GNV

Read First Before Installing

July 2004

This document describes the features and functionality provided with the OpenVMS GNV software. It explains how to install, set up, and use the software on your OpenVMS system.

Revision/Update Information:	This is a new document.
Operating System:	OpenVMS Alpha 8.2EFT and Version 7.3-2 OpenVMS I64 8.2EFT and Version 8.1
Software Version:	OpenVMS Alpha GNV 1.6EFT, OpenVMS I64 GNV X1.6EFT

Hewlett-Packard Development Company, L.P.
Houston, Texas

© Copyright 2004 Hewlett-Packard Development Company, L.P.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Linux® is a U.S. registered trademark of Linus Torvalds.

UNIX® is a registered trademark of The Open Group.

© Copyright 1989, 1991 Free Software Foundation, Inc.

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

This document was prepared using DECdocument, Version 3.3-1b.

Contents

Tables

Preface

This document includes information about the GNV software, including:

- Features and functionality
- Utilities provided with the software
- Installation and setup instructions
- Differences between the OpenVMS GNV utilities and their counterparts on UNIX® systems
- Reporting problems
- Documentation provided with the GNV software
- Sources of information on the Internet

Conventions Used in This Document

OpenVMS utilities, commands, file names, and directory names are shown in uppercase text, while UNIX utilities, commands, file names, and directory names are shown in lowercase text. If you are reading this documentation in an output format other than text (.TXT) — such as in PostScript or HTML output — UNIX utilities, commands, file names, and directory names are shown in `monospace` text.

The following additional conventions are used in this document.

Convention	Meaning
Italic[1]	Italic typeface indicates a place holder for information or parameters that you must provide. For example, if the procedure asks you to type *file name*, you must type the actual name of a file. Italic type also indicates titles of other documentation referenced.
`monospace`[1]	Monospace typeface indicates code examples, command examples, and interactive or system screen displays. In text, this type face indicates UNIX commands that you enter.
Ctrl/*x*	While you hold down the Ctrl key, press another key or a pointing device button.

[1]Applies only to documentation formats other than text, such as PostScript and HTML.

Reader's Comments

HP welcomes your comments on this manual. Please send comments to either of the following addresses:

Internet	**openvmsdoc@hp.com**
Postal Mail	Hewlett-Packard Company OSSG Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

1 New Features in GNV V1.6

Several new UNIX utilities have been added in GNV V1.6.

- du — Displays disk usage, including the size of directories in the current or specified directory. The utility recursively displays the sizes of all subdirectories. The size listed for a directory denotes the total size of all files within that directory.

  Note

  On an OpenVMS system, the size displayed by du represents only the highest generation of each of the files in the directory.

  By showing all directories in a directory tree, du allows a user to easily assess how disk space is being used.

  For more information, type:

  ```
  bash$ du --help
  ```

  or

  ```
  bash$ man du
  ```

- printenv — Displays an environment variable. If no argument is present, displays all environment variables.

  For more information, type:

  ```
  bash$ printenv --help
  ```

  or

  ```
  bash$ man printenv
  ```

- env — Runs a program in a modified environment.

  As with the printenv command, specifying the env command with no argument will display the set of environment variables. More importantly, you may specify one or more environment variables, and run a program with those variables set. The values of those variables stay in effect only for the duration of that program.

  For more information, type:

  ```
  bash$ env --help
  ```

  or

  ```
  bash$ man env
  ```

- which — Displays the location (path) of the specified utility. When you type a command to bash that is not a bash "internal" command, BASH searches for the executable for the command in the default path (environmment variable PATH). The which utility indicates where in the path the executable for that command is found.

  For more information, type:

  ```
  bash$ which --help
  ```

  or

  ```
  bash$ man which
  ```

- GNU tar (gnutar) — Most UNIX systems include a tar utility. Many Linux® systems ship GNU tar. Traditionally, GNV has included VMSTAR, which is a tar look-alike written for OpenVMS. Because GNU tar has more capabilities than VMSTAR, it has been ported to OpenVMS for use with GNV.

  GNU tar is especially useful for reading OpenSoure tarballs, which are often shipped compressed as GZIP files. Using VMSTAR, the tarball must be decompressed prior to reading with tar. For very large tarballs, this process can be time-consuming. GNU tar allows you to read tarballs compressed with either gzip or bzip2 in-line. That is, you need not decompress them in a separate command.

  For example:

  ```
  bash$ gnutar -xzf gnutar.tar.gz
  ```

  The current GNV kit includes both the new GNU tar image and the older VMSTAR. In this kit, the tar command still invokes VMSTAR. To run the new GNU tar, type gnutar as shown in the preceding example.

  Caveats

  HP recommends using tar (either tar (VMSTAR) or gnutar (GNU tar)) for reading OpenSource tarballs. You can also use tar to save files on your OpenVMS system to a tarball. However, this works well only for UNIX-style files. The tar utility works especially well for stream_LF files, either copied from UNIX or GNU systems, or created by UNIX programs with C Run-Time Library functions.

  The tar utility does not work well on some other OpenVMS file types. It can successfully copy variable-length text files; however, it corrupts other types of variable-length files. tar corrupts object files generated by compilers.

