

Chapter 22

Applications under Linux

This chapter introduces the user to various applications that are available under Linux, using both the CLI and GUI interface. Each application covers basic function and operation, but leaves additional research to the user to enhance it performance. Unfortunately, this chapter will never be complete.

Concepts Learned in this Chapter

- A review of a few of the many extra applications that are available to the Linux system

Table of Contents

Applications under Linux.....	1
22.1 The GIMP.....	3
22.1.1 Drop-Down Menus.....	3
22.1.1.1 File.....	3
22.1.1.2 Xtns.....	5
22.1.1.3 Help.....	5
22.1.2 Icon Tools.....	5
22.1.2.1 Selection Tools.....	5
22.1.2.2 Transform Tools.....	5
22.1.2.3 Paint Tools.....	5
22.1.3 Color Selection.....	5
22.2 Screen Capture.....	6
22.3 Data Backup.....	6
22.3.1 Do It Yourself.....	6
22.3.1.1 What to Back Up.....	6
22.3.1.2 Where to Back Up.....	8
22.3.1.3 Data Compression.....	8
22.3.1.4 Backup Process.....	9
22.3.2 Internal Backup Utility.....	12
22.3.3 Commercial Backup Utility.....	12
22.4 GAIM – Instant Messenger.....	12
22.5 Connecting to Cisco Console Interface.....	15
22.X Commands Used in this Chapter.....	15
22.X Chapter Review Questions.....	15

A multitude of different applications are available for use within Linux. The following is a review of just a few of the many applications.

22.1 The GIMP

The application GNU Image Manipulation Program (“The GIMP”) allows the user to create and manipulate images. It is often referred to as the “Photoshop”™ of Linux. Simply put, the power of the GIMP is immense – but the learning curve is steep. It is very powerful in the hands of the digital artist.

After starting the GIMP, the user is presented with the Toolbox window. This is the center of all further operations. Across the top are the three drop-down menus. Below that are a series of operational icon tools. Finally at the bottom are the color selection tools.

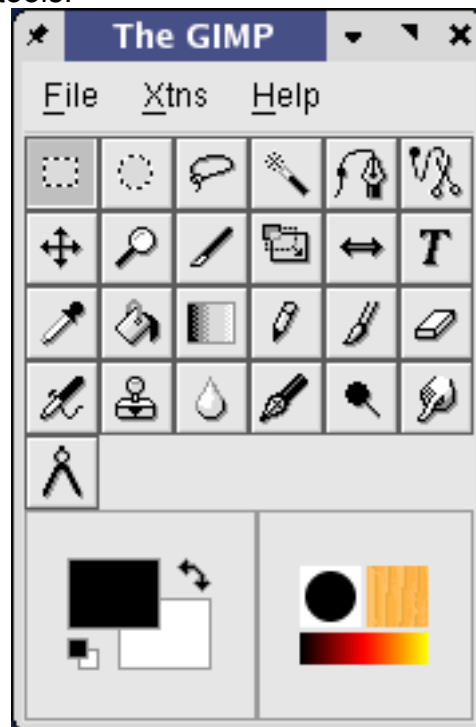


Figure 22.1: GIMP Opening Screen

22.1.1 Drop-Down Menus

The Drop-Down menus consist of three options – File, Xtns and Help.

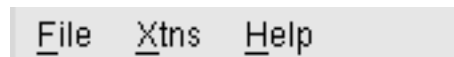


Figure 22.2: GIMP Drop-Down Menus

22.1.1.1 File

The File menu allows the user to either open a new or existing image. Either will open an Image window. Other properties of a file may also be configured. After the

