FileMan - A Complete Guide

FileMan - A Complete Guide

Simon Edwards

Version 1.1

Announcing "System Administration through Shell Programming"

"FileMan" is not the only project I have been working on... I have almost completed a book! Please do not judge the rest of this text as a judge of the quality of the book - see that for yourself via a download.

The book is aimed at intermediate experience system administrators, or system administrators with limited shell programming knowledge. It uses the Korn Shell to explain how scripts can be used to automate many tasks which would otherwise take precious time. If you have ever wondered whether a particular labourious task could be automated, this could be the book for you!

The book is still in draft and suffers from quite a few spelling and grammer mistakes - but the ideas and text are mostly there. I am still modifiying it, but am really looking for a publisher before completing the work, so if anyone has any interested in the book, please email me at the below address:

filem@bongo.demon.co.uk

The book, currently available as a "zip" format file can be downloaded into a web page by pointing you browser at the page indicated below. **Note: This is a Word 6 document (unfortunately) - it may be converted to a "lout" document if I really must!** Once the file has been downloaded, using the "Save As.." browser option and save to a file which can then be uncompressed and loaded into "Word 6".

http://www.bongo.demon.co.uk/script.zip

Simon Edwards
Primary FileMan Developer

About this book

This book has been written in an attempt to help the general public install, use and configure **FileMan** to ensure maximum benefit.

This text also aims to provide enough information on the various platforms which are supported to allow a useable, customised configuration to be readily achievable.

Further more, this book is an attempt at using a text processing language, rather than a word processor. In fact, the results are encouraging - since it means one more reason to ditch "NT" off the machine! The language I am using is not the usual "LaTeX", but rather a language called "lout" - thankyou **Jeffrey H. Kingston**!

Contents

Introduction

FileMan has been around in various guises on various systems for a number of years. It originated as a simple environment to display a tree based widget structure. This was before the times of such widgets were widely available - going back over three years. I wrote a small program to display an arbitary directory structure using the algorithm written in the "Tcl/Tk" language.

I then spent some time expanding on this to produce a very basic file manager which was entirely constructed in the "Tcl/tk" language, and this worked fine. However, there were two problems:

- It was quite slow
- It did not have any machine specific features

Yet it was useable, and thus I spent quite some time developing it - originally aiming to create a commerically based product, (it was being developed on UNIXWARE, HPUX, DRS/NX, among others.

After a year or so had passed things had changed greatly. I had left my previous employment, and thus the rights to FileMan had passed beyond my control. Yet I still spent time developing FileMan for my previous employer on a part-time basis for at least six months. After this time they decided to cease development since the returns from the product would still take time to appear.

Yet I knew the value of the product and thus continued developing it mainly under Linux. I also maintained portability with AIX version 4.x since I had access to a development system in my new employers.

Last year I released the Beta version onto the Internet, which met with mixed views, the main problem being the lack of documentation regarding how to install and use the software, and the occasional stability issue. However, I decided to keep developing because of all the file managers I had tried, none offered the level of functionality, configurability, or security that I required.

Currently the version of FileMan which is available contains many more enhancements compared to the original beta version. Many more of the commands have been converted to "C" to ensure improved performance. More items can be configured, and the look-n-feel of program has been greatly improved.

Towards the end of 1997, (just a few weeks ago), I released the official version 1.0 of the product, initially for the "Linux" operating system. When this document is complete, it will become part of the 1.01 version which should hopefully be simultaneously released for "Linux", "Solaris" and "Aix" operating systems.

The future holds many changes to FileMan, including the ability to manage and process on remote machines and file systems. This will be done in a secure manner to ensure that product can become useable in a hetrogeneous, distributed UNIX environment. I am continuing to look at ways of improving the software, and also looking for further platforms for porting, and thus if you have any comments, or available environments, please contact me!!

Hope you enjoy the software!

Simon Edwards
Primary FileMan Developer

Chapter 1. FileMan Installation

This chapter details how to get FileMan installed using the standard installation options. It will highlight any particular requirements for the various supported platforms.

1.1. Who is this chapter for?

Since FileMan is aimed for many levels of user, it might be reasonably safe to expect an installation procedure which could be classified as straightforward. As of version 1.0 this is exactly the case - however, it should be noted that since this is a product that resides in the "/usr/local" directory structure, you will need root authority to create the subdirectory required.

Of course, FileMan does assume some kind of knowledge as far as installation is concerned, these basically consisting of:

- Some UNIX System Administration Experience
- The ability to following written instructions

As you can see, these pre-requisites are very modest! Now you should know whether you personally can install FileMan, and if not, who you should approach to get the product installed. The next stage is to ensure you have the latest available version...

1.2. How to get the latest version

The first thing to check is to ensure that you have down loaded the latest version - if you have not, you may be missing out on the some bug fixes, or additional features. Hence the first thing to be done is to check the latest version that is publically available. For this information, check the following Web Page:

htpp://www.bongo.demon.co.uk/fileman

This web page should contain information of which versions are currently the latest for all supported architectures. Specific information should be available for all supported platforms, and pleas for platform environments to port to other machine types. The important information to confirm is the latest version, and if you have not got the latest version for your architecture, then the following site information will be relevant to retrieving it.

Operating System (Architecture)	Primary FTP Site
Linux 2.x (Intel)	ftp://sunsite.unc.edu/pub/Linux/X11/desktop
Aix 4.x (PowerPC)	ftp://ftp.ibm.edu/pub/contrib/X11/desktop
Solaris 2.x (SPARC)	ftp://???/pub/X11/desktop

From the previous table the required directory should contain various files which make up the distribution for a particular architecture. Note the actual file names which make up an distribution for a particular architecture are in one of the two following forms:

- fileman-*X.Y*.tgz
- fileman-*ArchitectureX.Y*.tgz

Note that if the name of the distribution does not contain the name of the *architecture*, then it will be stored in a architecture specific directory structure. In the above descriptions *X.Y* represents the release version of the distribution. Please try to tie this as closely as possible to the latest distribution as noted on the web page. At time of writing this is version 1.1.

From version 1.1 there are four different files in the distribution:

1. The "README" file
2. The "LSM" file (or equivalent if necessary)
3. The "DISTRIBUTION" file
4. The "MANUAL" file

It is obviously recommended that all of these files be down loaded. The README file should contain the relevant information for the unpacking of a particular archive, but more on that in later sections.

Of course, the fact that you are now reading this indicates that you may have already down-loaded the manual. However, the README documents are essential downloads for any architecture, since they contain the very latest information.

Finally, if you are unable to locate a distribution for a certain architecture, please contain the auther using the following email address:

filem@bongo.demon.co.uk

1.3. Requirements and Unpacking

Once you have the latest versions of the software to install we can actually start to think about any other requirements. First please ensure that the distribution files are moved to a temporary directory, such as "/tmp". The first requirement is based on the structure of the archive in which the distribution arrives - "gzip"[1]. This program is now as popular as the standard "compress" command, but is not yet a standard feature in commerical UNIX environments.

Thus the first test is to ensure that the "gzip" command is installed, which can be done by

[1]"gzip" is GNU Software

running the following command. Note, this expects the current shell to be either the "Bash" shell, or the "Korn" shell.

```
# which gzip
/bin/gzip
```

If this command produces an error, this is a clear indication that the "gzip" utility is not installed. Thus, before installation can preceed please locate, compile and install the relevant "gzip" distribution for your machine.

Once the relevant "gzip" has been installed the next step is to actually unpack the software. This is done by using the following command. Note that rather than unpacking the distribution first, and then extracting the files, we pipe the two commands together. Remember that the following commands must be run as "root". Type all the commands at the "#" prompt to ensure the correct extraction of the files. **Note: If you already have a version of the software installed please follow the instructions given in the next section.** The commands below presume that the distribution to install is called "fileman-linux.1.1.tgz", and this is temporarily installed in the "/tmp" directory.

```
# mkdir -p /usr/local/fileman
# cd /usr/local/fileman
# gunzip -c fileman-linux.1.1.tgz | tar xf -
# chown -R root.bin .
# chmod -R 775 .
# ln -s /usr/local/fileman/ufm /usr/bin/ufm
```

The above commands will create the required installation directory, unpack the files in that directory, change the permissions to ownerships to ensure no one without "system" type permissions can actually over-write the newly installed files.

Security Configuration

FileMan supports a security feature that allows you to configure which users from which systems can actually use the software. Although this could have been implemented more elegantly for each of the different versions of UNIX supported, it would not have been cross-platform. For this reason, we must configure secuity a configuration file.

From version 1.1 onwards the default configuration has the security feature turned off. The main reason behind this is that most of the questions asked have been due to the fact that this file has not been updated for your users, or has been configured incorrectly. Thus, if you don't mind anyone using the software, this section can be ignored.

However, use the following example of how to add a user to the configuration file if you really want to do things that way:

```
# echo root, <OS>,$(name -n),$HOME >>
> /usr/local/fileman/Config/admin/users
```

Licensing

The current versions of the software have license functionality built in. However, this is not used, so no other configuration should be necessary to make the program usable. Note: the above command assumes that *<OS>* is replaced by the name of the current operating system, which must be one of the following:

OS Value	Actual Operating System
LINUX	Linux versions 1.0.13 -> 2.0.x
SUNOS	Solaris Version 2.51 -> 2.6
AIX	Aix Version 4.1.4 -> 4.3

1.4. Installing a Later Version

You may wish to skip this section if you are doing the first installation of "FileMan" on your system. This section details additional steps to be undertaken if you already have an older version of the software already installed.

If you have an older version of the software installed, there are two different cases which must be dealt with:

- Installation exists in "/usr/local/fileman"
- Installation exists in another directory

In the first case the best case is to rename the directory rather than replace it unless, (for whatever reason), the newer version fails to function once installed. This can be done using the following commands. Note this also moves all the start-up configuration files from the root user's home directory to "/tmp" if they exist. Again, this allows the files to be restored later if the newer version does not work.

```
# mv /usr/local/fileman /usr/local/fileman.old
# cd $HOME
# for i in .filemrc .filemst .ufm_*
> do
> [ -f $i ] && mv $i /tmp
> done
# [ -f /usr/bin/ufm ] && mv /usr/bin/ufm /tmp
```

Once this has been done, the next stage is load in the new version as described in the previous chapter. Although it is likely that the previous user startup files are compatible, it is probably best to actually prove that the new software actually works before copying in the old startup files. In that way, it is easy to determine whether the new software works at all, and if it works with the older startup files.

In the case where the old version of fileman is not in the "/usr/local/fileman" directory, the the first stage is to install the latest distribution, as described in the earlier section and then use the commands in the previous box to move copies of the configuration files to "/tmp". Note: you may wish to not run the final "ln" command, since this may only be done once an existing link or file has been removed.

Now you are ready to test the new installation!

1.5. Basic Installation and First run

So, the installation is complete[1], now the next part is to check if it can be run correctly. Since you should now have a link in "/usr/bin" for the "ufm" command, you should simply be able to run the following command:

```
# ufm
```

This will obviously require you to be running Xwindows at the time. If this results in an error indicating the command has not been found, it indicates that one of the previous sets of commands in this chapter have not been completed successfully - please review the commands given so far, and try to spot and correct any commands that have not typed, or typed incorrectly.

Presuming the command has been found, "FileMan" should now be starting. Any versions currently available, (1.0 and later), support automatic user configuration default file generated, and thus, when the software is ran for the first time, you may get similar output to the following extract:

```
Creating .filemrc...Done.
Creating .ufm_types...Done.
Creating .ufm_pixies...Done.
Creating .ufm_menus...Done.
Creating .ufm_conf...Done.
WARNING: Trashcan directory did not exist, creating.
Creating Audit Log Directory...
```

These messages are simply for information only. If everything is OK, the application should pop to life on your screen. This does take some time depending on the hardware. A welcome progress window will be should whilst the application configures and loads itself.

If the application gets this far, this is good enough to indicate that the basic installation has been successful. You know may wish to use older configuration files if you saved them during the upgrade process described earlier. If the program does not run once they have been replaced simply remove all the configuration files and run the program again - this should re-create the relevant configuration files as required.

If this fails to work, hopefully any error messages which are output should help you to pinpoint the area the program files. If this is not the case then capture the output of the attempt in a trace mode, and email the results to the previously shown email address. To run the application in trace mode simply requires the following command:

```
# ufm -trace 2>/tmp/trace
```

[1] I told you it was simple!

At this point the file "/tmp/trace" can be sent to me via email for diagnosis - though hopefully this will not be necessary!

1.6. Tuning the Basic Installation

This section will not really go into any great detail regarding tuning of the configuration for a particular platform, but will explain a little of the basics about what can be changed. Many changes are trivial, though some are best left to the immagination!

In fact, the remaining chapters in this short book will describe many of the aspects of configuration introduced here in great detail. However,

1.7. Configuring access for other users

Although it may seem a little long-winded having to update the security file for every use that requires access, it is worth it since it allows you to control access at any level, independant of whether your particular operating system supports Access Control Lists (ACL's). If you are looking for a quick and dirty way of ensuring everyone on the machine has access, you could always use the following script:

```
#!/bin/ksh

# Re-create the security access file for FileMan
# Simon Edwards, 1997.

ARCH=$(uname | tr a-z A-Z)
SECFILE=/usr/local/fileman/Config/admin/users
>$SECFILE
awk -v FS=: '{print $1}' | while read x
do
   eval homedir=~$x
   print $x,$ARCH,$(uname -n),$homedir >> $SECFILE
done
```

This small script should solve most of your problems - run from "cron" if required to ensure the security file is updated on a regular basis. **Note: This has only been tested on Linux, other varients may require a little tweaking, since the "uname -n" may not give the expected output.**

Using the above program may look like a good idea - however, this program is only useful if user's from a single machine access this program. If the installation directory is NFS shared between other machines, (which is supported), even different architectures, such an updating script would have to be a bit more exotic in nature - though of course this is possible.[1]

[1]Indeed it would be nice if the future distributions included such a series of programs.

1.8. Common Problems During Install

Although I am reasonably certain that this installation guide is relatively complete, there are always possible problems. Below are a few situations which might arise, and possible results to resolve them.

1.8.1. "FileMan" runs, but lots of messages on screen

Some of the distributions were compiled with debug output turned on! This will basically mean that it will produce lots of meaningless messages on the screen. The easiest way to resolve this kind of problem is to run "FileMan" using the following invocation:

```
# /usr/local/fileman/ufm >/dev/null
```

This will resolve the problem in the simplest form - hopefully when a later release is made available this will "feature" will have been turned off!

