

FIRST STEPS LINUX BEGINNERS SERIES

Compression Ark and File Roller

In which a smallerified Andy Channelle spends too long playing with the excellent compression software available on Linux. Dial-up readers: let this be a warning to you.



LAST TIME
In LXF70 I ended the three-part mini series on system maintenance with a skip through recent Gnome updates. If you missed the issue, call 0870 837 4773 or +44 1858 438795 for overseas orders.



Everything is getting bigger. Office suites now routinely come in at 60MB-plus, websites are stuffed full of video and audio and most distros would put Hulk Hogan in the shade. To cope with this trend towards bigification you need compression software. This uses some incredibly clever mathematics to squash files down to a fraction of their original size, thus allowing them to be stuffed down a phone line or shoehorned in to an already overflowing CD.

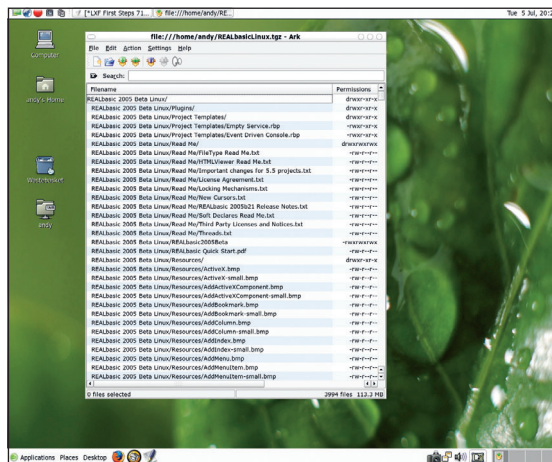
Compressed files, or archives, come in a number of guises, but all are signalled by their file extension, that is, the letters that follow the period (.) in a filename. These differ from the files we usually encounter; in fact they have more in common with directories, because they can contain more than one file. The file extensions you need to look out for include .zip, .tar.gz, .tar.bz, .rar, .jar and .war among others. Documents are most often compressed into a .zip file, whereas Linux software that you might want to build yourself is generally stored in a .tar.gz or

.tar.bz archive – otherwise known as a tarball. These are not hard and fast rules, but it does seem to be the way people like to work.

When faced with an archive, there are a number of things we can do. We'll start with the traditional command line option beloved of source code installers and then look at a couple of desktop applications that can do the same job with a little less typing and command memorising.

Getting the bends with tar and gzip

First we need to download a tarball and save it to a new directory within **/home**. I've chosen the latest version of *Gaim* and downloaded the file **gaim-1.3.1.tar.gz** from the project's website. The filename has three parts: the first is the application name and version; the second (**.tar**) tells us it's an archive of many files; and the third (**.gz**) tells us it has been compressed using *gzip*. I've saved the 8.1MB archive into **/home/gaim**.



Ark's user interface is divided into three areas: menu bar, tool/search bar and list area.

Now open a console and navigate to that directory like so:

```
cd gaim
```

Note that we don't have to specify that we're going into `/home`, because that's where we're running the command from. You can check the contents of the `/home/gaim` directory by typing `ls` or `dir`. The command we're going to use will follow the following format: **tar -<arguments> <filename>.tar.gz**. There are a few common arguments we can use:

- **x** Extracts the contents of the archive.
- **z** Unzips the archive.
- **v** Displays a commentary of what's happening.
- **f** Tells `tar` that a filename will follow.
- **c** Creates a new archive.
- **t** Lists the contents of the archive.

To decompress and extract the archive we will use this command with four arguments:

```
tar -zxvf gaim-1.3.1.tar.gz
```

This will populate the `/gaim` directory with the contents of the tarball – over 1,000 files. We could use an extra switch to extract the files to a different location. For example, the command

```
tar -c /tmp -zxvf gaim-1.3.1.tar.gz
```

will extract the contents of the archive into the `/tmp` directory. Our `Gaim` directory has now expanded to take up just over 30MB of disk space.

