

CHAPTER

1

Introduction to the OpenVMS Operating Environment

<i>What Is the OpenVMS Operating Environment?</i>	4
<i>SAS 9.2 Supported OpenVMS Platforms</i>	4
<i>Access to OpenVMS</i>	4
<i>Requirements for Accessing an OpenVMS System</i>	4
<i>Login Procedure</i>	5
<i>Files That Affect the Login Procedure</i>	5
<i>User Authorization File</i>	5
<i>LOGIN.COM File</i>	5
<i>Logout Procedure</i>	6
<i>Basics of the OpenVMS File System</i>	6
<i>Directories</i>	6
<i>Introduction to the OpenVMS Directory File Structure</i>	6
<i>Creating Directories</i>	7
<i>Changing Your Default Directory</i>	7
<i>Files</i>	8
<i>OpenVMS Filenames</i>	8
<i>Syntax for File Specifications</i>	8
<i>What Is a Fully Qualified Name?</i>	8
<i>Rules for File Specifications</i>	8
<i>Description of File Specification Fields</i>	8
<i>ODS-5 File Naming Conventions</i>	10
<i>Difference between the ODS-2 and ODS-5 File Naming Conventions</i>	10
<i>Benefits of Using the ODS-5 Volume Structure</i>	10
<i>Wildcards in OpenVMS Filenames</i>	11
<i>The Asterisk</i>	11
<i>The Percent Sign</i>	11
<i>OpenVMS File Types</i>	11
<i>OpenVMS File Types Used by SAS</i>	12
<i>Warning about Changing File Types</i>	12
<i>File Types for SAS Files</i>	12
<i>File Types for External Files</i>	13
<i>OpenVMS Logical Names and Logical-Name Tables</i>	13
<i>The OpenVMS Digital Command Language</i>	14
<i>Introduction to the OpenVMS Digital Command Language</i>	14
<i>The Command Line</i>	14
<i>How to Continue a Long Command on the Next Line</i>	14
<i>Interrupting Command Line Processing</i>	15
<i>Command Procedures</i>	15
<i>Creating a Command Procedure</i>	15
<i>Invoking a Command Procedure</i>	15
<i>Commonly Used DCL Commands</i>	15

<i>Commands for Controlling a Terminal Session</i>	16
<i>Commands for Managing Files</i>	16
<i>Data Security</i>	17
<i>Introduction to Data Security</i>	17
<i>File Protection</i>	17
<i>Specifying File Protection for a Process</i>	18
<i>Specifying File Protection for Individual Files</i>	18
<i>Access Control Lists</i>	18

What Is the OpenVMS Operating Environment?

OpenVMS is an interactive virtual-memory operating environment that runs on the HP Integrity server.

Like any operating environment, OpenVMS is designed to manage information. It accepts, stores, retrieves, and processes many types of information, such as data, text, programs (such as SAS programs), and output from programs. The OpenVMS system performs all data processing functions in response to DCL commands that you issue. These functions include the following:

- managing a terminal session
- submitting jobs for execution
- storing and retrieving data files
- allocating resources (for example, disk space, time, and internal memory) to individual jobs
- controlling peripheral equipment such as printers, plotters, and disk drives

For detailed information about OpenVMS, see the documentation provided by Hewlett-Packard. Also, the operating environment provides an online Help facility that you can access by using the DCL HELP command.

SAS 9.2 Supported OpenVMS Platforms

HP offers the OpenVMS version 8.3 operating system on AlphaServer and Integrity server platforms. SAS 9.2 supports the OpenVMS version 8.3 operating system for the Integrity server platform.

Access to OpenVMS

Requirements for Accessing an OpenVMS System

To access an OpenVMS system, you must have an OpenVMS *user name* and a *password*. In SAS documentation, the user name is usually called the *user ID*. Ask your supervisor or system manager for a user ID, a password, and any other information that you might need to access OpenVMS at your site. For example, if your system is part of a DECnet for OpenVMS network, then you also need to know how to access the appropriate computer system before you begin the login procedure.

Login Procedure

The login procedure differs from site to site, depending on how your system is configured.

