



Installing Linux Applications for the First Time

A Linux White Paper

Abstract

Although I have used and supported operating systems for many years (since before the IBM PC, in fact), when I began this project Linux was brand new to me. So it was with a mixture of eagerness and trepidation that I approached my assignment: Install Linux[®] and write about my experiences. (A companion paper, available from the same sources as this one, discusses installing the operating system itself.)

At first it sounded simple. After all, I have installed every version of DOS, Microsoft[®] Windows[®] and IBM[®] OS/2[®] that has ever existed up to this point (including quite a few beta versions along the way). However, they all have many things in common, thus simplifying the learning process: command names, directory structures, file and directory naming conventions and so on.

Once the operating system is up and running, the next step is to find and install applications for it. That is the subject of this paper: Where to find Linux applications and how to go about downloading and installing them. For someone new to Linux, this is not as simple as it may sound.

Note: There probably are many terms used in this paper that are unfamiliar if you are new to Linux. You may find the white paper called *A Brief Linux Glossary for Windows Users*, available from the same sources as this paper, to be a useful reference.

Contents

Abstract	1
Introduction	3
Where Are They Hiding All Those Linux Applications?.....	3
Compression/Decompression/ Packaging Tools	4
Tarred and Feathered	6
Zip It Up	8
Execute Me.....	9
Gee, More Zip	9
Rev Up Your Engine.....	9
A DEB Coming Out Party	10
I've BIN Had	11
Raiders of the Lost LHArc, and Other Rarities.....	11
Conclusion	11

Introduction

Linux is different from Windows in many ways, not the least of which is that there are many fewer commercial applications available for Linux than for Windows, and even fewer storefronts that carry them. This places a premium on being able to locate, download and install programs found on the Internet.

I was familiar with the methods of installing software on other operating systems, but the UNIX[®] world was new to me. (For example, I was used to .ZIP files in the DOS/Windows and OS/2 environments, and now I had to deal with a mélange of .tar, .tar.gz, .tgz, .gz, .z, .Z, .tar.bz2, .tar.bz, .tar.Z, .tbz2, .tbz, .rpm, .deb, .bin, .zip, .lzh, .lha and other file types.)

The following are my experiences braving the jungle of Linux application installation.

Note: I used Red Hat Linux 6.2 for my test platform, so the following procedures may not all work exactly the same way on a different Linux distribution, but at least they should give you an idea of how to proceed.

I decided to try installing applications packaged with each of the most common methods: tar, RPM (RPM Package Manager), gzip and zip, so I could compare and contrast those procedures.

Before I could test the various installation methods, however, I had to find some Linux software to download.

Where Are They Hiding All Those Linux Applications?

There are any number of sources for Linux software, both local and online, free and commercial. If you are not fortunate enough to live near a store that carries Linux applications, you will have to resort to companies that sell via phone, mail order or the Internet. Many of the free Websites are dedicated simply to providing a laundry list of software to download, while others add Linux news, online documentation and other helpful information. (For a list of useful applications to get you started, see the related white paper, *What good is a Linux Client?*, available from the same sources as this paper.)

Here are some sites to start you off in your search. (Many of them have links to other sites.)

First, free software sites:

- Freshmeat (freshmeat.net)
- Linux Apps (www.linuxapps.com)
- Linux Archives (home.linuxarchives.com/software.html)
- Linux Game Tome (www.happyenguin.org)
- Linux Games (www.linuxgames.com)
- Linux Software Encyclopedia (stommel.tamu.edu/~baum/linuxlist/linuxlist/linuxlist.html)
- Linuxberg/Tucows Linux (linux.tucows.com)
- Slashdot (slashdot.org)

Then online stores, for commercial software:

- Indelible Blue (www.indelibleblue.com)
- Linux Mall (www.linuxmall.com)

Preparing Today for Linux Tomorrow

- The Linux Store (www.thelinuxstore.com/perl-bin/index.pl)

Linux users in Australia can shop at:

- Everything Linux (www.everythinglinux.com.au)

Besides offering software, these commercial sites also carry Linux books, as well as hardware devices known to be Linux-friendly. Some of them even offer such things as caps, mugs, neckties and other Linux paraphernalia. If you are looking for an extensive list of Linux books, try Amazon.com (www.amazon.com), Barnes and Noble (www.bn.com), or O'Reilly & Associates (linux.oreilly.com). Each offers hundreds of books related to Linux, and the O'Reilly site adds Linux news and commentary.