Logically enough, you can also go the other way: it's possible to turn a collection of files or a directory into a tar archive from the command line. If we want to create a new archive called `archive.tar` and include the files `a.png`, `b.png` and `c.png`, we do the following while in the same directory as the files:

```
tar -cvf archive.tar a.png b.png c.png
```

This will create a single, uncompressed archive suitable for emailing or backing up to CD. However, if your intended recipient is on a dial-up connection or your CD is already packed, you can compress the archive using `GNU Zip`, aka `gzip`. The `gzip` command is simplicity itself, though you have to remember to execute from within the same directory as the `tar` archive:

```
gzip archive.tar
```

This should create a new file called `archive.tar.gz` in the same folder. The amount of compression achievable depends on a number of factors, including the type of files in the archive and whether any other form of compression has been applied beforehand. There will be no benefit, for example, in compressing previously compressed archives. No algorithm is that clever!

So that's the hard way to compress and decompress files. Fortunately, Linux developers have devised decent graphical applications to do the same job. These applications can handle a whole load of different compressed formats. As ever, there are two standard applications that you may come across. *Ark* is the KDE archive manager, while *File Roller* does the same job for Gnome. The big advantage for command line-phobes is that complex jobs such as adding, extracting and deleting stuff from existing archives are made really easy.

Squeeze into the Ark

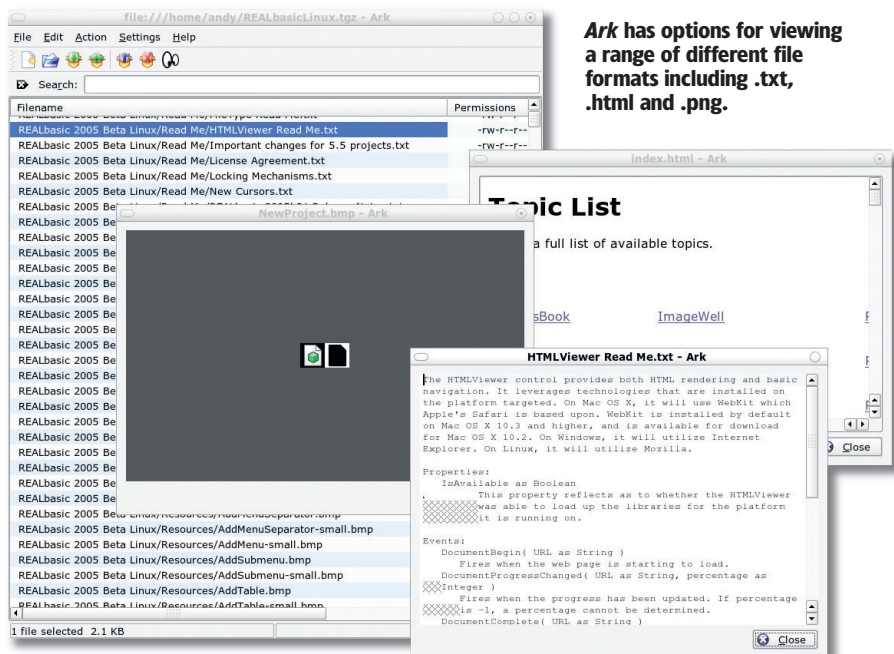
Ark follows the conventions of a typical KDE application. It has a menu bar giving access to various functions via drop-down menus; a toolbar, which has shortcuts for common tasks; and a work pane, which displays the contents of an open archive.

We can get access to the *Gaim* archive that we downloaded earlier using the `File > Open` menu option and navigating to the