  When a variable-length text file is written to a tarball and then restored to a directory on OpenVMS, the result is a stream_LF file with the same text contents as the original. This might not be suitable for some applications.

  Be very careful when using tar to back up non-stream_LF files. This warning applies especially to GNU tar, but can also apply to VMSTAR.

  Note about Image Names

  In this release of GNV, the new GNU tar utility is invoked by typing gnutar in the command line. Typing tar in the command line invokes VMSTAR.

  For more information, type:

  ```
  bash$ gnutar --help
  ```

  or

  ```
  bash$ man gnutar
  ```

- g++ added to GNV wrapper

The wrapper utility processes UNIX-style compile and link (and related) commands. It reads commands such as the UNIX cc command and invokes the appropriate OpenVMS DCL command.

For this release, g++ has been added as a synonym for cxx. This matches the gcc command to invoke the C++ compiler.

2 Overview of GNV Features and Functionality

The GNV software provides an open source, GNU-based, UNIX environment for OpenVMS. It provides UNIX application developers, system managers, and users a UNIX-style environment so that they can easily develop and port UNIX software to OpenVMS. (GNU is a UNIX-like operating system that is free software. Linux is basically a GNU system: the system is GNU, while the kernel is Linux.)

GNV provides a UNIX-like shell (command-line interpreter) environment and a C Run-Time Library (CRTL) supplemental library to provide utilities typically found on UNIX systems. The shell used by GNV is bash (Bourne-Again SHell, from GNU, using the POSIX.2 specification).

Note

To the normal user, bash performs like a command-line interpreter; however, it is not a replacement for the OpenVMS DCL (DIGITAL command language) command-line interpreter.

GNV for OpenVMS is a port of a series of GNU software intended for UNIX-like systems to OpenVMS. It is offered for your use under the terms and conditions of a GNU General Public License (GPL). You can see the latest GPL at the following location:

```
http://www.opensource.org/licenses/gpl-license.html
```

Several copies of the GPL are also included with the GNV kit. After you install the kit, you can find these copies in any of several directories, in a file named COPYING.; as in the following example, in OpenVMS terminology:

```
GNU:[SRC.GNV.FINDUTILS]COPYING.
```

Or, in UNIX terminology:

```
/gnu/src/gnv/findutils/copying.
```

OpenVMS engineering is very interested in your experiences using the port of the GNU software to OpenVMS and encourages all customers to report problems and issues. Your feedback can help us enhance future versions of the GNV software for OpenVMS. For information on reporting problems, see Section 8.

2.1 Using GNV utilities from DCL

It has come to our attention that some users like to run GNV utilities from the DCL prompt rather than from within bash. While this may work in many cases, HP does not support this method and does not guarantee that the utilities will work as expected. Most noticeably effected will be use of wildcards, which must be processed by bash. They will not work when a GNV utility is run from DCL. The following example illustrates the inconsistency of results obtained when running ls from the DCL prompt:

```
$ ls
A        PRINT_ENV.C  T.C     TEST.H     TEST_FWRITE.C  test
HELLO.C  SLEEP.C      TEST.C  TEST.H-GZ  t.t.t.t.t

$ ls *.c
/VMS$COMMON/GNV/bin/LS.EXE: *.c: no such file or directory

$ ls *.C
/VMS$COMMON/GNV/bin/LS.EXE: *.c: no such file or directory
$
```

2.2 ODS-5 Requirement

ODS-5 disks allow use of files with extended file names, including most UNIX style filenames. Such file names are common in Open Source software. GNV includes mostly ported Open Source (or free software) code, and as such contains numerous files with UNIX style file names.

While it may be possible to install GNV (or at least portions of GNV) on ODS-2 disks, HP recommends that you install GNV on an ODS-5 disk. Installing GNV on an ODS-2 disk is likely to result in numerous installation warning and error messages. GNV has been tested primarily on an ODS-5 disk. HP does not guarantee the functionality of GNV on an ODS-2 disk.

Likewise, you can operate GNV on an ODS-2 disk, but some file naming features will not be usable on such a disk. You will not be able to make use of case-senstive file names or file names with special characters, such as multiple dots, which are so common in UNIX environments.

3 Utilities Provided with GNV

This section lists the UNIX utilities provided with GNV and describes any differences observed between the utilities as used on OpenVMS systems and the counterparts as used on UNIX systems.

3.1 UNIX Utilities Provided for OpenVMS

Table 1 lists the utilities provided with GNV on OpenVMS. Included for some of the listed utilities are OpenVMS command equivalents as well as behaviors that vary from expected behavior on a UNIX system. HP does not guarantee that all variances have been observed and documented.