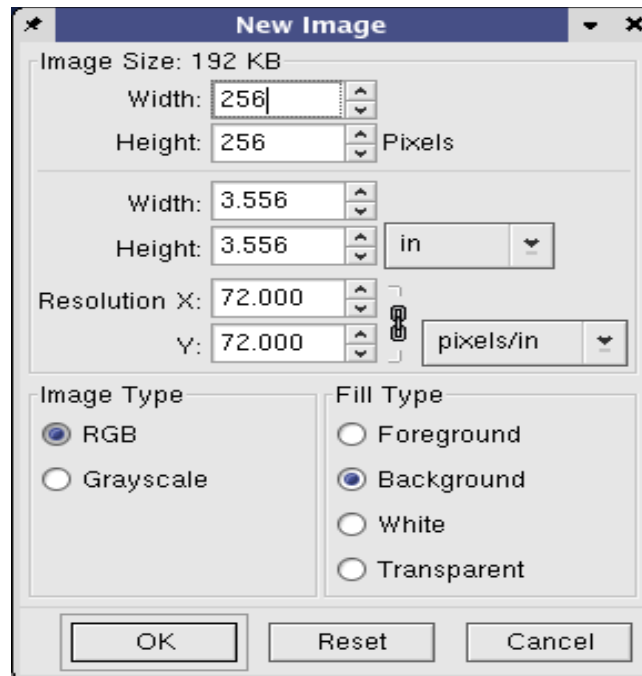


Figure 22.3: GIMP New Image

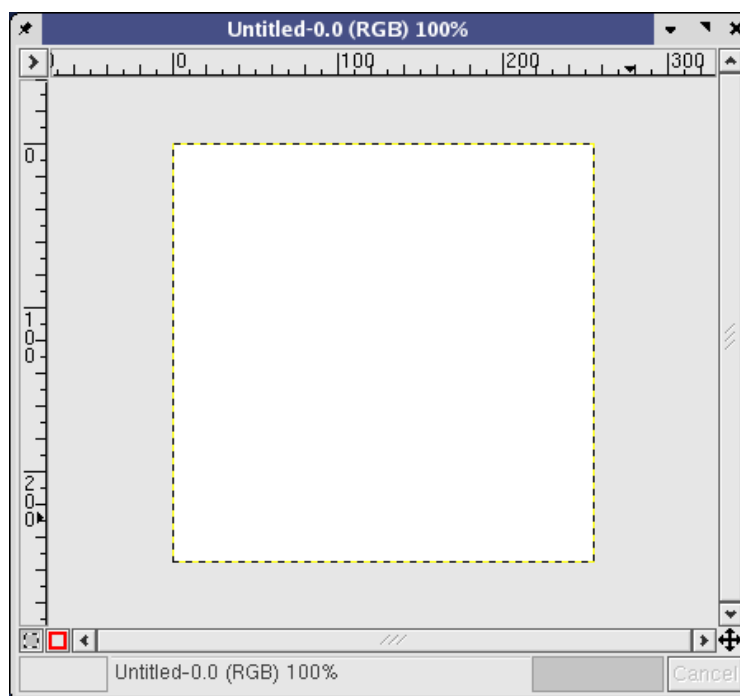


Figure 22.4: GIMP Image Screen

Image window is open, right clicking on within the window will open a menu, allowing one to perform the normal save and editing actions to the selected image. Prior to a new image window opening, a New Image size window is opened, allowing the user to specify the image size.

22.1.1.2 Xtns

The Extensions menu provides for various plug-ins or scripts that are in general not associated with an image.

22.1.1.3 Help

The Help menu provides various help tools to assist in using The GIMP.

22.1.2 Icon Tools

The tools are a set of icons that provide various drawing functions. The icons may be divided into three functional groups – Selection, Transform, and Painting.

22.1.2.1 Selection Tools

Each selection icon allows one to selectively chose a portion of an image that is to be modified. Various tools are available.

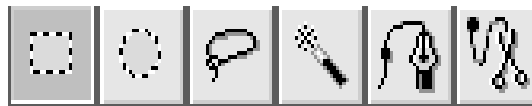


Figure 22.5: GIMP Selection Tools

22.1.2.2 Transform Tools

Each Transform icon allows the user to move, rotate, flip or modify the previously selected area.



Figure 22.6: GIMP Transform Tools

22.1.2.3 Paint Tools

Each Painting icon allows the user to modify an image or selected area by modifying the color, edges, lines, or text.



Figure 22.7: GIMP

Paint Tools

22.1.3 Color Selection

The Color selection tools allow the user to specify various colors that are to be used for either the foreground or background of an image, as drawn by the Paint tools.

