Inessential Document Formats[1]

Version 0.1

/afs/sipb.mit.edu/project/doc/idocformats/iDocs.dvi

The Student Information Processing Board

Omri Schwarz <ocschwar@mit.edu>, Camilla Fox <cfox@mit.edu>

November 7, 1999

Contents

1 Introduction

The aim of this document is to list the file formats you will most likely encounter as you use the Athena working environment, the purposes they fill, and how to convert from one to another.

1.1 What do I gain from knowing about document formats?

You won't find yourself hours from a deadline and stuck with documents given to you by your classmates or professors that you have no idea how to read, use, or edit. You will be able to convert documents you receive in inconvenient formats to the formats with which you will grow accustomed. You will be able to select the most appropriate file format for the work you generate.

1.2 Identifying what format you have

Suppose you have received a file named `foobar.baz`, and you aren't sure what format it's in. An easy approach to finding out what format it is, is to run `file foobar.baz`. This should tell you what kind of file you have, and what format it is in. The `file` command uses a common convention known as "magic numbers" to identify file formats. The magic number is an element in most file formats that is agreed upon as a mark of that format. For a list of the formats that `file` knows about, look at the file `/etc/magic`.

What the file is actually named need not have relevance to its content, but it's usual for many formats to use a particular suffix to the file name. It's a good idea to stick to the usual suffix, since it'll matter to some viewers, and keep you from getting confused.

1.3 Choosing what format to use

What format you will want to use or to convert your document into will depend both on how to produced it, and the expected uses of the end form. For example, if you intend to go back and edit your file again, it is likely that you will preserve the most information by leaving it in the format native to the program you were using. For example, FrameMaker's document format is useful for editing, but only to those who have access to FrameMaker. If you want to give your work to an end-viewer, who can only be presumed to have basic platform independant tools, PostScript and PDF formats would be good choices, since they can be printed and viewed by almost anyone, but they are not good for re-editing. Some formats preserve more information about the content, than just what should be displayed on the page. Formats such as these for example, HTML, LaTeXand roff, can be edited easily, but may be displayed differently by every person, and converted into printable formats in a variety of different ways.

1.4 Other sources of information

Because of its scope, this document makes certain assumptions about your knowledge of UNIX. If you want to learn more about UNIX, there are a variety of sources of information. Man pages should exist for all of the commands used in this file (if they don't, it's a bug, which should be reported using `sendbug`). Other SIPB Inessential guides and the publications from IS are also great sources of information; SIPB Inessential Guides can be found at `http://www.mit.edu/sipb/docs.html` or `/mit/sipb/doc/` and you can visit `http://web.mit.edu/is/pubs` for a listing of IS publications.

2 Common Formats for Printed Documents

For books, papers, tracts, instruction manuals, lab reports, theses, and so on and so forth, Athena users commonly use these formats:

2.1 PostScript

Adobe©PostScript©is a language used for the layout of both text and graphics. Among other uses for PostScript, most large networked printers, such as those in the Athena clusters, expect their input to be in

PostScript format. Whenever you print anything, such as a web page, a PostScript document is produced and then sent to the printer. For this reason a large number of programs are capable of outputting postscript files.

PostScript files usually have the filename suffix `.ps`. To view a PostScript file, you can use the program `ghostview` in the `gnu` locker (`add gnu; ghostview &`). To print a PostScript file from a command line, just run `lpr file.ps`, where "`file.ps`" is of course, the file.

The fact that a PostScript document consists simply of ascii text containing data and commands (PostScript is in fact a Turing–complete programming language) is convenient in many ways, and allows for many nifty features. It also means that if your PostScript file is corrupted in some way, you have a very good chance of being able to fix it.

Encapsulated PostScript (suffix `.eps` or `.epsi`) is a particular subset of the PostScript language, and also a standard for including PostScript in other documents, such as LaTeX and Frame documents. It excludes commands which alter the page layout in drastic ways, so that the document can reliably be included within a larger postscript document. Some programs, such as `xv`, write Encapsulated PostScript by default, while for many others, you need to use a utility such as `ps2epsi` to convert to Encapsulated PostScript. One problem that you might meet with Encapsulated PostScript is that nothing at all is displayed when you try to print it. If this happens, try adding '`showpage`' to the bottom of the document.

If an Encapsulated PostScript document causes problems, such as blanking a page after it has drawn it, that's an indication that it isn't properly encapsulated. One thing you can use to clean up such postscript is `pstoedit` which is in the outland locker. The usual command line would be `pstoedit -f ps oldfile.ps newfile.ps`. This has the effect of capturing the output calls of an interpreter running over your postscript. Note that it will output non-encapsulated PostScript, so you will want to run `ps2epsi` over it.