1.8.2. Type the command to start "FileMan", but nothing happens

This could be many problems, but first you must make sure the command you are typing is actually references the correct program. If all the installation instructions where followed, you should be able to replicate the command and output...

```
# ls -l /usr/bin/ufm
lrwxrwxrw 1 root root 22 Nov 29 01:45 /usr/bin/ufm -> /usr/local/fileman/ufm
```

If the output is different, or an error message occurs, then run the following two commands as "root" to ensure the relevant link is available.

```
# rm -f /usr/bin/ufm
# ln -s /usr/local/fileman/ufm /usr/bin/ufm
```

Now please try to run the command again. If this still does not work, or the link was correct then do not give up hope, there are still many things to check... The next being the installation location of the files. Try running the following command - the output should also look very similar:

```
# ls /usr/local/fileman
3Dpixmaps  Config    Docs    bin     etc
fsystems   helptext  html    libs    pixmaps
src        src-tcl   ufm
```

If this is not the output, then more than likely you installed the files in the wrong location. Please refer to the previous instructions and try to install the files again. **Note: The commands used during the install, include a "cd" command, which is vitally important.**

The final check is to ensure that the required shell is available. All major scripts associated with "FileMan" require use of the "Korn" shell. If this command has not been installed, then install it. Linux users may not have an installed "Korn" by default, but are quite likely to have a "Bash" shell. If this is the case. if you do not have a "Korn" configured, then run the following command, since the scripts are also "Bash" compatible.

```
# ln -s /bin/bash /bin/ksh
```

1.8.3. "FileMan" starts to load, then aborts

This could be any number of things actually! As of 1.01, if it fails in this manner, it should at least print some indication why it fails - there are also quite a few return codes, which will be shown along with a suitable reason in the table below. If the particular error does not appear in this table, then please email me at the address shown in the earlier pages. Please include the complete output of the failure, and information on the platform and version - I will hopefully resolve any problems, and update this document if necessary. **Note: Most errors are very similar to the samples in the pages, so should produce an error message that helps you resolve the problem.**

To get the error exit number, please run the following command immediately after the program has been run:

```
# echo $?
```

So, some sample errors codes, and how they can be resolved...

Error No.	Actual Possible Cause & Resolution
27	This is again due to a missing variable in the ".filemrc" file, please ensure that an entry exists for the "ufm_use_magic" variable.
28	The required variable for the internal representation of menus has been removed. Please check the ".filemrc" file in your home directory. If necessary copy the file from the "/usr/local/fileman/Config/GLOBALS" sub-directory appropriate for your architecture.
60	Another error indicating that the values in the startup file are not complete - set the "SORTING" variable in the ".filemrc" configuration file before trying again.
61	Again, another variable setting in the startup file! This time the variable in question is the "FILE_FORMAT".
133	This means that the program has been unable to find or create the required directory for the auditing logs. Please check to see if the directory specified in the "AuditLogFile" exists, and if not, either create it, or change the setting in the "/filemrc" file in your home directory.

1.8.4. "FileMan" uses different tcl/tk libaries than I installed!

Do not worry! "FileMan" binaries use of dynamic libraries will come prepared with all necessary files in the distribution. Please be reassured that the installation files installs only the

"/usr/local/fileman" directory structure, and no files[1] are stored in any "system" directories.

Indeed, some versions of the distribution do not even include shared libraries for "tcl/tk" - these are statically linked by default. This is the case for the "AIX" distribution, for example.

Please note, simpy by running the "ldd"[2] command, the results will not be correct. For example, as root, the following is output when this attempted on a Linux system. **Note: The location of the actual binary executed is determined by the architecture of the current machine.**

```
# ldd /usr/local/fileman/src/LINUX/ufmx
  libtix.so => /usr/lib/libtix.so
  libtk.so => /usr/lib/libtk.so
  libtcl.so => /usr/lib/libtcl.so
  libX11.so.6 => /usr/X11R6/lib/libX11.so.6.1
  libm.so.5 => /lib/libm.so.5.0.6
  libmagic.so => /usr/lib/libmagic.so
  libdl.so.1 => /lib/libdl.so.1.7.14
  libXpm.so.4 => /usr/X11R6/lib/libXpm.so.4.9
  libc.so.5 => /lib/libc.so.5.3.12
```

Notice the locations of the libraries used for the "tcl/tk" libraries. However, notice the difference when running the following two commands:

```
# export LD_LIBRARY_PATH=/usr/local/fileman/libs
# ldd /usr/local/fileman/src/LINUX/ufmx
  libtix.so => /usr/local/fileman/libs/libtix.so
  libtk.so => /usr/local/fileman/libs/libtk.so
  libtcl.so => /usr/local/fileman/libs/libtcl.so
  libX11.so.6 => /usr/X11R6/lib/libX11.so.6.1
  libm.so.5 => /lib/libm.so.5.0.6
  libmagic.so => /usr/local/fileman/libs/libmagic.so
  libdl.so.1 => /lib/libdl.so.1.7.14
  libXpm.so.4 => /usr/X11R6/lib/libXpm.so.4.9
  libc.so.5 => /lib/libc.so.5.3.12
```

Thus, the required libraries are loaded from the installation directory, rather than the default libraries on the system.

1.8.5. Yellow box with icons appears near trashcan

This yellow box should appear when the mouse pointer is moved over the trashcan. However, rather than containing icons it should contain information regarding the status of the trashcan. This is a problem with font name resolution, and the solution if fairly straightforward.

[1]Obviously apart from configuration files

[2]"Ldd" is the command to list dynamic dependancies (i.e. shared object libraries

Since this problem really depends on the fonts available for each display in use, the first point is to decide whether to change for just for the current user, or for all users. Obviously this can only be done based on the how many users have this problem.

Changing it for the current user can be done by running the following set of commands.

```
# cd
# sed 's/:font(3)=*-12-*/:font(3)=fixed/g'
> .ufm_types >ufm_types.new
# rm .ufm_types .ufm_types.tmp
# mv .ufm_types.new .ufm_types
```

This will change the font used for the trashcan to the "fixed" font, which is generally available on all systems. To change this for for all users, you could repeat the above steps in each user's home directory, or a better approach is to change the template used for creation of this file, and then remove the current version of the file in each user's home directory.

Chapter 2. Using the Standard Installation - Part 1

So "FileMan" is now installed and we think it is working... But what can it be used for? What features are genuinely useful, and what functionality can you not expect in the default installation?

This chapter will be used to describe what "FileMan" can be used as soon as it installed. This will probably also show limitations which you may wish to remove - which will then lead on to the following chapters after the standard configuration functionality has been discussed in depth.

2.1. Who should read this?

FileMan is easy to use and much of the functionality can be used without having to read any documentation. However, this chapter provides an overview of the basic functionality. It is worth reading this since it may show you the principals and ideas behind the functionality of the program. The chapters following this one are more for reference, and thus can be referred to, if necessary.

For those who are interested in customising the application in any way, it is recommended that they at least be familiar with the functionality that "FileMan" already offers - there may be the feature you require already built in, but not immediately obvious. Chapter four onwards deal with various levels of customisation for those really wanting a challenge!

2.2. Invocation Options

Although it is possible to simply type the name of the command, there are some options that you should really be aware of when using the product. It is unlikely that in day-to-day use these will not be used, but someone might find a use for them...

2.2.1. Specifying the "X" display to use

"FileMan" does support the standard "X" feature of being able to run your application on the display hosted by another machine. This can be quite useful in networked environments. For example, if you wished to run "FileMan" on a machine called "fred", but were actually twenty miles away on a machine called "mike", then you could use the "-display" option to actually view the application on the screen in front of you, using the following commands (including logging into the remove machine called "fred":

```
<mike> # telnet fred
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.

Red Hat Linux release 4.2 (Biltmore)
Kernel 2.0.30 on an i686
login:  simon
password:

<fred> # ufm -display mike
```

I use this functionality on a day-to-day basis at work to use "FileMan" to use the display as a local PC. Remeber that X servers are available to run on a variety of platforms.

2.2.2. Tracing what "FileMan" Does

Imagine there is a problem with "FileMan"[1] - and you wished to report the problem to the developer, hoping for a speedy fix. However, it is very difficult to report where the problem is actually occuring, since it simply keeps crashing!

There is an option which has been designed to help - it turns on command tracing in the "tcl/tk" environment. To use this, run "FileMan" in the following manner:

```
# ufm -trace >/tmp/output
```

It is often necessary to re-direct the output from the trace to a file since the amount of output generated is **Huge**!

2.2.3. Removing all output messages

It is often possible for "FileMan" to generate some messages to the terminal, even when debug mode is turned off. Although this is entirely harmless it can become irritating! If this does start to bother you remember UNIX re-direction - you can simply run the following command to remove all messages from the terminal:

```
# ufm -trace >/dev/null 2>&1
```

2.3. Contents of the Main Display

Once the application has started, several windows will appear.[2] These windows are reasonably straightforward to use, but an overview certainly is worth while. Because the

[1] However unlikely this may seem!

[2] Assuming the default configuration.

contents of the window are highly configurable it may look slightly different that the example shown, but the functionality should still be closely related.

Firstly, consider the main window, as shown in the diagram following:

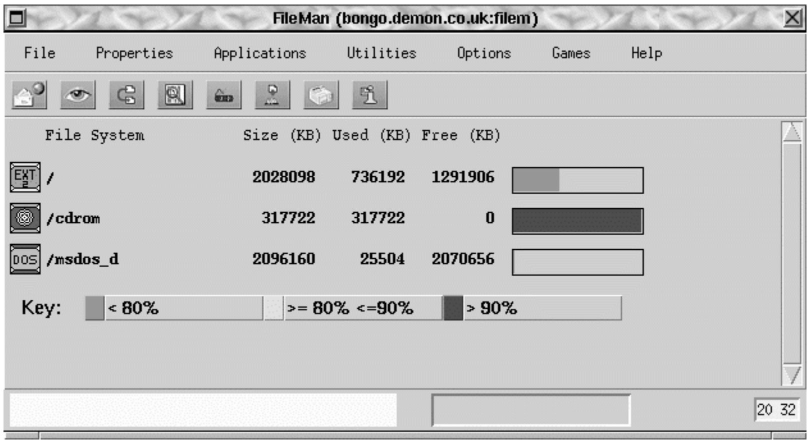

Figure 2.1. Main "FileMan" Window

As you can see, this main window presents the user with some useful information. The actual contents of the window can be broken down into a series of horizontal strips:

- A series of menu buttons
- A toolbar
- Disk Partition Table
- Progress Montior area

Each of these items has a clear purpose. The menus behave in a similar fashion to any other menus either on a PC or other "X window" based applications. **Note: The menu options are case sensitve, and this certain options may only be available at certain times (more on that later).** As usual, any menu which produces a requestor for further input has "..." at the end of the text - for example, "Delete Files...".

Below the menus are a series of button making up a "Toolbar". As you would expect the contents of the toolbar are fully configurable. Some of the buttons can be used in conjunction with selected files, others can be used at any time.

The most interesting and space consuming section is the Disk Partition information. This is the section where all available partitions[1] are shown. The details for each partition include the type, size (in kilobytes), and the amount of space used and available. To actually display the contents of a particular partition is simply a matter of clicking on the line representing any partition window.

The final section is the progress monitor area. This gives more information on what

[1]Well, those currently mounted anyway.

each button on the toolbar actually does, and also a progress bar for indicating the progress of performing window updates.[1]

2.4. Partition Windows

The contents of a partition will appear in a separate window when any part of the line for a partition is clicked on. The actual contents of the partition window really depends on many configuration options, but may look something like the following:

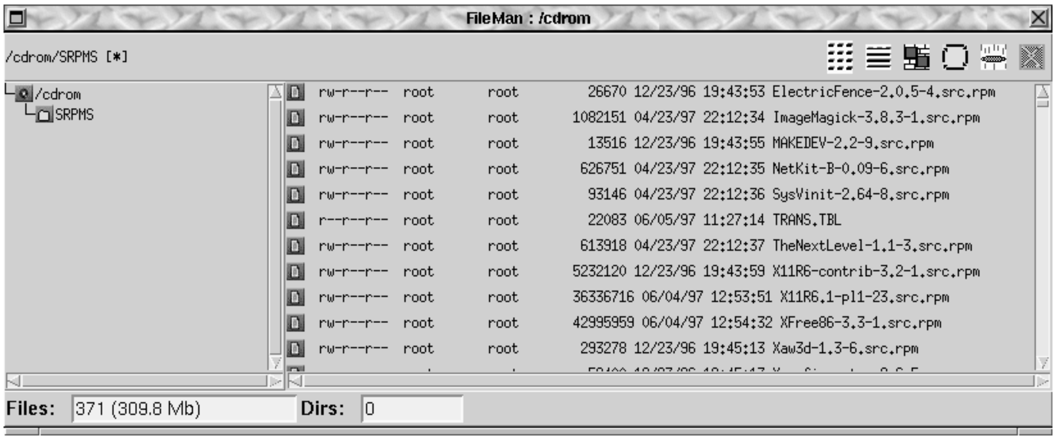

Figure 2.2. An Example Partition Window

Once a partition window has been made visible by clicking on the bar for a particular partition, you are ready to explore that particular partition. **Note: If you click on a partition in the main window, and the partition window is already available, it will "raise" this window, rather than creating a new one.** For those interested, it is possible to create multiple windows for the same partition - as shown in the next chapter.

The functionality of the window is separated into two distinct operations - which is tied to the scructure of the window displaying the partition. On the left of the window a directory tree is shown. Currently only a single click is implemented - **a double click may function, but not on all versions.** A left click on a directory will expand one level of directories within this directory, and this operation can be repeated on sub-directories to build a complete directory tree as required. If a directory contains many opened levels of sub-directories a single left click on the parent directory will result in any open sub-directories closing.

The majority of the parition window is allocated to displaying the files in the currently selected directory. The actual format and contents of the window really depend on many options, but the default will probably look similar to figure x. Both sections of the partition window should share the same size icons and font - there are three font and icon size combinations available.

Selecting files is a little more flexible than directories - though it all uses the left mouse button and a combination of keys if required. A selected file is made evident since the text is in blue rather than black, (i.e. slightly paler on grayscale devices).

[1]Window updates occur when any partition windows are currently open - the view on each partition is updated on a regular basis to ensure the contents reflect the reality of the current directory structure.

Single clicking on a file will select that file. This action will also de-select any other selected files in this, or any other, partition window. To select several files use the mouse button and the "CTRL" key. This can be used to select adhoc files. Finally, a range of files can be selected by selecting the first in the range, and then moving to the last file and using the "SHIFT" key and the mouse button. **Note: Which files are selected in a range is determined by the number of columns on the display - experiment!**

Once files have been selected, the menu options which required files to have been selected, magically become available! The only other basic input to the partition window is a double-left click which when used on a particular file will result in the default action occuring for that particular file.

2.5. Application Toolbox

There is an easy way of launching applications from within "FileMan", if this functionality is required .[1] The toolbox window allows you to place icons to represent any application you wish to launch. As you would expect the exact location, programs, folders and icons shown are entirely configurable.