Linux developers (and even some non-programmers) might be interested in the IBM developerWorks™ Web site at www.ibm.com/developer/linux. There you will find a number of free downloadable tools, including the Application Developer's Kit (which includes the Lotus® Domino™ server, IBM DB2®, IBM WebSphere™ Application Server, etc.), the Developer Kit for Java (including runtime modules), Bluetooth wireless technology development tools, IBM Journaled File System (JFS) source code, IBM VisualAge® for Java, and various device drivers, patches, ports and other Linux aids.

If you are looking for Linux application source code to play with, try the *SourceForge* (sourceforge.net/softwaremap). It is a repository of several thousand open source programs, written in a variety of languages and scripts (C, C++, LISP, Perl, Java, etc.) that can be downloaded for free and used or modified. Go to the home page for an explanation of open source, copyrights, licensing and other information about the SourceForge.

Finally, for a jumping off place to many other Linux Web sites, visit Linux Links (www.linuxlinks.com), FirstLinux (www.firstlinux.com) and Andover.net (andover.net).

Compression / Decompression / Packaging Tools

Now that you have found the applications that you need, the next step is to download and install them. Downloading in Linux is really no different from downloading with any other operating system. You select the application from your browser and specify the download location on your hard disk drive. Once that is done, the next step is not so simple. Most likely, the application files are combined into a “package” to simplify the downloading. (After all, you do not *really* want to have to individually download 473 different files, do you?)

There are a number of free and commercial programs that are used for creating “archives” of files, some with compression, some without. You will undoubtedly run into many different file formats, so you should be familiar with what they are and what tools are needed to unpack them. Some, but not all, of these tools have graphical interfaces to make them easier to use. It is a good idea to try out a few different utilities to see which you like best. Some are limited to unpacking one type of file while others can handle multiple types.

There are simply too many available tools to try to describe them all here. Nevertheless, there are a few common commands and utilities—some of which are likely to ship with your Linux distribution—that you can start with:

- **tar** — **tar** was the earliest and most common “packaging” tool for Unix/Linux. (It does the *unpacking* of files as well—there is no “*untar*” command.) The **tar** command creates a file that contains all the other files and directories, optionally in a compressed state (using **gzip** or **bzip2**) to save space. If you have used **zip/unzip** or PKzip with DOS/Windows, then you are already familiar with the concept. Files packaged with **tar** typically end with a **.tar** extension. If also compressed with **gzip**, the extension may be **.tar.gz**, **.tar.Z** or **.tgz**.

Tar files compressed with **bzip2** generally have a **.tar.bz2**, **.tar.bz**, **.tbz2** or **.tbz** extension. The **gzip** and **bzip2** compression utilities can also be used independently of the **tar** command to compress and decompress files (see below).