Table 1 GNV Utilities

Utility	Description	Comments/Exceptions
General Purpose Utilities		
basename	Returns actual base file name when given a path.	
bash	Invokes bash shell.	
bind	Equates a key sequence to a function.	
cd	Changes the current working directory.	OpenVMS DCL SET DEFAULT
date	Displays current date and time.	OpenVMS DCL SHOW TIME
dirs	Displays a list of currently remembered directories.	
env	Run program with modified environment.	
exit	Terminates the shell.	
help	Provides explanatory text about bash internal commands and features.	
history	Displays the list of previously executed commands.	OpenVMS DCL RECALL
hostname	Sets or displays name of current host system.	
id	Displays the current user and group IDs and names.	
logout	Logs out of the shell.	
popd	Modifies the current directory stack.	
printenv	Displays environment variables	OpenVMS DCL SHOW SYMBOL or SHOW LOGICAL
ps	Displays status of system processes.	OpenVMS DCL SHOW SYSTEM or SHOW PROCESS/SUBPROCESS
pushd	Modifies the current directory stack.	
pwd	Displays the current working directory.	OpenVMS DCL SHOW DEFAULT
sh	Invokes a shell.	
sleep	Pauses for a specified time.	OpenVMS WAIT
tee	Sends output to multiple destinations.	
times	Displays user and system times.	
uname	Displays the name of the system.	
wait	Waits for background processes to complete.	
which	Displays path of a command executable or utility.	

(continued on next page)

Table 1 (Cont.) GNV Utilities

Utility	Description	Comments/Exceptions
Command Manipulation		
`alias`	Provides command name translation.	Similar to the OpenVMS DCL line: $ symbol =[=] "string" see `unalias`
`break`	Terminates a loop in the shell.	
`builtin`	Runs a shell built-in.	
`case`	Executes commands selectively.	

(continued on next page)

Table 1 (Cont.) GNV Utilities

Utility	Description	Comments/Exceptions
Command Manipulation		
command	Executes a simple command.	
continue	Resumes execution at the top of a loop.	
declare	Declares variables and their attributes.	
echo	Echoes input to standard output.	Similar to OpenVMS DCL line: $ WRITE SYS$OUTPUT
enable	Enables or disables built-in shell commands.	
eval	Constructs a command.	
exec	Executes commands outside of the current shell.	
export	Sets the export attributes.	
expr	Reads an expression, evaluates it, and writes the result to standard output.	
false	Returns a nonzero exit value.	See true
for	Executes a command in a loop.	
function	Creates a simple command.	
getopts	Gets input options.	
hash	Provides direct access to utilities.	
if	Executes commands selectively.	
let	Evaluates arithmetic expressions.	
local	Creates a local variable.	
read	Reads a line from standard input.	OpenVMS DCL READ
readonly	Sets the readonly attribute on shell variables or functions.	
return	Returns from the current shell function.	
select	Processes commands selectively.	
set	Sets shell flags and positional parameters.	
shift	Shifts positional parameters.	
source	Executes commands from a file.	
test	Evaluates an expression.	
trap	Intercepts exception conditions.	
true	Returns a zero exit value.	
type	Displays command information.	
typeset	Assigns attributes and values to variable.	

(continued on next page)

Table 1 (Cont.) GNV Utilities

Utility	Description	Comments/Exceptions
Command Manipulation		
unalias	Removes command name translation.	OpenVMS DCL DELETE/SYMBOL; see alias
unset	Unsets values and attributes of variables and functions.	
until	Executes commands in a loop until a condition is reached.	
while	Executes commands in a loop until a condition is reached.	
Program Creation		
cc	Compile links; invokes C compiler.	DECC must be installed
cxx	Compiles links; invokes C++ compiler.	CXX must be installed
gcc	Compiles links; invokes C/C++ compiler.	DECC must be installed
g++	Compiles links; invokes C++ compiler.	CXX must be installed
as	Assembler.	
ld	Invokes the linker.	
make	Builds programs. Maintains up-to-date versions of target files and performs shell commands.	Similar to MMS (Module Management System).
User-Level Administration		
chmod	Changes file permissions.	OpenVMS DCL SET SECURITY or SET PROTECTION
chown	Changes owner of a file.	OpenVMS DCL SET FILE/OWNER
df	Displays amount of free disk space in a system.	OpenVMS DCL SHOW DEVICE/FULL
share	Displays list of NFS shares.	
touch	Updates the access and modification dates of a file.	
ulimit	Displays and sets file size limit.	
umask	Displays and sets the file creation mask.	
System-Level Administration		
chgrp	Changes group ownership.	OpenVMS DCL SET ACL

(continued on next page)

Table 1 (Cont.) GNV Utilities

Utility	Description	Comments/Exceptions
File Manipulation		
ar	Archives files.	OpenVMS LIBRARY
bzip2	Compress/Decompress a file.	
cat	Concatenates files; displays or prints files.	OpenVMS DCL TYPE; see head, more
cksum	Displays checksum and byte count of a file.	See sum

(continued on next page)