Any application can be started using a simple left click. Currently dragging and dropping files onto the icons is not supported.[2] The number of times an application can be simultaneously launched is only dependant on the limitations of the current operating system and the actual application. This allows applications such as "xterm" to be placed in this window if required.

Further to allow large numbers of applications to be dealt with, the application toolbox has the concept of "folders". The following picture shows two folders in use - which is the current default configuration.

Figure 2.3. Application ToolBox Window

2.6. The Trashcan

UNIX has always been unfriendly to users - and this is no more obvious than in the infamous "rm" command. A novice user (or administrator), comes along and types a command such as:

```
# rm * .o
```

Notice the fatal space? So did they once the complete contents of the current directory (apart from any sub-directories), have been deleted. The icon below may not luck much, but could save

[1]This may also be provided by your window manager.

[2]This has been recognised as an area for improvement.

you from this problem ever again!

Figure 2.4. Application ToolBox Window

The trashcan window allows deleted files to be retained and restored if required. The application allows a certain amount of disk space to be allocated to deleted files and ensures that no more space than that allocated can be used for deleted files. More information on the actual management of the trashcan files will be presented later.

An instant status report[1] is available simply by moving the mouse over the window. Clicking on the window will produce a window which can be used to manage the contents of the trashcan manually - again, more on this subject a little later.

2.7. Basic File Manipulation

So now on to actually manage to recognise what each window in the application actually does - the next stage is to use "FileMan" as a day-to-day tool! Thus this section explains the most basic options to allow you to view, edit, remove and print files. These are some of the options on the *File* menu. Use of the basic functions will quickly familerise you with the basic character of the application and ensures that the more "Advanced"[2] features.

2.7.1. Viewing Files

The most important point to remember is that the application requires you to have selected one or more file and directories before certain menu options can be used - this is the case for many of the *File* menu options. Hence it is important to ensure that the files you wish to view are selected!

Once one or more files have been selected, then the menu option which must be selected to view these files is:

`Files ⇒ View Files`

Once selected the application required to view each of these files should automatically start. You will notice that each file will invoke a separate instance of the application for each file. The default file viewer, and the viewer for text files is simply a small default script which uses the UNIX "more" command. There are some interesting points which you may not have recognised at this point:

- Selecting a directory will also invoke a viewer - which is obviously not that sensible.
- Each different file selected invokes a new instance of the application.

[1]Which may be out of date, since this information is updated on a 30 second basis.

[2]Perhaps this is the wrong term to use since the functionationality is very easy to use.

- For one of the files selected either an error occured, or the viewer used has a better alternative.

The first point is interesting - this is because no specific action for a directory has been specified. This will be used as a customisation example in a later chapter. The second point is currently a limitation of "FileMan".[1] The final point is mainly due to the nature of UNIX environments - they differ quite considerably compared to a typical PC environment. This means that the default application used as a viewer for a particular file may not be suitable to the environment - this is why "FileMan" is so configurable, as will be shown later.

2.7.2. Editing Files

Once you have learnt the basic interaction required by "FileMan" to view files, other actions is just as simple. Again, once the relevant files have been selected, simply select the following menu option:

`Files ⇒ Edit Files`

Again, you will notice that for each file that has been selected a separate edit window will appear. The command or application used for edit each particular file will depend on the filetype - though the default editor is a text editor.

2.7.3. Deleting Files

To actually remove files, simply select the relevant files, and then select the following menu option. You may notice that if you have selected directories in the list of files to remove, these will be ignored by this menu option. Indeed, there is a separate menu option which is used to remove directories.

`Files ⇒ Remove Files`

Once this menu option has been selected, if any files have been selected for deletion, the following window will appear, asking whether you wish to delete the first of your selected files.

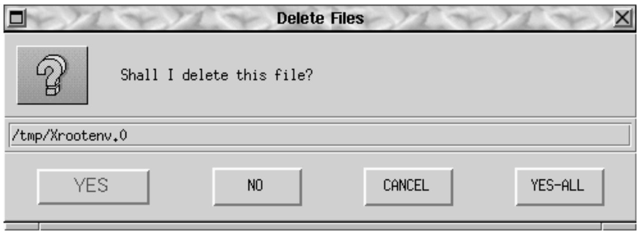

Figure 2.5. Delete File Requester

[1]Currently this is seen as a sensible default action, however improvements are likely in later versions.

If you wish to cancel the delete operation, clicking on the "Cancel" button will immediately abort the operation. However, any files have already been deleted, will remain lost , or in the trashcan. If you click on the "Yes" button the file being displayed will be removed, and the next file shown in the requester.

If you do not wish to delete this particular file, simply click on the "No" button. This will skip the current file and move to the next. Finally if you have selected a large number of files, which you definately wish to delete, clicking on the "Yes-all" button will delete all the files without further interaction. In this instance the requester will still appear on the screen until all files have been deleted.[1]

2.7.4. Deleting Files without using Menus...

In trying to reduce the amount of effort required for standard operations, the current version of "FileMan" allows you to delete files in two other ways rather than using the relevant menu option.

The first way is to simply press the "Delete" key when some files have been highlighted. The other way is to keep the left mouse button down on the last file to delete and to move the mouse over to the trashcan icon - sitting in the top left of the window. As you move the mouse, an icon will follow it. Once over the trashcan icon, simply release the mouse button.

Either of these methods will result in the requester appearing allowing you to delete one or more of the selected files.

2.7.5. Renaming Files

As you know already probably realise, this is another straight forward option. Again, select the series of files that you wish to rename and choose the following menu option. **Note: This option can also be used to rename directories if these have also been selected.**

`Files ⇒ Rename Files`

Once selected, a simply requester will appear where you can are shown the original name of the file, and are asked to type in the new name. This is used to rename the files in the same directory, but you can place a path on the new name to move the file to a different directory. The picture below shows the requester which will appear.

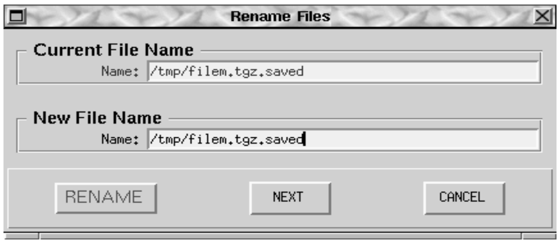

Figure 2.6. Rename File/Directory Requester

[1]An actual progress meter will replace the requester in this circumstance in later versions of "FileMan".

As usual, clicking on the "Ok" button will rename the file, whilst the "Cancel" will abort the operation. If you wish to ignore the current file and rename the other files which have already been selected, click on "Next" to advance the requester to another file.

2.7.6. Copying and Moving Files

The final "basic" operation is to copy or move files around the various directories or even file systems. Again, firstly the relevant files must be selected prior to using the menu option:

`Files ⇒ CopyMove ...`

Once this has been selected, a requester will appear allowing you to select the destination directory for the files you have selected. As the figure below shows, there is an icon to the right of the entry which , when clicked apon, will display a requester allowing to navigate the directory tree to select a destination this way. Whether you type the directory, or use the navigator, clicking on the "Ok" button will move to the next stage of the operation, whilst the "Cancel" is by now obvious.

Figure 2.7. Rename Copy or Move Files Requester

Once the "Ok" button has been selected, another requester will appear, this time allowing you to select whether to copy or move the files. However, as you can see, there are also several options:

Figure 2.8. Rename Copy or Move Files second Requester

Option	Meaning
Confirm Over-write	This is by default set, it ensures that if the operationn is about to over-write an existing file a warning will appear allowing you to a abandon this particular operation. If you do not really care this can be de-selected.
Create backup	This is not selected by default. When selected, if a file is about to be over-written then a backup of the original file is created. The name of the backup is the original file name with ".bak" appended. **Note: If a backup file already exists, this will be overwritten at this point.**
Indicate all errors	If many files are being moved, you may not wish to be shown every failure as it happens - and this is the default setting. However, clicking on this select it will show all errors as they occur, rather than waiting until the end of the operation.
Preserve Information	This option is currently only shown on Linux systems[1] When this is set, and it is by default, the copy of the original file is given the same permissions.

Once you have chosen the option to you to use, (the main one being whether to copy or move the files), clicking on the "Ok" button will proceed. The "Cancel" button requires no mention, but the "Help" button will pop up a small window giving a few hints regarding this requester.

And so finally we can actually do the required action... well almost! Yet another requester will appear again, similar to when you wish to delete files. In this case you can selectively indicate which files to move or copy, and again, use the "Yes-all" option to quickly perform the action on all selected files.

2.8. Restoring Files

So, you have found out how to delete files, but what if you have mistakenly deleted a file - what do you do? Luckily, the trashcan is configured by default, and assuming you have not overfilled the trashcan[2], then the file should be available for recovery.

The file recovery option is always available, since it does not require any previous information, (since it knows where to restore the files to). The menu option to start this process is:

```
Files ⇒ Restore Files...
```

Once selected, two windows will appear. The one on the left is used as a reference to contain the list of files you wish to restore, whilst the one on the right allows you to navigate the trashcan to select the files to restore.

[1]Simply because Linux is the only system where the copy command has an option to preserve permissions.

[2]The installation default limits the size of the trashcan to 250Kb.

The filesystem within the trashcan depends on the files which have been deleted, but typically only includes directory paths which contain deleted files. It is also important to notice that the starting point for the directory tree is the current directory.[1] Any directories which are shown can be double-clicked on to drop into that directory. A ".." directory entry is available to move back up the directory tree. **Note: Any directory entry in the list is clearly identified, since a "/" is placed on the end of the entry.**

When you have reached the correct directory is restore files from, all available files in the trashcan, in that directory, will also be shown. Select one or more files, using a single click and slight drag, and then the names should then appear in the window on the left. This process can continue until you have all the files you wish to restore in the left hand window. Once this has been completed, click on the "Ok" button on the right hand of the window, and all files will be restored. Whilst the files are restoring, a progress meter will be shown.

The buttons on the two windows shown are quite obvious, but just in case, here are explanations of the two buttons on the left hand window. The "Clear" will clear the complete list of files which have currently been selected for restoration. The "Delete" button will only remove those files in the list which have been highlighted. Highlighting files can be done by using the mouse button and the "CTRL" and "SHIFT" keys if you wish to highlight more than one file at once.

2.9. Removing Directories

Because removing directories can be so dangerous in UNIX, it has been given a separate menu option which must be invoked if a directory is to be deleted. So, if you wish to remove a directory, use the following menu option:

`Files ⇒ Remove Directory`

Once this option has been choosen, you will be presented with a requester allowing you to select the directory to delete. If is possible to select the name of the direectory by either typing it in, or by using the icon to the right of the entry field. Once you are happy, select the "Ok" button to continue, otherwise the "Cancel" will abandon the operation. A picture of the window presented is shown below. **Note: The default functionality is to use the trashcan, beware that if you change this any files which have been deleted, will not be recovered.**

Figure 2.9. Requester used for Remove Directory Option

[1]The directory which you last clicked on in the left hand of an active window.

Once "Ok" has been selected, the application will actually examine the directory structure you wish to delete. It does this to determine the amount of work it must do, and also to ensure that you realise the full extent of your actions. Thus, after a short delay you will be shown another requestor, such as the following:

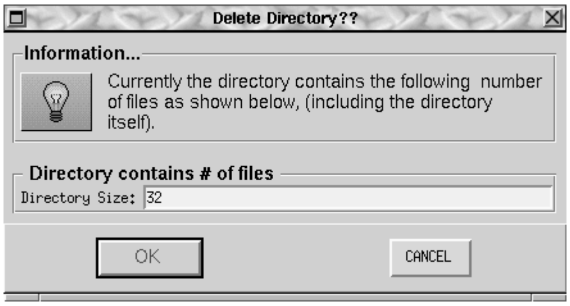

Figure 2.10. Requester showing extent of Remove Directory

Again it is possible to abandon the whole operation at this point, but if you do wish to go forward, then again select the "Ok" button. Since this operation may take some time, during the file deletion a progress monitor is shown along with the name of the current file being deleted.[1]

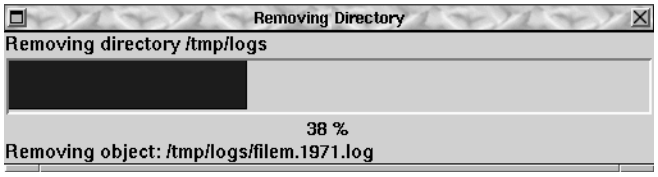

Figure 2.11. Progress Monitor of Directory Remove

2.10. Creating Directories

Creating directories is very straightforward. Once you are in the directory you wish to create, select the following menu option, and a requster will appear. This requester can be used to type the name of the new directory.

`Files ⇒ Create Directory`

Once a directory has been created if it is an already expanded directory window, it will appear shortly after the operation completes.[2] If the directory could not be created, (since it already exists, or you do not have permissions to create it), a window will appear which includes the relevant error.

[1]Eventually this or a move stylish one, will become more widespread through out the program.

[2]This delay may be up to 30 seconds.

2.11. Exiting "FileMan"

So once you have had enough of "FileMan", or you want to to go back to doing things the hard way... Whatever the reason, "FileMan" can be exited by using the following menu option:

`Files ⇒ Exit`

Once selected, you will be prompted with a requester asking if you are sure, and if you are, simply click on the "Ok" button, otherwise the "Cancel" button will abort the operation.

If you do not wish to use the menu option there are two other recommended ways of exiting "FileMan".

2.11.1. Using the ToolBar

The toolbar has a button on it which can be used to exit "FileMan" - a button with dynamite on it! If this is not immediately obvious, then leaving the mouse button over any button for a reasonable period of time will result in a "tooltip" being displayed.

2.11.2. Using the Window Manager

Whichever window manager you use, they all contain an option on the title bar allowing you to close an application. If "FileMan" appears to lock up,[1] using this method should again allow you to exit gracefully.

Of course, you could exit the application by using the "kill" command, or by using "CTRL-C" in the window containing the running command, but neither of these two methds are recommended. **Note: Exiting the application in this manner may cause corruption in the setup files, and thus beware!!**

2.12. Using the Menu Functionality

Now that you are familar with the way "FileMan" actually works, the other simple to use menu options should make perfect sense. The following chapter includes details on the other interesting menu options, but the rest of this chapter is used to ensure that the functionality of the menus, toolbar, trashcan, and applications toolbox are fully understood. This is expected to be enough to use "FileMan" effectively, and the following chapters are more for reference.

By now you will realise that the menu options are context sensitive - it is possible to access some options all the time, others when at least one directory has been expanded, and most of the others when files are currently selected. The idea behind this is to ensure the functionality of the application is as logical as possible - what is the point of being able to select a menu option when it logically does not make sense?