- **gzip/gunzip** — **gzip** (GNU zip) is the tool traditionally used for compressing tarred files and its companion **gunzip** is used to decompress those files. More recent tools, such as PKunzip, can generally uncompress files created by **gzip** as well as those compressed by other tools, such as **PKzip** and **zip**. Due to this flexibility, and because **PKzip** and **zip** are also available on other operating system platforms, they have become more popular than **gzip**. Files compressed with **gzip** usually end with **.gz**, or in the case of some older files, **.z** or **.Z**.
- **bzip2/bunzip2** — Think of **bzip2** as a newer, faster version of **gzip**. It has a very similar command structure (including many of the same parameters), but it tends to be faster at compressing and decompressing than **gzip/gunzip**. Bzipped files end with a **.bz2** or **.bz** extension.
- **zip/unzip** — **zip** and **unzip** are a pair of command-line tools that will package and compress (and decompress) files and directories. In addition to being very fast, they are free and available for other platforms, including DOS/Windows, OS/2 and other UNIX OSes, including SCO. The **unzip** command can also unpack files created by **PKzip**. Files compressed with **zip** end with a **.zip** extension.
- **PKzip/PKunzip** — **PKzip** is a popular shareware/commercial tool, primarily in the DOS/Windows world. **PKzip** is very fast and offers varying degrees of compression vs. speed. Newer versions include a graphical user interface. **PKunzip** is compatible with files created by **PKzip**, **zip**, **gzip** and **tar**, as well as the UUencode, XXencode, MIME and BinHex file formats. Like **Zip**, **PKzip** files end with a **.zip** extension.
- **Kpackage** — The RPM Package Manager (**kpackage**) was developed by Red Hat for its Linux distributions, but it proved so popular that now other distributions use it as well. RPM-packaged applications are self-extracting and self-installing. RPM also maintains a table of the programs installed by RPM, and their interdependencies. It notifies you if another software product that is required by the one you are installing is missing or down-level, and tells you that you will need to obtain and install it. It also simplifies uninstalling software previously installed with RPM. Files packaged with RPM have a **.rpm** extension.
- **Dpkg** — The Debian Package Manager (**dpkg**), developed by VA Linux for its distributions, is somewhat similar to the RPM Package Manager and likewise has a graphical interface. These files are identified by a **.deb** extension.

To summarize which tools can be used to unpack these and other, less common, file types:

File Extension	Tools
.arc	See <i>Raiders of the Lost LHArc</i> , below
.arj or .ar	TkZip or Unarj
.bin	See <i>I've BIN Had</i> , below
.bz or .bz2	Bzip2, Gnozip, gxTar, or Karchiveur
.deb	Dpkg
.exe	Self-extracting; just type the filename
.gz	Gzip, Gnozip, Karchiveur, or PKunzip
.lha or .lzh	Karchiveur or KPackViewer
.rar	GnomeRAR or Karchiveur
.rpm	RPM
.spm	See <i>Raiders of the Lost LHArc</i> , below
.tar	Tar, gxTar, or PKunzip
.tarbz or .tar.bz2	Tar, gxTar, or PKunzip
.tar.gz or .tar.gzip	Tar, gxTar, or PKunzip

.tbz or .tbz2	Tar, gxTar or PKunzip
.tgz or .tar.Z	Tar, gxTar or PKunzip
.z or .Z	Gzip, Gnozip, Karchiveur or Pkunzip
.zip	Pkunzip, Unzip, Gnozip or Karchiveur
.zoo	Gnozip

In addition, there are a number of downloadable utilities, including **guiTAR**, **gxTar**, **KPackViewer**, **Qutar** and **tkArchive**, that enable you to run some of these tools from a graphical interface. To find these or other tools, see the *Where are they Hiding All Those Linux Applications?* section, above, for a list of Web repositories of downloadable Linux utilities.

In the following sections we will see examples of how to use these tools to unpack and install various programs. To complete the installation after unpacking the files, follow the instructions for the program. There should be an installation program to run once the files are unpacked.

Tarred and Feathered

Okay, okay! I admit that there were no feathers involved, only tar. To find out how to install a program packaged with the **tar** utility, I began by downloading StarOffice 5.1a, an office suite, as one 70MB file called **so51a_lnx_01.tar**. (It was also available as a series of 7MB-15MB files, but I had a fast T1 Internet connection to work with, so I went with the simpler single-download method. See below for information about reintegrating a multipart download.)

The browser defaulted to putting the file into my **/home/mark** directory, and that was fine by me. After the file downloaded the next question was, “What do I do with a **.tar** file?” I knew that it was roughly similar to a **.ZIP** file in the Windows world and I knew that there was a **tar** command but not how to use it.

I opened a command shell, using the “terminal emulator” icon—the one that looks like a little computer screen—on the KDE Panel (Taskbar, to Windows users) and changed to the directory where the **.tar** file was located:

```
cd /home/mark.
```

I typed the command

```
tar --help
```

to find out the syntax for the command. Then I used

```
tar -xvf so51a_lnx_01.tar
```

to extract the files from the Tar archive.

The **x** parameter tells **tar** to **extract** the contents of the “archive,” the **v** says to be **verbose**, or list the files as they are extracted, and the **f** means to use the archive filename that follows. The **v** parameter is purely optional and can be omitted if you prefer. Other arguments may be included as needed.