When OpenVMS prompts you for your user ID, type the user ID, and then press the ENTER key. Next, the system prompts you for your password. Type the password, and then press ENTER. OpenVMS accepts and validates the user ID and password, even though you cannot see the password on the monitor. Next, you might see messages welcoming you to the OpenVMS system. If you enter an incorrect user ID or password, you must start the login procedure again.

The following sequence illustrates the login procedure:

```
Username: user-ID <ENTER>
Password: <ENTER>
. . . system messages . . .

$
```

When you log in, OpenVMS defines the environment in which it responds to your DCL commands. This environment is called your OpenVMS *process*.

Certain default characteristics are associated with your process, such as a default disk, a directory name, and resource quotas. This information is taken from the user authorization file. A command interpreter (usually the DCL interpreter) is also associated with your process.

After you have logged in successfully, the system typically displays a dollar sign (\$) to indicate that it is ready to accept a command. However, a different symbol might be used at your site, or you might use a menu interface. In this document, all examples assume that your OpenVMS system uses the \$ symbol as the system prompt and that it uses the DCL command line interpreter.

Files That Affect the Login Procedure

User Authorization File

During the login procedure, OpenVMS accesses a file called the *user authorization file* (UAF) to validate your user ID and password. The UAF is maintained by your system manager, and it contains a record for every person who is authorized to use the system.

Besides the user ID and password, the UAF record for each user specifies the user's default disk and default directory. Thus, each time you log in, your session is attached to the default disk at a location called your default (or home) directory. The UAF also specifies the access privileges and quota limits that are associated with your user ID.

LOGIN.COM File

At some sites, the system manager creates a login file, LOGIN.COM, in your home directory when you are authorized to use the system. This file contains DCL commands and utilities that are commonly used at your site.

When you log in, the OpenVMS system automatically searches for the LOGIN.COM file in your home directory. If one exists, the system executes the commands in the file before you receive the first DCL prompt (often the \$ symbol).

If you find that you regularly use certain DCL commands and utilities to customize your process, you can avoid entering these commands every time you log in by storing them in the LOGIN.COM file in your home directory.

Check with your system manager first before deleting or moving your LOGIN.COM file. You can create or modify the LOGIN.COM file using an OpenVMS editor.

For example, suppose that user John Smith has the following LOGIN.COM file stored in his home directory [SMITH]:

```
$ ! Login command file for John Smith
$ SHOW TIME
```

Each time Mr. Smith logs in, the command file executes automatically, and displays the current date and time. Note that the \$ prompt is generally in column 1 of each line that contains a DCL command. (A file containing just '**SHOW TIME**' will also work.) To improve readability, you can insert one or more blanks after the \$.

Note: If you use assignment statements in your LOGIN.COM file to create symbols, make sure that they are global assignments by using a double equal sign (==); otherwise, the symbols will be local to the LOGIN.COM file. In other words, an assignment statement that uses a single equal sign (=) creates a symbol that exists only while the LOGIN.COM file is executing. Δ

The following is an example of a global symbol assignment statement. Mr. Smith modifies the LOGIN.COM file by deleting the SHOW TIME command line and adding the following assignment statement:

```
$ DT == "SHOW TIME"
```

With this modification, Mr. Smith can request the current date and time during his terminal session by entering the following line:

```
$ DT
```

For more information about global and local assignments and about the LOGIN.COM file in general, see *OpenVMS User's Manual*.

Logout Procedure

When you are finished using the system, enter **LOGOUT** at the DCL prompt. The system displays a message to confirm that you have logged out.