Thus, if you are not able to use a menu option when you think it would make sense, try selecting a file, and try to use the menu option. If the option will require input, then if should

[1]This occasionally happens and is down to focus problems, rather than anything too sinister.

then be available. Otherwise, try selecting a directory by either expanding it (i.e. clicking on it in the left hand part of a window), or by selecting in as a normal file.[1]

2.13. Using the toolbar

The buttons on the toolbar act in a similar function to menus - some may be available only under certain conditions. Unfortunately they do not currently "grey" out when they are not available.[2] The default installation only has a few buttons (less than ten), but it quite easy to add additional buttons, as later chapter will show. The icons should hopefully give an indication of what they are actually used for, but if you are still confused, leaving the mouse over a button for a couple of seconds will result in a "tooltip" being displayed.

A "tooltip" is a small yellow box which displays one or two words to hopefully identify the meaning of the icon. If this is not enough an actual sentence will be displayed in the "status area" showing what the actual icon does. This is the case for both the icons on the main window, and the icons on the partition window.

Finally, it is worth remembering that the functionality of a menu option can be invoked by simply clicking on the relevant button, (with files or directories selected, if required), or by dragging and dropping the selected files on to the button. Take your choice!

2.14. Using the Trashcan

As explained in a previous section, the trashcan (the icon in the top left of the display), is used to delete files, gain status reports on the trashcan, and manage the contents of the trashcan. This this can be accomplished using the small icon!!

The first visual clue to the status of the trashcan is the icon itself. If it is a normal trashcan, this indicates that the trashcan is empty. If the icon is a bulging trashcan, this indicates that the trashcan does contain some deleted files.[3] However, to gain a more accurate listing of the trashcan statistics, simply move the cursor over the trashcan icon. A small window should appear, containing the following information:

```
Trashcan Max     : 250Kb
Trashcan Used    : 56Kb
Trashcan Entries: 12
```

This immediately informs you on the maximum size of the trashcan, the actual number of files stored in the trashcan, and the amount of space actually used in the trashcan.[4] As of version 1.01, when the trashcan becomes full, the icon will turn pink, indicating the trashcan is almost full.

[1]This functionality should be turned on by default - but can be removed, see the next chapter for further details.

[2]Though this should be added quite soon.

[3]Note: From version 1.01 the trashcan icon actually contains a little more visual information - it shows an approximation on how full the trashcan actually is.

[4]This is still not fully accurate since the amount of space used includes information on directories - even if they are empty. This is expected to be resolved in version 1.1.

The trashcan can be emptied at any time simply by clicking on the trashcan icon. This will present a window which will allow you to select one or more files you wish to remove from the trashcan. **Note: Clicking on the "Remove Selected" button will permenantly remove the files from the system.** When you have finished simply click on the "Finish" button. Finally, if you wish to remove all the files in the trashcan, simply click on the "Remove All" icon.

So what happens when the trashcan becomes full? There are two possible options - "Manual" and "Automatic". The default is manual. When set to manual you will be presented with a requester shortly after the trashcan exeeds it's maximum size, and will only be able to use the "Finish" button on the requester once you have deleted as many files is required to reduce the usage of the trashcan space to acceptable levels.

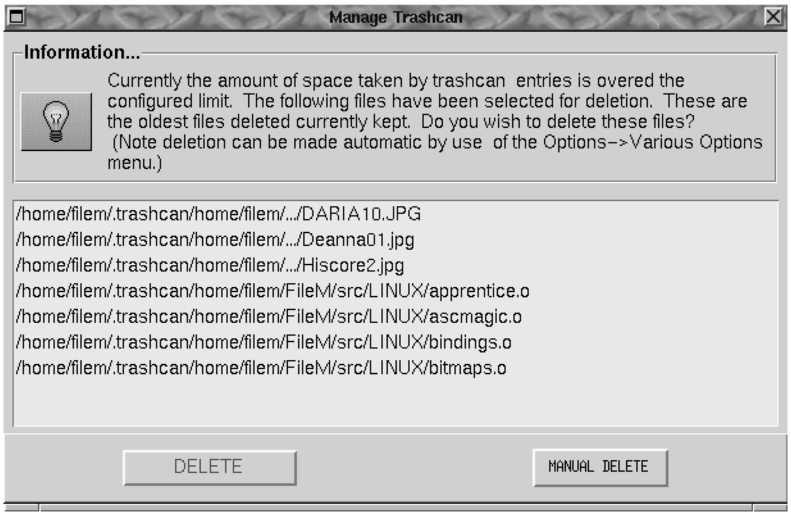

Figure 2.12. Suggested files for removal from trashcan

As can be seen, the requester will show which files it recommends to be removed from the trashcan - clicking on the "Delete" button will remove these files, allowing the files in the trashcan to occupy only as much space as configured. **Note: The files which are selected to delete are those which were removed the longest time ago.** The recommendation is to use the "Manual" option so you can always be sure on what exactly is being deleted. Once you are happy you could choose the "Automatic" option - though beware the this will mean files will be removed from the trashcan without your knowledge.

Figure 2.13. Trashcan managment - user initiated.

The trashcan can be managed at any time by using a left click when no files have been dragged to it. This will cause the above requester to appear. Since this is user initiated it can be abandoned at any time. The information shown includes the sizes of all files which are in the trashcan.

Files can be highlighted by clicking on them and then using the "Remove Selected" button. To completely empty the trashcan, use the "Remove All" button. To exit at any point, use the "Finish" button.

Chapter 3. Using the Standard Installation - Part 2

Now that the basic functionality has been discussed the next stage is to document all available options in the standard configuration which will then lead nicely onto the following chapters which enlighten on how to change or extend the configuration to your own specific requirements.

3.1. Surely I know all I need to?

Well, anyone who feels that they are not confident, and do not like the idea of experimenting with the functionality. However, for those people a recommend creating a dummy directory with several files in, and then using "FileMan" to manipulate those files.

This section is more of a reference than a guide, and so presumes that the user is able to carry out all the basic tasks detailed in the previous chapter.

3.2. Using the Partition Window

From the previous chapter the basic functionality and use of the partition window should already be obivous. However, this section is a quick overview of the toolbar options given on each of the partition windows. You may have already experimented with these already - however, they may have been something you missed, so please continue to read if you can spare the time!

3.2.1. Using the "Basic Listing" button

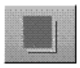

Figure 3.1. "Basic Listing" icon

This button is quick way of changing the type of file listing shown for **all** partitions. By clicking on this button the view for all partitions will be shortly updated with the listing of this type. This listing will show simply the filenames and no details. Multiple columns are used - though the number of columns is dependant on the length of the filenames in each directory.

When you change the view of the partitions this information is changed immediately in the configuration, meaning that whenever "FileMan" is re-started, this type of content view will be used for all partitions.

3.2.2. Using the "Details Listing" button

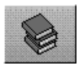

Figure 3.2. "Details Listing" icon

As with the previous button is this another way of changing the contents shown for the files in the partition window. This one shows all details for all files (and directories, if required), including the owner, group, size, modification time.

Again, once this button has been selected, there will be a short delay before all the partition windows are updated. Once this has been set, this view type will be used until another is chosen, for this and future invocations of "FileMan".

3.2.3. Window Duplicate button

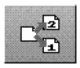

Figure 3.3. "Window Duplicate" icon

This is a very useful feature - it will create another window showing you the details of the current partition. It functions in exactly the same way as the original window - the only difference is that the window title bar will include the words "(Copy of)".

Why would you require two windows with the same partition? The simplest answer is that this can be very useful when you wish to drag-n-drop files - it might be that the due to the location of the directory is difficult to show the names of both directories on the screen at once, and though it is possible to move or copy the files, sometimes two partition windows can make the task much easier - try it!

3.2.4. Refresh Windows button

Figure 3.4. "Refresh Windows" icon

Many of the operations which modify the contents of a directory will force a refresh of the partition windows to ensure they are still relevant. However, there are several (three) differing levels of update - and on the highest a refresh will only occur on user request - this is where this button must be used to refresh window contents.

This functionality is especially useful when the contents of a directory take a long time to update - it can be a real pain having to wait for it to perform the automatic update - in this case simply change the refresh option (described later), and use this button when you require an updated window contents.

3.2.5. Window Filter button

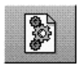

Figure 3.5. "Window Filter" icon

By default all files in a directory for a partition window are shown.[1] However, if a directory contains 15,000 files (possible) then actually searching through for a file, (or even allowing "FileMan" to attempt to draw all files), would take an age... This is where this button comes in useful.

If you have a particular series of files you wish to see, then the filter button can be used to select a pattern by which only matching files are shown. Currently the pattern syntax is not so great, but here are a few examples of what is possible.

Filter	Purpose
*.h	Show all files in the current directory which have a ".h" extension.
d*.tar	Show all files have begin with the letter "d", and have an extension of "tar".
[a-e]*.[ch]	Shows only those files which an extension ".c" or ".h" which begin with any of the lower case letters, "a" through to "e".

3.2.6. Window Close button

Figure 3.6. "Window Close" icon

When you no longer wish to use a partition window, you can either choose the window close button, as made available by your window manager[2] , or by using this button. There may be a slight delay before the window closes, (usually less than one second), but this is normal.

Remember that rather than closing a partition, most window managers support the idea of "minimised" windows, where the window will appear as an icon on the desktop. "FileMan" supports the idea of minimised windows - indeed if a window is minimised clicking on the main window to create a window for that partition will automatically un-minimise a window for that partition, rather than creating a new one.

[1]"Hidden" files can be turned on and off by using an option - documented later.

[2]Most window managers show this as a button with an "X" in it as part of the title bars. Others will have a "close" option on a menu accessed by the title bar.

3.3. Using the "Utilities" menu

The "Utilities" menu is where you should find those commands which are typically available in a UNIX environment. If you are not able to find access to the utility of your choice through this option, or have an idea of improving any of the options, please feel free to email me with your comments.[1]

3.3.1. Using the "Split" menu option

The "split" utility allows you to cut a large **text** file into several smaller parts. This is particulary useful when cutting files down to size to fit on floppy disks, or even transferring large amounts of text over email accounts (some of which still may be limited to transferring 64Kb per message).

Once you have selected one or more files you wish to split, simply select the relevant menu option, and you will be presented with a requester as shown below. Although initially this may appear quite complex it is actually very straightforward. It basically gives you two options:

- It allows you to choose *how many parts* you wish to split the file into. Obviously the sizes of the parts are roughly the total size divided by the number of parts.

- The actual maximum size of each part (in Kilobytes). It will then calculate how many lines are in the file, and thus are many parts the file must be split into to actually meet the criteria. Obviously it should be noted that if a small file size is given for a large original file, a **large** number of files may be generated!

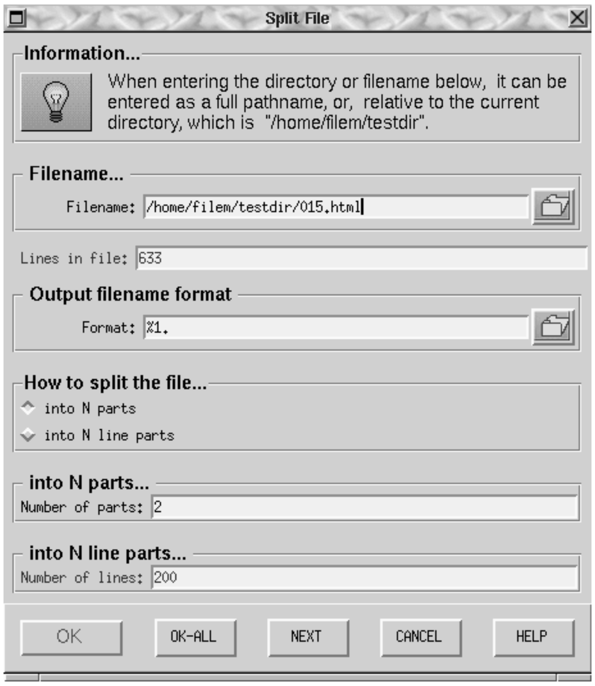

Figure 3.7. "Split Requester" icon

[1]My email address can be found in the Preface.

Of particular interest in this case is the names of the files that the "split" process actually creates. Notice that the filename is shown as "%1.". The "%1" will be replaced by the actual file which is being split. The split command will append a suitable suffix on the end of the file to ensure all split sections have unique names.

This is particularly useful when you have selected more than one file to split. Using the "Ok" button will make the requester re-appear for each file that is to be split - which can be a real pain. However, by keeping the filename to use for the split files to be "%1.", you can use the "Ok-All" button to ensure that the file names generated are based on the actual filename that is split.

There are several occasions when this facility is utilised by FileMan to aid tasks that would otherwise become too repetitive.

3.3.2. Using the "Word Count" menu option

Fairly obviously, this allows you to show statistics of text files. When one or more files have been hi-lighted the menu option can be used. The requester allows you to select whether to show any combination of a line count, word count, and character count.

The actual first file which has been selected will be shown in the "Input FileName" section. The icon on the right of this entry can be used to select another filename - click on it and another easy to use requester will appear allowing you an option to change the filename to view the statistics of.

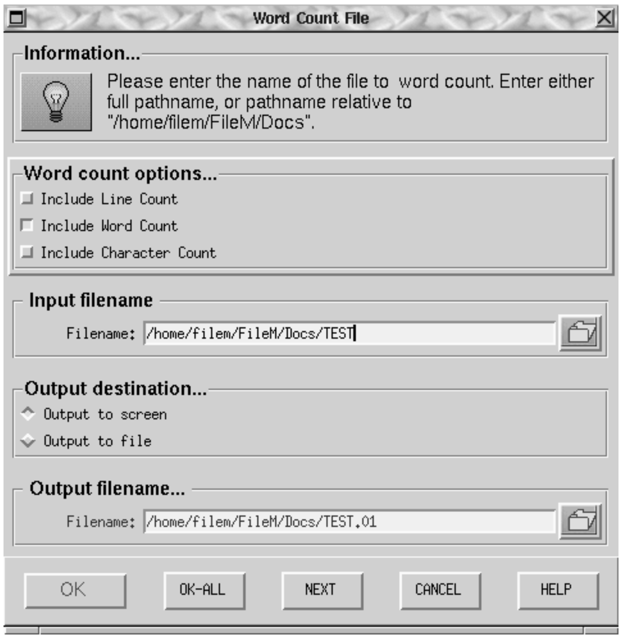

Figure 3.8. "Word Count Requester" icon

It is possible to output the results either to the screen, or to a file. When you choose to output to a file the "Output Destination" entry can be edited to select a correct filename.

Note: When using the "Ok-All" option, only more file will be generated, not one for each file selected.[1]

3.3.3. Using the "Spell Check" menu option

This option will allow you to scan one or more text files and will display a list of words which are thought not to be spelt correctly. For this function to work, the "spell" package must be installed on your system.