Note: UNIX/Linux commands are case-sensitive, so be *sure* to use **xvf**—not **XVF**—or you will get completely different results.)

Preparing Today for Linux Tomorrow

The extraction process created a new directory, **/home/mark/so51inst**, and under that, **/home/mark/so51inst/office51**. This latter directory included hundreds of files, including one called **setup.ins**. This is the installation program.

The simpler, graphical method of reaching the same point is to click on the KDE icon on the left side of the Panel (Taskbar), then *Disk Navigator*, then **so51a_lnx_01.tar**. Clicking (or double-clicking, depending on the particular file manager you are using) on the filename automatically starts the *tar* process to unpack the files. Much easier than the command line method, is it not? (On the other hand, using the command gives you more flexibility in terms of how you want to unpack the files and where you wish to put them.)

If you have a *tar* file that was compressed with **gzip** (i.e., the file name ends with **.tar.gz**, **.tar.Z** or **.tgz**) things get a bit more complicated. There are several ways you can unpack the files, using either the **tar**, **gunzip**, or **gzcat/zcat** commands. (Some Linux distributions include **gzcat**, while others provide the similar **zcat** utility.) Reading the online **tar** and **gunzip** manual pages (command: `info tar` and command: `info gunzip`), may leave you more confused than when you started (I know I was!), because although each of those methods may work, some are more error-prone than others. Rather than show all the possible methods and explain the reasons not to use most of them, I will simply present the easiest and safest way to do it.

If, for example, the file is named **MyProgram.tgz**, simply use the following command:

```
tar -xfvz MyProgram.tgz
```

The **xfv** parameters have the same meaning as in the earlier **tar** example. Adding the **z** parameter tells *tar* to also decompress with **gzip**.

Note: You may have noticed that I used **-xvf** in one example and **-xfv** (plus **z**) in the other. This was to illustrate that while Linux may be case-sensitive, it is not usually fussy about the order in which you use the arguments (parameters).

Likewise, if the *tar* file was compressed with **bzip2** (i.e., the file name ends with **.tar.bz2** or **.tar.bz**), you can unpack the files using either the **tar** command or the **bunzip2** command. If, for example, you had a file named **MyProgram.tar.bz2** the simplest command to use is:

```
tar -xfvi MyProgram.tar.bz2
```

This time, the addition of the **i** parameter tells *tar* to use **bzip2** to perform the decompression.

Multipart Downloads

Earlier I alluded to the fact that some large packages may be provided as multipart downloads for users with slow Internet connections. If you were to download a package this way, how would you stitch those pieces back together before installing?

In the StarOffice 5.1 example, above, the product was offered as seven files named **so51a_lnx_01_n1.tar** through **so51a_lnx_01_n7.tar**.

To turn those seven files into one big installable file, use the following procedure:

First, to ensure that you do not accidentally corrupt the original file, copy it to a new file called **staroffice.tar**:

```
cp so51a_lnx_01_n1.tar staroffice.tar
```

Preparing Today for Linux Tomorrow

Then use the **tar** command as follows:

```
tar --concatenate -file=staroffice.tar so51a_lnx_01_n2.tar
```

*This will append the second file to the end of the first one (while removing the end-of-file marker from the end of the first file—do *not* try to use the **cat** command to concatenate the files, because it does not know to strip out the end-of-file marker).*

Repeat this command for each of the remaining five files, replacing the **so51a_lnx_01_n2.tar** filename with the third filename, then the fourth and so on. Each will be appended in turn onto the end of the **staroffice.tar** file:

```
tar --concatenate -file=staroffice.tar so51a_lnx_01_n3.tar
tar --concatenate -file=staroffice.tar so51a_lnx_01_n4.tar
tar --concatenate -file=staroffice.tar so51a_lnx_01_n5.tar
tar --concatenate -file=staroffice.tar so51a_lnx_01_n6.tar
tar --concatenate -file=staroffice.tar so51a_lnx_01_n7.tar
```

When you are done with all seven parts, run the **tar** command as you normally would on the **staroffice.tar** file (which, if you executed the previous steps correctly, should contain the entire seven-part package) to unpack it:

```
tar -xvf staroffice.tar
```

Note: Be *sure* you append the files in the correct order; otherwise, you will not end up with a working package.

