Release Notes for HP-UX 9.0

HP 9000 Series 300/400

Printing History

New editions of this manual will incorporate all material updated since the previous edition.

The manual printing date and part number indicate its current edition. The printing date changes when a new edition is printed. (Minor corrections and updates which are incorporated at reprint do not cause the date to change.) The manual part number changes when extensive technical changes are incorporated.

August 1992 ... Edition 1.

Contents

Index

1

Introduction

This document contains information about HP-UX release 9.0, which runs on Series 300/400 computers.

This chapter:

- Describes the release notes.
- Explains the purpose of the release notes.
- Points to other sources of information about HP-UX.
- Describes differences between this document and other release-notes documents.

The Release Notes and Release Notes Files

The release notes are contained in this document and in a file that is part of the core HP-UX system. After an installation or update, whether you have one release notes file or several files depends on the product or products you purchased.

All release notes files exist in a directory named:

    /etc/newconfig/90RelNotes

Under that directory, there is a release notes file for the 9.0 HP-UX system. The name is:

    hpuxsystem

If you install (or update to) the HP-UX 9.0 release and subsequently update the 9.0 version by adding several products, the etc/newconfig/90RelNotes directory could contain release notes files such as:

```
hpuxsystem
C
Debugger
Fortran
Pascal
runtime
```

The directory for housing release notes files is named 90RelNotes so future update processes do not overwrite any existing release notes files.

Release Notes Purpose

The release notes give you information about new, changed, and removed functionality (features). By reading them, you can see the changes in relation to moving from an 8.x release to the 9.0 release. Where appropriate, the release notes also inform you about changes in the support of products.

The release notes do not fully document functionality. Instead, they contain cursory information about the functionality and provide pointers to documentation.

Other Sources of Information

Besides the release notes, which focus on changes from the previous release, you have other sources of information about HP-UX and its documentation.

README documents

Many products you can add to your system have a README document. These documents contain information about the product's install/update process that might not appear in the install/update document for HP-UX. Consequently, you might have several README documents. For the 9.0 release, some products that previously had README documents no longer have them. Instead, the README information might appear in release notes files. This varies across the range of products you can obtain.

/etc/newconfig/README

This is a file that lists all files in the /etc/newconfig directory. This file describes the files, their function, and their relationship to customizable files in the 9.0 system. The information can affect you because the files in /etc/newconfig can be changed when you perform an update.

Finding HP-UX Information

A document that describes HP-UX documents.

HP-UX Documentation Set

A set of documents that describes how to use HP-UX. For example, you get information about system administration, shells, networking, windows, programming, etc.

Differing Release Notes

This release notes is an ASCII file that is an integral part of the HP-UX 9.0 release. Prior to performing an update, you might have received a printed version of this file (this document). Because the printed document was published before the file was integrated with the system, there might be small differences in the two versions. However, this document is sufficient for you to study the new, changed, and removed functionality before you perform an installation or update.

2

Hardware (SPUs, Peripherals/Devices, Cards/Interfaces)

SPUs

No new computers were introduced at the 9.0 release.

Peripherals/Devices

The Series 300/400 systems now allow the use of autochanger cartridges that are numbered greater than 127. This features now matches that of the Series 700.

Cards/Interfaces

EISA System Board Configuration Files

Summary of Change

The E/ISA system board configuration files, which are located in the directory etc/eisa, were modified to prevent the eisa_config command from allowing E/ISA cards to overlap that portion of EISA memory space that is used by internal hardware. Since an overlap should have been caught by the driver, a valid EISA system configuration in a release prior to 9.0 remains valid because no cards were mapped on top of this hardware. This change detects the problem earlier and, if possible, automatically chooses a working configuration. This allows better sharing of card configuration files among workstations and PCs.

Warnings

When `eisa_config` runs in automatic mode from `etc/bcheckrc` and when the physical E/ISA system configuration has not changed, no warning that the E/ISA system board configuration file has changed is displayed and the system configuration remains valid. This is allowable behavior since nothing in the system board configuration file has changed that would invalidate a previously working system configuration.