When one or more files have been highlighted this menu option is available to show you any spelling mistakes in those files. Obviously the files specified should be normal text files to ensure that meaningful results are shown.

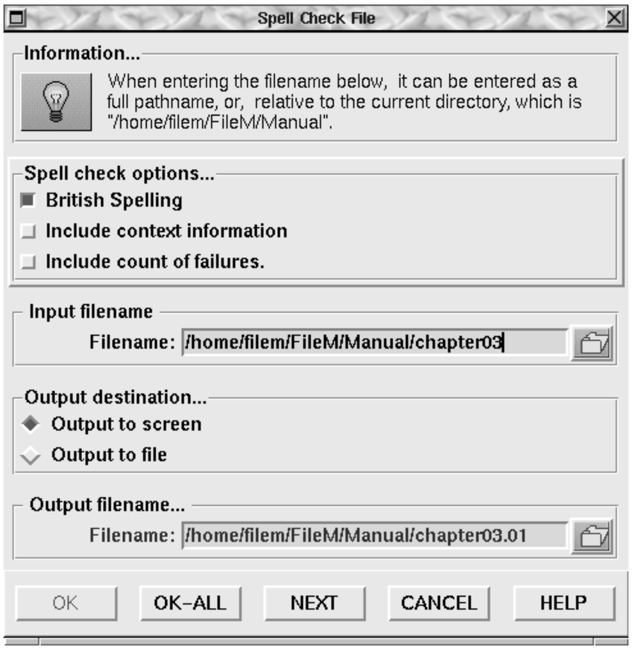

Figure 3.9. "Spell Check Requester"

When the list of files is shown, there are three options which can be used to modify how this command operates:

- **Use British Spelling** - When this option is checked the "-b" option will be used on the "spell" command to try to use the British English Dictionary.
- **Show Errors in Context** - When this option is checked the list of mis-spelt words is fed through a "grep" filter, which then shows each incorrect word in the line in which it occurs. If a word is spelt incorrectly more than once, every line will be shown.
- **Count of Failures** - In this instance, for each word spelt incorrectly, a number is shown indicating the number of times it occurs in the file.

Below the list of options, the first of the selected files is shown. The icon on the right can

[1]There appears to be a bug with the "Ok-All" option, even with 1.01 - this should be fixed in the next release - currently 1.1.

be used to select another file if required. It is also possible to select the destination of the results - either to the screen or to a file. The default is to the screen. When outputing to a file, you will be able to alter the destination file name.

Finally at the bottom of the requester various buttons are available:

- **Ok** - Go ahead and perform the spell check for the selected file. If more than one file is selected, the requester will display on the screen with details of the next file to check, once the current file has been dealt with.
- **Ok-All** - Go ahead and spell check all the files. The output window, if used, will be separated into separate sections for each file. As usual, the filename specified such include a % to indicate the original filename, otherwise each invocation of the spell check will generate and overwrite the original file.
- **Next** - Click on this if you want to ignore the spell check on the current file, and move to the next file to check. If only one file was selected, or this is the last file, the requester will be removed.
- **Cancel** - Abort this operation - the requester will disappear and you will be allowed to continue to interact with the main application.

3.3.4. Using the "Find Files" menu option

This option allows you to recursively scan a directory structure and produce a list of matches. There are various possible filters to add to the search including the name of the object, type of the object, size of the object, and modification times.

When this requester is shown, the latest selected directory is shown as the starting point for the recursive search. As can be seen, the file selector button is available, allowing you to change the start location if you so wish.

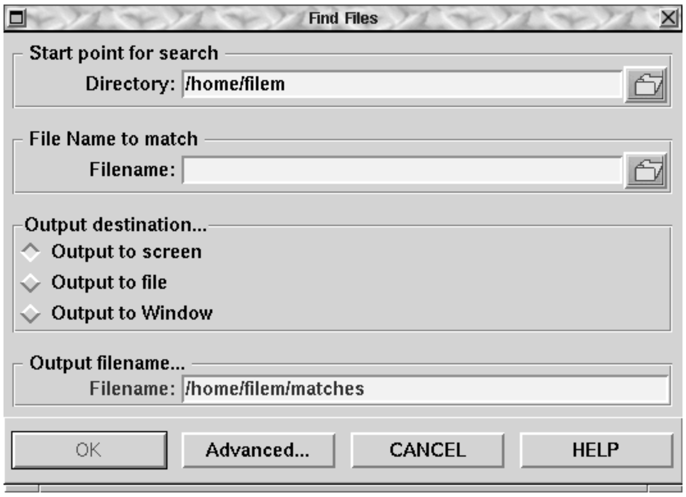

Figure 3.10. "Basic Find Options"

It is possible to specify a pattern match which will be used by the "find" command to match all files. All shell expansions are acceptable, so changing this to *.[ch] will produce a list of all objects which have a ".h" or a ".c" extension.

The output destination for the results should be either the the screen, or a file. Currently the window option is not yet supported, and can not be selected. If outputing to a file, the output filename entry will be available to choose the name and location of the file to create.

The buttons available on this requester are similar to those shown previously, accept for the "Advanced..." button, which allows you to access the more advanced functionality available.

Figure 3.11. "Advanced Find Options"

On the advanced menu it is possible to specify the following additional criteria to add to the search:

- **File Size** - Used to specify the size of the file. If this is required, choose one of the buttons on the radio option, and then enter the number in the entry frame. This number is given in bytes.
- **Modification Time** - Allows specification of the modification times of files/objects to match against. If required, select one of the radio options - this will allow the number to be edited. This is the number of days. A "day" is specified as 24 hours.
- **Object Type** - This option can be used to specify whether the results should just include all objects, just files, or just directories.

Once any advanced options required have been set, simply press the "Ok" button to return to the main Find Files requester.

3.3.5. Using the "Grep Files" menu option

This option, (not named very well really), allows you to search through one or more files for a text string, or a regular expression. As usual the filename to search through is shown, which can be changed using the file selector if required.

There are some options with which you can modify the functionality of the search:

- **Ignore Case** - Usually the search is case sensitive, but if you check this option case will not be taken into account, and all matches will be shown.
- **Don't include filename** - Usually when more than one file has been chosen to match against the filename is prefixed for each match line. Checking this option will remove this from the output if required.
- **Include line numbers** - Each match shown will include a line number, which can sometimes help locating the matching lines in an editor.

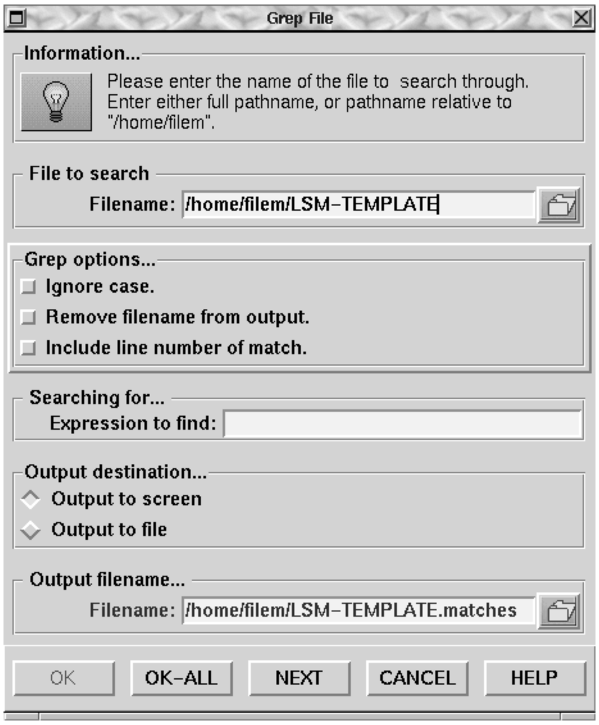

Figure 3.12. "The Grep Files Requester"

The expression to search for can be a text string, or an extended regular expression, as used by the "grep" command. So, for example to search for all the lines which contain the word "fred" and / or the word "mike", specify the following as a search pattern:

fred|mike

The destination for the output can be either the screen or a file as usual. Also, the buttons at the bottom of the requester work in the same fashion as with the other requester.

3.3.6. Using the "UUencode files" menu option

Although MIME enabled email is widespread, sometimes it is necessary to uuencode the file before it can be sent. This option will allow you to do this. This menu option is available when one or more files are currently selected.

When selected, the requester will allow you to set the output filename, as shown below. When setting the filename, the default is currently "%1.uu", which will create a file with the same name and directory of the original file, but will have a "uu" extension.

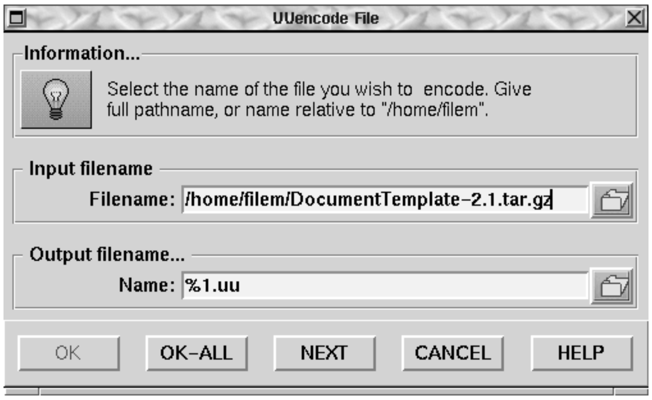

Figure 3.13. "The UUencode Files Requester" icon

The buttons at the base of the requester are similar to those shown on other requesters. The "Ok-All" can be used to uuencode all the selected files, one after the other.

3.3.7. Using the "UUdecode files" menu option

This is possibly a little more useful than the "UUencode" menu option, since many attachments will be saved to disk in "UUencoded" format. Note: The default action on a file with an extension of ".uu" will be to UUdecode it, and so, as well as the menu option you could simply double-click on this file.

3.3.8. Using the "Directory size" menu option

This option can be used to estimate the size of a directory structure. Although the partition window will show you the total size of all files in the current directory, (well those matching the filter), this is not as accurate as checking the disk blocks occupied by all entries including contents of all sub-directories. For this reason a separate menu option has been provided to allow access to this information. Once a partition window has been opened, and a directory selected, the "Directory Size" menu option will be available. When selected you will be prompted with a requester with various subsections.

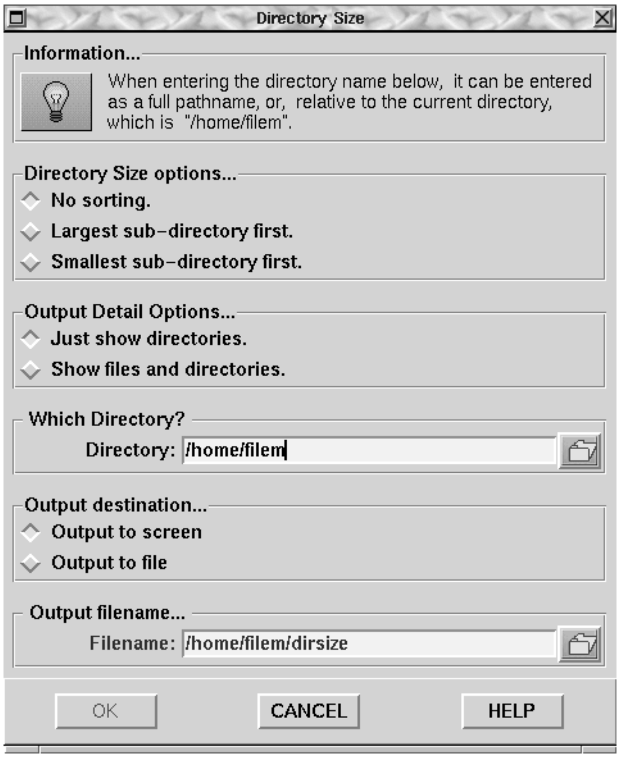

Figure 3.14. "The Directory Size Requester" icon

The sorting option allows you to specify how you wish to display the results. It is often useful to sort the results to show those files and directories which occupy the most space first or last. The default is to use no sorting which will show results as it naturally processes them.

Note: Unless you are processing a very large directory structure, (such as "/"), there is very little overhead in using the sorting facility.

The detail output option allows you to specify whether the output should include all files, rather than the more usual output of showing the size of each directory structure only. The default is to just show directories. Again, be ware that showing the complete contents of a directory structure including the files can result in vast amounts of output.

Obviously the directory start point can be modified at any point to ensure that it is valid. Finally, it is possible to select the output destination for the results - either to the screen, or to a file.

Once the "Ok" button has been selected the results will be generated. Depending on the size of the directory structure and the speed of the machine, there may be a considerable delay before the results are shown, or the interface unlocks if the output is to be sent to a file.

3.3.9. Using the "Compress Files" menu option

This option should be used to compress files. It allows you to choose which compression method to use. As of version 1.1 there is currently a choice of three compression methods:

- **Compress** - This is the standard UNIX compression tool which is available on all flavours of UNIX on which FileMan actually runs. It runs much quicker than the other two, but the compressed files can be much larger than if the file were compressed with "gzip" of "bzip".
- **Gzip** - Although not standard, GNU zip is possibly the most widely used compression method. It offers far better compression than "compress" but can take much longer to run.
- **Bzip** - This is a relatively new compression utility, which (as of 22/07/98), has not reached version 1.0. However, it does offer very good compression, often besting "Gzip". The down side of this is that compression can take longer, and it does require more system resources for handling the compression.

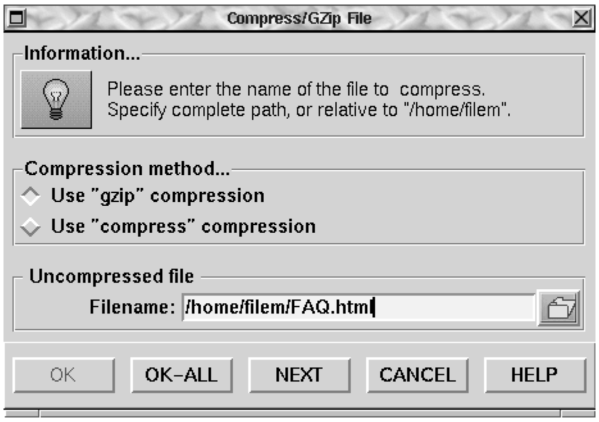

Figure 3.15. "The Compress Files Requester" icon

Once the compression method has been selected ensure the file to compress has been selected, and click on the "Ok" button. If more than one file has been selected, all files can be compressed once the compression method has been chosen, by clicking on the "Ok-All" button.

3.3.10. Using the "Uncompress Files" menu option

When one or more files have been selected, and this menu option is selected, you will be given a choice of which uncompression method you wish to use. Obviously the uncompression method to use must be the same one the file was compressed with! There are several uncompression options - which are the opposites to the compression methods available, mentioned previously.

