Table of Contents

12. Maintenance 12-3

12.1 All about Maintenance 12-3

Personnel 12-3

Training 12-3

Documentation 12-4

Document Classifications 12-5

Support 12-6

Enhancement Report (S- ENH) 12-6

Document Enhancements (S-DOCE) 12-7

Maintenance Contracts 12-7

12.2 Files 12-8

Listings, Finding and Displaying Files 12-8

File Structure 12-8

Configuration Directory 12-9

Graphics Directories 12-9

History Directories 12-9

12.3 Environment Variables 12-11

Motif Resources Fine Tuning 12-12

File Handles 12-12

Backing up Configuration Data 12-12

Restoring Configuration Data 12-13

Resetting the Configurator 12-13

Configuration Reports 12-13

Error Windows 12-13

Backing up and Restoring Trend Data 12-14

A typical backup tar command to backup trends to a floppy drive 12-15

Recreating Indexes 12-16

Tar command 12-16

System Files 12-17

12.4 Network Considerations 12-18

12.5 Programs (Processes) 12-18

Display processes (using the ps command) 12-18

MacroView Programs 12-19
Start-up Programs 12-20
PC Start up Programs Example (SUN shown) 12-21
12.6 xops3 Command Line Options 12-22
Troubleshooting 12-22
Performance checks (Using ps) 12-23
System Requirements 12-24
Performance Checks (Using Vmstat) (Unix Only) 12-25
Meta file Performance Checks 12-26
Meta file Performance Reports 12-26
Disk Usage (df command) 12-27
Source Diagnostics 12-27
Network Diagnostics 12-28
who 12-28
ping 12-28
testterm 12-28
Further Reading 12-29

12. MAINTENANCE

12.1 All about Maintenance

Like all process control systems, your ***MacroView*** system will continue to evolve as new points and functions are added.

This chapter describes how you maintain the system and provides background information on file structures, programs and the utilities provided with the system.

Personnel

We recommend that one person is appointed the System Administrator.

The System Administrator is responsible for the integrity of the system and is assigned the super-user status with the login name **root** (Administrator for NT).

The super user is charged with:

- Assigning the various user accounts.
- Maintaining the security of the system.
- Assigning the resources - e.g. consoles, software licences etc.
- Other system administration tasks.

This chapter on Maintenance is of special importance to the System Administrator.

The System Administrator assigns the other users in the system, (Engineers, Operators etc.) the resources they need to carry out their day to day procedures.

Training

Regular training programs are offered by your distributor both on-site or in the training school.

The table below shows the `Strongly Recommended', `Recommended' and `Desirable' training classes for the various types of users.

Table 1: Training Class Recommendations

User	Engineering 4 days	Operations 1 day	Advanced Engineering 4 days	System Admin 5 days
System Admin	***		*	**
Engineers	***		**	
Operators		***		
Managers	*	***		

Legend: Strongly Recommended : ***

Recommended : **

Desirable : *

Your distributor can provide more details on the individual training courses. The table below summarises the content of each of the training classes.

Table 2: Training Class Content Summary

Course	Objectives
Operations Course (1day)	The student becomes comfortable with all the operations of the system and is able to use the facilities automatically and efficiently.
Engineering (4 days)	The student can configure a complete ***MacroView*** system from scratch to a working process control environment. Covers hands-on configuration of the subjects in this document including Entities, Sources, Graphics etc. (Also introduces UNIX and dBase.)
Advanced Engineering (4 days)	The student consolidates the knowledge gained in the Engineering course and can harness the advanced features of ***MacroView*** in practical high-return applications such as Advanced Control, Logging and Reporting, Optimisations, Interfacing, SPC etc. (based on hands-on exercises).
System Administration (5 days)	The student can comfortably set up and manage a UNIX based network. The course is an intensive hands-on course and it covers all the aspects likely to be encountered in a typical UNIX system. (The course is usually held by the hardware vendor.)

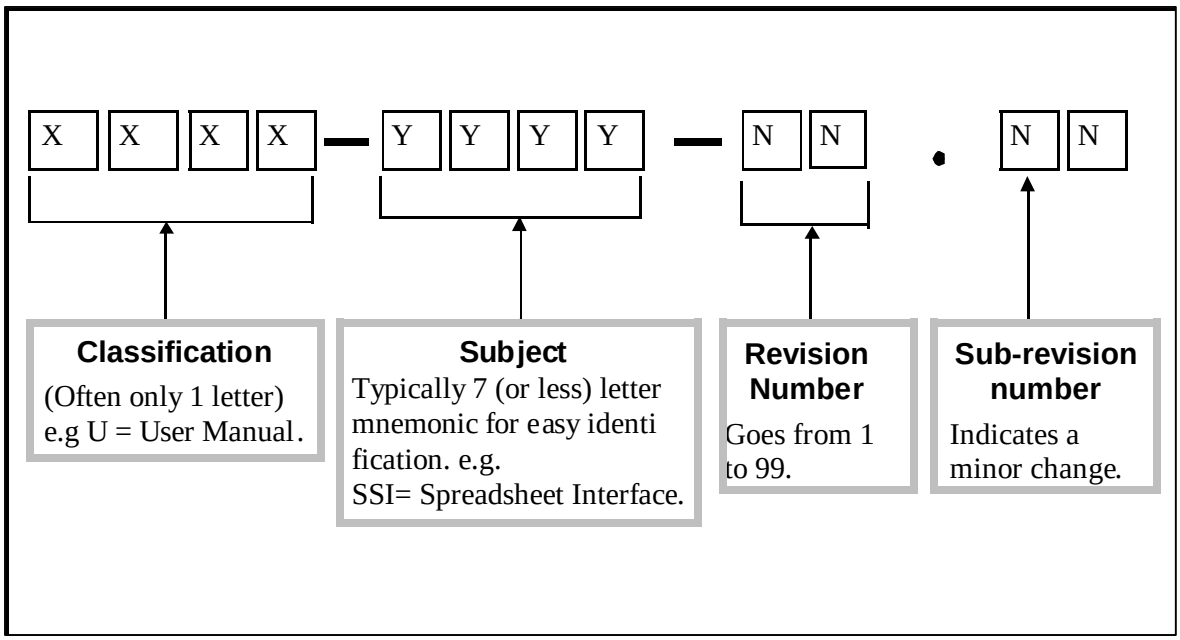
Documentation

You can obtain support documentation from your distributor.

The range of documents is broken up into various sections, for example:

Support, Training, Applications etc.

Each document has a unique number that identifies its classification, subject and version number as follows:



You can use the document number as follows:

- Use the classification to find the level of documents. For example, **P** for Packages, **G** for General.
- Use the subject code to identify the document itself, and then
- Use the revision number to check that you have the latest document.

Check the next page for a summary of the classification codes.

Document Classifications

The table summarises the various classifications of documents you are likely to use.

The Appendix "***MacroView*** Documentation" lists some of the documents that are available at the time of printing. For an up-to-date list of documents, please consult your distributor.