Table 1 (Cont.) GNV Utilities

Utility	Description	Comments/Exceptions
File Manipulation		
cmp	Compares two files, showing first difference only (for a quick check).	OpenVMS DCL DIFFERENCE; see comm, diff
comm	Compares two sorted text files; output is three columns showing (1) lines in *file1* only, (2) lines in *file2* only, (3) lines common to both files.	See cmp, diff
cp	Copies files to a new destination.	OpenVMS DCL COPY
csplit	Splits a file into new files containing segments of the original file.	
diff	Compares two files, showing all differences.	OpenVMS DCL DIFFERENCE; see cmp, comm
du	Displays disk space usage.	
egrep	Searches for text in a file.	OpenVMS DCL SEARCH; UNIX: grep -e
fgrep	Searches for text in a file.	OpenVMS DCL SEARCH; UNIX: grep -f
find	Searches down directory trees for a file (no need to know device!).	OpenVMS DIR [...]*filename*; see ls
grep	Searches for text in a file.	OpenVMS DCL SEARCH; see egrep and fgrep
gnutar	GNU tar utility to archive files. Reads OpenSource software distribution tarballs and archives and restores UNIX-style files.	
gunzip	Decompress a file.	See gzip
gzip	Compress/Decompress a file.	See gunzip
head	Displays the beginning of files.	See cat, lpr
join	Joins lines of two files (fields) in output.	See paste
less	Displays a file one screen at a time.	OpenVMS DCL TYPE/PAGE; see more
ln	Link (creates an alias filename).	OpenVMS DCL SET FILE/ENTRY
ls	Displays contents of a directory.	OpenVMS DCL DIRECTORY; see find
mkdir	Creates a directory.	OpenVMS DCL CREATE/DIRECTORY
mkfile	Creates a file.	OpenVMS DCL CREATE
more	Displays a file one screen at a time	OpenVMS DCL TYPE/PAGE; see less
mv	Moves files to a new location or renames files.	OpenVMS DCL RENAME

(continued on next page)

Table 1 (Cont.) GNV Utilities

Utility	Description	Comments/Exceptions
File Manipulation		
paste	Joins corresponding lines of several files, or subsequent lines in one file.	See join
rm	Deletes files.	OpenVMS DCL DELETE
rmdir	Deletes a directory.	
split	Splits a file into multiple files.	
sum	Displays a checksum for a file.	
tar	Archives files.	This runs VMSTAR, a tool similar to tar. It is not a genuine UNIX tar utility. Similar to OpenVMS BACKUP. For more information, see Section 3.2. See gnutar.
unzip	Retrieves archived files.	
zcat	Expands compressed files.	
zip	Archives files.	
Text Processing		
cut	Locates specified fields of each line of a file and writes the characters in those fields to standard output. Displays columns of a file.	
expand	Replaces tab characters with spaces in the named files or in the standard input, and writes the result to the standard output.	See unexpand
fmt	Formats text of a file to a specified width.	See fold
fold	Formats file by wrapping lines at the specified width.	See fmt
gawk (awk)	Text processing language.	
nl	Numbers lines in a file.	
od	Writes the contents of a file to standard output, in a specified format.	
sed	Invokes a stream editor.	
sort	Sorts the lines of a file.	
tr	Finds and replaces characters in a file.	
unexpand	Replaces spaces with tab characters in the data from the standard input, or in the named files, and writes the result to the standard output.	See expand
uniq	Removes duplicate lines from a file.	
vi	Invokes a text editor similating the standard UNIX vi text editor.	OpenVMS EDIT; for more information, see Section 3.2
wc	Displays the number of lines, words, and characters in a text file.	

(continued on next page)

Table 1 (Cont.) GNV Utilities

Utility	Description	Comments/Exceptions
Printing		
`lp`	Prints a file.	

3.2 Utilities Exhibiting Variant Behavior

The following GNV utilities have been observed to exhibit behavior that varies from their UNIX counterparts.

- `tar` (this runs the OpenVMS utility VMSTAR, which differs slightly from the the genuine UNIX `tar` utility. Try `gnutar` instead.)
- `vi` (this runs the OpenVMS utility TPU, which differs slightly from the genuine UNIX `vi` utility)
- The intent of GNV is that all utilities accept UNIX-style file specifications (that is, those specifications containing slashes "/", single or double dots ".", "..", and so forth.) Several utilities included with GNV were ported to OpenVMS previously and can accept OpenVMS-style file specifications. In fact, some may not accept UNIX-style file specifications, such as the following:
  - `unzip`

4 Installation Instructions

This section provides directions for installing GNV onto your OpenVMS system. The latest GNV software is included in [GNV.KIT] on the Open Source Tools CD-ROM. You can also downline load the GNV software from either of the following locations:

`http://h71000.www7.hp.com/OPENSOURCE/opensource.html`

or

`http://gnv.sourceforge.net.`

4.1 Installing the GNV Software on Your System Disk

To install the GNV software on your system disk, follow these steps:

1. Load the installation kit media onto an available drive.
2. Log in to the SYSTEM account (at the login prompt, enter user name SYSTEM and the appropriate password), or an account with equivalent privileges.
3. At the DCL prompt ($), type the following command, as shown, where *device-name* is the name of the device containing the kit (a CD-ROM drive).

   ```
   $ PRODUCT INSTALL GNV /SOURCE=device-name:[GNV.KIT]
   ```

   To install your GNV software on a location other than the system disk, see Section 4.2.

4. When you enter the PRODUCT INSTALL command, the system responds with a display similar to the following:

```
The following product has been selected:
    DEC AXPVMS GNV V1.6             Layered Product

    Do you want to continue? [YES]
```

Continue the procedure by pressing the Enter key for the default (YES). The system responds as shown in the example below. The procedure might take several minutes and numerous messages might be displayed on the screen. In response to each prompt displayed by the system, choose the default answer.

```
Configuration phase starting . . .:

You will be asked to choose options, if any, for each selected product
and for any products that may be installed to satisfy software
dependency requirements.

DEC AXPVMS GNV V1.6

Do you want the defaults for all options? [YES]
```

Note that the only option with GNV is whether to include the source files with the installation. The default is YES.