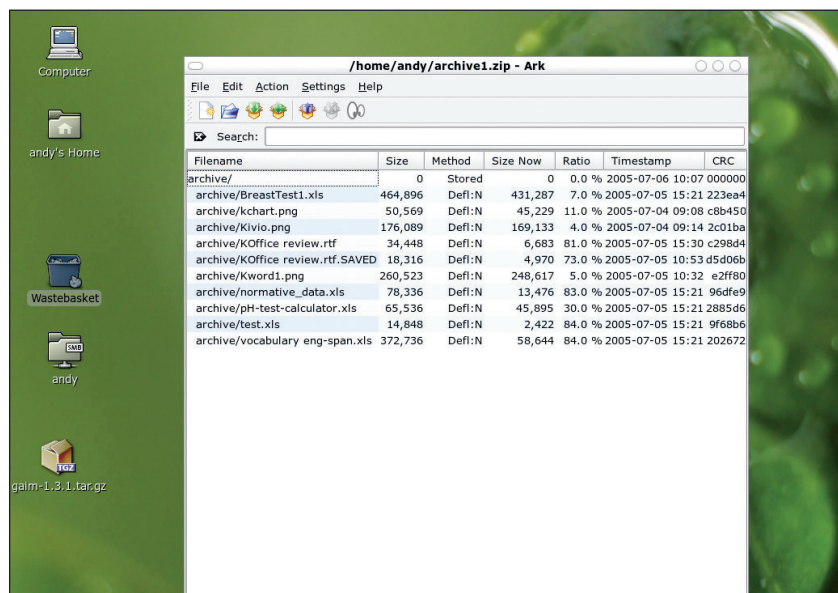
“ADDING, EXTRACTING AND DELETING STUFF FROM ARCHIVES IS MADE REALLY EASY.”

correct file using the browser. After a little while (and extracting the contents of a compressed archive can take some time depending on its size and the speed of your machine), we will be shown a list of all the files in the archive. Though it may not be immediately apparent, there is a hierarchy here that will mirror the final directory structure of the extracted folder. These folders are defined, as might be expected, using forward slashes, like this: `/directory/subdirectory/file.ext`.

The most obvious thing to do is extract everything. You can do this by making sure nothing is highlighted in the list view, and doing `Action > Extract` or hitting the Extract button, which is third from the right on the toolbar. This will do the same job as the `tar -zxvf` command we used earlier, and that's to decompress all the archived files into a directory. One difference is that it's much easier with *Ark* to specify a different location for the extracted files. Just add the path in the Extract To... field or click on the folder icon to open a file browser to locate the right directory. Hit OK and the files will be decompressed.



Ark has options for viewing a range of different file formats including .txt, .html and .png.



Compression ratios vary between file types.

« We're not, however, restricted to extracting the tarball in its entirety – we can specify a range of files and directories to decompress without affecting the rest of the archive. This isn't particularly useful when it comes to archived applications, but it can be used for pulling out individual files from a backup. With the relevant archive opened in *Ark* you can select single files by clicking on them and hitting the Extract button. This file will retain any positioning within the file structure, so if you only want to extract the file **blist-signals.docx** (which is in **/gaim-1.3.1/doc**), you would find a directory called **gaim-1.3.1** within home, and within that there would be a new directory call **doc** that would contain the decompressed file.

“RTF AND XLS FILES OFTEN SHRINK BY OVER 70%. IMAGE FILES COMPRESS LESS WELL.”

A contiguous range of files or folders can be selected by using the Shift key in combination with the left mouse button, and you can select any number of non-adjacent files using Ctrl+click. By the way, next to the Extract button is a Delete button which – surprise! – can be used to remove files from the archive. Be careful with this button, as there's no 'recycling bin' from which you can recover deleted files. Once it's gone from the archive, it's gone for good.

Finally, there is a handy viewer applet for the extraction process, which enables us to select a file and then look at its contents without having to go through the whole decompression process. This is especially important if you've packaged up an archive of images (photos, web graphics and so on) with unintuitive names. Select the file, hit the Eyes icon and the viewer will be launched. This also works for XML, HTML, text and a range of other file formats.

You may also notice under the Action menu options called Open With... and Edit With..., and these do exactly what you'd expect. The former will decompress the chosen file and open it with an application of your choosing. The latter does the same thing, but attempts to write the changes back directly into the archive. Note that many applications don't support this function.

Of course, extracting archives is only one part of the process. We also need to be able to create new archives and add files to

existing ones. Fortunately, *Ark* can handle it all. Say we have a folder (**/home/archive**) containing 11 files that need to be compressed into an archive. First we need to launch the application and do File > New to open the New Archive dialog. This is the same as a normal KDE file browser, so we define where we want the archive to go and give it a name. Select the archive type using the drop-down list, and mark the radio button, which automatically adds the relevant file extension. We are now presented with the normal *Ark* window ready for populating with files.

