
1 Background Information

This section is designed for writers who do not know how to use a Unix shell or have no experience using OSS reference pages or Unix man pages. The section covers the following topics:

- Working on a Unix System
- The OSS System
- Reference Pages
- The Text Formatters
- Additional Resources

After reading this section and referencing recommended documentation, you will be able to go on to the remaining sections and perform updates on OSS reference pages.

Working on a Unix System

This subsection describes what you need to know about Unix in order to build OSS reference pages. The following topics are covered in this subsection:

- What is Unix?
- The Unix Shell
- The Unix File System
- Unix Shell Commands
- The `.cshrc` File
- Unix Text Editors
- Printing From a Unix System

What is Unix?

Unix is an operating system that runs on large workstation computers such as those from Sun Microsystems or Silicon Graphics. It is designed for multiple users on the same processor, and is accessed from a terminal. It can also be accessed from a Windows NT system or Macintosh through network connection software. You must have an account on the Unix system to logon to it. Accessing the Unix system is discussed in more detail in [Setting Up Your UNIX Work Environment](#) on page 2-1.

The Unix Shell

The Unix *shell* is the user interface to the Unix operating system. Unlike a Mac or Windows user interface, the shell is not graphical. It was designed to be accessed from a terminal and can also be accessed from a *terminal emulation window* (a window that looks and acts like a terminal screen), which can be run on a PC or Mac. The Unix shell

accepts keyboard input commands at a shell prompt, and looks similar to a DOS screen on a PC.

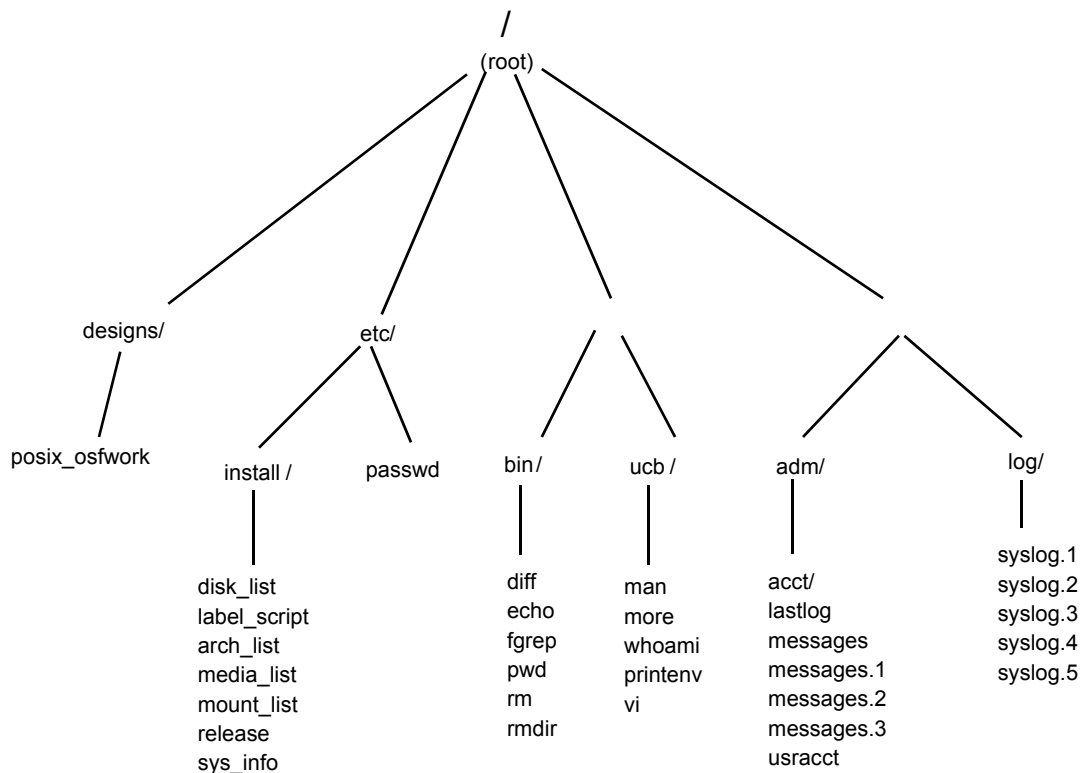
The Unix File System

The Unix file system is arranged in a hierarchical tree structure. It is analogous to the file structure on a PC or Mac, where you store files in folders. What would be called a folder on a PC is called a *directory* in Unix. A directory can store files or other directories. The nesting level of directories within directories can get quite deep.