Table 3: Documentation Classification Codes

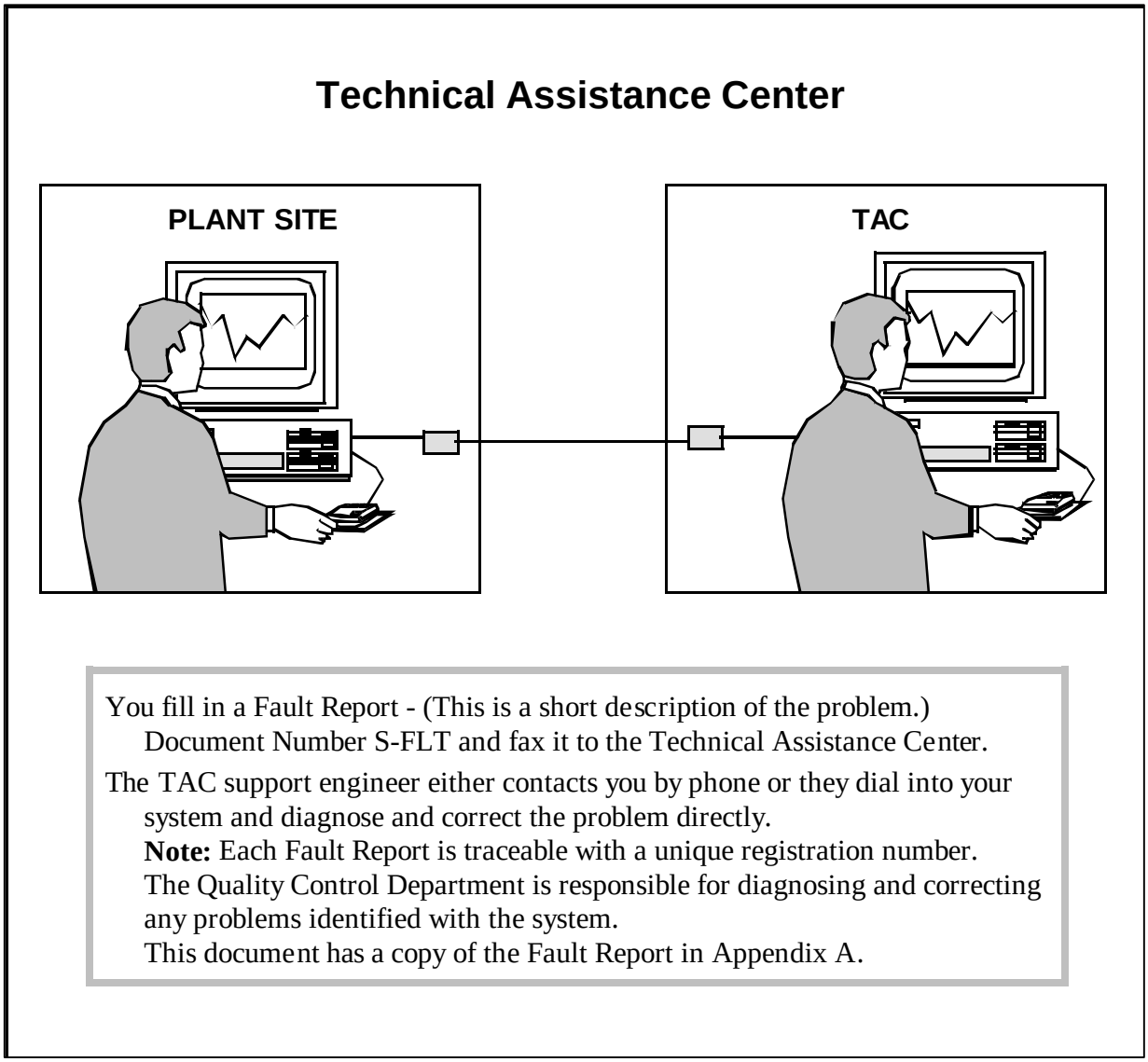
Code	Classification	Description	Example
G	General Overview	Broad coverage of subject. (Usually marketing.)	G-PWB Plant-wide brochure.
P	Package	Document that describes a package	P-SSI Spreadsheet interface document.
U	User Manual	Document designed to provide all the information required to use the package	U-NAV The Navigator User manual
D	Drivers	Driver document	D-MODF MODBUS FEP description.
T	Training	Training document	T-XCOM Common Xenix commands.
A	Applications	Application notes	A-CRON Scheduling using Cron.
C	C Type Documents	Detailed technical data on programs	*Setpoint(C)* Setpoint driver.
F	Files	Technical data on file structures	*Sources(C)* Sources file structure
Q	Questions & Answers	Questions & Answers document	Q-LTBI Lotus buy-ins
S	Support Issues	Support Functions	S-VEND Vendor lists.
SS	Specification Sheets	Small specification sheets for each package	SS-SSI Spreadsheet interface specification
AD	System Administration	Detailed system administration	AD-XIN Xenix installation
SP	Specification	Design specifications	SP-PEG Pegasys Interface specification

Support

The ***MacroView*** system is supported using the TAC philosophy and the Fault Reports (S-FLT). Refer to the Appendix.

The TAC (Technical Assistance Centre) looks directly into your system via a dial-up modem.

The diagram below shows how the TAC system operates.



Enhancement Report (S- ENH)

- If you have any suggestions as to how you would like to see the system improved, then you may document your suggestions by filling in the Enhancement Report (document number S-ENH) and fax this to our Development Manager.
- All such Enhancement Reports are used by our development team to create an improved product which is more aligned to your requirements.

- You can get a copy of the enhancement report from Appendix A.

Document Enhancements (S-DOCE)

- If you have any suggestions as to how to improve this or any other ***MacroView*** document, please complete the Document.
- Improvement Form (S-DOCE) and fax it to our Quality Assurance Manager.
- Documents are continually being revised and improved and your assistance is greatly appreciated.

Maintenance Contracts

- Various levels of Maintenance Contract are available from your distributor.
- These range from a high level of support with periodic visits to site, down to a "pay-as-you-use" TAC support contract.
- Some contracts also include software upgrades and enhancements.
- Please contact your distributor for details on these Maintenance Contracts.

12.2 Files

This section describes the various files that are important to the ***MacroView*** system.

It also describes the back-up and restore procedures associated with these files.

Listings, Finding and Displaying Files

- To **list** the files in a given directory, use the `ls` command as follows:

  `ls` lists the files in the current directory.

  `ls <path> <filespec> <arguments>` lists files in the path that comply with the filespec.

- The arguments that you can use with the `ls` command are:

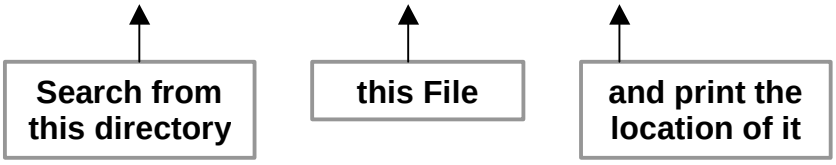
  `-l` give a long listing. i.e. show permissions, owners etc.

  `-a` show invisible files.

  For a more detailed description of the `ls` command, just type `man ls` at the UNIX prompt.