If the E/ISA system configuration is changed or if `eisa_config` is run interactively, the following message is displayed:

        HP-UX E/ISA CONFIGURATION UTILITY

The system board `cfg` file (`/etc/eisa/!HWPCXXX.CFG`) has changed since the configuration was changed to NVM. The previous NVM contents (if any) are disregarded. If you want any E/ISA boards in your configuration, you must add them using an interactive `eisa_config`. Refer to the E/ISA Configuration Documentation.

In this state, the current E/ISA system configuration is invalidated. This is the correct behavior because `eisa_config` does not know that, in this specific case, the system board configuration file change can be safely ignored. Also, the warning message may be misleading in that the configuration of E/ISA cards can be done by using either automatic or interactive mode. ISA cards or special non-default configurations of E/ISA cards still require the use of interactive mode.

3

Topics Related to the HP-UX System

This chapter contains information about topics that relate to the HP-UX system (for example, the kernel, I/O, and standards).

Clusters

For the 9.0 release, the Series 800 systems do not support clusters. The Series 300/400 and Series 700 systems do support clusters.

Compatibility: Fast Symlinks (S300/400/700 Only)

Series 300, 400 and 700 systems now support a new method of storing symbolic link information that is faster and uses less disk space than the previous method. This feature is referred to as `fast symlinks` and is disabled by default. You may enable this feature by following the instructions below.

Once a disk contains `fast symlinks` you cannot move it to an older revision of HP-UX, or to a Series 800 system, since these systems do not support the `fast symlinks` feature and do not recognize the new symbolic link format. This is the reason that this feature is disabled by default.

To enable this feature, modify your kernel dfile to include the line:

```
create_fastlinks 1
```

and then regenerate your kernel and reboot. Existing symbolic links are not affected by this change, but any symbolic links created when `create_fastlinks` is 1 will be `fast symlinks` if they are 59 bytes or smaller. You can convert any existing symbolic links to `fast symlinks` by removing the

symbolic link and then recreating it after you have enabled the `fast symlinks` feature.

With the `fast symlinks` feature enabled, the Series 300, 400 and 700 store data for short symbolic links (59 bytes or less) within the symbolic link's inode. Previous versions of HP-UX stored the data for symbolic links in a data block, which used additional disk space (typically 1 KB) for each symbolic link. By storing the symbolic link data in the inode, less disk space is used, and fewer disk reads are necessary to reference the symbolic link. This is why the feature is termed `fast symlinks`. Symbolic links that are longer than 59 bytes are still stored in a file system data block, since the inode only has room for 59 bytes.

Any Series 300, 400 or 700 running HP-UX 9.0 (or later) recognizes `fast symlinks` even if the `create_fastlinks` parameter is not set to 1. The `create_fastlinks` parameter only controls how new symbolic links are created. This feature is transparent to NFS and HP-UX Diskless.

Core Commands

Commands discussed in this section adhere to the IEEE P1003.2 standard. (except HP proprietary commands)

Any command line parameter that begins with a - is treated as an option. To override this, use `--` to terminate the options list. For example:

        mv -- -a b

moves the file `-a` to `b`.

Interactive Performance

9.0 provides improved interactive performance of HP-UX while background process do paging and swapping to the disc. To maximize the interactive performance of the system, use the `nice` command when starting batch or background processes. See the `nice` section in this topic.

Environment Variables

The following environment variables may affect the operation of the command or system interface. Not all environment variables affect all commands or system interfaces. The HP-UX Reference lists the environment variables affecting a specific command or system interface.

- LC_ALL - This variable determines the values for all locale categories. The value of the LC_ALL environment variable has precedence over any of the other environment variables starting with LC_ (LC_COLLATE, LC_CTYPE, LC_MESSAGES, LC_MONETARY, LC_NUMERIC, LC_TIME) and the LANG environment variable.

- LC_MESSAGES - This variable determines the locale category for processing affirmative and negative responses.

Localized User Interface

Summary of Change

HP-UX commands, CUE, and TSM have localized user interface.