There are many different parameters available for the **tar** command, so it is best to read the manual (“man pages”) for **tar**, using the command: `info tar`.

Zip It Up

Files with a **.zip** extension have been packaged by either **Zip** or **PKzip**. The programs are generally interchangeable (as long as you are using a recent version), in terms of being able to unpack each other’s files, and relatively simple to use.

To unzip a file called **Whoopie.zip**, the simplest method would be to use one of the following commands, depending on which tool you are using:

```
unzip Whoopie
```

or

```
pkunzip Whoopie
```

(The **.zip** extension is optional.) This will unzip the files into the current directory. There are a number of arguments (parameters) that may be used if needed—for example, if you are unpacking a **.zip** file that spans multiple floppies.

The instructions that come with the file you are attempting to unzip should tell you what arguments to use, if any. **Zip** and **Unzip** may be included with your Linux distribution; if so, read their manuals using the following commands: `info zip` and `info unzip`. For more information on **PKzip/PKunzip**, read the documentation that comes with **PKzip**.

Execute Me

A variation on the **.zip** file is the **.exe** file, which is an executable self-unpacking form of the **.zip** file. When you run the **.exe** file, not only does the file unpack itself, but it already knows which arguments are needed to create subdirectories, etc. For instance, to unpack the file **MyProgram.exe**, simply run:

```
MyProgram.exe
```

from whatever directory it is currently in and the unpacking process will take care of the rest.

Gee, More Zip

If a file has a **.gz**, **.z** or **.Z** extension, it was compressed with **gzip** and you will need **gunzip** to unpack it. (Possibly another tool, such as **PKunzip**, can unpack it as well.) If it has a **.bz2** or **.bz** extension, it was compressed with **bzip2** and requires **bunzip2** to uncompress it.

To decompress a file called **MyProgram.gz**, compressed with **gzip**, use the following command from within the directory containing the file:

```
gunzip -cd MyProgram.gz
```

Similarly, to decompress a file called **MyProgram.bz2**, you can use this command:

```
bunzip2 -cd MyProgram.bz2
```

For an example of using the **gunzip** and **bunzip2** commands to unpack a file that was packaged with **tar** and compressed with **gzip** or **bunzip2**, see the *Tarred and Feathered* section, previously.

You can read more about the **gzip/gunzip** commands using: `info gzip`. For more information on using **bzip2**, try the: `info bzip2` command (in case your distribution included it), or see the Web page at fetter.org/Bzip2-HOWTO/Bzip2-HOWTO.html.

Rev Up Your Engine

Well, maybe not with *this* type of RPM. The RPM Package Manager simplifies the unpacking, installing and uninstalling of software packages on Linux distributions that include the RPM Package Manager—but not all distributions do. (Distributions that do provide it include Caldera, Red Hat, SuSE and TurboLinux, among others.)

For this example, I downloaded **AbiWord** version 0.7.10 as a file called **abisuite-0.7.10-1.i386.rpm**. I opened a file manager and navigated to the **/home/mark** folder where the file was located and then simply double-clicked on the file icon. This started the **Kpackage** (RPM Package Manager) program. (*Note that you may have to be logged on as the root operator to install RPM packages.*) The check boxes on the left side of the panel let you specify whether you are upgrading over an existing version of the product, wish to look for interdependencies with other programs, and so forth. The *Check Dependencies* feature is useful, as it will determine whether the product you are installing first requires other products to be installed (or upgraded to a newer version).

On the right side of the panel are two tabs: *Properties* and *File List*. The *Properties* tab describes the product you are installing, including the version number, the software build date, the name of the vendor and so on, while the *File List* tab shows just what the name implies: the complete file and directory contents of the package.

When I pressed the *Install* button, RPM proceeded with the installation. There is a progress measurement graph at the bottom-right side of the Kpackage window. When the blue bar reaches the end of the graph the installation is complete. (There is no message displayed to indicate this, however.) At this point, the left side of the panel displays a list of all the RPM packages installed on the system. To find out more about any of these products, simply left-click on a folder name (such as Editors) and then the product name (in this case: "abisuite"). The right side of the panel displays the *Properties* and *File List* tabs for that package. When you are done, close the panel as you would any other window using the X button in the upper-right corner of the panel.