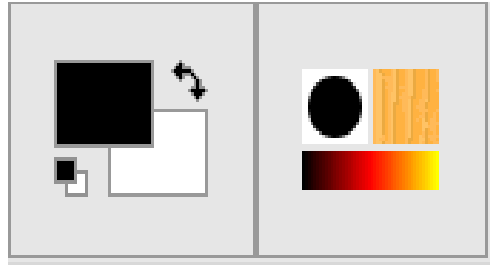


Figure 22.8: GIMP Color Selection

22.2 Screen Capture

The application Screen Capture is a utility for capturing either all or a designated window of the screen. A specified delay allows the user to move the mouse cursor over the desired window, which may be then either printed or saved to a file. The default saved file type is “png”, but other formats may be used. This is an excellent utility for capturing a screen window and then inserting it into one favorite word processor.

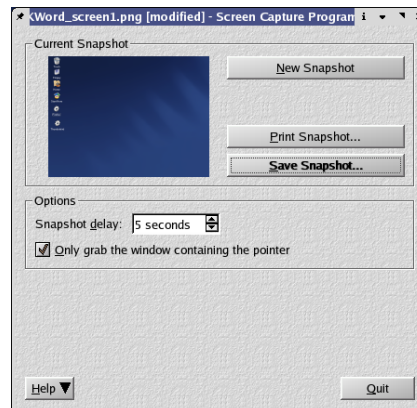


Figure 22.9: GIMP Screen Capture

22.3 Data Backup

In spite of what one should do, one of the least desirable tasks that should have the highest priorities is the backup of user data. An important question is what should be backed up?

22.3.1 Do It Yourself

The easiest way to approach a backup of information is to have it performed on an automatic basis. This is an excellent application for the **cron** utility that performs a task on a specified periodic basis.

22.3.1.1 What to Back Up

Before implementing **cron**, one needs to decide what information should be backed up. Assuming that the Operating System can be easily re-installed, there is no need to back it up or the multitude of applications that come with the installation. What is required is data that is subject to change. So again, what is subject to change? Instead of specifying what information changes, we should

really specify what directories have information that changes. The following are suggestions:

/home	This stores the personal information of each user.
/etc	This stores the configuration files for the system. Although they rarely change, an occasional backup should be made to insure the server configurations are maintained.
/var	This stores variable server data and logs. Very important that this data be backed up.
/usr	User installed programs are normally installed under the /usr directory. This needs to be backed up so the user installed programs do not need to be re-installed.
/opt	Although commonly empty, it may be used to store data and applications used by users.
/data	This is not a standard directory, but one that might be created to store user data, downloads, or other information. Information in this directory would typically be available to all users.

Note that during installation is many files and applications are pre-installed in several of the above noted directories – there is no need to back them up (but configurations should be backed up). The major question here is to know what directories need to be protected. This fact was observed during the development of this routine, where the initial backup of directories even when zipped was in excess of 500 Mbytes, and one that was in excess of 1.6 Gbytes. This issue requires additional investigation. The best way to decide which directories need to be investigated is to do a raw backup, then edit the list down. In our example this has been done and the following was learned.

```
[/backup]# ll
-rw-r--r-- 1 root root 798704419 Jul 3 01:35 data040703.tar.gz
-rw-r--r-- 1 root root 3139457 Jul 3 00:10 etc040703.tar.gz
-rw-r--r-- 1 root root 61815315 Jul 3 00:08 home040703.tar.gz
-rw-r--r-- 1 root root 188498576 Jul 3 01:23 opt040703.tar.gz
-rw-r--r-- 1 root root 1664996280 Jul 3 01:19 usr040703.tar.gz
-rw-r--r-- 1 root root 381292335 Jul 3 00:17 var040703.tar.gz
```

As an example, lets look at the **/var** directory. First do a listing of its contents:

```
# ls
account      ftp          log          nis          tmp
arpwatch     gdm          mail         opt          tomcat4
cache        kerberos    mailman      preserve     tux
crash        lib          mars_nwe    run          webmin
db           local       named       spool        www
empty        lock        net-snmp    state        yp
```

Now we need to decide which directories are variable and need to be backed up. This can be a task that depends upon each individual system. For this example, on review it is believed that the following **/var** directories should be backed up:

/var:

ftp	lib/pgsql	log	net-snmp	www
lib/mysql	lib/tripwire	mail	spool	
lib/news	e	named	webmin	

In like manner, we need to view the other directories. The following options were elected:

/etc:	3 Mbytes zipped – back it all up.
/home:	User variable data – back it all up.
/opt:	189 Mbytes variable data – back it all up.
/usr:	This is probably the most difficult directory to evaluate. There is an immense amount of installed data in this directory – but this is also where the user-installed programs are installed. After review, the following evaluation was found:
/usr/dic:	May initially be empty, but with word processing may contain a user dictionary.
/usr/doc:	User application documentation installed here – along with all other data. Hard to differentiate.
/usr/games:	Well, you are sure to install additional games.
/usr/java:	This is subject to change / improvement.
/usr/local:	This is where user installed programs are normally placed.
/usr/include:	This directory contains mostly system C libraries, but user installed libraries may also be installed here. Only the user can decide if this should be backed up.
/usr/lib:	This directory contains mostly system installed libraries, but user installed libraries may also be installed here. Only the user can decide if this should be backed up.
/usr/share:	This directory contains virtually all system installed applications that are available to all users, but user installed applications are also installed here. It is recommended that the administrator keep track of installed applications and only back up those specific directories. (This may be hard to do.) In this example I have chosen to back up two directories /usr/share/lyx and /usr/share/java

22.3.1.2 Where to Back Up

The next question needed is where to store the data? A recommendation for ease is to copy the backup files to a separate drive, that way if the primary drive should fail, the data is still there. It is fast and easy, and simple to restore if a problem should exist. If remote off-line storage is desired, a tape backup should be used. (A removable drive might also be used.) A tape may seem to be antiquated, but it is still an excellent backup media.

22.3.1.3 Data Compression

Almost there, next question. Should the data be copied as is or should it be compressed? In order to save space, it needs to be compressed. If one is not

concerned it may be copied as is. Several alternatives are available for compression, but when using the **tar** utility, compression may be included in the process.

22.3.1.4 Backup Process

In our example, the data will be compressed and stored on a separate drive. It is assumed that the partition that maintains the backup data has been built into our directory structure as **"/backup"**.

Our first requirement is to backup a given directory in a compressed format. The general command is:

```
tar czvf /backup/$filename ./dir/*
```

In this case, all files and subdirectories in the specified directory (**./dir**) are created (c), zipped (z), and written to the file (f) **/backup/\$filename**. For a starter to verify that everything is working as we want, as a test, we have the process operate in verbose (v) mode.

This copies the specified directory to the specified filename. If we wish to keep a log of past backups, it would be nice to give each file a unique name.

To make each file name unique, lets create a filename that uses the directory-name and the date, in the format of **yymmdd**, by using the command:
filename=dir\$(date +%m%y%d).tar.gz

Thus we would have a filename (from the above example), **home040702.tar.gz**. The backed up directory is home, the copy was made on 2 July 2004, it was tarred, and gzipped.

We now have the information to write a simple script to backup our data. Initially, to make it more interesting, we will only backup one directory at a time on different days.

less backup.sh This is our initial file – later to be modified

```
#!/bin/bash
# /usr/local/backup/backup.sh - backup shell of system data
# When calling the backup script, follow it with a
# variable of 1 through 6, which specifies which directory
# that is to be backed up.
#
#
case $1 in
1)
filename=home$(date +%y%m%d).tar.gz
cd /home
tar cvfz /backup/$filename ./*
;;
2)
filename=etc$(date +%y%m%d).tar.gz
cd /etc
tar cvfz /backup/$filename ./*
;;
```

```

3)
  filename=var$(date +%y%m%d).tar.gz
  cd /var
  tar cvfz /backup/$filename ./*
;;
4)
  filename=usr$(date +%y%m%d).tar.gz
  cd /usr
  tar cvfz /backup/$filename ./*
;;
5)
  filename=opt$(date +%y%m%d).tar.gz
  cd /opt
  tar cvfz /backup/$filename ./*
;;
6)
  filename=data$(date +%y%m%d).tar.gz
  cd /data
  tar cvfz /backup/$filename ./*
;;
esac

```

Save the above file to the **/usr/local/backup** directory (you will need to create the backup directory).

The above script copies everything in a specified directory to the designated file. As we noted above, this can lead to some very large backup files. We therefore need to modify some of the **tar** statements. If we add the feature of creating a tarred file from a file list, we have the following command:

```
tar cvfzT /backup/$filename /root/bkfst/dir-list
```

In this case, the directories to be backed up will be specified in a separate file, such as those for the **/var** directory, we need to create the file

```

/root/bkfst/var-list
/var/ftp/
/var/lib/mysql/
/var/lib/news/
/var/lib/pgsql/
/var/lib/tripwire/
/var/log/
/var/mail/
/var/named/
/var/net-snmp/
/var/spool/
/var/webmin/
/var/www/

```

Be careful that the directories are correctly specified, otherwise an error will be incurred and the backup will fail. In this example, the dir-list is being stored in the administrator's home directory, in the subdirectory **/bkfst** (which has to be created), making it easy to find and update if necessary.