Message catalogs of HP-UX commands are localized into 6 languages:

```
French, German, Japanese, Korean,
Simplified Chinese, Traditional Chinese
```

CUE message catalog is localized into 3 languages:

```
French, German, Japanese.
```

TSM menus and messages are localized into 2 languages:

```
French and German.
```

Selected HP-UX Reference entries in Section 1 (about 50 files) are localized into Japanese.

Configuration

NLIO product is required to use localized user interface in Asian languages.

Size Requirement

Localized user interface requires less than 0.5 MBs of disk space for each language except Japanese. Japanese user interface requires less than 1 MB.

Accounting

Features

The limitations on the number of users and connection sessions for all system accounting commands are removed.

Summary of Change

Specifically, the following commands have their limitations removed:

- `diskusg(1M)` can now support more than 1000 users.

- `acctcon1(1M)` can now support more than 500 simultaneously connection sessions.

- `acctprc1(1M)` and `acctprc2(1M)` can now support more than 500 users and 500 sessions respectively.

basename(1)

Features

The basename command now conforms to POSIX.2 standard.

Summary of Change

The following changes are made to basename to conform to the POSIX.2 standard.

If a string passed to basename consists entirely of slash characters, the string is set to a single slash character. Previously, basename returned an empty string.

If a string contains trailing slash characters, basename removes them. So, if you pass a string that ends with a slash to basename, it returns only the string preceding the slash. Previously, basename returned an empty string.

If a suffix operand is present, and is identical to the characters `basename` extracts, then `basename` returns the suffix. Previously, it returned an empty string if the suffix matched the remaining string.

Impact

The changes may impact any script that uses `basename` because, for the cases listed above, `basename` returns a string that is different from the string it returned in previous releases. If your string contains trailing slash characters or if the suffix matches the extraction, the result is now different.

chmod(1)

Features

A new permission mode, X , permits conditional changes to the file's access mode.

Summary of Changes

The octal form of the mode option is now listed as obsolete. It is still supported, but symbolic modes should be used instead of octal modes.

If the who part of a symbolic mode is omitted, then the permission is modified based upon the file mode creation mask that chmod inherits. A permission is changed only if the corresponding bit(s) in the creation mask are clear. In past releases, when who was omitted, then it defaulted to User, Group, and Other.

chown(1)

Features

A file's group can be changed at the same time as its owner using the new:

        group synt

cksum(1)

The cksum command is a synonym for sum -p. The 9.0 cksum implements a
new CRC check sum algorithm as specified in the POSIX 1003.2 Draft 11.3
standard. The new algorithm is different from the one implemented in the
release 8.0 version of cksum (sum -p). Consequently, the check sum output
in release 9.0 is probably differ from the check sum output in release 8.0.
This affects only the output of cksum (sum -p). The check sum generated by
executing sum without options is not affected (and should be the same across
the two releases).

Summary of Change

The CRC checksum generating polynomial has changed to

```
G(x) = x^32 + x^26 + x^23 + x^22 + x^16 +x^12 +x^11 +x^10 + x^8 +
    x^7 + x^5 + x^4 + x^2 + x^1 + x^0
```

The previous generating polynomial was

```
    x^32 + x^7 + x^5 + x^3 + x^2 + x^1 + x^0
```

For each file processed, the cksum utility writes out the checksum, the number
of octets, and the pathname, to the standard output. This is a change from the
previous implementation where the number of blocks were written out instead
of the number of octets.

Impact

Due to the change in the generating polynomial, the calculated checksum
differs from the one that was calculated in the older versions. Application that
use this can be impacted. Applications that use the number of blocks that were
previously output, along with the checksum, are also impacted because this was
changed to the number of octets.

Alternatives/compatibility

The number of blocks can be obtained from the number of octets by dividing
the number of octets by the Block Size (512).

Any application that uses this to generate the checksum and then use it for
checking purposes has no problem, but those that used the older version to

generate the checksum and use this version for verifying or checking find a mismatch. There is no workaround.