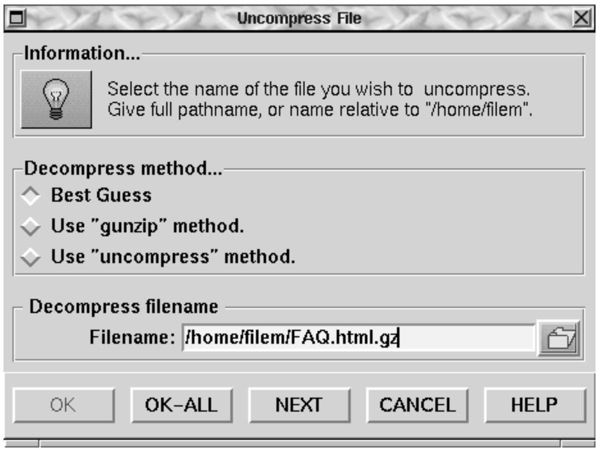

Figure 3.16. "The Uncompress Files Requester" icon

However, the best choice is to leave the compression method to Best Guess. This will choose the decompression method based on the extension of the file. This is useful when several compressed files have been selected, and you wish to decompress them all. In this case, pressing the "Ok-All" button will uncompress all files, choosing the relevant decompression method for each file.

As usual, the requester has buttons to process the current file, (the "Ok" button). Also if you wish to ignore this file, and move on to the next, click on the "Next" button.

3.3.11. Using the "Archive Files" menu option

An "archive" file is defined as a single file which contains many other files. This is the method by which much of the software on the internet is distributed. This option allows you to select a series of files and create a single archive file, containing all these files.

Currently "FileMan" supports creating archives in the following format:

- **CPIO** - this is a standard archiving format which is widely used on many UNIX environments. The only problem with this type of archive is that there are many optional flags, and if you get one wrong trying to unarchive the files, you get an error! This is probably why it is less widely used then the other alternative ...

- **TAR** - The most widely used of the archive methods available. It it is easy to unarchive the files, and requires few options. The disadvantage is that it is not as flexible as "cpio" - for example, "cpio" can be used to create "tar" archives, but not the other way round!

Once this menu option has been selected, the following requester will appear:

Figure 3.17. "The Archive Files Requester" icon

It is possible to type the name of a file to which to create the archive. Note: If you end the name of the file with a "." a suitable extension will be added to the filename to ensure that the archive type is standard - and recognised by "FileMan".

Alternatively, it is possible to select either a floppy or tape device - this will then backup to the devices specified in the configuration file as tape or floppy devices. Beware that if many files are selected, selecting a tape or floppy disk device may take some time to complete, and beware of the limitations of size on floppy disk devices.

Once you have specified the type of archive, and the name of the archive file, click on the "Ok" button to create the archive. Currently there is no feedback on how this operation is progressing, and so if a large number of files are being archived, expect a short wait before this operation completes - especially when writing to tape or floppy disk.

Note: If you wish to archive a complete directory structure, (including all subdirectories), there is an easy way to do this. Firstly ensure that the option allowing you to show directories in the file list is set, and then select the directory you wish to archive. When this requester is selected, type the new of the new archive, and click on "Ok". Note: Since this program works with absolute pathnames, the archive created may contain absolute pathnames - beware of this fact. Once an archive has been created use the "view" option on the file to check whether this is the case. "Gtar" will automatically remove a leading "/" from the filenames, whilst many other archiving programs may not...

3.3.12. Using the "Unarchive Files" menu option

If you have an archive you wish to extract, you can select the archive file, and select the "Unarchive Files" menu. Once selected, you are presented with a requester as shown below.

Figure 3.18. "The Unarchive Files Requester" icon

As with the archive option, you are able to choose from restoring from a tape device, floppy disk, or any file. You are also able to specify the type of archive which the file actually is.

If the archive method removed the leading "/" from the pathnames of the archived files, you can use the entry for selecting the directory to restore the files to.

Note: If youre archive and unarchive programs do not remove absolute pathnames, then this option will be ignored by the unarchive program.

As with archiving, the amount of time this takes depends on the size of the archive, and the source of archive itself - extracting from tape can be slow.

3.4. Using the "Options" menu

The options menu provides a variety of functions which modify the behaviour of "FileMan" in some way, or allow access to the configuration or logging information. Currently some of the menu options are not 100 percent implemented, but mostly this functionality is actually complete.

3.4.1. Using the "View Log" menu option

This menu allows you to examine the commands which have been run during this session. If required this menu option, and the next, can easily be removed from the interface. In fact, the whole of the "options" menu can be removed if required. Of course, if you wish to create a system which can not be customised by the user’s, it is quite easy to do - see the next chapter for details. To view the log use:

`Options ⇒ View Log`

If this menu option is available, the output will be shown in the following format:

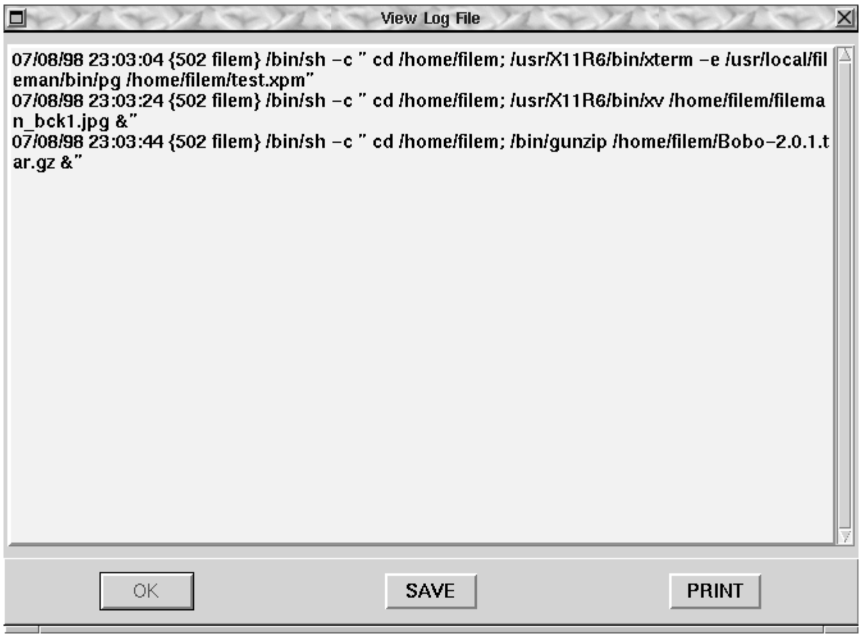

Figure 3.19. The View Log Display

The output is in the following format:

- **Time and Date** - the time and date the command was run. Please be aware that a new log is generated every time the application is restarted. At the moment, the logging is done to a separate file for each user as well as each session.
- **User id and name** - this is really used by site administrators when all user's logging information is going to separate files. Otherwise not really useful!
- **Command string** - the actual command which was run to perform a certain action. This is the final command string after passing through the parser - so it you typed the following command:

```
gife filename
```

in the Run command menu option, you would probably see a command such as the following:

```
/usr/local/bin/xv /home/filename
```

Note: Some of the command strings are quite complex, and can include several commands - usually they set the directory relative to the file in question. Note: Also the startup process will remove all log files for the current user which are over 24 hours old. This means that the old log files will not fill up log file system. The default location for logs is "/tmp/logs".

3.4.2. Using the "View Error Log" menu option

When an external command returns an error this should be recorded in the error log file. The entries in this file are the same as the standard log file, but also include the error message. The error message is sometimes not as appropriate as it could be, but it is better than nothing. Note: Not all commands actually log error messages in this way, but as of version 1.1, this is gradually improving.

3.4.3. Using the "Tile Partition Options" menu options

These three menu options allow you to tile the partition windows in various ways. These menu options work as follows:

- **Tile Vertically** - All the existing partition windows will be re-sized and re-positioned to fill the screen, with each window resized so that all fill the screen.
- **Tile Horizontally** - All the existing partition windows will be re-sized to fill the entire screen. The windows will be "short and wide", rather than "tall and thin".
- **Tile Partitions** - The partition windows will be reduced to a small box and arranged into a square. This option is really only useful when you have lots of partition windows and possibly a high resolution screen.

3.4.4. Using the "Configure Menus" menu options

This option is not currently fully functionality, and hence is not being documented until completed. Once completed it will allow you to add and remove menu options from the interface via a standard requester.

3.4.5. Using the "Customise Buttons" menu options

This menu has not been implemented as of version 1.1 - however this feature should be available from version 1.2 onwards.

3.4.6. Using the "Associate Type" menu option

This menu option is only available when a file has been selected, and allows you to choose which type the file should be recognised as. If the file is already recognised, then the existing type will be shown.

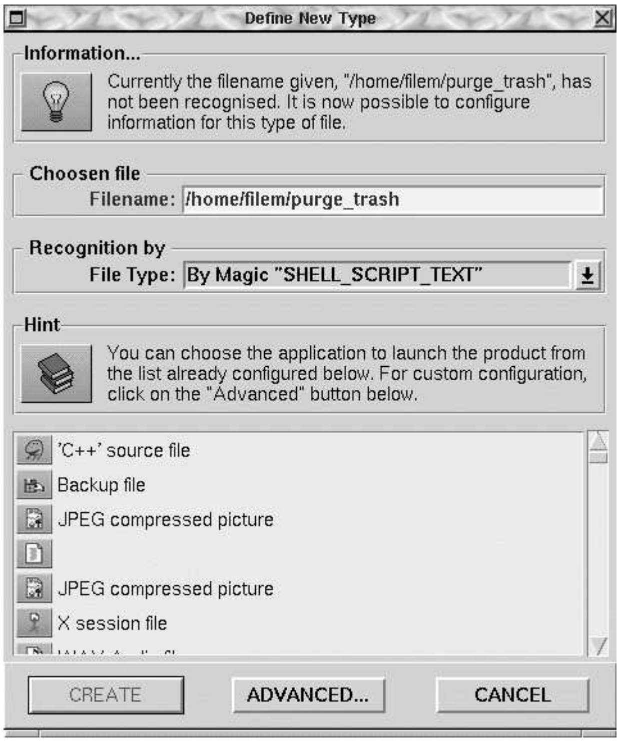

Figure 3.20. The Associate Type Requester

When choosing a type, please choose a currently defined file type which is as close as possible to the new type. Once a file type has been selected, you can click on "Advanced..." button to further configure the type.

If the choosen file type already does exactly what should be done, simply click on the "Create" button, and you will be then asked to type a short description of the file type.

The "Advanced" settings..

If you really wish to change the functionality for a type, you will need to access the advanced menu to change the details. Here it is possible to change the following settings:

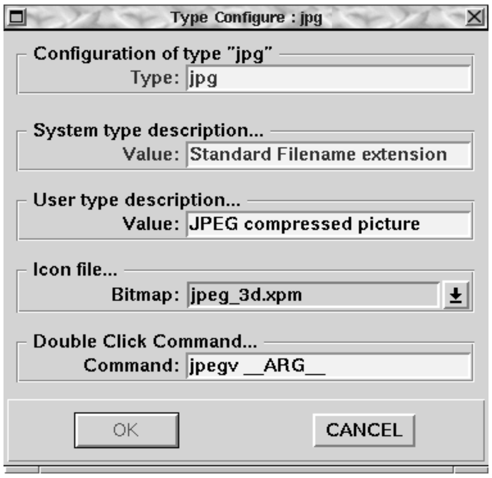

Figure 3.21. The Advanced Associate Type Requester

- **Type** - Shows you the type of the file which is being recognised. It is not recommended that you change this, since the results could be unexpected.
- **System Description** - This can't be edited and is actually an indication of how the system recognises this particular type - such as by its extension, or magic of type.
- **User Description** - This is the description of the file type - you can change this to include any text for the type representation. However, please use only standard characters, and not brackets.
- **Icon** - Allows you to specify the icon to use to distinguish this type of file. The list of available icons is taken from the icon directory - so if you have installed additional icons, they should be in this list.
- **Command** - Specifies the command to be run. If the command is prefixed by a " **:** ", then this indicates that the command is an internal "tcl" command. If the command is prefixed by a " **#** ", this means that when the command has been completed, refresh the partition windows - useful when the command creates new files, or removes them. There should also be an " **__arg__** " word, which will be replaced by the relevant file name.

3.4.7. Using the "Sorting Scheme" menu option

This menu option allows you to choose the ordering of the files which appear in the partition windows. There are several options available. The sorting of information does have a small overhead of the system when displaying partition windows, but unless the directories are very large, this can be considered insignificant.

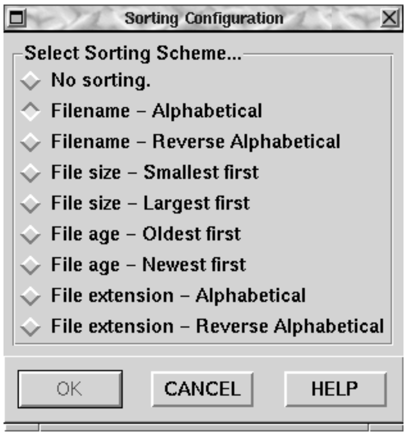

Figure 3.22. The Sorting Scheme Requester

The sorting scheme is accessed by using the following menu option:

```
Options ⇒ Sorting Scheme
```

You will then be presented with the following available sorting schemes:

- **No Sorting** - this is the default and in this case there is no sorting of the files. You may find that the files come out sorted, it will depend on your system, and is not guarenteed.
- **Filename - Alphabetical** - This will display all the files in alphabetical name order, and is probably the most common method of displaying information.
- **Filename - Reverse Alphabetical** - The contents of each partition window will be displayed in reverse alphabetical order.
- **File size - Smallest first** - The smallest file will be shown first. In this instance the directories will probably all be bunched together since they will have a file size of "512" or "1024".
- **File size - Largest first** - Again the files are ordered by the size in bytes, but this time the largest files are shown at the top of the partition window.
- **File age - Oldest first** - In this instance the files which have the oldest modification times, (the time which changes when a files contents are altered), will be shown first.
- **File age - Newest first** - In this case the files which have been modified the most recently will be shown first.
- **File extension - Alphabetical** - This will sort files by the named extension - the characters after the last, (or only "." in the filename).
- **File extension - Reverse Alphabetical** - This will sort files so those with an extension starting with "Z" will be shown before those starting with a "A".

Please notice that when sorting is done "alphabetically", this means that soreting will be done based on ASCII ordering, so when sorting "Daddy" and "baby", the first word will be "Daddy".