For the remaining dir-list – usr-list:

```
/root/bkfst/usr-list
/usr/dic/
/usr/doc/
/usr/games/
/usr/java/
/usr/local/
/usr/include/
/usr/lib/
/usr/share
```

Now we need to go back and modify our original script for the usr and var cases:

```
3)
filename=var$(date +%y%m%d).tar.gz
cd /var
tar cvfTz /backup/$filename /root/bkfst/var-list
;;
4)
filename=usr$(date +%y%m%d).tar.gz
cd /usr
tar cvfzT /backup/$filename /root/bkfst/usr-list
;;
```

One final note for our script. After everything is tested and working, you will generally want to the verbose (v) option of the tar statement.

OK, we now have the script for backing up the data. Now we need to set up cron to periodically run the script. Issue the command:

```
crontab -e
```

Enter the following lines (recall that you are using the vi editor):

<u>Min</u>	<u>Hr</u>	<u>DoM</u>	<u>Mo</u>	<u>Day</u>	<u>File</u>	<u>Opt</u>
0	22	*	*	1	/usr/local/backup/backup.sh	1
0	22	*	*	2	/usr/local/backup/backup.sh	2
0	22	*	*	3	/usr/local/backup/backup.sh	3
0	22	*	*	4	/usr/local/backup/backup.sh	4
0	22	*	*	5	/usr/local/backup/backup.sh	5
0	22	*	*	6	/usr/local/backup/backup.sh	6

Min = Minute

Hr = Hour

DoM = Day of Month

Mo = Month

Day = Day of Week,

0=Sunday

Thus, the following backup will be made on:

1. Monday 10:00 PM of every week and month /home directory
2. Tuesday 10:00 PM of every week and month /etc directory
3. Wednesday 10:00 PM of every week and month /var directory
4. Thursday 10:00 PM of every week and month /usr directory
5. Friday 10:00 PM of every week and month /opt directory
6. Saturday 10:00 PM of every week and month /data directory

22.3.2 Internal Backup Utility

22.3.3 Commercial Backup Utility

22.4 GAIM – Instant Messenger

As the web site specifies: ¹

Gaim is a multi-protocol instant messaging (IM) client for Linux, BSD, MacOS X, and Windows. It is compatible with AIM (Oscar and TOC protocols), ICQ, MSN Messenger, Yahoo!, IRC, Jabber, Gadu-Gadu, and Zephyr networks.

Gaim users can log in to multiple accounts on multiple IM networks simultaneously. This means that you can be chatting with friends on AOL Instant Messenger, talking to a friend on Yahoo Messenger, and sitting in an IRC channel all at the same time.

Gaim is a very feature rich Instant Messenger providing many features – and is under continual improvement. It is capable of running on multiple Operating Systems, including Linux and MS Windows.

Gaim is opened through the X menu, typically on the Internet menu.

¹ www.sourceforge.net



Figure 22.10: Gaim Opening Screen

To create a new account, click on the **Accounts** button.