- To **find** the files use the **find** command as follows:

```
find /u/macro -name histmsgs.dbf -print
```



- You must use a `\` for wildcards e.g.

```
find /u/macro -name hist\* -print
```

  In this case, the system will find all files with `hist` as the first four characters.

- To **simplify** the command, you can put it in an alias in your `.cshrc` file. i.e.

```
alias f `find ./ -name \`echo $1\` -print`
```

  In this case, just type `f <filename>,` and the system will find the file if it is in the current directory path.

- To **display** the text file, you use the more command. I.e. `fill9 | more`

  This will display the file on your screen.

File Structure

The diagram on the next page shows the layout of a typical ***MacroView*** file system.

Configuration Directory

Note: that all the configuration files (other than Navigator Application files) are held in the "Configuration Directory".

In the diagram, the directory has arbitrarily been given the name `config`.

You may call this directory any name you choose - for example, if you had two sites configurations in the one computer, you could call one configuration directory `site1` and the other `site2`.

This directory is given the environment variable MACRODIR. All ***MacroView*** programs use this environmental variable to find the configuration data.

To change configurations, just change MACRODIR and the new configuration will take effect for your current login shell.

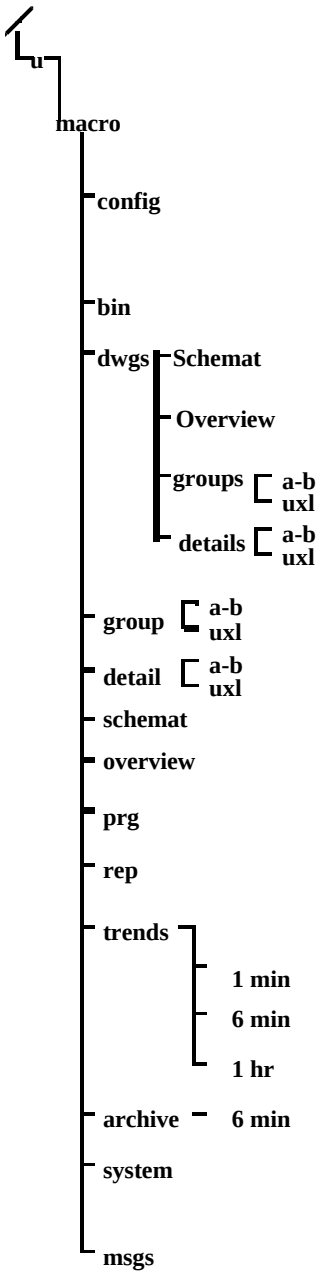
Graphics Directories

The **dwgs, groups, detail, schemat, overview** and **help** directories hold the graphics information. For more details about these files, refer to the Graphics chapter.

History Directories

The files in the **trends** and **archive** directories hold the historical data. For more information, refer to the History chapter.

Table 4: *MacroView* Files



Desription	Files
The root file system.	
The user file system, sometimes called /usr.	
The home directory of MacroView. Holds login start up files.	.cshrc .login .Xops
The configuration directory MACRODIR. Holds all the configuration files. You may call this directory any name you like. (See "Configuration Files" in the Introduction.)	entities.dbf alarms.dbf hlist.dbf
Holds all executable files: both system and user.Refer to "Programs (Processes)" on page17.	xops3, mv_histr
Holds Schematic, Overview and Help skd and dxf files before conversion.	boiler.dxf boiler.skd keys.dxf oview1.dxf
Faceplate skd and dxf files held in the directory name which is the same as the source name before conversion to the metafiles.	pid.dxf pid.skd
Detail faceplate skd and dxf files held in the <source name> directories before conversion.	pid.skd pid.dxf
Faceplate metafiles (.dgt) after conversion from dxf format.	pid.dgt
Detail faceplate metafiles (.dgt) after conversion from dxf format.	pid.dgt
Schematic metafiles.	boiler.dgt, schemat.dbf
Overview metfiles.	ov1.dgt, overview.dbf
System dBase programs.	configur.prg
Report programs (often dBase programs). Also report output files.	daily.prg daily.rep
Trends main directory. The subdirectory names depend on your historical specifications.	
One minute samples files.	199210031212
6 minute files.	1992091304
1 hour average files.	19920901
6 minute compressed archive files.	1992100330
System files and example files.	Start .Xops
Alarm and Historical messages files. Refer to the chapter on Alarms.	histmsgs.dbf alarms.dbf guides.dbf

12.3 Environment Variables

The table below shows the environment variables used by ***MacroView***. To get the values of these variables at any time, just type `env` at the UNIX prompt.

To set a variable:

(i) At the prompt type:

```
setenv <variable name> <values>
```

E.g. `setenv MACRODIR /u/macro/demo`

(ii) In a Bourne shell enter:

```
<Variable name>=<value>
export <Variable name> e.g.:
MACRODIR=/u/macro/config
export MACRODIR
```

Table 5: MacroView Environment Variables

Environment Variables	Description	Typical Value
MACRODIR	The directory name in which all the configuration files are held. Most references to files are made relative to this directory.	`/u/macro/config` `C:\users\macro\config`
DISPLAY	The name of the X display to send the graphics to.	`pc21:0` (not used in NT)
SCANRATE	The rate in seconds at which the screen is updated. This value greatly affects the loading of the graphic display running in a system. The higher the value, the less the processor is loaded. The variable is very useful in tuning a system.	`2`
FASTRAMP	A multiplication factor that indicates how much faster the Fast ramp key changes the value than the normal ramp key.	`5`
USDATE	If set to 1, a US date format is used (MM/DD/YY). If not, then a date format of DD/MM/YY is used.	
HOME	The home directory of the user.	`/u/macro` `C:\users\macro`
MV_HELP_PATH	The home directory of the on-line help system.	`/u/macro/help/lib`
EZENGDIR	The home directory of the EZ Eng application files.	`/u/macro/ezeng/dgt`
ENGDIR	The home directory for the engineering configurator.	`$HOME/eng/engdir`
NAVDIR	The Navigator home directory.	`../nav`

Motif Resources Fine Tuning

Some of the xops3 widget attributes are controlled by X resources. Several like the menu sizes, scrollbar colours and Window Title Size, (not a Xops3 widget) are important to operators, particularly text sizes which, many operators prefer to be very large. The sizes may be set in several places, there are in fact 8 methods of setting resources. The easiest location is the file `$HOME/.Xdefaults.`

The following is a typical .Xdefaults file set up to fine tune these resources. **Note** that comments start with a !.

```
! Set the window title to a large font.
Mwm*client*title*fontList:lucidasans-24
! Set the menu text to a large font.
MacroView*fontList:lucidasans-18
! Set the scrollbars and menubar colors to grey.
! MacroView*foreground:grey
```

File Handles

Any operating system has a restriction on the number of file handles that are open at any one time. The version III system uses many files and the file handle limit can be easily exceeded. If this occurs, a "Too many files" error message will be produced. Other less obvious messages may also occur.

To correct this problem, you can:

(i) Re-configure the kernel so that it has more file handles or

(ii) Re-organise the application so that it uses fewer files. In this case, the DROP VIEW command is useful.

(iii) In Solaris use the command;

```
mvserver% limit descriptors 256
```

Backing up Configuration Data

You can back up all or parts of your configuration onto floppy disk, magnetic tape or other devices using the Engineering Configurator.