Basics of the OpenVMS File System

Directories

Introduction to the OpenVMS Directory File Structure

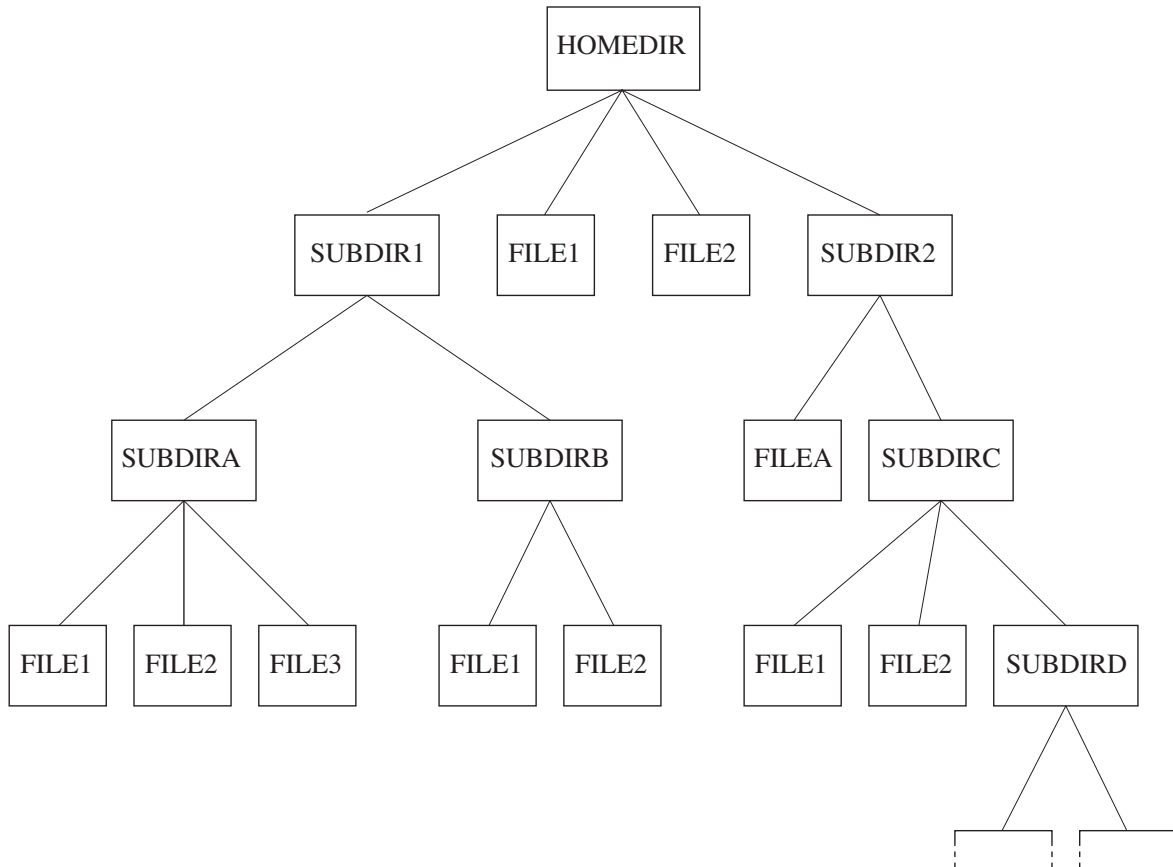
In the OpenVMS environment, files are organized into directories. A *directory* contains a list of all the files that are organized within that directory. When you log in, OpenVMS attaches your session to a directory on a default disk that is associated with your user ID. This directory is called your *home directory*. For each user ID, there is only one home directory. Often the home directory name is the same as the user ID.

The home directory can contain both files and other directories called *subdirectories*. Subdirectories can also contain files and subdirectories. The terms directory and subdirectory refer to the same type of file unit. The term subdirectory conveys the

relationship of one directory to another in the hierarchical structure that begins with the home directory.

This method of structuring files enables you to group sets of related files within directories. You can organize your files however you like. The following figure illustrates the directory file structure.

Figure 1.1 Directory File Structure



The directory that you are working in at any given time is called your *default directory*. For example, when you log in, your home directory is usually your default directory.

Creating Directories

To create a directory, use the DCL `CREATE/DIRECTORY` command.