In the postscript locker (`add postscript`) are several utilities for printing PostScript files more than one per page, such as `psnup`, and `epsmerge` and manipulating these files in other ways. Any time that you are using an application that doesn't give you the flexibility to print in the format you want, you should select the option "print to file" (rather than "to printer") and specify a file name. This will give you the file in PostScript format, which you can then manipulate with other utilities before sending it to a printer. This technique is also useful if you are using a windows application, and wish to be able to view and print the file on athena.

If you need to extract the text from a PostScript file, in the consult locker is the utility ps2ascii (`add consult; man ps2ascii`). You should note that there's no way of converting PostScript into a format that you can edit, without losing the formatting information.

For more information, Adobe's website, at `http://www.adobe.com/prodindex/postscript/main.html` might prove interesting to you. You can also find a lot of examples of hand written PostScript in the postscript locker.

2.2 DeVice Independent (DVI)

This is the file format used for the output of the LaTeX typesetting program (suffix `.dvi`). These files can be printed or converted to PostScript with `dvips`, or viewed with `xdvi`.

You can give `dvips` options to make it rotate its output (landscape) or use another paper size, as well as reordering the pages.

2.3 Portable Document Format (PDF)

Adobe©Portable Document Format is the product that Adobe is now promoting over postscript. It shares many of postscript's properties but is somewhat more common on the web, because it's more often readable from windows machines than PostScript is.

Unlike PostScript, while PDF comes in an ASCII format, it's much more common for it to come in a binary form, which is considerably smaller than a PostScript file containing the same information. Another important differance between PostScript and PDF is that a PDF document may contain hyperlinks to other parts of the document, or even to URLs on the web.

You can view and print PDF files using acroread (`add acro; acroread &`). Also in the `acro` locker, `distill` will generate PDF from PostScript, while `ps2pdf` and `pdf2ps` in the `gnu` locker also do conversions.

Also, the consult locker's `ps2ascii` utility extracts the text from a PDF file. Using this to extract the text, then editing that, is usually the only way to edit the text content of a PDF file.

2.4 LaTeX

LaTeX is a typesetting language used to write lab reports, theses, textbooks, and things such as this document. When the time comes for you to deal with LaTeX documents, you can pick up Inessential LaTeX from the SIPB. The command `latex` processes LaTeX files (suffix `.tex`) into DVI files (see above). Your LaTeX work will most likely begin with some lab report, for which you will quite likely generate figures using Matlab. The `print` command in Matlab lets you generate an Encapsulated PostScript file (with the option `-depsi`) which you can then import into LaTeX . A brief note you should recall is that there are two main versions of LaTeX (Latex and Latex2e) and they differ by a few details. Versions of LaTeX exist for both of these so you can make your DVI files.

2.5 HyperText Markup Language (HTML)

HTML is probably familiar to most people as what web pages are written in. It provides simple markup of text, which can either be logical, or oriented towards display, as well as providing the facilities to embed images and links. Later versions of the specification also allow such features as frames and cascading style sheets.

You can edit HTML with any text editor, as well as several more fancy web page creation environments. If you use emacs, check out the font lock mode for HTML, which will highlight tags and help you catch many errors. HTML is often a good format to choose if you want to add simple formatting to a text file, and print it out. For larger projects that don't require a lot of links, however, you'll probably be happier with something else.

One thing to keep in mind when editing HTML is that there are things which will display in your web browser which are not strictly HTML, and what looks good in your browser may not look the same in someone else's. For this reason, it's a good idea to make sure that your pages are strictly HTML compliant. You can visit `http://www.w3.org/` for tutorials, specifications, and verification services (this is a service which will tell you how compliant your webpage is). A tool on Athena to help you fix up unsightly HTML is `tidy` in the outland locker.

2.6 Standard Generalized Markup Language (SGML)

SGML is not a format in itself, but it deserves mention here because it is a format for defining a markup language. A DTD, or Document Type Definition, defines what markup is allowed and what is required. A document is then written with markup that follows the rules set by the appropriate DTD, and some other program uses the DTD, the document and some set of display rules to generate output in some desired format.

SGML is often used to get around the limitations of HTML. Because the markup in SGML is usually descriptive, rather than procedural, more information is maintained in the SGML copy, and is used to output the content in a variety of formats, including HTML.