To backup a system, pull down the **Maintenance** menu and click on the "**Backup Configuration**" option. A "tar" file will be created that holds the configuration.

Note: Depending on which operating system you are using, the media options will be different from that shown above.

Be careful when using the Floppy option that there is sufficient space for the files. (The tape option is preferable because of its capacity.)

With some systems, (e.g. SUN) you may prefer to use the tape utilities that come with the system.

The data is stored in standard format. I.e. it can be restored to any UNIX machine. If you want the data in DOS format, you should use the DOS emulator that comes with the

system to store the data or use the Sun `pcfs` utility and mount the drive as a DOS drive.

Restoring Configuration Data

Once you have backed up your configuration data onto a floppy disk, magnetic tape or other devices, you can restore it into the system using the Engineering Configurator.

To restore a configuration, pull down the **Maintenance** menu and click on the "**Restore Configuration**" option. A warning message will appear informing you that you are about to overwrite your existing configuration. Once you have acknowledged the warning, the configuration that was previously backed up will be restored.

Resetting the Configurator

You can reset the configurator (views and screen) by selecting the **Maintenance** pull down option and clicking on the "**Reset Configurator**" option.

Configuration Reports

The configurator provides a method of generating reports on your configuration. Reports are created and stored in text files in the $MACRODIR directory. To create a configuration report, pull down the **Maintenance** menu and click on the "**Configuration Reports**" option. You will be prompted to enter which aspects you want reported. When you have selected the options, a text file is produced. You may view the text file or print it out to the printer.

Error Windows

There are two kinds of error windows that relate to the configurator:

(i) **System Errors:** This is a window that holds a log of all stem and internal errors that are generated. Call up this pop-up window by selecting the **Maintenance:System Error** option.

(ii) **File Errors:** This is a window that holds a log of all file integrity errors that are generated. E.g. if you have files that are missing or are in an incorrect path etc. Call up this pop-up window by selecting the **Maintenance:File Error** option.

Backing up and Restoring Trend Data

The ***MacroView*** History Manager will automatically archive your historical data to the directory you choose in the historical specification part of the configuration. (See Archiving in the History chapter.)

This directory could be a removable WORM disk or it could be another drive mounted on another computer elsewhere in the network.

Once archived, the data is still available online for trending or for other applications.

If you need to backup the data to a backup device then either use one of the tape utilities provided with the system or use the `tar` command.

The tar command line arguments are consistent across platforms but the Unix devices that should be used differ.

The following table lists the common device names across the operating system platforms available.

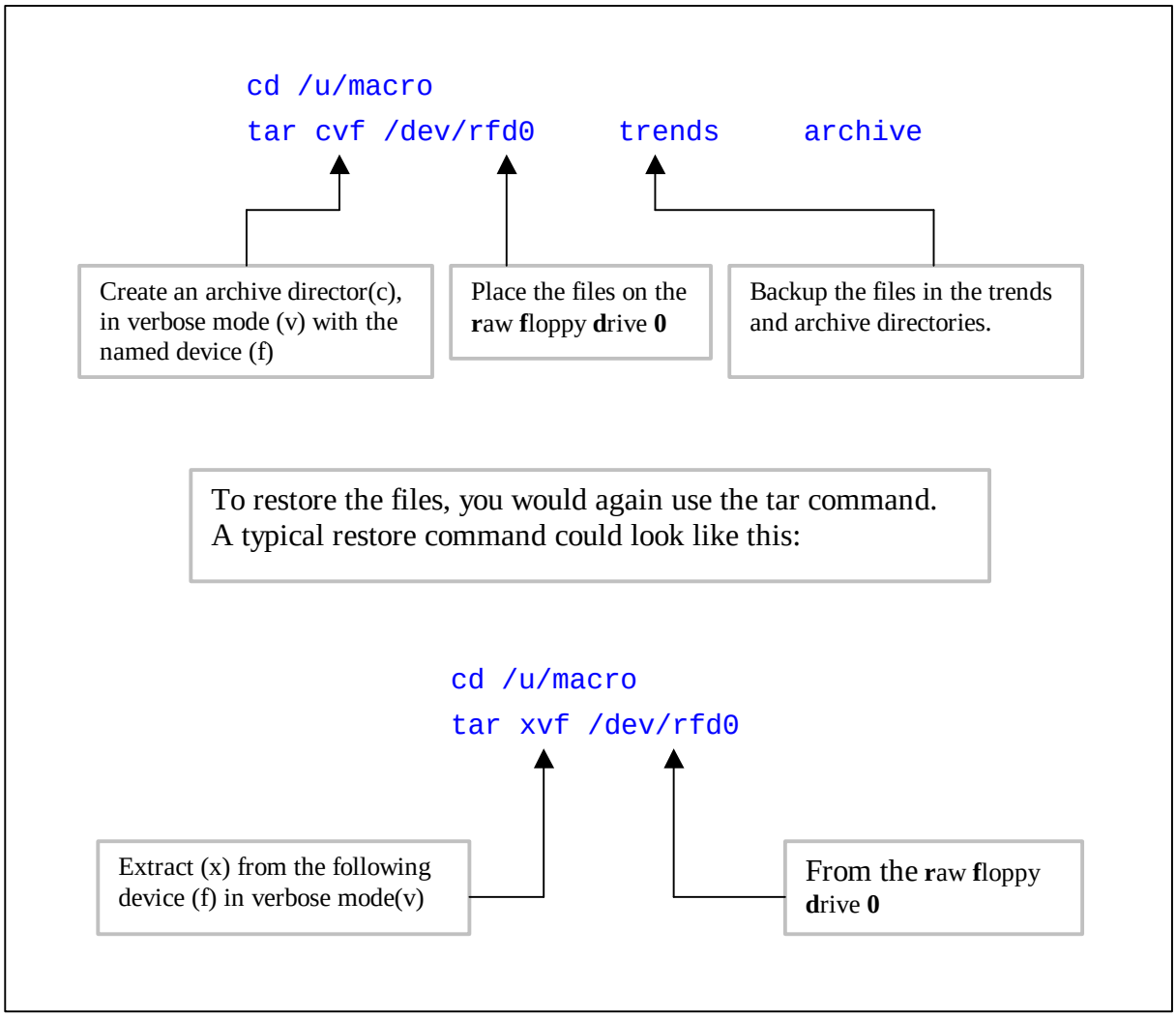
Table 6: Typical Device Names

Operating System	Device	Description
SunOS	`/dev/rfd0`	`3 1/2 inch floppy drive`
SunOS	`/dev/rst0`	`Tape drive.`
SCO Xenix/Unix	`/dev/install`	`Floppy drive A: Can be written and read from but cannot be formatted.`
SCO Xenix/Unix	`/dev/install1`	`Floppy Drive B:`
Ultrix	`/dev/rct0`	`Tape drive.`
Ultrix	`/dev/rmt0`	`Tape drive.`
Ultrix	`/dev/rrz5c`	`Floppy drive.`
HP-UX	`/dev/dsk/0s0`	`Floppy disk at SCSI address 0.`

Note: These are examples only. The devices on your system may differ depending on how it is set up.