5. Accept the default for all options by pressing the ENTER key. The system responds by asking whether you want to review all the options:

```
Do you want to review the options? [NO]
```

6. Proceed with the GNV installation by pressing the ENTER key to accept the default (NO, skip review of options). The installation proceeds with a progress report on the product installation as in the following example:

```
Execution phase starting . . .:

The following product will be installed to destination:
    DEC AXPVMS GNV V1.6     DISK$ALPHASYS:[VMS$COMMON.]

Portion done:0%...10%...20%...30%...40%...50%...60%...70%
...80%...90%..100%

The following product has been installed:
    DEC AXPVMS GNV V1.6    Layered Product
```

When the installation procedure is complete, the system returns you to the DCL prompt ($).

7. Finally, perform the steps described in Section 5 to set up GNV properly.

4.2 Installing Your GNV Software on a Location Other Than the System Disk

You can use the PRODUCT INSTALL command to install GNV on a location other than the system disk by specifying the location with the /DESTINATION qualifier. The target disk must be an ODS-5 disk.

The procedure for installing GNV on a location other than the system disk is equivalent to the steps outlined in Section 4.1, except the PRODUCT INSTALL command format in step 3 would be:

```
$ PRODUCT INSTALL GNV /SOURCE=device-name:[GNV.KIT] /DESTINATION=device-name
```

Note that the GNV$STARTUP.COM file is placed in the [SYS$STARTUP] directory on the target disk. You must either copy it to your SYS$COMMON:[SYS$STARTUP] directory or invoke it on the target disk from your SYS$MANAGER:SYSTARTUP_VMS.COM file. For more information on the SYS$STARTUP.COM file, see Section 5.

4.3 If You Move the Target Disk After Installation

The installation procedure creates file GNV_DESTINATION.COM in the [SYS$STARTUP] directory. This file points to the target location of the GNV kit, including the physical device specification of the target disk. If you should move the disk on which you installed GNV, edit the GNV_DESTINATION.COM file so that the device specification correctly reflects the new location.

5 Setup Requirements

GNV provides the following command procedures for use with your system.

1. SYS$STARTUP:GNV$STARTUP.COM — This file, located in the [SYS$STARTUP] directory of the disk on which you installed GNV, should be executed automatically at startup time. To have it executed at startup time, add the following line to your SYS$MANAGER:SYSTARTUP_VMS.COM file if you installed GNV on the system disk:

   ```
   $ @SYS$STARTUP:GNV$STARTUP.COM
   ```

   If you installed GNV on a disk other than the system disk, add the following line instead, where *device-name* is the device specification of the disk where GNV was installed.

   ```
   $ @device-name:[SYS$STARTUP]GNV$STARTUP.COM
   ```

   This startup file makes certain required systemwide definitions. Most importantly, it defines the GNU logical to point to the GNV top level directory.

2. GNU:[LIB]GNV_SETUP.COM — This file is to be executed by each user who will be using GNV. A user can have the file executed automatically at login by incorporating it in the user's LOGIN.COM file. If all users of a system will be using GNV, then this file may be executed in the systemwide LOGIN.COM file: SYS$MANAGER:SYLOGIN.COM.

   This file defines certain process-private symbols that cannot be implemented systemwide. These must be implemented on a per-user basis.

6 Using GNV

To use GNV, simply enter enter the `bash` command at the OpenVMS DCL prompt. The `bash$` prompt will then appear, as in the following example.

```
$ bash
bash$
```

Enter `bash` commands at this prompt, as shown in the following example, in which the `ls` command is entered:

```
bash$ ls
```

Alternatively, you can enter a single bash command at the OpenVMS DCL prompt, in the following format:

```
$ bash -c bash-command
```

For example, to enter the ls command from the OpenVMS DCL prompt, type the following line. After the contents of the working directory are displayed, the OpenVMS DCL prompt appears again.

```
$ bash -c ls
accountng.dat                          desktop.dir
errorlog.                              app.exe
help.dir

$
```

If the command you are entering has two or more components, the command and components must be surrounded by double quotes as in the following example:

```
$ bash -c "ls -al"
```

7 General Environmental Differences Between GNV/OpenVMS and UNIX Systems

This section discusses some of the differences observed between GNV features and utilities on OpenVMS and their counterparts on UNIX systems.

7.1 Root Directory

OpenVMS systems do not have a root directory similar to that of UNIX systems. The UNIX root directory (/) is the top level of the system file hierarchy. All directories on the system, irrespective of the physical device, are located under the root. On a native OpenVMS system, the closest entity to the UNIX root directory is the toplevel directory of a specific device. The character "/" is not recognized as a directory.

GNV makes use of a feature of the CRTL to implement a substitute UNIX root. GNV points this root at the primary GNV directory, generally on the system disk. Furthermore, GNV creates a number of directories commonly found immediately under a UNIX root directory: /etc, /usr, /bin, /lib,. and so forth.

This root directory is the top level of the GNV directory tree. You may use it to locate numerous files and directories. However, unlike a UNIX system, it is not true that all files and directories in the system can be found under the root.