For more information on both SGML and XML (a similar concept) see `http://www.oasis-open.org/cover/`

2.7 FrameMaker and Other WYSIWYG Editors

Frame Maker (`add frame; maker &`) is Athena's most often used word processor. It has its own file format. (We advise you to use the suffix .fm.) Other word processors used in Athena are Applix and StarOffice, each with their own formats. Frame Maker can read several kinds of files. To see, start a document, click on the file menu, click on import, and then look at the long list of available formats. To take a look at Applix's range of available file formats, look at the dialog box *File* → *Import.* These are the document formats it imports. Applix also deals with graphics formats (to see which ones, click on *Insert* → *Object from File* → *Other...*).

In general, if you plan to edit the document again, save a copy in the native format of the program you are using, and if you will need to share it with someone in an electronic form, or view it when you aren't on Athena, output a copy in something portable such as PostScript, PDF, or RTF, as well as keeping the origional.

2.8 Microsoft Word

You may encounter or receive documents in one of Microsoft Word's file formats. Among the problems you may encounter are that there are numerous Word formats out there, and more being used with every new version of Word. The utility mswordview (add consult; mswordview &) supports some formats, as do FrameMaker and Applix (`add applix; applix &`), but there will always be more formats than are supported on Athena. If someone is sending you Word files, it is best to ask him or her to change the files to some other format and save yourself much trouble. If he or she cannot, ask for a version of the document in RTF format (Rich Text Format), which is the Word format easily imported into Athena's main word processors.

2.9 Rich Text Format (RTF)

Another Microsoft product, RTF was deseigned with portability in mind (only portability between Microsoft applications, but still), and it is thus less hard to handle than many other Microsoft formats. It is basically an encoding of text, formatting information, and graphics, as ascii, so if you open it with a text editor or viewer, you may be able to make some sense of it.

It's also the format to choose if you want to output a document from Microsoft Word, or almost any other Microsoft application, and edit it on Athena, since FrameMaker and others will read it.

3 Graphics formats

The section outlines the file formats used for generating images. They can be generated by numerous MIT software utilities in the graphics locker (`add graphics`), such as XV, Xpaint, and many others. (That is, `xv, xpaint ..`) They can also be generated by the GIMP (`add gimp; gimp &`), and the ImageMagick utilities.

3.1 Graphics Interchange Format (GIF)

GIF files (suffix `.gif`) use the LWZ compression algorithm for their format. They can be edited to label one color setting as transparent, and to create animations, and for that they are popular among web page authors. The compression algorithm, however, is patented by the Unisys Corporation, and this may in the future cause problems. GIF files use a colormap to encode 24-bit colors into 8 bits, so the format is well suited for images with few distinct colors. One form of the GIF format, GIF-98a, allows for animations in the form of a sequence of images. This makes the format popular on the Web. To make animated GIFs, you can use the GIMP, or you can use command-line utilities from the graphics locker, such as `gifsicle`.

3.2 Tagged Image File Format (TIFF)

TIFF (suffix `.tiff`) files may also be found in many web pages. The format is also used by services that forward fax messages into email. Since the format allows for multiple pages, you may sometimes need one of the other pages of a TIFF file when `xv` only shows one. The utilities `tiffinfo` and `tiffsplit` come in handy (`add graphics`). The former displays a list of the pages of the file, and the latter splits a multi-paged TIFF file into several of one page each.

3.3 Portable Network Graphics (PNG)

The PNG format (suffix `.png`) is a good alternative to the GIF format, and allows the same functionalities and then some, but without any patent problems. The specification and features for the format are listed in `http://www.cdrom.com/pub/png/png.html`.

3.4 Joint Photographic Experts Group (JPEG)

The JPEG format (suffices: `.jpeg, .jpg`) uses cosine transforms to obtain very high compression. The higher the degree of compression, however, the higher the loss of detail. This format is often used for scanned or digitally photographed images. It is best suited for images of many distinct colors.

3.5 XCF

This format (which comes from Berkeley's eXperimental Computing Facility), saves all of the intricate details (such as layers) which you may put into an image for future editing if you use the GIMP.

3.6 Other formats, and going between formats

`xv` lets you convert in and out of the image formats listed above, as well as several other formats you may encounter, such as PBM, PGM, PPM, XPM, BMP, Sun Rasterfiles, IRIS RGB, Targa, FITS, and many others. The GIMP also is useful for that purpose. However, the resolution and quality of the converted images is dependent in the such things as your monitor screen, especially if PostScript is involved. You will get better results using the various command line utilities in the graphics locker. One of the better utilities is `imconvert`. To see all the things you can do with `imconvert`, read the man page. For example, to convert an image named foo.pbm to GIF format, the command would be `imconvert foo.pbm gif:foo.gif`. This utility also performs a plethora of image processing manipulations. You can also use xv as a command line (only) tool. You can find a copy of it's extensive documentation on Athena in `/mit/graphics/src/xv/docs/xvdocs.ps`

4 Documents over email

4.1 Multipurpose Internet Mail Extensions (MIME)

MIME is the standard for attaching various kinds of files to email messages (as described in the RFC 1341). There may be several different parts to a MIME encoded message, each with a different content type and encoding. You should use mime whenever you need to send a file which isn't simply ascii text.