A typical backup tar command to backup trends to a floppy drive

A typical backup tar command to backup trends to a floppy drive would be something like this.



Note: It is a good idea to back up data using relative naming conventions (like `trends` or `./trends`) and not absolute file names (like `/u/macro/trends/`). It makes it easier when restoring files to restore them in any directory required.

If required, you can schedule a backup program that contains the appropriate `tar` commands so that the backup is regular and automatic. (See the application note A-CRON.)

Recreating Indexes

All the configuration files are standard dBase format. This means that users can access the files directly using dBase.

It is possible to corrupt the indexes if you make changes to the database without using the index.

If this occurs, you can recreate the indexes by selecting the Maintenance reCreate option from the Engineering Configurator.

You can format the floppy drive by selecting the Maintenance, Format, floppy options from the Engineering Configurator.

Tar command

You can use the tar command from the Unix prompt to do various file tasks. The format of the tar command is:

```
tar <options> <filesystem> <filenames>
```

The options include:

c Create a new file system.

t List a table of contents.

x Extract the files from the file system.

f Use the next argument as the tar file device.

v Use verbose mode to log the events occurring during the tar operation.

u Update the tar file.

For example:

```
tar cvf /dev/rfd0 spc3.Z
```

This means create (c) a tar file on the device /dev/rfd0 (f) in verbose mode (v) and add the file spc3.Z.

System Files

Various UNIX and DOS system files are important to the System Administrator for maintenance purposes. The table summarises these files for the various operating systems.

Table 7: System Files

Filename	Description	Comments
/etc/rc	Program that runs when the system boots up.	
/etc/hosts (UNIX)	Names and Ethernet addresses of all systems on the network	
\nfs\hosts(DOS)	Names and Ethernet addresses of all systems on the network.	
/u/macro/.login	This program is run whenever the user logs in to the system. It sets environment variables among other things.	
/u/macro/.cshrc	This program is run whenever a new session is started by the (already logged in) user. Like .login, it sets environment variables.	
/etc/default/tar	Names of tar devices on the system.	SCO Xenix/Unix only
Cron Files	Please see the application note A-CRON for the cron special files.	
/exports	Which file systems may be exported to the network.	
/etc/ttys	List of terminal devices in the system.	SCO Xenix only
/etc/ttytab	List of terminal devices in the system.	
/etc/gettydefs	List of terminal communication settings.	SCO Xenix only
/etc/get/tytab	List to terminal communication settings.	
/usr/lib/uucp	Directory where UUCP configuration files are kept.	
/etc/uucp	Directory where UUCP configuration files are kept.	SUN OS 4.X and Solaris 2.X only
/etc/hosts.equiv	A list of all systems on a network that can execute programs "remotely" on the local machine.	

12.4 Network Considerations

There are a number of Maintenance issues you should consider when you network multiple computers together:

- Which file systems, users on the network can access from which computers? (For example, you can mount drives across the network onto Windows machines.)
- Which resources are available to which uses? - E.g. printers etc.

These considerations are described more fully in the Application notes on Networks.

12.5 Programs (Processes)

There are a number of programs that should be active at various stages in a typical ***MacroView*** system.

This section describes these processes briefly and indicates just where you may get more information about them.

Display processes (using the ps command)

You can get a list of processes running at any point in time by typing in the ps command at the UNIX prompt. I.e.

SCO Unix/Xenix: `ps -e`

Solaris: `ps -e`

SunOS: `ps -vax`

Ultrix: `ps -vax`

HP-UX: `ps -e`

`ps -e` Lists all programs running, which terminal the programs are associated with and the processor time used by the programs since they started.

`ps -vax` Lists all programs running and a large amount of other information.

The information of most use is which terminal the programs are associated with, the percentage of processor time used by the program over a short period of recent time and the amount of memory used by the program.

Note: That the programs (processes) are listed in order of highest loading.

The loading is determined by taking both the processor usage and memory usage into account.

The memory usage can vary during program operation as memory is allocated and freed as necessary.

Use the `|` and more command to display one page of information at a time.

E.g. `ps - vax | more`

Use the grep command to print the information about a single program.

E.g. `ps - vax | grep <progname>` (This is a most useful command and you can put it in as an alias in your `.cshrc` file.)

Use this, to get the information about a program (e.g. `exproc`), just type `p exproc`

E.g. alias p `'ps - vax | grep $1'.`

MacroView Programs

The table below summarises the more important programs that run during the normal ***MacroView*** operations. Comments on the processor usage time have been added.

Table 8: *MacroView* Programs

`mvcat`	A program that passes messages on to printers etc. The amount of processor time that the `mvcat` program uses is directly related to the amount of messages being processed.
`alarms`	The alarm status scanning program. The processor time this program uses is related to the number of alarms configured and the alarm scan rate used in the `system/start` file. The default is an alarm scan rate of 1 second. The alarm scan rate can be set to less than 1 second but, if it is, then the **alarms** program can use up a large amount of processor time. The alarms program will not run if no alarms have been configured. (Refer to the Alarms chapter.)
`mv_histr`	The history collection program. The processor time this program uses is related to the number of values being historized and the lowest sample rate used. Typically the processor time used by the `mv_histr` program is minimal compared to the alarms program. If there are no values configured in the `hlist.dbf` database then the `mv_ histr` program does not run. (Refer to the History chapter.)
`drivers`	Each **source** is associated with a driver. Drivers vary significantly in their effect on processor time. Please refer to the driver documentation for any special items of note.
`mshell`	The program that executes meta script programs from the UNIX prompt.
`mv_msgs`	The program that handles all ***MacroView*** messages and alarms.

Start-up Programs

Start-up programs have the function of initialising variables and starting other programs such as drivers etc.

The table below summarises the main ***MacroView*** start-up programs.

Table 9: Start-up Programs

Program (Relative to the home directory)	Start-up Functions
`system/start`	This program is run when the system first boots up. It is responsible for starting the drivers, Alarm Manager, History Manager, XL-Term daemons, `mvcat`, `exproc` etc. For more details, please see the sections "Starting Up" in each chapter.
`.login`	The log in start up program: This program sets up environment variables etc when the user first logs in.
`.cshrc`	The C shell start-up program: This program starts up whenever a new session is started by the user (unlike `.login` which is only run when the user first logs in). The `.cshrc` file is also used to set various environment variables.
`bin/restart`	This program is initiated when you select Maintenance Restart from the Engineering configurator. It is used to restart the alarms and history programs when new configuration data has been added. Refer to the History and Alarms chapters.

`bin/pcxops`
`bin/pcxterm`
`bin/pcconfig`
	These are a series of terminal programs that set up the X environments for the PC terminals and then call the main ***MacroView*** programs.
	
	These are normally started from an icon in the PC windows environment.
	
	See the section "PC Start up Programs Examples (SUN shown)" on page20.