Note: You will also find a new button, *Uninstall*, at the bottom of the right side. Use this to remove any software installed via RPM. Kpackage can be started from one of the Utilities menus, depending on which Linux distribution you are using.

If the *Check Dependencies* feature indicates that one or more other products need to be installed or upgraded it will tell you what they are, but it will be up to you to find those products on the Internet or elsewhere, and install them.

If you would rather work from a command line, or find yourself forced to by necessity, the command to install an application, such as **AbiWord** using the same example, is very simple:

```
rpm -i abisuite-0.7.10-1.i386.rpm
```

The `-i` parameter tells RPM that you are installing the package for the first time. If you are upgrading to a newer version, you should use `-U` (for Upgrade—be sure to use a *capital U*) instead. If you need to *reinstall the same version* of the program, rerun the command with one additional parameter in order to override the error message telling you that it is already installed:

```
rpm -i --replacepks abisuite-0.7.10-1.i386.rpm
```

If it turns out that the new version is buggy and you would rather return to the previous version (called "downgrading"), you can do so as follows:

```
rpm -U --oldpackage abisuite-0.7.09-1.i386.rpm
```

(Again, notice the *capital U*.)

Should you decide to remove **AbiWord** at a later time, the command to use would be:

```
rpm -e abisuite
```

The `-e` argument means Erase. Note that typing the rest of the file name (consisting of the version number and other information about the software) is unnecessary. RPM will get the rest from the installation log file, which was updated when you first installed **AbiWord**.

The RPM command offers many more command line features, including the ability to find all of the files, or all the documentation, owned by a package; or the converse: which package owns a given file. For more on the RPM command read the manual page using: `info rpm`.

A DEB Coming Out Party

Some Linux distributions (Debian Linux and CoreLinux, for example) offer an alternative to the RPM Package Manager, called the Debian Package Manager. *If you are using a distribution that comes equipped with the Debian Package Manager, you would be looking for software that uses this package manager instead of RPM.* These files are identified by a `.deb` file

extension. The basic command to use is: `dpkg MyProgram.deb` (or whatever the name of the file is). Use: `info dpkg` to determine what parameters are needed to extract the program you are installing, or consult the instructions that come with that program.

If you are using one of the distributions that ship with RPM Package Manager (see *Rev Up Your Engine*, above), then you should be looking for software that has a **.rpm** extension instead. *In general, software with a .deb extension is intended only for Debian-based distributions and not RPM-based ones.* It is possible that **.deb** files will install and run correctly on RPM-based distributions, or vice versa, but there are no guarantees.

I've BIN Had

This is not actually a package file type, but the package you download may have a **.bin** file extension to identify it as a type of binary file. If your system is set up to default to read/write (but not execute) permissions on downloaded files, you must run the **chmod** command on the file to make the file executable. For example, when I downloaded the update to StarOffice—version 5.2—the file was called **so-5_2-ga-bin-linux-en.bin**. Before I could install it, I first had to login as the root operator (Superuser) with the command

```
su -
```

(and enter the root password, when prompted). Then I used the command

```
chmod +x so-5_2-ga-bin-linux-en.bin
```

to make it executable (the **+x** parameter grants execution permission for the program).

After making the file executable, I went to a file manager, found the file icon and clicked on it to begin the installation. (Again, you may have to double-click, depending on the file manager you are using.)

Raiders of the Lost LHarc, and Other Rarities

Once upon a time **LHarc** was a popular archiving tool, but with the advent of newer tools it has fallen out of favor. However, it is still possible that you will encounter files with a **.lzh** or **.lha** extension, so it is helpful to know what those files are. Tools that can decompress **LHarc** files, among others, include **Karchiveur** and **KPackViewer**.