7.2 Multiple Versions of a File

OpenVMS operating systems maintain multiple versions of a file, with the highest version number being the most recent. UNIX maintains only the most recent version of a file. With a few exceptions, GNV supports this UNIX feature. For example, the rm utility removes all versions of a file. The following are some of the utilities that still act only on the most recent version of a file, leaving earlier versions in place:

- mv
- chmod
- chown
- ln

For example, if you use `mv` to move (or rename) a file, only the highest version number file is moved. The lower versions (older) files are left in place.

7.3 Case Sensitivity in File Names

Normally, OpenVMS systems are not case sensitive. However, on ODS-5 devices you can enable case sensitivity for file names by using the following command at the OpenVMS DCL prompt or in a login command file:

```
SET PROCESS/CASE=SENSITIVE/PARSE_STYLE=EXTENDED
```

7.4 Variances in Interpretation of Characters and Unsupported Characters

The following subsections describe differences in the way OpenVMS GNV and the UNIX operating system interpret certain characters and lists characters that are unsupported.

7.4.1 File Names Beginning with a Period

OpenVMS lets you create a file name beginning with a period. The OpenVMS DIRECTORY command will list such files. UNIX systems consider such files as hidden. The UNIX `ls` command does not list such files unless, for example, the `-a` option is used or the file name is specified in the command line.

7.4.2 Control Characters

The following control sequences do not work as expected:

- CTRL/C (the response depends on the circumstances; if you do not get the expected response, try using CTRL/Y to bring you to the DCL prompt, and then enter the DCL CONTINUE command to bring you back to the original prompt or interrupted process)
- CTRL/Y (instead of bringing you to the previous command entered, this sequence might bring you back to the DCL prompt, stopping the `bash` program; try entering CTRL/Y a second time)

The CTRL/@ sequence has not been tested.

8 Reporting Problems

All normal problem-reporting channels are available for GNV users to report problems with the GNV software provided with OpenVMS Alpha 8.2EFT and OpenVMS I64EFT. Please note that OpenVMS engineering cannot guarantee resolution of all reported problems in this kit but will do its best to address all reports in a timely manner.

9 Bug Fixes

The latest GNV software includes fixes to bugs found on the preceding version of GNV (V1.5-6). The fixes will apply to GNV running on either Alpha or I64 systems, except where specified otherwise.

- CONFIGURE scripts and `config.h`

  GNV is often used on OpenVMS systems to port and configure OpenSource software. Most OpenSource packages are distributed in source form. A configure script is run to determine the features of the system it is running on, and to configure the build files, especially the make files and one or more include files (`.h`).

The configure script often generates an include file called config.h. This file contains the results of many of the tests performed by the configure script. The configure script runs numerous tests to determine features implemented by the local system. It then writes records to config.h based on the results of these tests.

In the past, many of the results written to the config.h file have been wrong. This has been traced to a bug in bash. This problem is resolved in the version of bash shipped with this release of GNV. The problem with bash is illustrated by the following example, where the wrong branch of the if statement was taken.

```
echo foo | true

if test `eval echo no` = yes
    then
        echo Writing to confdefs.h
    else
        echo Not writing to anywhere.
    fi
```

With this bug fix, configure scripts can be expected to generate more useful config.h files.

- Redirection of DCL commands on OpenVMS I64 systems

  Redirection of DCL commands now works reliably.

  OpenVMS on I64 systems tends to generate mailbox devices with five-digit unit numbers. On Alpha, these unit numbers were limited to four digits. Code used for redirection of DCL commands depended on four-digit unit numbers. For consistency with I64 systems, Alpha unit numbers have been increased to five digits.

- Definition of SYS$POSIX_ROOT

  The GNV startup file (SYS$STARTUP:GNV$STARTUP.COM) defines the logical name SYS$POSIX_ROOT, if that is not already defined. This allows GNV utilities to make use of a UNIX-like system root.

  This release defines SYS$POSIX_ROOT in a format that more closely matches the format expected by RMS. No directory is specified in the definition of this logical.

- Pipe to cut -f bug fixed

  In previous releases of GNV, piping output to a cut -fn command would hang. For example:

  ```
  bash$ cat nosuchfile | cut -f2
  ```

  This problem is resolved in GNV V1.6.

10 Known Problems and Restrictions

- Pipe operations hang

  Certain forms of pipe operations are known to have problems. For example:

  ```
  bash$ (cat t.txt) | less
  ```

  This command works fine for a small enough t.txt. If the file is larger, the command hangs.

One workaround to this and certain other pipe hangs is to use the Run-Time Library's feature to increase the buffer size of the mailboxes used to implement pipes, as in the following example:

```
$ DEFINE DECC$PIPE_BUFFER_SIZE 65000
```

Do not use a value much larger than 65000. For example, 65535 does not work.

For more information, refer to the discussion of the DECC$PIPE_BUFFER_SIZE feature in the Introduction Chapter of the *HP C Run-Time Library Reference Manual for OpenVMS Systems*.

This allows the command to work for moderately large files. However, sufficiently large files will still hang.