3.4.8. Using the "Formatting Options" menu option

If you wish to change what details are shown for the files and directories in the partition window, then this is the menu option you need to use:

```
Options ⇒ Formatting Options
```

Once this menu option has been chosen, the following requester will appear:

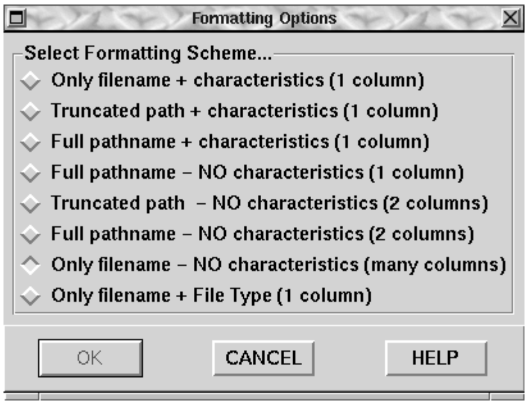

Figure 3.23. The Formatting Options Requester

There are various different formatting schemes which are available, as explained in the list below.

- **Only filename + characteristics (1 column)** - This shows information in a format similar to the **"ls -l"** command. Information shown includes the permissions, owner, group, size, and modification time for each file and directory.

- **Truncated path + characteristics (1 column)** - Similar to the previous option, however, in this case, the pathname of the file is shown. Once the pathname becomes too long it will be truncated to show the last part of the path.

- **Full pathname + characteristics (1 column)** - Again, similar to the previous optoin, but this time the complete path is shown for each file, whatever the length of the path.

- **Full Pathname - No characteristics (1 column)** - This option currently only shows the file name only - it has not been properly implemented yet!

- **Truncated path - No characteristics (2 columns)** - This will show two columns of information - the name of each file, and part of the path. The amount of levels of the path shown for each file depends on the length of the path items... Try it, you'll soon see what this does!

- **Full pathname - No characteristics (2 columns)** - This should show similar information to the previous option, but not show truncated pathnames - however as of version 1.1 this option is still not functioning correctly!

- **Only filename - No characteristics (many columns)** - This will show the filename only. The number of columns used depends on the length of the filenames in the current directory, the size of the font, and the width of the screen. This is possibly the other common format used to display filename information in partition windows.

- **Only filename + File type (1 column)** - This shows the file name, and the file type description (if the file is recognised). This can be quite useful, especially if you've assigned illogical icons to your filetypes...

3.4.9. Using the "Various Options" menu option

This is possibly the most useful requester as far as configuration is concerned. Here it is possible to alter many of the characteristics of the program, and it is worth using this for a while to get the characteristics you are most happy with.

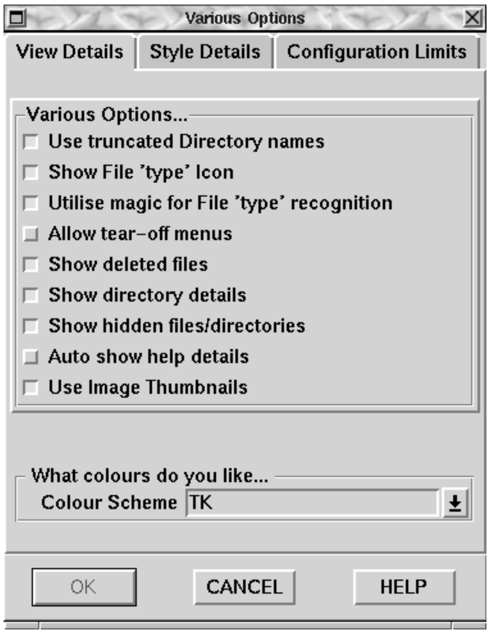

Figure 3.24. The Various Options Requester

This requester uses a "notebook" style window, the various tags at the top of the window can be used to access different aspects of the configuration information. The uses of each tag are as follows:

3.4.9.1. View Details

Various Options

This is a series of buttons which can be independently altered to change the details of the what typically is seen as the product is used. As of version 1.1 there are currently nine settings available.

1. Use truncated directory names

When set the names of the directories on the left hand section of a partition window are simply shown as the directory names. If this is unset, rather than showing the name of the directory, the full pathname is shown, as can be seen from the picture below. – Picture Here —

2. Show file "type" icon

When this is set, the file type icon is shown for each file - which is the default. When this is turned off, no icons are shown for the files. This has the advantage of reducing the system overhead in terms of memory, colourmaps, and time used to draw the items on the window! However, most people feel happier with an icon to look at!!

3. Utilise magic for file "type" recognition

FileMan comes with a built in "magic" type recognition library which currently matches over 1,800 different file types. This option allows you to choose whether or not to use this library. When used, each file that is not recognised by other methods is examined by the magic recognition process. Again, setting this option does add to system overhead. The benefits really depend on how many files aren't recognised by other methods, (such as name, type or filename extension).

4. Allow tear-off menus

Currently FileMan does not fully support tear-off menus. This option allows you to turn on this functionality, allowing you to detach menus from the window. However, such menus are not context sensative at the moment, and so it is currently not recommended that you use this feature. – Picture Here – The picture above shows how the menus appear when they can be torn off. Simply click on the broken line to "tear off" a menu.

5. Show deleted files

When selected, if a trashcan is in use, and contains any files, whenever a file which is deleted is in the current path, it will be highlighted in red. As can be seen from the below menu, if you have removed a file in "/usr/bin" then when in "/usr", the directory details for "bin" will be shown in red, indicating that somewhere in that directory, (or descendent sub-directories), there will be deleted files.

There is very little overhead in this functionality, and it really depends on whether you wish to know where deleted files are.

6. Show directory details

When this option is selected, directories as well as the other types of file system object will be displayed in the right hand side of partition windows. This means that it is possible to show the directory attributues, rather than just the name.

7. Show hidden files/directories

When selected, any files or directories which begin with a "." will be shown in both the left and right hand sections of partition windows. By default this option is turned off.

8. Auto Show help details

When turned on, if a requester has some associated helptext, this will appear in a window. Currently this is a little crude - hopefully a better version will appear in version 1.2. Also, not all requesters actually support this option.

9. Use Image Thumbnails

This enables the new version 1.1 feature of showing thumbnail versions of supported graphic files. By default, "GIF" format files are recognised, but only those with the correct "gif" extension.

However, if the img package has been installed, (which should be included in the distribution for version 1.1), then you may have access to other file types - this can be used to recognise both "JPEG" and "PNG" files.[1]

---

[1]This additional file types depend on their being suitable shared object libraries for the format in question - i.e "libjpeg" for jpeg type files.

3.4.9.2. Colour Scheme

Allows you to change the current colour scheme. At the moment there are only a few colour schemes to choose from. Hopefully additional ones will be added in the future - personally I'd like a light green based colour scheme...

3.4.9.3. Style Details

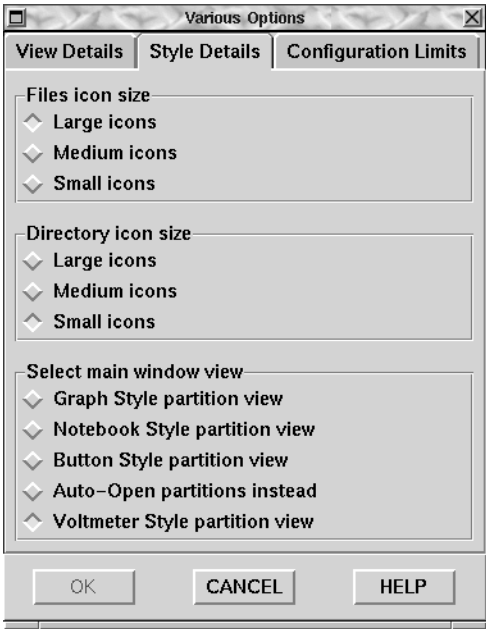

Figure 3.25. The Various Options Requester (2)

Files icon size

This allows you to choose from three sizes of icons for the files to display in the right section of partition windows. Please notice that only when the large icons are chosen will the "image thumbnails" functionality actually function.

Directory icon size

Similarly, this allows you to choose the size of the icons for the directories on the left side of the partition window. The two icons sizes can be different if you wish - a feature which is new for version 1.1.

Main Window View

This allows you to change the initial window - currently there are four supported views. The first one, the partition view shows information on the type of filesystem, and graphically how much space is available and used on each node.

The second available view is a "note-book" style view. When this is used, a page for each file system will be available.

The third type of view is the "button" view, where the information is shown as a "button" for each available partition. Clicking on the button will pop up a partition window for each particular file system.

The fourth type of view which is available is the "Auto-Open". When this is used a window will pop open as for each file system as soon as the software starts - to be used with caution if you have more than a few partitions!

The final type of main window display is the piechart type view. This is only available when the following packages can be found, (they are contained in the distribution anyway). This will show a 3D piechart for each file system which is available. Currently the utilisation of filespace is shown, along with the partition name. As you may expect, clicking on the piechart icon for a file system will pop up a partition window for that particular filesystem.

3.4.9.4. Configuration Limits

This page allows you to change the limits for various settings in FileMan. Please notice that changing these limits can increase the overheads of using FileMan. To ensure that no stupidly large values are configured, there are currently certain limits on the settings which can not be exceeded. Whatever changes you make, please be aware of any warnings given in the information below!

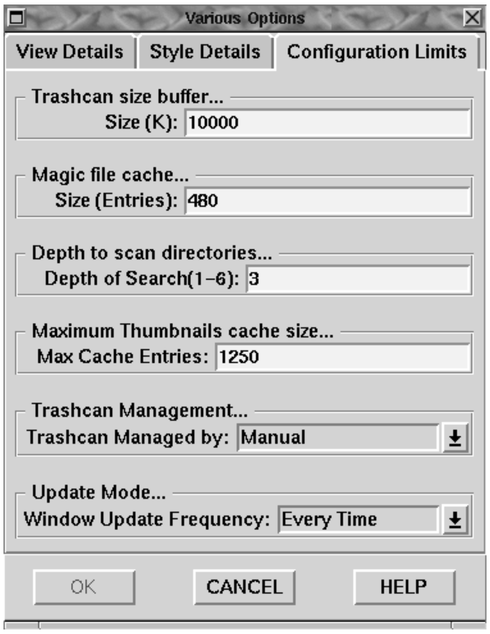

Figure 3.26. The Various Options Requester (3)

Transcan Buffer Size

This allows you to specify the maximum size for the trashcan buffer before it is considered "full". The recommended range of sizes is between 250 KB and 10 MB. Whichever size is chosen, please be aware that the trashcan for each user resides in a subdirectory in the home directory. Thus, if you have 10 users with a 10 MB trashcan each, expect up to 100 MB of storage to be allocated to trashcans!

In this case, you may wish to remove this menu option from the user's configuration - or change the maximum trashcan size allowed in the global startup file. Currently if you set the maximum above the limit this will only be effective for the current session of FileMan. When

FileMan is restarted the trashcan will be resized down to 10MB with the following startup message:

```
Defaulted Trashcan buffer to 10000
```

Magic File Cache

The "Magic" File feature has to examine the contents of each file to try and match against the 1,800 patterns of header information it recognises. Obviously, this can be quite slow, so to ensure the minimum overhead is incurred with this functionality, the results of this process are cached. Each cache entry is simply the text of "Magic" processed ready to be a recognised file type.

Thus it is possible to specify the size of the cache. **Please note that currently the cache is not cleared down to this size, and grows continuously.** The limiting of the cache size will be a feature implemented in version 1.2. However, this is only likely to be a concern when the "Magic" facility is used on small memory machines. However, any machine which can meet the requirements specified in the "Requirements" section of the manual should not have any problems.

Depth to Scan Directories

Currently this facility is not available, (it was available in the earliest versions), but should make it back for version 1.2. It allows you to specify the depth of recursion when running the "Recursive Directory Expand" button in the partition window.

It can be set between the values of 1 and 6. When implemented it is recommended that a value of 3 be used unless you wish to wait a long time when you use this feature!

Maximum Thumbnails Image Cache size

This is used to specify how many thumbnails imaged may be cached between cache clear downs. The cahce is cleared down to half the size of the full cache if it has overflowed. It clears down the cache entries by removing the oldest ones. It is recommended that the number of entries in the cache be set to at least **twice the maximum number of images in a single directory**.

However, each image is cached in a directory called ".cache" in the user's home directory. Each thumbnail image is cached as a PPM file, (Portable PixMap), and typically takes 10 KB. Thus, setting an image cache of 1000 images, (which is realistic), you could you more than 10 MB of space for the cache!

For security reasons, each user has their own cache. However, in version 1.2 there should be the ability to share the cache. This means that the cache size will not be able to be changed in this manner.

Thus, please be aware that setting this to a large number could result in a large amount of temporary files being generated in the file system used for users' home directories. Currently there is no limit to the size of the cache you can allocate - it will be limited to 10,000 in version 1.2 - which is 100 MB - more than enough for most users!!

Trashcan Management

This allows you to specify how you wish the trashcan be to managed. There are cyrrently two supported options - "Manual" and "Automatic". When set to "Manual", when the transcan becomes full a requester will pop up and ask you to delete enough files to reduce the utilisation until it is under the specified limit. This functionality is explained elsewhere in the manual.

The other option, "Automatic", means that once the trashcan utilisation increases beyond the size of the trashcan, the oldest files will be auotmatically deleted until the utilisastion drops below the threshold. Obvioulsly this is much less hassle, but does mean that you might loose files from the trashcan without any knowledge.

The advantage of the "Manual" option is that it will suggest which files should be removed, (the oldest ones), and allows you to remove them all with a single click.

Chapter 4. Customising your Configuration

This chapter attempts to explain some of the available facilties which the package offers to allow each user to change the configuration and functionality of the program. This is closely related to the following chapter on how to set system defaults which each user will not be able to override.

4.1. What can actually be customised?

Actually most things! One of the main reasons I actually have attempted to write a file manager was to ensure that it could be customised to suit most peoples need. That means to say they sure you can add a new type, so that ".pcx" files are handled correctly. You could ensure that the text editor used everyway is "emacs" rather than then "vi".

However, such things are quite common among file managers anyway. Ignoring all the standard configurable options via the menu interface, it is possible to go beyond the conventional. You could add a new menu option, take an existing one away, or simply change an existing one in a variety of ways.

If you want you can ensure that only certain users can use the product, and not others, you can! If you want to ensure that user's are only able to use "Fileman" on their home directory and "/tmp", you can!

This section is quite short at the moment, but will be expanded as later versions of the software are released. However, it should give you enough information to allow enough customisations to suit most people!!

4.2. Customisation Example 1: Viewing Directories

As mentioned in the second chapter, the default action when a directory is selected and the "View Files" menu option is used used is to try and use the "pg" program to show the directory contents. The aim of this first example on customisation is to show you how this functionality could be improved. At the moment customising menu options is a matter of changing the actual menu configuration file by handing - not as daunting as it might seem!