Other file types that are older or rarely encountered in the Linux world include **.arc**, **.arj**, **.rar**, **.spm** and **.zoo**. If you run into some of these, there should still be utilities available that can unpack them, such as **TkZip** and **Unarj** (**.arj**); **GnomeRAR**, **Karchiveur**, **TkZip**, **Unrar** (**.rar**); and **Gnozip** (**.zoo**). If needed, check out the Linux software sites listed in the *Where Are They Hiding All Those Linux Applications?* section, previously, for some of these tools.

Conclusion

The preceding information about the various packaging file types and how to unpack them should help lessen some of the confusion surrounding the installation of the vast majority of downloaded software you are likely to encounter for Linux.

Should you become utterly confused, there are a number of avenues to pursue for possible sources of help, in addition to all the online manuals included with Linux.

For informational publications about Linux, try these Web sites:

Preparing Today for Linux Tomorrow

- Linux Documentation Project (www.linuxdoc.org) — A collection of Frequently Asked Questions (FAQs), “how to” documents, manuals and online magazines.
- Linux Journal (www.linuxjournal.com) — It contains links to newsgroups, chat rooms, online manuals, FAQs, vendor support sites, local Linux user groups and more.
- Linux Man Pages (linux.ctyme.com) — A collection of the “man” (manual) instructions for various Linux commands and utilities.
- Linux Newbies (www.linuxnewbies.org) — Help files for “newbies” (those new to Linux).

To contact the distribution vendors directly for questions about what tools they provide and how to use them, try:

- Caldera Linux support page (www.caldera.com/support)
- Debian Linux support page
(www.valinux.com/services/support/?session_hash=7d7ccc8f28c0f42cddc98da037d7875a)
- Red Hat Linux support page (www.redhat.com/apps/support)
- SuSE Linux support page (www.suse.com/us/support/index.html)
- TurboLinux support page (www.turbolinux.com/support)

Some, if not all, of the distribution vendors offer telephone support for a fee. Consult the documentation that came with the distribution (either printed or readme file) for the appropriate telephone number. Commercial application vendors may offer technical support for their products, as well.

Finally, the following Usenet newsgroups can provide a significant amount of user-to-user help:

- *alt.uu.comp.os.linux.questions* — General Linux questions and answers.
- *comp.os.linux.announce* — Announcements of new products, updates, bug fixes, etc.
- *comp.os.linux.answers* — Linux FAQs, How-To's, READMEs and other documents that answer questions about Linux.
- *comp.os.linux.misc* — A catchall for whatever does not fit in one of the other groups.
- *comp.os.linux.setup* — How to install and configure Linux and add-on software products.

There are other Linux newsgroups provided for other topics, such as vendor-specific questions (*alt.os.linux.caldera*, *linux.debian.user* or *redhat.rpm.general*, for example), hardware issues and those for programmers.

Mark T. Chapman
IBM Server Group
August 15, 2000



© Copyright IBM Corporation 2000

IBM Server Group
3039 Cornwallis Road
Dept. LO6A
Research Triangle Park, NC 27709

Produced in the USA
9-00
All rights reserved

References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates. IBM reserves the right to change specifications or other product information without notice.

IBM, the IBM logo, DB2, developerWorks, OS/2, VisualAge and WebSphere are trademarks of IBM Corporation in the United States and/or other countries.

Linux is a registered trademark of Linus Torvalds.

Lotus and Domino are trademarks or registered trademarks of Lotus Development Corporation.

Microsoft, Windows, Windows NT and the Windows logo are trademarks or registered trademarks of Microsoft Corporation.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

All other trademarks and registered trademarks are the property of their respective owners.

THIS PUBLICATION MAY INCLUDE TYPOGRAPHICAL ERRORS AND TECHNICAL INACCURACIES AND MAY BE CHANGED OR WITHDRAWN AT ANY TIME. THE CONTENT IS PROVIDED AS IS, WITHOUT WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. SOME JURISDICTIONS DO NOT ALLOW DISCLAIMER OF EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS, THEREFORE THIS DISCLAIMER MAY NOT APPLY TO YOU.

This publication may contain links to third party sites that are not under the control of or maintained by IBM. Access to any such third party site is at the user's own risk and IBM is not responsible for the accuracy or reliability of any information, data, opinions, advice or statements made on these sites. IBM provides these links merely as a convenience and the inclusion of such links does not imply an endorsement.