set to **Yes** when auto archiving is enabled. The option is not available in the manual archive mode, i.e., when the user manually clicks the **Start** button.

Note: When you specify "Enable" sigbru waits for 10 seconds before polling the directory to give you a chance to set the Delete Files field.