In the default configuration on Athena, `show` will deal correctly with mime attachments, and either display the file, or prompt you to save it. Likewise, `exmh` is capable of decoding attachments, although printing them from `exmh` often proves tricky (it's probably best to simply use `show` if you need to print the contents of the attachment).

To send attachments, `exmh` is usually the simplest tool. Begin composing a peice of email, and then look in the menu under *More...* → *Attachments*. If you want to do it from the command line, you can use `mimencode` from the mime locker.

4.2 uuencode

This is one of the commonly used encodings for email attachments. Uuencode takes each 3 bytes of input, splits them into 4 6-bit characters, and makes 4 bytes of output of them so that they are all printable characters. The commands `uudecode` and `uuencode` are good for manipulating files into and out of the format.

4.3 base64

This is another encoding format. The command `mmencode` encodes and decodes data with it

5 File Packages and File Compression Schemes

Sometimes people need to exchange large numbers of individual files, and often it is convenient to the recipient for the files to end up stored in the same directory structure as that used by the sender. Software packages are one example, but it has also become common for people to exchange such packages of HTML files (e.g. when they arrange the replication or transfer of a Web site) and LaTeX files (e.g. when they work on large-scale publications). Also, people often use packaging for storing regular backups.

5.1 Tape Archives (TAR)

Tape archives, noted by the .tar filename suffix, (computer types refer to them as "tarballs") use a simplified file system format originally meant for tape drives (hence the name). The advantage of this format is that it preserves the directory tree structure of the files it stores. Tape archives are assembled and disassembled using the very versatile `tar` command. The manual page for `tar` provides the information you need for more special circumstances, but most commonly you will find yourself using it for the following two purposes:

When you receive a tarball named "foo.tar", you can place it anywhere (say /var/tmp/foo.tar), change your current working directory to where you want to the root of the directory structure to go , and then type `tar xvf /var/tmp/foo.tar` The argument `''xvf''` means "eXtract, Verbosely, the File I specified."

To make a tape archive of the files in one directory (foo) and all directories below it, `cd` to the directory right above it, and then run `tar cvf /var/tmp/foo.tar foo` The arguments "`cvf`" mean "Create, Verbosely, the File I specified." In /var/tmp you will then find the tape archive, which you can attach to email, move elsewhere, post on the Web, et cetera.

5.2 DOS Executables (EXE)

Sometimes people use "self-extracting archives", which are Microsoft DOS executables. When a user runs one of these executables (on a Microsoft DOS or Windows machine), it writes out data from its own contents into files and directory structures which were specified by the writer.

There is no place on Athena where you can run DOS executables. You will need a Microsoft Windows or DOS machine to do so. We advise you take care before doing so. What you receive advertised as a self extracting archive may or may not actually be one. It may be a virus (several such viruses are circulating around the Internet, propagating themselves through email attachments and advertising themselves as Windows screensavers), or a Trojan horse. If you receive one by email, send a reply asking for confirmation that the sender really did intend to send you this archive, and then and only then run it. As a matter of rule, do not accept self-extracting archives from strangers.

5.3 Compression Types

Tape archives and file packages are usually large, as are many other types of files, so people compress them. Some common compression schemes are `gzip`, `zip`, and `bzip2`. Of these, `bzip2` (suffix `.bz`) gets extremely good compression but is newer and isn't widely used, while `gzip` (suffix `.gz`, or, when it's a tar archive that is compressed, `.tgz`) is the standard across UNIX systems. For windows, `zip` (suffix `.Z`) is the most commonly used (and the only one you can count on people having access to).

- Use `gzip` and `gunzip` from the sipb locker, for `gzip`.
- Use `compress` and `uncompress` for `zip`.
- Use `bzip2` and `bunzip2` for `bzip2`.

Remember that compressing things more than once is unlikely to improve your compression much, nor will you get much benefit from compressing a file for which compression is already built into the format, such as a jpeg or mp3. If you are compressing with gzip or bzip2, you can use a command line option, such as `-9` for it to compress better, at the expense of taking longer to compress.