Next use Action > Add Folder... to select the **/archive** folder. The main part of the application window will now show the files in this archive, but there's some other important information here too. If you look along each row, you will see the original size of the file, the compressed size and the compression ratio as a percentage. As the image on the left shows, the compression ratios possible vary dependent on many factors. Rich text format (.rtf) files, which contain only text, often shrink by over 70%, as do text- or number-only .xls files. Image files, and those with a lot of image data embedded in them, tend to compress less efficiently.

Note that once you get to this stage, the archive is treated as above: you can extract, delete or view parts of it as before. It's also possible to add files or folders to an archive – the

SMALL PICTURES

The most common compression format you are likely to come across is JPEG. This format, created by and named after the Joint Photographic Expert Group and given the .jpg extension, was designed to allow large images such as photographs to be compressed for distribution over the internet. It is also the *de facto* standard for digital cameras.

The thing about JPEG is that it's a 'lossy' format; that is, it works by removing information that would normally not be seen by the human eye, just as MP3/OGG removes sound frequencies outside human hearing to make audio files smaller. When saving images in this format, users must work out the best compromise between size and quality: the better the image, the bigger the file, and the longer it will take to download.

Gimp has a ton of options that can be used to optimise an image: these range from the sledgehammer-like Quality slider to more subtle space-saving options such as Smoothing or removing the EXIF data that digital cameras usually append to each photograph.

JPEG works by comparing regions of colour – if they are similar enough (this is where the Quality setting comes into effect) it treats the two regions as a single colour. This is why heavily compressed images look blocky and pictures with lots of solid colour can have very large compression ratios.

A lossless alternative to JPEG is the Portable Network Graphic (.png) format. This uses the *zlib* compression library and was created as a response to patent issues regarding the compression algorithms in the .gif image format. It works using magic.



The shot of Novell boss Jack Messman on the right is a fraction of the size of the non-compressed version on the left. But it looks a lot worse.

QUICK TIP

Working with .zip
Decompress .zip files from the command line with

unzip filename.zip

You can zip a file up with

zip archivename.zip

filename.txt and whole directories can be zipped up by using

zip -r archivename.zip

/path/to/directory

The -r switch ensures that

all subdirectories are added

to the archive.

Find out more by typing

man zip at the command line.

ARCHIVE FORMATS SUPPORTED BY ARK AND FILE ROLLER

	.a	.ar	.bz	.bz2	.deb	.ear	.gz	.jar	.lha	.lzo	.rar	.tar	.tar.bz	.tar.bz2	.tar.lzo	.tar.gz	.tar.z	.war	.z	.zip	.zoo	.7z
Ark	✓		✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓
File Roller		✓				✓		✓				✓		✓		✓		✓		✓		

procedure is the same as before: Action > Add Folder... or Action > Add File.

Before we move on we'll take a brief look at Ark's configuration, as this allows us to set up better integration with the KDE desktop. To get to the appropriate dialog, do Settings > Configure Ark... The settings dialog is quite small and easy to manage, and is divided into three sections: General, Addition and Extraction.

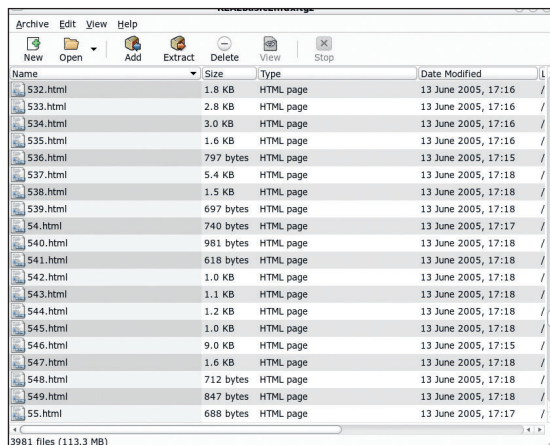
■ **General** allows you to set up *Konqueror* integration, which makes it possible to add or extract files using *Konqueror*'s right-click 'service' menu. You can also set Ark to use its own viewer applet to audition files before extraction. Switching this off will cause Ark to open files selected for viewing in the desktop's default application.