Changing Your Default Directory

To move from one directory to another directory in the file structure, use the DCL `SET DEFAULT` command. Each time you use the DCL `SET DEFAULT` command, your default directory changes to the directory that you specify. Be sure to specify a pathname to the target directory. (A full pathname follows an unbroken path from the first-level directory down to the target subdirectory. However, a partial pathname, such

as [.SUBDIR], can also be specified.) Using Figure 1.1 on page 7 as an example, the following command specifies the full pathname to the subdirectory SUBDIRC:

```
$ SET DEFAULT [HOMEDIR.SUBDIR2.SUBDIRC]
```

When this command executes, SUBDIRC becomes the default directory.

If you have any doubt about your current location in the file structure, use the DCL SHOW DEFAULT command to show your default directory. Keeping track of your default directory helps you keep track of files. For example, some programs write output files to the default directory.

Files

Files contain various types of data, programming statements, or program output. Under OpenVMS, you can create files with several editors, including the EVE and EDT editors, the Text Processing Utility (TPU) editor, and the SAS text editor. Regardless of which editor you use, each file must have a unique name within that directory.

OpenVMS Filenames

Syntax for File Specifications

What Is a Fully Qualified Name?

A *fully qualified name* indicates how a file fits into a structure (a system of directories and subdirectories) that contains all the files stored under the OpenVMS system. The following type of file specification is a fully qualified name:

```
node::device:[directory]filename.file-type;version
```

Rules for File Specifications

A file specification cannot exceed 255 bytes. The *directory* and *file-specification* can each consist of up to 39 characters. The *file-type* can consist of up to 38 characters, although most of the default file types have only 3 characters. Permissible characters are the letters A through Z, the numbers 0 through 9, an underscore (_), a hyphen (-), or a dollar sign (\$). You can enter OpenVMS filenames and file types in uppercase, lowercase, or mixed case, but all characters are stored in uppercase format. For more information about file specifications, see *OpenVMS Guide to Extended File Specifications*.

Note: The default for OpenVMS filenames is the traditional ODS-2 syntax. However, SAS supports the extended ODS-5 syntax on ODS-5 enabled volumes. For more information, see “ODS-5 File Naming Conventions” on page 10. \triangle

Description of File Specification Fields

In many cases, you can uniquely identify a file even without specifying all of the fields in a fully qualified name. The following definitions give default values for fields that can be omitted from a file specification:

node

specifies a node name in an OpenVMS network. The node specification is always followed by a double colon (::). The default value for *node* is your OpenVMS system node; therefore, include *node* in the fully qualified name only when you require access to a file that is located on a different node in your OpenVMS network.

device

specifies the name of the physical or logical disk that contains the file. The device specification is always followed by a single colon (:). The default value for *device* is your current disk. When you log in, the default disk is the disk that is associated with your user ID. Include *device* in the fully qualified name only when you need to access a file on another disk.

directory

specifies the name of a directory or a sequence of directories. The directory specification must be enclosed in brackets; for example, [DIR1.DIR2.DIR3]. The directories that follow the first directory in the sequence are called subdirectories. In the previous example, DIR2 is a subdirectory of DIR1, and DIR3 is a subdirectory of DIR2.

The default value for *directory* is the default directory. Include *directory* in the fully qualified name only when you need to access a file that is not in your default directory. (For more information about the default directory, see “Directories” on page 6.)

Note: You can substitute angle brackets (< >) for square brackets ([]) in directory specifications. Δ

file-specification

specifies the name of a particular file within the directory. If the file is a SAS file, the filename must also comply with SAS naming conventions. (For details about SAS naming conventions, see *SAS Language Reference: Dictionary*.) The *file-specification* field has no default value unless you use a wildcard character.

file-type

usually describes the contents of the file. The *file-type* must be preceded by a period (.).

The default value for *file-type* depends on how the file is created or used. For example, some DCL commands assume default file types. You can assign a file type when you create a file.

version

specifies the version number of the file. Each time you modify or create and save a file, OpenVMS increments the highest existing version number for that file, and adds the new version to the directory. Version numbers can range from 1 to 32,767. If you request a file without specifying the version number, then you access the latest version of the file by default. (The latest version of the file is the one that has the highest version number.) If you specify a version number, you must precede it with either a semicolon (;) or a period (.).