PC Start up Programs Example (SUN shown)

PC ICONS	Icon Setup (All rsh, user name macro, hostname victrn) Client	SUN daemons

XTERM	/u/macro/bin/pcxterm/ victrn:0	**Pcxterm**
		`LD_LIBRARY_PATH=/usr/openwin/lib`
		`export LD_LIBRARY_PATH`
		`DISPLAY=$1`
		`export DISPLAY`
		`stty erase '^H'`
		`/usr/bin/xterm -fn 8x13 &`

OPS	/u/macro/bin/pcxops3 victrn:0	**pcxops3**
		`testterm -d > /dev/console`
		`MACRODIR=/u/macro/config`
		`export MACRODIR`
		`/u/macro/bin/xops3 $* -geometry 560x420+0+0 &`

CONFIG	/u/macro/bin/pcconfig victrn?:0	

dBase	/u/macro/bin/pcdb4 victrn?:0	**dBase**
		`DISPLAY=$1`
		`export DISPLAY`
		`TERM=xterm`
		`export xterm`
		`LD_LIBRARY_PATH =/usr/openwin/lib`
		`export LD_LIBRARY_PATH`
		`DBTERM=xterm`
		`export DBTERM`
		`XAPPLERESDIR=/usr/openwin/liob/xll/app-defaults`
		`export x APPLRESDIR`
		`DBPATH=/usr/local/lib/dbaseiv`
		`export DBPATH`
		`MACRODIR=/u/macro/config`
		`export MACRODIR`
		`cd $MACRODIR/../dbase`
		`/usr/bin/xterm -e /usr/local/db4 &`

Editor	/u/macro/bin/pcedit victrn0	**Editor** Same as pcxterm except the last line is: `/usr/openwin/bin/xview/textedit -fn 8x13 &`
Sketch		**Sketch** No SUN daemon but /u/macro is mounted on the PC as the S drive i.e. `net use: sparc:/u/macro`

12.6 xops3 Command Line Options

The table below summarises the various command line options that can be used when starting up the operations program xops3.

Argument	Description
`-edit`	Activates graphic object editor mode. In this mode the functionality described in the Graphics chapter is enabled. Otherwise, graphic objects can not be edited on line.
`-defaultGlobal`	By default, any references to variables that are not stated as being GLOBAL or LOCAL are assumed local to the metafile. This will result in unexpected behaviour in any applications developed using Version 3.0.2 or earlier as all variables in these versions are global. As a backward compatibility measure, the `-defaultGlobal` results in variables being assumed global unless otherwise specified.
`-reportTimes`	A ***MacroView*** engineer can request that a metafile performance report be sent to the standard output of a xops3 process be specifying the `reportTimes` option on the command line.
`-highlight`	This option results in all Meta Sprite objects being highlighted on metafile windows when they are displayed. Refer to the Graphics chapter for details on Meta Sprite objects and how they are used.
`-updateRate`	Default rate in seconds at which metafiles should be scanned. Defaults to 1. Higher numbers reduce CPU usage.
`-noConfig`	Do not require configuration databases to be present in start-up directory. A xops3 started with this option will not have access to any entities.

Troubleshooting

If you have problems with the system, there are various simple checks you can carry out to determine exactly what the problem might be.

The problems you check for and the tools you can use are shown in the table below.

Table 10: Troubleshooting Tools

Checks	Tools
Performance checks	`Ps, vmstat, kill`
Disk usage checks	`ls, du`
Source Diagnostics	`Imgstat, plcstat`
Network Diagnostics	`Ping, who, testterm`

Performance checks (Using ps)

If the system appears to be getting very slow, use the `ps` command to find out:

- If there are any processes that are running that shouldn't be running. (E.g. badly exited programs.)
  - (i) Read the process id from the `ps` listing (usually the first column) and
  - (ii) Use the **kill** command to terminate the processes i.e. `kill <pid>.`
- If there are any processes taking too much time and resources you can:
  - (iii) Compare the loading of the current processes against a printout of the `ps` command taken when the system was healthy.
  - (i) Also have a look at the comments relating to each program in the "***MacroView*** Programs" section of this document.

The following is a printout of the `ps -vax` command from a healthy SUN4.X based system.

Obviously, this printout could be very different from your system and it is a good idea to get your own "healthy" printout for comparison.

```
sparc2% ps -vax
  PID TT STAT   TIME SL RE PAGEIN SIZE  RSS   LIM %CPU %MEM COMMAND
  703 co R     1:47  0 99    718 3056 2556    xx 16.0 23.8 xnews
 1095 co S     0:05  0 86     13  336 1036    xx  0.8  9.6 xops
 1087 co S     0:05  0 99     21  340  976    xx  0.8  9.1 xops
 1083 co S     0:04  0 99     79  316  920    xx  0.0  8.6 xops
  717 p0 S     0:04  0 15    151  192  844    xx  3.6  7.8 cmdtool
 1091 co S     0:01  0 98     30  176  652    xx  0.0  6.1 xspc
 1099 p0 R     0:00  0  0      0  180  408    xx  0.0  3.8 ps
  145 ?  S     1:29 14 99    165  160  364    xx  0.0  3.4 exproc
  139 ?  S <   1:00 14 99    387  180  356    xx  0.0  3.3 alarms
  143 ?  S     6:40  0 99    107   64  216    xx  0.0  2.0 localsrc
  137 ?  S <   5:36 14 99    428   44  168    xx  0.0  1.6 mvmsgs
  723 p0 S     0:00  0 13     28   64  156    xx  0.0  1.4 csh
  117 ?  S     0:00  0 99     85  148   92    xx  0.0  0.9 atlmd.sun4
  138 ?  S     4:08 14 99      9   24   28    xx  0.0  0.3 mvcat
   86 ?  IW    0:00 99 99     19   60    0    xx  0.0  0.0 syslogd
   69 ?  IW    0:02 23  1     54   36   48    xx  0.0  0.0 in.routed
  100 ?  IW    0:00 99 99      1   56    0    xx  0.0  0.0 rpc.mountd
  691 co IW    0:00 99 99     21   68    0    xx  0.0  0.0 csh
   61 ?  IW    0:00 99 99      0   40    0    xx  0.0  0.0 keyserv
  696 co IW    0:00 99 99      6   28    0    xx  0.0  0.0 sh
  125 ?  IW    0:00 99 99     24   56    0    xx  0.0  0.0 cron
  130 ?  IW    0:00 99 99     63   56    0    xx  0.0  0.0 inetd
  107 ?  IW    0:00 99 99      3   52    0    xx  0.0  0.0 rpc.statd
  108 ?  IW    0:00 99 99      0   92    0    xx  0.0  0.0 rpc.lockd
   56 ?  IW    0:01 99 99    183   68    0    xx  0.0  0.0 portmap
  133 ?  IW    0:00 99 99      1   52    0    xx  0.0  0.0 lpd
    1 ?  IW    0:00 86 61    160   52    0    xx  0.0  0.0 init
   94 ?  IW    0:00 99 99    133  100    0    xx  0.0  0.0 sendmail
  220 ?  IW    0:00 99 99    301   92    0    xx  0.0  0.0 rpc.frameusersd
  140 ?  IW<   3:29 30  1   1289  292  428    xx  0.0  0.0 mv_histr
  141 ?  IW    0:00 99 99      3   36    0    xx  0.0  0.0 mtopsd
  714 p3 IW    0:17 74 53     94  200    0    xx  0.0  0.0 cmdtool
  716 p1 IW    0:00 99 99     20  184    0    xx  0.0  0.0 cmdtool
  721 co IW    0:00 99 99      7   64    0    xx  0.0  0.0 sv_xv_sel_svc
  705 co IW    0:00 99 99      1   28    0    xx  0.0  0.0 sh
  724 p3 IW    0:00 99 99     10   64    0    xx  0.0  0.0 csh
  722 p1 IW    0:00 99 99      6   64    0    xx  0.0  0.0 csh
  702 co IW    0:00 99 99      2   40    0    xx  0.0  0.0 xinit
  712 co IW    0:02 73 42     46  136    0    xx  0.0  0.0 olwm
  224 ?  IW    0:00 99 99    221   96    0    xx  0.0  0.0 rpc.frameusersd
  715 p2 IW    0:03 99 99     57  184    0    xx  0.0  0.0 cmdtool
  727 p2 IW    0:00 99 99      0   64    0    xx  0.0  0.0 csh
  122 ?  S     2:34 11 12     11   12    4    xx  0.0  0.0 update
   73 ?  I     0:00 99 99      0   16    0    xx  0.0  0.0 biod
  104 ?  I     0:00 99 99      0   28    0    xx  0.0  0.0 nfsd
   72 ?  I     0:00 99 99      0   16    0    xx  0.0  0.0 biod
   75 ?  I     0:00 99 99      0   16    0    xx  0.0  0.0 biod
  101 ?  I     0:00 99 99      0   28    0    xx  0.0  0.0 nfsd
  103 ?  I     0:00 99 99      0   28    0    xx  0.0  0.0 nfsd
   74 ?  I     0:00 99 99      0   16    0    xx  0.0  0.0 biod
  102 ?  I     0:00 99 99      0   28    0    xx  0.0  0.0 nfsd
    2 ?  D     0:15  0 99      0    0    0    xx  0.0  0.0 pagedaemon
    0 ?  D     0:05  1 99    545    0    0    xx  0.0  0.0 swapper
sparc2%
```