■ **Addition** is used to set various options for adding files to an archive. We can, for example, set things up so that old files in an archive are overwritten by new versions – this is useful for backups. You can also make sure that 'symlinks' in a folder – that is, a file that links to another file in a different location, like an alias – are included as real files. This is also important in backups, especially if you have files on a remote server with symlinks set up within your `/home` directory. There's nothing more frustrating than realising you have accidentally backed up ten empty folders that link to now non-existent directories!

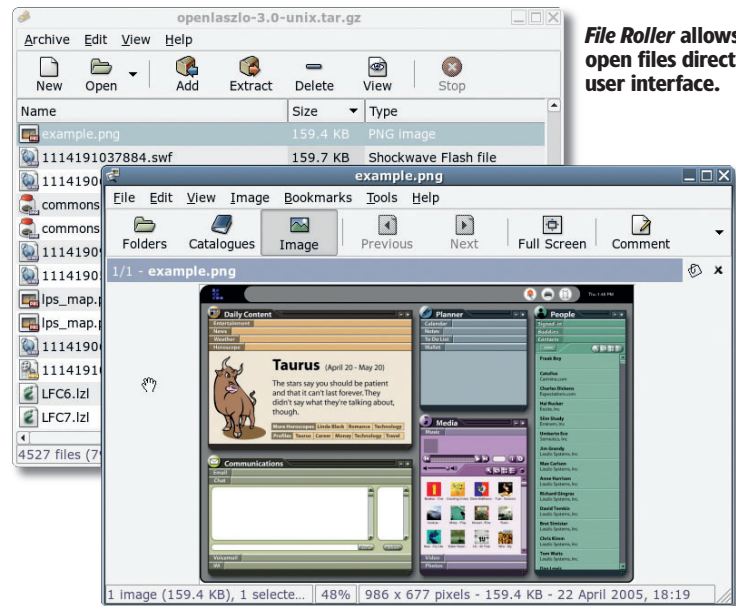
■ **Extraction** contains options that affect the extraction process. For example, we can set the application to preserve file permissions once extracted, or overwrite files of the same name. However, many of the options within this section are limited to particular archive formats.

Secrets of File Roller

File Roller is the archive/zip manager included with the Gnome desktop. Some versions of the desktop put it under the Utilities > Archiving > Archive Manager title rather than just calling it *File Roller*. Once launched, the basic user interface is remarkably similar to Ark's, and it works in pretty much the same way. However, buried within the menu bar are a few more options that are worth looking at.



File Roller is very similar in look and feel to other archive applications but it has some useful security tools.



File Roller allows you to open files direct from the user interface.

You may notice, once you've opened or created an archive, that the file window layout is a little different. *File Roller* doesn't include the full path to the file name in the first column, but puts it in the far right Location column. You'll also notice the lack of 'before' and 'after' sizes and the compression ratio column. You can find the overall compression ratio under the **Archive > Properties** menu.

Still, the process of extracting and building archives is the same in *File Roller* as it is in *Ark*, though the functions that *Ark* locates in the Action menu, *File Roller* files under Edit. Also remember that the Add icon on the tool can only add individually selected files: to add folders you will need to do Edit > Add a Folder... There are also options for cutting, copying and pasting as well as the ability to rename files within an archive (Edit > Rename). There is an option to rename the entire archive under the Archive menu.

This application doesn't have its own viewer applets, so selecting the View option will launch the application associated with a particular file type.

There are also a couple of security options available through *File Roller*: One you've created an archive you can test the integrity of the compression method to ensure everything is as it should be. Do Archive > Test Integrity and wait for confirmation that everything is OK. You can also encrypt your archive and add a password: this is easily done through the Edit > Password entry. When you come to extract the archive, you will be prompted for the password – choose something memorable, as an inaccessible archive is as bad as none at all.

Remember that, as with *Ark*, once a file is deleted from an archive it's gone forever. However, if you use the Move To Wastebasket option under the Archive menu, the whole archive will be moved to the wastebasket, and so is retrievable at a later date. **LXF**

NEXT MONTH

I'll be stripping away the mysteries of the Linux printing system. I'll also try to find space to cover network and PDF printing issues.