Your system manager sets the maximum number of versions of a file that are saved at any given time. For example, if you have edited a file named [DIR1] PROG.DAT 1,000 times and your system is set to keep four backup versions, then the directory [DIR1] contains the following versions:

```
PROG.DAT;1000
PROG.DAT;999
PROG.DAT;998
PROG.DAT;997
```

When you edit PROG.DAT the next time (version 1001), PROG.DAT;1001 is created and PROG.DAT;997 is deleted.

ODS-5 File Naming Conventions

Difference between the ODS-2 and ODS-5 File Naming Conventions

Although the default file naming convention is ODS-2, SAS accepts the extended ODS-5 syntax on ODS-5 enabled volumes. This extended file naming convention enables users to create filenames similar to those in the UNIX and Windows environments. The ODS-5 file naming convention behaves like the Windows convention. For example, if a filename exists in all uppercase, then any lowercase or uppercase reference to that file will resolve to uppercase.

Benefits of Using the ODS-5 Volume Structure

Some benefits of using the ODS-5 volume structure include the following:

- ODS-5 allows longer filenames. Filenames can be up to 236 8-bit or 118 16-bit characters in length. File specifications longer than 255 bytes will be abbreviated by unmodified applications.
- ODS-5 supports more characters from the 8-bit ISO Latin-1 and the 16-bit Unicode (UCS-2) character sets. However, a few special characters in the ISO Latin-1 character set are not allowed in OpenVMS file specifications.

The following characters are invalid:

- C0 control codes (0x00 to 0x1F)
- Double quotation marks (“
- Asterisk (*)
- Backslash (\)
- Colon (:)
- Left and right angle brackets (<>)
- Slash (/)
- Question mark (?)
- Vertical bar (|)

The remaining special characters in the ISO Latin-1 character set are valid when preceded by the circumflex (^), which acts as an escape character.

Note: If you use the apostrophe (') in a file specification, you must enclose it in double quotation marks (“ ^ ’ “). If you enclose it in single quotation marks (' ^ '), SAS will state that there is an error in the filename due to an uneven quotation mark. \triangle

- ODS-5 file specifications are no longer converted and stored in all uppercase. ODS-5 syntax supports mixed-case and lowercase file specifications.
- ODS-5 supports deeper directory structures. Directories can be up to 255 levels. The naming conventions for directories follow that for filenames. A directory name can be up to 236 8-bit or 118 16-bit characters in length.

For more information about the extended ODS-5 syntax, see the *OpenVMS Guide to Extended File Systems*.

Wildcards in OpenVMS Filenames

OpenVMS supports two general-purpose wildcard characters: the asterisk (*) and the percent sign (%). In DCL commands, you can use these wildcard characters in file specifications to operate on a group of files instead of on a specific file.

Note: Using wildcard characters in file specifications can degrade the performance of your operating environment. △

For additional rules for using wildcard specifications in operations across nodes in a network, see *OpenVMS Networking Manual*.

For information about using wildcards in file and directory specifications, see “Using Wildcard Characters in External File Specifications” on page 180.

The Asterisk

The asterisk (*) replaces zero or more characters in one or more of the *directory*, *file-specification*, *file-type*, and *version* fields in a file specification. It causes the DCL command to act on all files whose names match the fields that you include in the specification. For example, all of the following file specifications match CAT.SAS:

- *.SAS
- CA*.SAS
- CAT*.SAS
- *A*.SAS
- CAT.*

The asterisk (*) often references sets of files in DCL commands such as PRINT, TYPE, and COPY. For example, the following command prints all versions of all files in directory [DIR1] that have the file type .SAS:

```
$ PRINT [DIR1]*.SAS;*
```

The Percent Sign

The percent sign (%) replaces a single character in *directory*, *file-specification*, and *file-type* fields in the file specification. For example, the following command prints all versions of every file whose name has five characters beginning with the letters PROG and whose file type is .SAS:

```
$ PRINT PROG%.SAS;*
```

In other words, you can use the previous example to print files PROG1.SAS through PROG9.SAS, but not file PROG10.SAS. To print all versions of every file whose name begins with the letters PROG and whose file type is .SAS, enter the following command:

```
$ PRINT PROG*.SAS;*
```