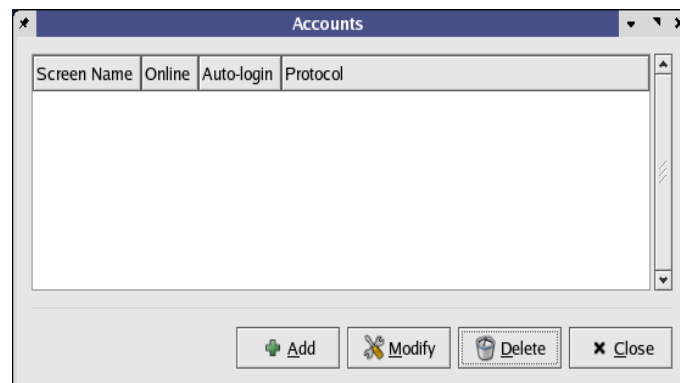


Figure 22.11: Gaim Accounts

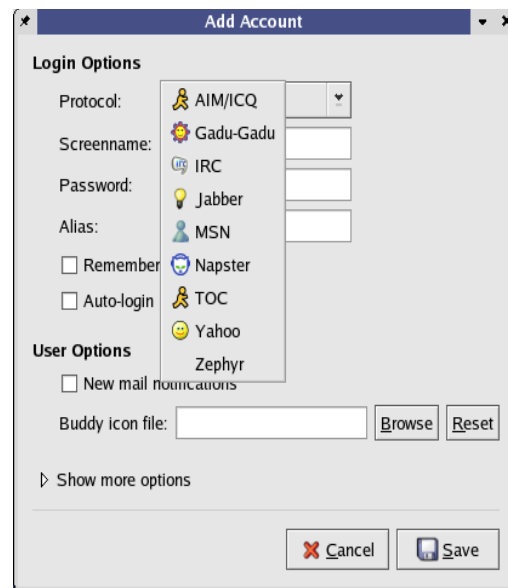


Figure 22.12: Gaim Adding New Login

Click on the **Add** button to create a new account.

Select which protocol you wish, then fill in your screenname, Password, and Alias. As an option, one may select to remember the user's password and to auto-login on startup. Finally, click on the **Save** button.

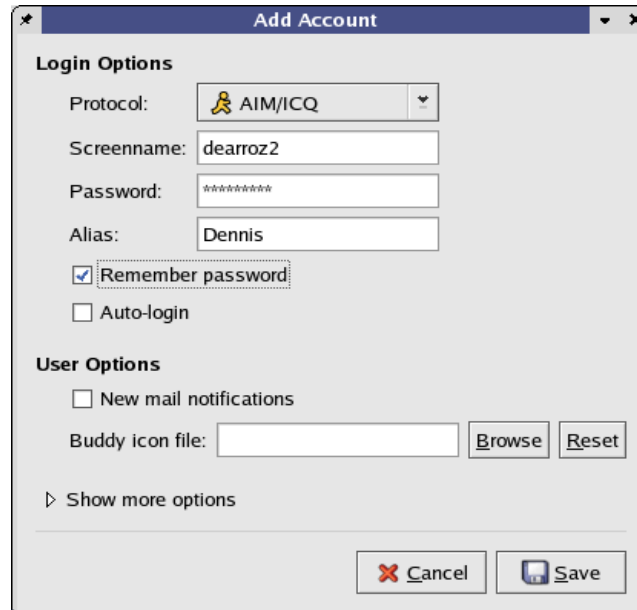


Figure 22.13: Gaim Adding Account Name

22.5 Connecting to Cisco Console Interface

When one first purchases a Cisco system (router, LAN switch, WAN switch, etc.), it is delivered in an un-configured condition. Before the system is placed into service, it must be configured via the Console interface. This is performed via the serial interface using the minicom program.

In order to connect to the Cisco system, one must use the Cisco supplied cables (or else make their own). This is because the Console port utilizes non-standard cabling and must connect to a computer using their special cabling. Cisco typically provides a db9F connector to RJ-45 converter and the special reversal cable, although with newer systems they now provide a single cable that has the special db9F connector and cable all in one.

After connecting the converter and cable to the desired serial interface of the computer, connect the second end of the cable to the Cisco Console port. The default configuration of Cisco devices is:

Data Rate:	9600 bps
Bit per character:	8
Stop bits:	1
Flow Control:	None

The CLI application **minicom** must be used to interface to Cisco device. Unfortunately, at this time, there is no known GUI serial port application.

22.X Commands Used in this Chapter

case	Script command to test for a given condition
crontab	Utility to modify the cron configuration
date	Read the date
gaim	Versatile Internet Messenger
minicom	Serial port application
tar	Create or concatenate a list of files

22.X Chapter Review Questions

Chapter Index

			D	
Data Backup	6	Screen Capture	S	6
Backup Process	9		T	
Backup Script	9	The GIMP		3
Creating filename	9	The GIMP		
crontab	11	Color Selection		5
Data Compression	8	Drop-Down Menu		3
Do It Yourself	6	Icon Tools		5
Initial Example	7		U	
TAR backup using list	10	Utility		
Using TAR Command	9	case		9
What to Back Up	6	GAIM		12
Where to Backup	8			