The first part of actually changing the configuration is to recognise which file must be changed in the first place. The configuration data for menus is usually kept in your home directory, with the following file name:

```
.ufm_menus
```

This configuration file is a text file and can be edited by your favourite editor - you could even use the "Edit files" option in "FileMan" to edit it. The file is fairly structured, and the section you must change is shown in the picture below:

```
File O "\""View Files"\""  F "\""Use relevant tool to for
each file individually."\""
        " __DEFAULT__ {xterm -e more __ARG__}  "
        " tar {xterm -e do_then_wait { tar -tvf __ARG__} }   "
        " tgz {xterm -e do_then_wait { tar -tvzf __ARG__} }  "
        " ps {gs __ARG__}  "
        " eps {gs __ARG__}   "
        " dvi {xdvi __ARG__}  "
```

As you can see, the actual contents of the file are reasonably structured. To add the extra functionality we want, add the following file below the line with the "__DEFAULT__" word starting it.

```
" !directory {xterm -e do_then_wait { ls -l __ARG__} }  "
```

Once this file has been saved with the changes, restart "FileMan" and notice the difference when the "View Files..." option is selected when a directory is highlighted!

Admittidly this is a very simply example, others can be a lot more complex, and the idea behinding adding additional functionality to an existing menu, can be easily extended to adding or removing extra menu options. However, before we continue too far, we should examine the syntax of the example just given.

The first word is way in which a match to a particular file or directory selected should be matched. The next chapter shows all possible values, but usually this is the extension of the file type to match, such as "doc" for files like "review.doc". In this instance it is set to the special word "!directory", which simply matches an object **type** of directory.

Following the text used to represent the type of filesystem object to match comes the actual command to run if this type of object has been selected for this particular menu option.[1] The command must be a single word, and must be enclosed in braces. Within the braces the command to be run should be given. This can be any UNIX command.[2] You will also notice that since the command we wish to run does not produce it's own X window, we must run it from an X terminal.

The "-e" argument to "xterm" means that all following arguments are treated as the command to run - in this case it is

```
{do_then_wait ls -l __ARG__}
```

Why the brackets? The reason for these is to ensure that the command parser in "FileMan" interprets the contents following the "-e" flag as a single command. The brackets will be removed from the final command. To get a clue why all this nonsense is actually need it, use the "View

[1]By now you are probably realising how much effort is required to added lots of new menu options, file types. Later versions of "FileMan" hope to address this by being able to define all "object" information in the same location.

[2]Which should be mapped to your ".ufm_conf" file - again see the next chapter...

Log" option on the "Options" menu, and check the command that this actually runs - it will lool something like the following:

```
12/12/97 00:33:56 {502 filem} /bin/sh -c \"cd /;"
   /usr/X11R6/bin/xterm -e
/usr/local/fileman/bin/do_then_wait
   ls -l /boot\""
```

Notice how all the commands have been given full pathnames? This is the reason for the internal parser - it allows the configuration files across multiple architectures to be the same, and maps the commands to the relevant command name - it can even by a different command! For example, under AIX the above command would map to "/usr/bin/X11/aixterm". How these actual commands are mapped is shown in the next chapter, but if you tried to use a command which did not have a suitable map entry, you would probably have seen the below error...

Finally, all the "do_then_wait" command actually does is to run the command passed to it as arguments, which in this case is "ls -l __ARG__". Finally the "__ARG__" value is replaced by the current matching file or directory before the actual command is run... That's the basic concept, other examples will hopefully add to your understanding and as stated endlessly, the next chapter also contains a complete reference to the menu file syntax.

4.3. Customisation Example 2: Removing the "Games" menu

You have decided that you don't want to play games .. ever. You also use "FileMan" but are constantly tempted by the "Games" menu option and have decided it must go!! How do we achieve this goal? **Easy!** Like the previous example we go back and edit the ".ufm_menus" file. Near the bottom of the file, (the file is parsed top to bottom, surprise, surprise), you will reach the following section...

```
Games O "Tetris" A
        __ALL__ {xtetris}
 Games O "Mahjongg" A
        __ALL__ {xmahjongg}
```

Simply remove these lines from the configuration file. On re-starting "FileMan" you will notice that the "Games" menu has been completely removed. It is just as easy to add a menu option. For example to run "xdoom" you could add the following menu option:[1]

[1]Remember that the command must be registered ... refer to the next section for details.

```
Games O "Tetris" A
        __ALL__ {xtetris}
 Games O "Mahjongg" A
        __ALL__ {xmahjongg}
 Games O "Doom" A
        __ALL__ {xdoom}
```

Please notice that any changes to this configuration file will only make changes to your personal settings. Removing the "Games" menu from everyone's menu is possible, but a little more complex.

4.4. Customisation Example 3: Registering New Commands

The example in the previous section regarding adding the "Doom" menu option would only add the menu option. When you tried to select it you would get the following window appearing, indicating that the command is not currently registered!

Thus the command must be registered as a command. To do this you must add an entry to the following file:

```
.ufm_conf
```

Why the hassle? Basically the use of command registering is that it allows users the ability to customise the standard features without effecting everyone else, and it allows the administrator the ability of ensuring that no-one can run certain commands - if you define a mapping to doom, he can override it. Alternatively, since the command sessions are logged, he could let you use it, but be able to monitor what you use...

The format of the user's configuration file which contains all the commands which they are able to run, can be seen from the following fragment:

```
" man           /usr/X11R6/bin/xman "
" file          /usr/bin/file "
" mkdir         /bin/mkdir "
" sh \t-c\t/bin/sh "
" ksh\t-c\t/bin/ksh "
" rm            __BASEDIR__/bin/my_rm"
```

As can be seem, the format of the file is fairly easy to understand. The format of each line is the same, and consists of between one and three fields. The first field is the name of the command to register. This typically is the name of a standard UNIX command such as "cat" or "wc". Following this there usually should only be one field - the actual command to run when the command is referenced.

The command to run is checked as part of the startup procedure. If it does not exist, the command is removed from the list of valid commands to run. It is therefore important that if the argument is given, it is given correctly!

If the line only has one argument, or the other argument appears to be a "flag", such as "-c", then the startup sequence will try to locate the command in your current **PATH**. Again, if the command can not be found, you will not be able to use it.

The optional "flag" argument only needs to be used when commands can execute sub-commands. As another example, consider the entry for the `xterm` command, in an "AIX" environment:

```
" xterm   -e      /usr/bin/X11/aixterm  "
```

The "-e" flag is used with the `xterm` command to indicate what follows should be taken as the command to run in the `xterm`. This can be seen in another of the configuration files, (which will be discussed next), called "`.ufm_types`". Consider the following entry:

```
" :ext(c)=gfm_c_3d {ForestGreen} {xterm -e {txtedit __ARG__}} {'C' source file}"
```

If this looks a little complicated don't worry - simply note that how the flag in the `xterm` command is followed in brackets by another command. By knowing the flag from the `.ufm_conf` file, the parser is able to realise that the command in brackets is actually another command to parse.

Following this, the `.ufm_conf` file it is quite simple in comparision. To actually change or add an entry to the `.ufm_conf` configuration file is as easy as follows: To register the "xdoom" command for your new menu option, you could add a line such as the following:[1]

```
" xdoom   /usr/bin/X11/xdoom  "
```

4.5. Example 4: Adding an archive manager

The aim here is to change the default "double-click" functionality for tar archives. Rather than extracting all files from the archive, it should pull up an GUI front-end, allowing individual files to be extracted. The solution should also be generic enough to add additional types of archive support.

What needs to be changed?

The first change is to look at the current ".ufm_types" entry for a "tar" archive.

```
:ext(tar)={gfm_tar_3d} {black} {#tar xvf __ARG__} {Standard
Tar Archive}
```

[1] Of course, if you did not know the path, you could simply have added "xdoom" to the file. However, it is recommended that you put the full, correct path in - makes startup a little quicker when you have a large number of commands!

So, the default action at the moment is to run the external tar command and extract all the files into the current directory. We are going to replace this with the following entry:

```
:ext(tar)={gfm_tar_3d} {black} {#archandle tf,xf,__ARG__}
{Standard Tar Archive}
```

This means that it will attempt to run the external command "archandle", with a long complex argument, which includes the name of the archive to manage. The next stage is to register this new command in the ".ufm_conf" file as follows:

```
archandle {__BASEDIR__/src/__ARCH__/ufmx -X
__BASEDIR__/bin/archive_mgr.tcl -Xargs }
```

This command may look a little bizare, but what it is actually doing is registering "archandle" as a command to run the ufmx executable for the current architecture! The "-X" flag means that it should be run as an interpretter for the tcl program following the "-X" argument.

Finally the "-Xargs" command means that the single argument following the command will be passed through to the interpretted script as a global variable "Xargs".

Obviously the final thing to do is to actually write the script which handles are archiving. Below is the listing of this script. It is basically "tcl", but does include use of some of the extensions, and also makes use of some of the procedures available in the auto-load modules.

```
# This is a tcl file which should be run from the
# ufmx executable - ensures it has a valid
# environment...

tix config -scheme TK
tix addbitmapdir /usr/local/fileman/fsystems
tix addbitmapdir /usr/local/fileman/bitmaps/system
tix addbitmapdir /usr/local/fileman/bitmaps/toolbox
tix addbitmapdir /usr/local/fileman/bitmaps
tix addbitmapdir /usr/local/fileman/pixmaps
tix addbitmapdir /usr/local/fileman/3Dpixmaps
```

This first section simply sets the colour scheme and ensures that "Tix" has the list of required directories to look for pixmaps. Obviously this needs to appear in all scripts written using "ufmx".

```
# Split the Xargs variable into its relevant parts ...
global Xargs
set arglist [split $Xargs ","]
```

Now we are given access to the Xargs varialbe. Since this will only be passed a single argument (no spaces), we use commas in the "archandle" command to pass in several arguments. Here we split this argument in a list for further manipulation.

```
# Now we might as well gain access to the standard tcl
# functionality existing in the function libraries...
# By sourcing the auto_index entries required.
set dir "/usr/local/fileman"
DesSource /usr/local/fileman/src-tcl/tclIndex
```

These section will ensure that all auto-load procedures exported by the standard product are available to this script. Not necessary, but useful it you want to use some of the widgets generated by that code.

```
set cdir        [pwd]
set command     [lindex $arglist 0]
set viewargs    [lindex $arglist 1]
set extractargs [lindex $arglist 2]
set archive     [lindex $arglist 3]
wm title . "Archive Manager: $archive"

# puts "command=$command"
# puts "viewargs=$viewargs"
# puts "extractargs=$extractargs"
# puts "archive=$archive"

# Process ID - required for command parsing,
# as well as null log files.
```

This section is specific to this script - it takes the arguments and puts them into variables. Note the argument passed through Xargs has four parts:

Command The command to run to manage the archive (i.e. tar)

viewargs The arguments required to the command to get a list of files contained in the archive.

extractargs The arguments used to extract files from the archive, using the command.

archive The name of the archive that is to be managed.

```
rename exec oldexec
set PID [pid]
set ErrorLogFd ""
set AuditLogFd ""

# Command registry - just include the details
# from the user's home directory for the moment.'

global COMM
set COMM(__DEFAULT__) __DEFAULT__

if { [file exists $env(HOME)/.ufm_conf] } {
        set_COMM_entries $env(HOME)/.ufm_conf
}
```

This section is required. The standard "exec" command is renamed, allowing the command parser to function correctly. It requires PID to be set, as well as blank auditing and error logging descriptors.

Finally, we run the "set_COMM_entries" command to register all the commands in the ".ufm_conf" file from the user's home directory.

```
# Load in type details to access the font setup

global env
catch {source $env(HOME)/.ufm_types.tmp}
```

Load information on the support types - just to be complete. Best to include it, just in case we need it. From here on down, it is just a tcl script as you would expect it.

We use commands like "execlog" and "mess_frame" to run commands and pop up message frames from the auto-loading functions. There are also others used for progress monitors, displaying entry frames etc.

More of the available procedures will be described in an appendix for later releases of this documentation.

```
mess_frame .mes
"This is the archive manager - a program that allows you \"
"to extract specific files from an archive. Currently this system \"
"supports archives created by tar." 1

proc getcontents {} {
global viewargs command archive

        set com "$command $viewargs $archive"
        regsub "@" $com "<" com2
        set x [execlog "$com2" "Unable to view archive contents!"]

        return [lrange $x 1 end]
}

# Show all the files in the archive ...

proc showcontents {files} {
global listbox

        set_cursor hourglass
        foreach ff $files {
                $listbox insert end "$ff"
                update
        }
        set_cursor {}
}
```

As you can see from the next procedure, extracting the files is simply a matter of using the commands passed on the command line.

```
# Extract all the selected files to the current directory

proc extract_files {} {
global command archive extractargs

        if { [catch {set sel [selection get]}] } {
                return
        }

        set cc [llength $sel]
        makeprogmon "Extracting Files" "Extracting file ..." $cc

        # List of files which need to be restored ...
        # Since we don't know whether the command'
        # takes multiple arguements, lets do one at
        # a time (painful, but reliable...)

        set cd [get_entry_value .cdir]
        puts "cd=$cd"
        set ccn 0
        foreach y $sel {
                updateprogmon "Extracting file $y" $ccn
                set com "cd $cd; $command $extractargs $archive $y"
                regsub "@" $com "<" com2
                [execlog "sh -c \"$com2\"" "Could not extract $y"]
                incr ccn
        }
        destroyprogmon
}
```

Now actually create the widgets containing the details. Here we use functions available rather than building everything from scratch.

```
entry_frame .archive "Filename:" $archive "Archive Managed"
entry_frame .cdir "Directory:" $cdir "Extraction Directory"
CHOOSE_DIR

frame .listf
pack .listf -padx 5 -pady 3 -fill both -expand 1
listbox .listf.l -width 60 -selectmode extended \
        -yscrollcommand ".listf.s set"
scrollbar .listf.s -orient vertical -command
".listf.l yview"
pack .listf.l -side left -expand 1 -fill both
pack .listf.s -side right -expand 1 -fill y
set listbox .listf.l

# Add the buttons at the base of the window ...

set split_v2 0
set a(0) "Restore"
set a(1) "Exit"
set s(0) 1
set s(1) 1
set v3(0) "set split_v2 1";
set v3(1) "set split_v2 2";
global split_v2
button_frame .buts a s v3
```

Now we have actually created all the widgets required, including the buttons using the "button_frame" functioni, we can get on with actually displaying the contents of the archive.

```
set files "[getcontents]"
showcontents $files

# Now go into a loop and wait for choices

while { 1 } {

        set split_v2 0
        tkwait variable split_v2
        if { $split_v2 == 1 } {
                extract_files
        }
        if { $split_v2 == 2 } {
                exit
        }
}
```

That's about it for customisation now. More in later revisions of this text!!

Index