OpenVMS File Types

The *file-type* portion of a filename often indicates the contents of the file. Both OpenVMS and SAS use default file types for output files. DCL commands that create files often assign default file types if you omit *file-type* in the file specification. When you assign file types to files you create, keep in mind that either OpenVMS or SAS

might require a certain file type, depending on how a file is used. The following is a list of commonly used OpenVMS file types:

.COM	is usually a DCL command file that can be executed with the DCL @ command or submitted for batch execution with the SUBMIT command. (For information about submitting a SAS job in batch mode, see “Batch Mode under OpenVMS” on page 25.)
.DIR	is usually a directory. The DCL CREATE/DIRECTORY command assigns the file type .DIR by default.
.LIS	is usually a file called a listing, which might contain output of a SAS session.
.LOG	is usually a file called the OpenVMS log, which contains batch job output, or it is the log of a SAS session.
.MAI	is usually a file containing messages that were entered with the OpenVMS Personal Mail Utility (MAIL).

For a complete list of OpenVMS file types, see *OpenVMS User's Manual*.

OpenVMS File Types Used by SAS

Warning about Changing File Types

SAS uses unique file types to distinguish between SAS files and OpenVMS external files in a directory.

CAUTION:

Do not change these file types. The file types in the following sections are an integral part of how SAS accesses files under OpenVMS. Changing the file types can cause unpredictable results. △

File Types for SAS Files

Most of the file types that SAS uses are assigned to files that are described as either temporary or permanent *SAS files*. A SAS file is stored in a SAS library and is referred to as a *member* of a library. Each member has a *member type*. SAS equates some OpenVMS file types with a general set of SAS member types that it uses under all operating environments.

The names of these file types have the following form:

engine-name-filetype

engine-name

is the name of the SAS engine, such as SAS7B for the BASE engine and SAS7S for the sequential engine.

filetype

is the type of file. For example, the sas7bdat file type is a data file that is accessed by the BASE engine.

For information about SAS engines, see Chapter 6, “Using SAS Engines,” on page 157.

In addition, two types of SAS temporary files exist, with OpenVMS file types of .SAS7BUTL and .SAS7BPUT. These files appear only in the SAS Work library.

File Types for External Files

External files can be processed by other programs and by the FILENAME function and the %INCLUDE, FILE, and INFILE statements in the SAS DATA step. (For more information about external files, see Chapter 7, “Using External Files and Devices,” on page 177.) SAS uses the following OpenVMS file types for external files:

.DAT

is an external file that contains data lines. This is the default file type that SAS uses when it is reading and writing lines with the INFILE and FILE statements. (For more information, see “Default File Types” on page 183.)

.LIS

is an external file that contains SAS procedure output. By default, the filename of the .LIS file matches the filename of the .SAS program file that generated the output.

.LOG

is the external file that contains the SAS log. By default, the filename of the .LOG file matches the name of the SAS program file that generated it.

.SAS

is a SAS program file—that is, an external file that contains SAS statements. Use this file type when you create a file that contains a SAS program. This is the default file type for the FILE command and for the %INCLUDE statement.

.TLB

is an OpenVMS text library. SAS can access text libraries as external file aggregates. Text library files typically store data or SAS programs that are related. For example, you might want to store all SAS programs that are associated with a particular application in one text library. (For more information about OpenVMS text libraries, see *HP OpenVMS Command Definition, Librarian, and Message Utilities Manual*.) OpenVMS text libraries are also often used to store SAS macros. (For more information about SAS macros, see Chapter 21, “Macro Facility under OpenVMS,” on page 525.)

OpenVMS Logical Names and Logical-Name Tables

Under OpenVMS, logical names are used extensively in place of part or all of a file specification or to see devices or queues. For details about logical names and about logical-name tables, see *OpenVMS User’s Manual*. You might also want to see the following sections for information about how OpenVMS logical names are used by SAS or about how you use them in SAS programs:

- “Customizing Your SAS Session Using OpenVMS Logical Names” on page 42
- “Assigning OpenVMS Logical Names” on page 146
- “Assigning OpenVMS Logical Names to External Files” on page 179
- “Commonly Used DCL Commands” on page 15

The OpenVMS Digital Command Language

Introduction to the OpenVMS Digital Command Language

When you communicate with the OpenVMS operating environment, you can use the OpenVMS Digital Command Language (DCL). Like other languages, DCL has grammar rules and a vocabulary. The vocabulary is a set of commands, and the grammar rules determine how you specify the commands.