Another workaround to the hang is to use a temporary file as in the following example. Of course, this requires editing the script file.

```
bash$ (cat t.txt) > s.txt; less s.txt; rm s.txt;
```

The real problem is that `bash` should execute both the `cat` and the `less` child processes asynchronously. However, it does not currently do this. Instead, the `less` command does not get processed until the `cat` has completed. The output of the `cat` command is buffered in the pipe. If the pipe is not large enough to hold the entire output, then `cat` waits for the `less` function to read, but `less` does not run until the `cat` function completes. The result is a hang.

HP is attempting to resolve this problem in a future release of GNV.

- `bash` Command Line Editing

  There are numerous problems with the `bash` History File and Command-Line Editing.

  HP hopes to resolve these in some future release of GNV.

- $! shell variable returns a bad process id

  $! is supposed to return the process id of the most recently executed background job. Currently, it instead returns a job number, which is not useful for most purposes.

  HP hopes to resolve this problem in a future release of GNV.

11 Documentation

Table 2 lists documentation provided with the GNV kit. Once you have installed the GNV software, you can find the documentation files in the directories indicated.

Table 3 lists sources of information on the Internet.

GNV includes the MAN utility, ported from GNU MAN. Also included are MAN page files for most of the utilities included in the GNV kit. Again, these MAN page files are extracted from GNU sources and might not exactly represent the utilities included with GNV.

Note

Most of the MAN page files have filenames with multiple dots. If you attempt to install GNV on an ODS-2 disk, these files will be missing, and MAN will be unable to find them.

Disclaimer

The documentation listed in Table 2 is derived from a variety of sources and presented as is. HP has not reviewed these for correctness, accuracy, nor usability. These documents might not represent the version of the software provided with the GNV kit. For example, the `bash` reference manual (GNU:[src.GNV.BASH.DOCUMENTATION]BASH_REFERENCE_MANUAL.TXT) provided with this kit documents `bash` Version 2; this kit provides `bash` Version 1.14.

In addition, most of these documents are specific to GNU, not to the GNV kit or the utilities provided with OpenVMS.

At least several documents are .TEX source files (.TEXINFO). Many of these documents might not print or display correctly. Nevertheless, they might contain valuable information. Many of the .TXT files are nroff files. The .INFO files are generally clean and printable.

If you are looking for documentation of some GNU (or UNIX) command, and it is not available with the GNV kit, you can often find a MAN page or other document either on your favorite UNIX system or on the Internet (see Table 3 for several pointers). Obviously, such documents might not match the version of the software provided with GNV, but they will at least give you a general idea of how the software works.

In the following table, page counts are provided where appropriate and are approximations.

Table 2 GNV-Supplied Documentation

File Name	Description and Comments
GNU:[000000] or `/gnu`	
GNVREADME_FIRST. (.HTML, .PDF, .PS, .TXT)	This document, which provides GNV product information and installation/setup instructions. Also available on the Open Source Tools CD documentation area. 30 pages.
GNU:[SRC.GNV.BASH] or `/gnu/src/gnv/bash`	
INSTALL.	Provides build and installation instructions for `bash` on GNU; not necessarily applicable to GNV on OpenVMS. Four pages.
NEWS.	Lists recently fixed `bash` bugs. One page.
RELEASE.	Describes new features of `bash`. Five pages.

(continued on next page)

Table 2 (Cont.) GNV-Supplied Documentation

File Name	Description and Comments
GNU:[SRC.GNV.BASH.DOCUMENTATION] or /gnu/src/gnv/bash/documentation	
ARTICLE (.MS, .PS, .TXT)	White paper on bash. 11 pages.
BASH (.1, .PS, .TXT)	MAN page. Source unknown. The .TXT file contains some unprintable text. 37 pages.
BASH_REFERENCE_ MANUAL.TXT	Reference manual describing the features and functionality of bash. This manual documents bash Version 2. This kit contains bash Version 1.14. This manual documents features not present in the provided software. For example, two such features are (1) arrays and (2) several invocation options (for example, bash -r). Approximately 100 pages.
BUILTINS (.1, PS, .TXT)	MAN page that documents the bash built-in commands. 11 pages.
FAQ.	Contains a set of frequently-asked questions concerning bash. 12 pages.
FEATURES (.DVI, .INFO, .PS, .TEXI)	A 1994 document describing bash features and functionality.
READLINE (.3, .PS, .TXT)	MAN page documenting the readline( ) API. Useful for understanding bash command-line processing.
GNU:[SRC.GNV.BASH.CWRU] or /gnu/src/gnv/bash/cwru	
POSIX.NOTES	Discussion of the bash POSIX mode. One page.
GNU:[SRC.GNV.BASH.LIB.READLINE.DOC] or /gnu/src/gnv/bash/lib/readline/doc	
HISTORY.PS	Documents the history function (command-line recall) of readline. For both users and programmers. 18 pages.
HISTORY.INFO	Technical guide to the history library. 12 pages.
HSTECH.TEXINFO	Programmer's guide to the history library.
HSUSER.TEXINFO	User's guide to the history library.
GNU:[SRC.GNV.BASH.LIB.TERMCAP.GROT] or /gnu/src/gnv/bash/lib/termcap/grot	
TERMCAP.INFO*	This series of files provides programmer's information about the termcap library. 100 pages.
GNU:[SRC.GNV.FINDUTILS.4_1] or /gnu/src/gnv/findutils/4_1	
README.	Technical usage notes for the find utility. One page.
INSTALL.	Build and installation instructions applicable primarily to GNU on UNIX systems. Four pages.
NEWS.	New features in recent versions of the find utility. Four pages.
COPYING.	GNU General Public License regulations. Six pages.