At the top of the file hierarchy in the Unix file system is a directory called the *root*. The root directory is symbolized by the forward slash (/) character. Every file in the system has a *path* that starts from the root and follows a unique chain of directories down to the file itself. This is also called the file's *full pathname*. The concept of a path is central to understanding the Unix file system.

[Figure 1-1](#) shows what the file structure would look like on a sample Unix system. Directory names are indicated with a '/' character after them; filenames are plain.

Figure 1-1. Unix File System Hierarchy



The topmost directory in this sample file structure is the *root* directory. Under *root*, there are four subdirectories: *designs*, *etc*, *usr*, and *var*. Each of these directories has files and/or directories underneath it, and some of those directories also have more files and directories underneath them.

To distinguish between one directory and the next in a pathname, the forward slash character is placed between directory names and between the lowest directory name and the file's name. In [Figure 1-1](#), the full pathname of the file `disk_list` is `/root/etc/install/disk_list`. Some pathnames can be very long.

You can move around from directory from directory. The directory you are currently in at any specific time is called your current working directory. One special directory you will use a lot is your home directory. This is the main directory for your own account. When you first logon to a Unix system, your current working directory is set to your home directory -- that is your starting place.

Unix Shell Commands

The following is a short summary of the most commonly used Unix shell commands. This list contains all the shell commands you'll need to work with reference pages.

The man Command

The most important Unix shell command you will use is the `man` command. The reason this is so important is that you can use it to find out information about all the other Unix shell commands discussed in this document. The `man` command is the Unix system's online help, and it displays *man pages* for Unix shell commands, system commands, utilities, program functions, and data structures. The man pages are files that describe the syntax, usage, parameters (flags), and known problems for each command.

The syntax of the `man` command is as follows:

```
$ man command-name
```

where *command-name* is the name of any Unix system or shell command.

Example:

```
$ man ls
```

will display the reference page for the Unix `ls` command on the screen. The `ls` command is described in the next section.

Other Shell Commands

Other important shell commands are:

<code>cat</code>	Displays the contents of a file to the screen
<code>cd</code>	Changes the current working directory (similar to navigating to a different folder on a PC or Mac)
<code>cp</code>	Makes a duplicate copy of a file
<code>grep</code>	Searches files for occurrences of text strings inside files. This command is extremely powerful and useful
<code>ls</code>	Lists the files and directories that live in the current directory (similar to the "dir" command on DOS, or to opening a Windows or Mac folder and displaying its contents)

mkdir	Creates a directory
more	Displays the contents of a file, but uses pagination (displays one screenful at a time), and has other useful features
mv	Changes the name of a file to another name. You can also use this command to “move” a file from one directory to another
pwd	Print working directory. This shows you which directory you are currently in.
rm	Removes a file
rmdir	Removes a directory (this can only be done if the directory is empty).

All of the above shell commands are documented online in the Unix man pages. See [The man Command](#) on page 1-3 for details on how to view the man pages.

The .cshrc File

An important file in your home directory is called the `.cshrc` file - the *run command* file for your Unix shell. This file is executed whenever you start the shell. (It is analogous to the `autoexec.bat` file on DOS and Windows systems.) It is used to set up your Unix environment so that you can run programs and access files that you need. You can add items to this file to customize your working environment.

When you first get a Unix account, this file might be empty or might contain a few default items. The DCNS Unix Helpdesk can help you set up this file properly. Later in

this document, there will be some instructions for modifying this file so that you can access tools and files needed to build reference pages.