The Command Line

A completely specified DCL command is called a command line. The general form of a command line is the following:

\$ command/qualifier parameter/qualifier

dollar sign (\$) prompt is required in all DCL command lines. When you execute DCL commands interactively, the OpenVMS system supplies the system prompt, which by default is the \$ prompt. When you enter DCL commands in a command file, such as LOGIN.COM, you must enter the \$ prompt in column 1.

command is a DCL command that identifies the action to be performed. Use *OpenVMS DCL Dictionary* as a general reference to DCL commands and rules of grammar. Some commonly used DCL commands are described in “Commonly Used DCL Commands” on page 15.

Some DCL commands are called verbs. These are commands whose names indicate the command’s function.

qualifier is an optional keyword that modifies or expands the action of the command. It is always preceded by a forward slash (/). If a qualifier requires a value, it is given in the following form:

qualifier=value

parameter is either a keyword or a file specification (depending on the command) that is acted on by the command. For information about how to enter an OpenVMS file specification, see “OpenVMS Filenames” on page 8.

If the command requires one or more parameters, then OpenVMS prompts you for them, unless you include the parameter on the command line. Notice in the command line syntax that both commands and parameters can be qualified.

How to Continue a Long Command on the Next Line

If a command line does not fit on one line, enter a space and a hyphen (-) as the last element on the command line. Then, press the ENTER key and continue the command on the next line:

*\$ command -
_ \$ parameter/qualifier*

When you end a command line with a hyphen, the next prompt is preceded by an underscore (_).

Interrupting Command Line Processing

You can interrupt command line processing by simultaneously pressing the CTRL key and the Y key (CTRL+Y).

Command Procedures

A command procedure is an OpenVMS file that contains sequences of DCL commands. The OpenVMS file type for command procedures is COM. The LOGIN.COM file is an example of a command procedure. (For more information, see “LOGIN.COM File” on page 5.) To save yourself time when you perform complicated or repetitive tasks, you can create command procedures.

Creating a Command Procedure

Use your favorite editor to create the file of commands. For example, the following series of commands might be included in a command procedure named RENAME.COM:

```
$ PURGE *.*
$ RENAME *.*; *.*;1
```

The first line deletes all but the latest version of all files in your default directory. The second line renames all files in your current directory to version 1.

Invoking a Command Procedure

To invoke the RENAME.COM command procedure, you would enter the following command at the \$, or system, prompt:

```
$ @RENAME
```

The at sign (@) indicates that you want to execute a command procedure. For more information about creating and using command procedures, see *OpenVMS User's Manual*.

Commonly Used DCL Commands

Some of the most commonly used commands in an interactive command language are those that control your terminal session and those that manage files. The following subset of commands is divided into these two categories: controlling a terminal session and managing files. Many of these commands are used in examples throughout this documentation. The common abbreviation for a command is shown in parentheses beside the fully specified command. These abbreviations are the defaults; if these commands or symbols have been redefined, they might have a different action. See your system manager if the abbreviation does not work as you expected.

Note: You can issue any DCL command from within a SAS program by specifying the command in the SAS X statement or X command. For more information, see “Issuing DCL Commands during a SAS Session” on page 44. Δ

Commands for Controlling a Terminal Session

Use the following DCL commands to control your terminal session:

DIRECTORY (DIR)

displays a list of files and subdirectories in the specified directory, or in the current directory if no directory or file specification is given. For example, the following command produces a list of files in your default directory:

```
$ DIR
```

HELP

activates the OpenVMS HELP facility.

LOGOUT (LO)

terminates a terminal session and deletes your process.

SET DEFAULT (SET DEF)

changes the default directory. If you move to another disk, this command also changes the default disk device. For more information about the default directory, see “Directories” on page 6.