For more information on the `ps` command, please see the section "Display processes (using the `ps` command)" on page17 or type `man ps` at the Unix prompt.

System Requirements

The ***MacroView*** package requires sufficient resources to function correctly. These resources may be divided up into hardware and system resources as follows:

Hardware

Because ***MacroView*** operates in such a variety of networks, platforms and configurations, it is best to contact your local dealer for a suggested hardware configuration for your particular requirements. The hardware recommended will be sized according to the number of users, the amount of historical data, the expected update time etc. The distributor should also be able to give you a good indication of your future upgrade migration path with the recommended hardware.

Software

It is important that the operating system is set up to handle the requirements of ***MacroView***. In particular, you will require a minimum of 256 file descriptors. In some of the older operating systems, this requires that the kernel is re-built (UNIX only).

For more information on these requirements, please contact your systems administrator or your local ***MacroView*** distributor.

Performance Checks (Using Vmstat) (Unix Only)

The virtual memory statistics (`vmstat`) program contains a large amount of information about your Unix system:

- You can get more information about `vmstat` by typing:

  `man vmstat`

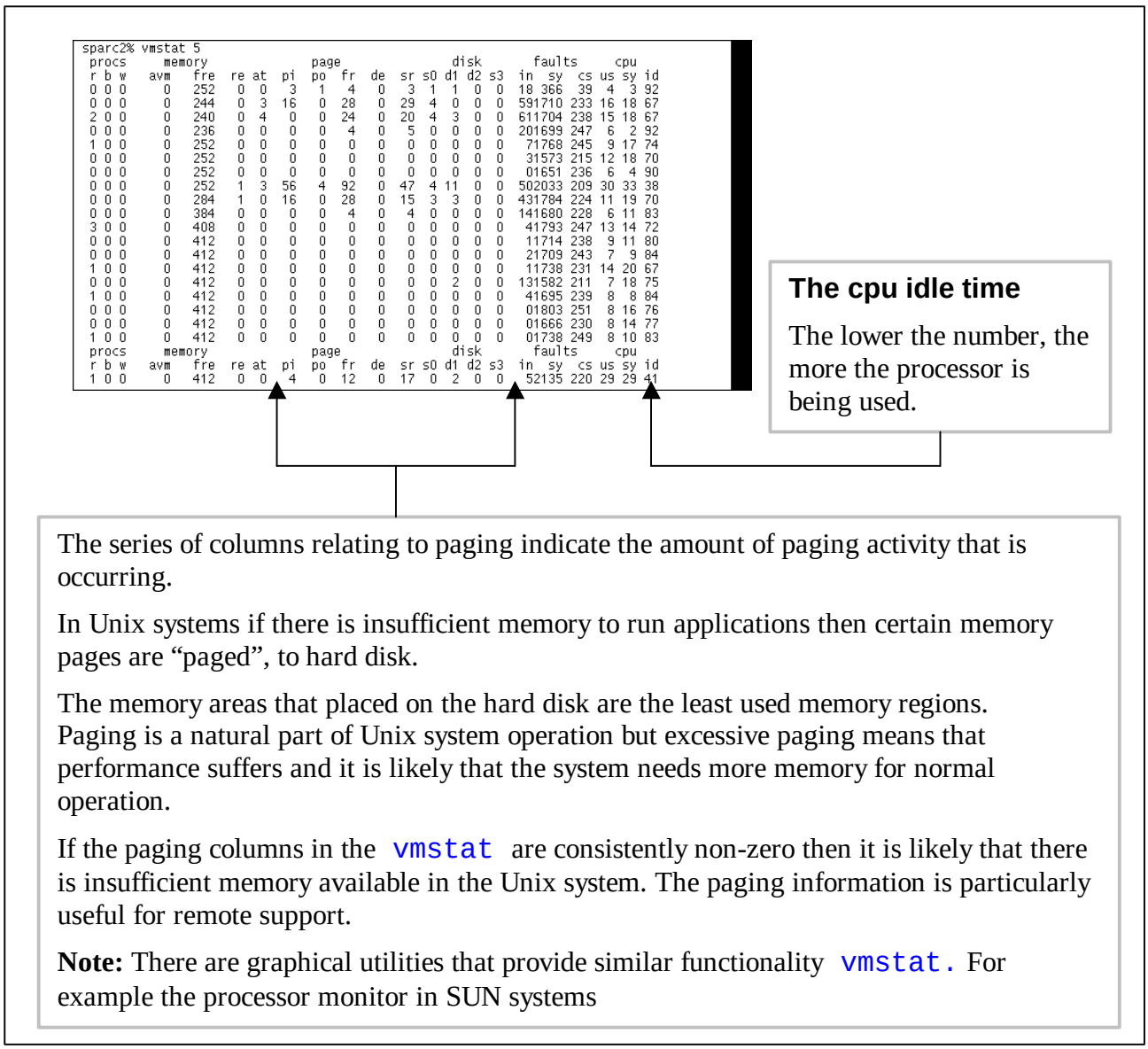
  at the Unix prompt.

- To use the `vmstat` program, type:

  `vmstat <n>`

- Where `n` is the interval in seconds the command will be repeated. Press Control C (or the delete key in SCO Xenix/UNIX) to interrupt the printout.