Example 1-1. Sample .cshrc file

```
# Set path
set path = (~ $lpath /usr/local/bin /designs/posix_osfwork/pubtools/bin /usr/ucb
/usr/bin /usr/5bin /usr/tandem/bin/ /usr/vendor/eroff/bin /usr/vendor/netscape
$FMHOME/bin $BUILD_TREE/Release_tools .)

# Set manpath
setenv MANPATH
/designs/posix_osfwork/pubtools/man:/usr/man:/usr/local/man:/usr/vendor/eroff/man:/us
r/tandem/man:/designs/posix_osfwork/dte/man:/usr/local/perl5/man:/designs/tuxpubs/man

# Set environment variables
setenv SRC /designs/posix_osfsrc/osf-build-dir/osfl/R1.2.1/doc/src/os_books/prog.rf
setenv WORK /designs/posix_osfwork/users/donl
setenv DTE_LIBDIR /designs/posix_osfwork/dte/libdata
setenv EROFF_DIR /usr/vendor/eroff
setenv EROFF_FONTDIR /usr/vendor/eroff/lib/font

# Set local printer
setenv PRINTER loc3_1305
# Turn off the printer banner page in the man page tools
setenv NOHEADER true
```

Don't be alarmed if you list the contents of your home directory and you don't see the .cshrc file listed among other files. In the Unix system, filenames that start with a dot are not visible with the plain `ls` shell command. If you list your directory contents using `ls -a`, the .cshrc file will be visible.

Note. It is important to make sure that your Unix account and its environment is set up properly. If you find that things aren't working the way you expect them to, ask the DCNS Helpdesk for more assistance.

Unix Text Editors

A text editor allows you to view and modify a text file.

The following is a list of text editors available on Unix systems and a brief description of each.

vi	A screen editor that works only on text files and accepts keyboard commands only
emacs	Another screen editor that accepts keyboard input only. Emacs is both more powerful and more difficult to use than vi
textedit	A text editor used only with a graphical front-end such as the X Window System. This editor accepts both mouse and keyboard commands.

Printing from a Unix System

Before you attempt to print a reference page from your Unix account, you must make sure that the printer you want to use is configured for the Unix machine you are going to use. The DCNS Unix Helpdesk can set this up for you.

After DCNS configures your printer, you must modify the PRINTER environment variable in your `.cshrc` file. To do this, put the following line in your `.cshrc` file:

```
setenv PRINTER printer-name
```

where *printer-name* is the name of your printer.

Both plain-text and PostScript files can be sent to your local printer. To print the files, you use the Tandem tools `dman` (to print plain-text files) or `dmant` (for Postscript files).

For example:

```
dman -pr fork.2
```

prints a text file of the `fork(2)` reference page to your printer.

```
dmant -pr fork.2
```

prints the PostScript equivalent.

The OSS System

You don't need to know a lot about the OSS system to build and release OSS reference pages, because all the work is done on a Unix server. However, this small subsection will explain briefly what the OSS system is and how it is related to the OSS reference pages.

Open System Services (OSS) is Tandem's implementation of the Posix standard of Unix. It is a subsystem of the Guardian operating system, and it provides a shell interface that looks and behaves very much like the Unix shell.

The reference pages you are building are used in the OSS subsystem. Although you create, build and submit the files from a Unix system, the files are published for the OSS system and are viewed there.

Once you have learned how to navigate your way around a Unix system, you'll probably find it easy to use the OSS system, although there are some differences between the two. A very good resource for learning the basics of the OSS system is the *Open System Services User's Guide*.

Reference Pages

This subsection describes what reference pages are and what they are used for, how to display them, and a brief overview of what is needed to modify them for new releases

What is a Reference Page?

In the Unix system, the `man` command displays the Unix man pages, a form of online help available in the Unix shell. The OSS system also has the `man` command, which works in the same way it does in Unix. However, on the OSS system, the pages that are displayed are called *reference pages*. Informally they may also be referred to as man pages, but they are called reference pages in all Tandem documentation. As in Unix, you can access OSS reference pages from the OSS shell prompt using the `man` command. For a description of the syntax of the `man` command, see [The man Command](#) on page 1-3.

Creating or Modifying Reference Pages

You actually create and build OSS reference pages on a Unix system. This is because the OSS system does not have the proper tools for creating and building reference pages. (See [Working on a Unix System](#) on page 1-1 for an overview of the Unix shell and the commands you need to know in order to work on reference pages. [Setting Up Your UNIX Work Environment](#) on page 2-1 explains how to get a Unix account and how to access it.)