(continued on next page)

Table 2 (Cont.) GNV-Supplied Documentation

File Name	Description and Comments
GNU:[SRC.GNV.FINDUTILS.4_1.DOC] or /gnu/src/gnv/findutils/4_1/doc	
FIND.INFO-1	User's guide for find utility. 25 pages.
FIND.INFO-2	Continuation of the user's guide. 18 pages.
FIND.TEXI	Source file for the user's guide.
PERM.TEXI	Discussion of UNIX file permissions. Six pages.
GNU:[SRC.GNV.GREP.GREP-2_4_2] or /gnu/src/gnv/grep/grep-2_4_2	
NEWS.	New features in recent versions of grep. Three pages.
README.	Readme file for grep. One page.
AUTHORS.	Acknowledgements. One page.
THANKS.	Acknowledgements. One page.
COPYING.	GNU General Public License regulations. Six pages.
INSTALL.	Build and installation instructions for the GNU version of grep. Four pages.
GNU:[SRC.GNV.GREP.GREP-2_4_2.DOC] or /gnu/src/gnv/grep/grep-2_4_2/doc	
GREP (.1, .INFO, .TEXI)	User's guide for grep. 13 pages.
GNU:[SRC.GNV.SED.SED-3_02] or /gnu/src/gnv/sed/sed-3_02	
COPYING.	GNU General Public License regulations. Six pages.
INSTALL.	Build and installation instructions for the GNU version of sed. Four pages.
NEWS.	New featuers introduced in recent versions of sed. One page.
README.	Readme file for sed. One page.
GNU:[SRC.GNV.SED.SED-3_02.DOC] or /gnu/src/gnv/sed/sed-3_02/doc	
SED (.1, .INFO, .TEXI)	User's guide for sed. 10 pages.
GNU:[SRC.GNV.MAKE.MAKE] or /gnu/src/gnv/make/make	
README.VMS	Release notes for the OpenVMS version of make. One page.
GNU:[SRC.GNV.LESS.LESS-358] or /gnu/src/gnv/less/less-358	
LESS.HLP	OpenVMS help file for the less utility. Four pages.
LESS.MAN	MAN page text file for the less utility. 30 pages.

(continued on next page)

Table 2 (Cont.) GNV-Supplied Documentation

File Name	Description and Comments
GNU:[SRC.GNV.TAR] or /gnu/src/gnv/tar	
AAAREADME.TXT	Build instructions and release for OpenVMS tar. Three pages.
GNU:[SRC.GNV.VITPU.DOC] or /gnu/src/gnv/vitpu/doc	
HOW-VI-WORKS.	Information on the internals of the vi text editor. Notes from developers. Three pages.
README.	Readme file for vi. Four pages.
TUTOR (.MEM, .RNO, .RNT, .RNX)	User's guide for novice vi text editor users. Six pages.
VI (.HLP, .HLB, .MEM, .RNO, .RNT)	Help file and guide to using the OpenVMS VITPU utility that emulates the vi text editor. 27 pages.
GNU:[SRC.GNV.GZIP] or /gnu/src/gnv/gzip	
GZIP (.1, .DOC, .INFO, .TEXI)	User's guide for gzip. Six pages.
GNU:[SRC.GNV.ZIP.MAN] or /gnu/src/gnv/zip/man	
ZIP.1	User's guide for zip in source format.
GNU:[SRC.GNV.ZIP] or /gnu/src/gnv/zip	
ZIP.HLP	Help file (OpenVMS) for zip.
GNU:[SRC.GNV.UNZIP] or /gnu/src/gnv/unzip	
UNZIP.HLP	Help file (OpenVMS) for unzip.
UNZIP.TXT	User's guide for unzip.
README.	Readme file for unzip.
GNU:[SRC.GNV.UNZIP.MAN] or /gnu/src/gnv/unzip/man	
UNZIP.1	User's guide for unzip.
GNU:[LIB-LIBGDBM] or /gnu/lib-libgdbm	
README.TXT	Describes GNU C libgdbm library components provided for OpenVMS Alpha and I64 systems. Explains how to build libgbdm, and lists the files provided, restrictions, and changes made for OpenVMS. Three pages.

(continued on next page)

Table 2 (Cont.) GNV-Supplied Documentation

File Name	Description and Comments
GNU:[LIB-LIBHASH] or /gnu/lib-libhash	
README.TXT	Describes GNU C libhash library components provided for OpenVMS Alpha and I64 systems. Explains how to build libhash, and lists the files provided, restrictions, and changes made for OpenVMS. Two pages.
GNU:[LIB-LIBREGEX] or /gnu/lib-libregex	
README.TXT	Describes GNU C libregex library components provided for OpenVMS Alpha and I64 systems. Explains how to build libregex, and lists the files provided, restrictions, and changes made for OpenVMS. Two pages.

Table 3 External Sources of Information

Description	location
GNU Project Website	http://www.gnu.org
GNV OpenVMS Website	http://gnv.sourceforge.net