The diagram shows an example of what the printout will look like.



The series of columns relating to paging indicate the amount of paging activity that is occurring.

In Unix systems if there is insufficient memory to run applications then certain memory pages are “paged”, to hard disk.

The memory areas that placed on the hard disk are the least used memory regions. Paging is a natural part of Unix system operation but excessive paging means that performance suffers and it is likely that the system needs more memory for normal operation.

If the paging columns in the `vmstat` are consistently non-zero then it is likely that there is insufficient memory available in the Unix system. The paging information is particularly useful for remote support.

Note: There are graphical utilities that provide similar functionality `vmstat.` For example the processor monitor in SUN systems

Meta file Performance Checks

You can check the performance of your meta file using the command:

```
SEND "ActivatePerformanceMeasurements" TO "MetaFile";
```

This command switches on the performance measurement mode of the object. Various statistics are stored in global variables for your use. The table below summarises the variables and their meaning. To de-activate the performance checks, use the following command:

```
SEND "DeactivatePerformanceMeasurements" TO "MetaFile";
```

Meta file Performance Reports

A performance report can be printed to the standard output at regular periods using the

`-reportTimes` option in the command line. This will send a report on the various global variables defined in the last section to the standard output whenever a metafile exits. The table below lists the Global variables that are reported.

Table 11: MetaFile Performance Global Variables

Variable Name	Description of Contents
`MetaFileName`	The name of the metafile for the window that has just exited or been closed.
`MetaFileSize`	The size of the metafile being reported on (in bytes).
`MetaFileDrawTime`	A summation of the metafile read time and the metafile paint time.
`MetaFileReadTime`	The time taken to read in all of the object information from a metafile. This time includes the time needed to interpret the metascript programs defined in that metafile.
`MetaScriptInterpretTime`	A summation of the time taken to interpret all of the meta scripts defined in a metafile.
`MetaWidgetCreateTime`	The time taken to create the widgets defined in a metafile window. Note that widget creation does not necessarily result in graphic display but does require a negotiation/communication between MacroView and the graphics subsystem.
`MetaFilePaintTime`	The time taken to paint all of the metafile graphic objects on the window. Note that this does not include the time needed to internally create widgets.
`MetaFileRemovalTime`	The time taken to internally destroy the metafile objects when a metafile window is exited or closed.
`MetaFileUpdateTime`	The average time needed to perform an update cycle. This time includes the meta script execution time and the time needed to paint any changes to graphics. Note: If a metafile window has an update rate of 2 seconds then it will attempt to perform an update cycle once every 2 seconds.
`MetaScriptExecTime`	The average time needed to execute all of the meta scripts in a metafile. This time is averaged over each of the update cycles that a metafile window performs.

Disk Usage (df command)

- You can get an idea of the amount of disk usage using the `df` command.

  Just type `df` at the Unix prompt and you will get a printout that looks similar to the following:

```
sparc2% df
Filesystem            kbytes    used   avail capacity  Mounted on
/dev/sd0a               7735    4400    2561    63%    /
/dev/sd0g             143965  113840   15728    88%    /usr
/dev/sd0h               7255    1250    5279    19%    /home
/dev/sd3g             189858  155180   15692    91%    /u
sparc2%
```

- You should periodically remove all files that are of no use to optimize the space on your disk.
- You could also write a small "watchdog" script that runs once a day (say) to alarm with `sendmsg` when the disk usage exceeds 80%.

Note: The command line parameter for `df` vary between operating systems. The suggested parameters for various operating systems are:

Solaris	`df -k`
SunOS	`df`
SCO Xenix/Unix	`df -v`
Ultrix	`df`
HP-UX	`df -t`

Source Diagnostics

To check out sources, you should use the `plcstat` routine for image-based drivers (type 6) and `imgstat` routine for **sorted** image based drivers (type 5).

These routines provide a comprehensive range of diagnostic and statistical assistance.

You can get more details on these routines by looking at the `plcstat` and `imgstat` sections in the Sources chapter.

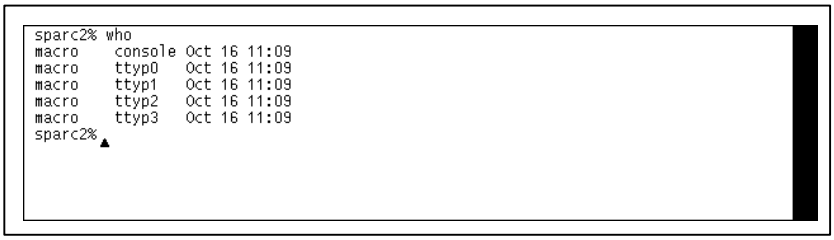
Alternatively, you can get more information from the `plcstat(C)` and `imgstat(C)` documents from your distributor.

Specific information about the driver itself will be available in the Driver Document itself also obtainable from your distributor.

Network Diagnostics

who

To find out who is on the network, just type **who** at the Unix prompt. You will get a printout that looks like this.



ping

To find out whether a computer is connected to the network and is responding:

(i) First find out the hostname of the computer by looking in the /etc/hosts file with:

```
more /etc/hosts
```

(ii) Then type:

```
ping <hostname>
```

If the computer connection is healthy, the system will typically respond with:

```
<hostname> is alive
```

The actual output is operating system dependent.

testterm

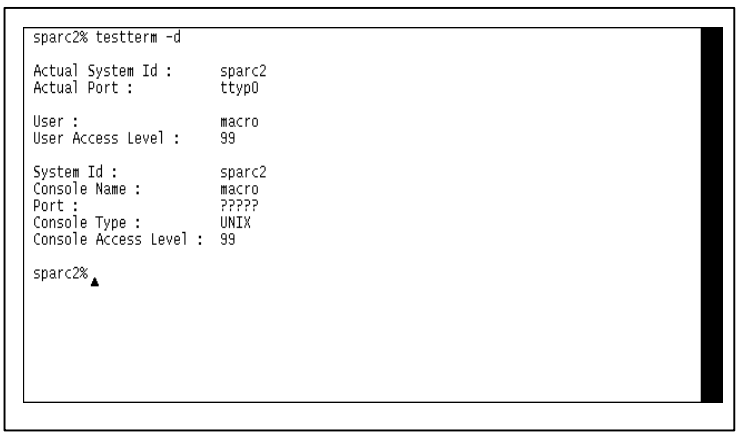
The `testterm` utility is most useful in determining a number of parameters about a terminal or user session.

These parameters include the hostname, port, console name, etc.

```
cd $MACRODIR; testterm -d
```

```
testterm -d
```

At the Unix prompt and you will see a screen that looks similar to the diagram below.



Further Reading

We recommend the following books that are useful in getting to know Unix better.

Table 12: Recommended Reading

Title	Author	Publisher	Reference
Unix Shell Programming	Lowell Jay Arthur	John Wiley & Sons	ISBN 0-471-51821-2
Mastering SUNOS	Brent D Heslop	Sybex	ISBN 0-89588-683-9
Using UNIX	Que Corporation	Que Corporation	ISBN 0-88022-519-X