SET DISPLAY (SET DISP)

indicates where to send the interactive display of an OpenVMS DECwindows application. You need to use this command when you are running interactive SAS on a nonlocal device such as a PC monitor running emulation software.

SHOW DEFAULT (SHO DEF)

displays the default directory and the default disk device.

SHOW QUEUE (SHO QUE)

displays all queues including the batch queue.

SUBMIT (SUBM)

places one or more jobs (command procedures) in a batch queue for processing.

Commands for Managing Files

Use the following DCL commands to manage your files:

COPY (COP)

copies one or more files to one or more specified files.

CREATE/DIRECTORY (CRE/DIR)

creates a directory.

DEFINE

associates a logical name with a file specification or equivalence name.

DELETE (DEL)

removes access to a file. The file specification must reference the version or versions.

EDIT</editor> (ED</editor>)

invokes the specified OpenVMS editor to create or modify a file.

PRINT (PR)

prints a file on the default system printer or on a specified device.

PURGE (PUR)

deletes all versions of a specified file or files except the latest version.

TYPE (TY)

displays the contents of a file.

Data Security

Introduction to Data Security

SAS protects data sets, catalogs, and external files by using the normal OpenVMS access-control measures: file protection and access control lists. These access-control measures apply equally to SAS data sets, catalogs, and external files. SAS data set security measures, such as data set READ and WRITE passwords, are also available.

File Protection

File protection is a generalized way of controlling access to files, based on the relationship of the user that is accessing the file to the file owner. A typical file protection is specified as a list of user classes and their permitted operations.

File protection consists of four classes of users who can access a file:

SYSTEM

refers to a user who has a system user identification code (UIC) or SYSPRV privilege. This class is typically reserved for the system manager and operations staff.

OWNER

refers to a user who has the same UIC as the user who created the file. At most sites, each user has a unique UIC, so the OWNER is always the person who created the file.

GROUP

refers to a user who has the same group number in their UIC as the creator of the file.

WORLD

refers to any other user on the system.

Permitted operations are the following:

READ access

is required to read a file. For example, you must have READ access to a file if you intend to browse, copy, or move it.

WRITE access

is required to change the attributes of a file, including its file protection, as well as to modify data in the file.

EXECUTE access

is required to execute files such as command procedures. It does not apply to OpenVMS data files such as SAS data sets, catalogs, or external files.

DELETE access

is required to delete or move a file.

A typical file-protection specification might be the following:

```
(S:RWE, O:RWED, G:RE, W)
```

This example permits a SYSTEM user to READ, WRITE, or EXECUTE the file. The OWNER is permitted any operation on the file. Members of the same GROUP as the user are permitted to READ or EXECUTE the file. All other users are denied access to the file.

Specifying File Protection for a Process

To specify file protection for the duration of your process (or until you change the specification), use the SET PROTECTION/DEFAULT DCL command. The /DEFAULT parameter tells OpenVMS to apply the specification to every file that you create during your OpenVMS process except those for which you explicitly specify file protection. For example, the following command specifies new protection values for files that you subsequently create:

```
$ SET PROTECTION=(S:RWE,O:RWED,G:RE,W)/DEFAULT
```

Specifying File Protection for Individual Files

To specify file protection for a particular file or set of files, use the SET PROTECTION command without the /DEFAULT parameter. For example, the following SET PROTECTION command sets new protection values for the file MYDATA.SAS7BDAT:

```
$ SET PROTECTION=(S:RWE,O:RWED,G:RWED,W:RE) -
_$ MYDATA.SAS7BDAT
```

For more information about file protection, see the *OpenVMS User's Guide* and *HP OpenVMS Guide to System Security*.

Access Control Lists

Access control lists (ACLs) provide more specific control over who can perform operations on a file. ACLs allow each file to have a list of identifiers, which can be either UICs or keyword names that are assigned by the system manager. Only users with matching UICs or resource identifiers can perform operations on the file.

The description of ACLs is beyond the scope of this document. For more information, see *HP OpenVMS Guide to System Security*.