There are a number of Tandem-specific Unix tools that you need to use to produce a new release of the reference pages. These tools perform several functions, such as building a formatted page from your text input, creating online copies and hard copies of your pages, archiving them into the version control system, submitting the finished release for publishing, and performing many auxiliary tasks. All these tools are accessible at the Unix shell prompt. [Section 2, Before You Start Building a Reference Page](#) describes the syntax and usage of each of these tools. [Section 3, Building OSS Reference Pages](#) explains the procedures for creating or modifying reference pages using the tools.

Formatters for Reference Pages

You format the OSS reference pages with one of two text-formatting programs, depending on whether the output file is for online viewing or for a printed manual. This subsection describes how to prepare your reference pages for the text formatters.

Source Files and Formatting Tags

The command codes (also called the formatting *tags*) of the text formatters specify the page layout, typographical features, and other formatting characteristics of the output. When you change a reference page or create a new one, you need to understand how to use the formatting tags.

The file that contains the formatting tags is called the *source file*, a plain text (ASCII) file that can be edited with a text editor. The formatting tags are embedded in the file's text. The formatting programs read the source file, interpret the tags, and produce a formatted output file.

The tags used by the text formatters are derived from the Unix text formatters `nroff` and `troff`. You do not run either `nroff` or `troff` directly, but instead you run Tandem-specific formatters called `dmann` (used to produce an online version of the reference page) and `dmant` (which produces a Postscript version for printing). You use the same set of formatting tags no matter which utility you are using.

The embedded commands in the source file all begin with a dot and start at the very beginning of a line.

Command List

Formatting tags encode the text for reference pages. The following list describes the tags that you will use:

Note. These tags are all case-sensitive.

<code>.TH <i>name</i> #</code>	Topic heading. <i>Name</i> refers to the reference page, and usually corresponds with the filename. After name, the logical manual section number appears.
<code>.SH</code>	Subheading.
<code>.sS</code>	Syntax start. Signifies the start of a syntax definition. Must be followed by <code>.sE</code>
<code>.sE</code>	Syntax end. Follows <code>.sS</code> .
<code>.ss</code>	Applies default formatting for the entry.
<code>*L <i>term</i>*O</code>	Bold font, where <i>term</i> is the emphasized word. Used for computer literals. The term must be followed by <code>*O</code> , which returns the text to the default font.
<code>*V <i>term</i>*O</code>	Italic font, where <i>term</i> is the emphasized word. Used for computer variables. The term must be followed by <code>*O</code> , which returns the text to the default font.
<code>.nL</code>	Signifies a new line, single-spaced.
<code>.br</code>	Signifies a text-formatting break where a page break is allowed.

.PP	Plain paragraph (causes 1.5 line spacing and allows a page break).
.ne <i>n</i>	Need <i>n</i> lines. Used in conjunction with .PP or .br. This is a “keep together” command that allows a page break before the next <i>n</i> lines or after the next <i>n</i> lines, but not during.
.VL	Signifies the start of a variable list where .LI is immediately followed by the term being described in quotations.
.ML	Signifies the start of a bulleted list where .LI is on a separate line preceding the list item text.
.AL	Signifies the start of a numbered list. Used in the same way as .ML
.LI	List item. Must precede any item in a list.
.LE	List end. Must follow the last list in an item to close the list
.iS	Input start. Bold font. Signifies the start of an example.
.iE	Input end. Signifies the end of an example.
.cS	Comments start. Use to comment out text.
.cE	Comments end. Must use in conjunction with .cS after comments or else all text will be commented out.

Additional Resources

This section has touched on various topics very briefly, enough to give you an idea of the tools and systems you will be working with. If you are very new to any of these concepts, you can find additional help from the following resources:

Unix	<i>Unix in a Nutshell</i> , O'Reilly & Associates, Inc., 1990
Unix shell commands	Read the Unix man pages on individual commands.
The vi editor	<i>Learning the vi Editor</i> , Linda Lamb. O'Reilly & Associates, 1990
The emacs editor	<i>The GNU Emacs Manual</i> , Free Software Foundation
The OSS system	<i>Open System Services User's Guide</i>