Obsolescence

It is recommended that the command **cksum** be used instead of **sum -p**, which has become obsolete.

cp(1)

Features

The **-R** Copy file hierarchies option was added.

Summary of Change

With the **-R** and **-r** options, cp replicates the source directory tree into the destination directory. With the **-R** and **-r** options, in addition to regular files and directories, cp also replicate FIFOs, character and block device files, and symbolic links. Only **root** can copy device files. All other users get an error.

Symbolic links are copied so the target points to the same location that the source did. If **filea** is linked to **fileb**, than **cp filea filex** results in **filex** being linked to **fileb**.

Hard links are copied so the hard link relationship within the source directory tree is maintained within the destination directory.

```
Given:

dir0        dira
----        ----
fileZ       filex (hard linked to fileZ)
            filey (hard linked to filex)

            cp -R dira dirb will result in

        dirb/filex
        dirb/filey (hard linked to dirb/filex)
```

- The -r option now copies special files. The `warning: skipped XXXX` messages have been removed.

- The -p option attempts to preserve the last modification and last access times. It preserves the UID and GID. It attempts to preserve the permission and S_ISUID and S_ISGID bits.

The -p option does a `chown(2)` to set the ownership information on the destination file before it does the `chmod(2)` call. If the original file does not belong to the user doing the cp the `chmod(2)` fails and the permissions for the destination file are left at zero (0).

- The current cp command did not check for failure on various file system calls. All such calls now produce an error message if they fail.

- If there are multiple file operands and the last operand does NOT exist, it is assumed to be a directory. It is created and the other `file operands` copied into it.

Given files A and B. If c does not exist `cp A B c` creates directory c and copies files A and B into directory c. If c exists, an error message is printed.

- If an error occurs while copying a file from the source tree, the copy operation continues with the next source tree element.

- The following error message was changed:

      8.0 message: cp: %s directory

      9.0 message: cp: %s directory. Need -R option.

Impact .

- With 9.0, a recursive copy (`cp -R`) replicates special files. File copy attempts that previously would have generated warning messages, now execute. And, if the user is not root and an attempt is made to copy a device file, a different error message is issued than occurred at 8.0.

- Scripts checking for the `cp: %s directory` error message may need to be changed.

- Copying a file owned by another user using `cp -p` does copy the file. It also sets the ownership of the file to the owner of the source file, sets the permission bits to zero, and issues an error message. An error message is

issued because the chmod(2) call fails since the user is not the owner of the file.

csh(1)

Features

The csh shell now supports SIGWINCH. SIGWINCH properly interprets the resizing of a window. This feature was previously available for Series 700 computers running 8.07.

Summary of Change

Prior to this release, when getting its input from a file, csh would exit immediately if it was unable to execute a command (for example, if it was unable to find the command). Now, csh continues on and attempts to execute the remaining commands in the file. This is the behavior that seems to occur on most other systems. However, to provide backward compatibility, the user may set the environment variable EXITONERR to 1 to get the old behavior.

cue(1)

Support of new terminals

CUE and TSM support of new terminals (700/60, 700/96 and 700/98) is as follows:

Since 700/96 and 700/98 are direct replacements for existing/supported terminals (700/92 and 700/94, respectively), there is no special consideration. With TSM, however, users should ensure that the TERM environment variable is set to either 70092 or 70094.

The 700/60 terminals are supported only if they are in the emulation mode of an existing/supported terminal. Some examples include Wyse50 and VT100. For a complete list of TSM supported terminals please refer to the TSM User's Guide (Part Number B2351-90000). The current list of CUE supported terminals is:

        2392, 2394, 700/92 and 700/94.

cut(1)

Summary of Changes

The cut(1) command no longer treats the backspace character in any special way. This means that the combination char-back_space-char is treated as 3 distinct characters. Previously, this combination would have been treated as a single (possibly bold) character.

Impacts

Existing scripts relying on special treatment of backspace characters embedded in an input file to cut(1) do not work properly. Please see the Alternatives section for a workaround.

Alternatives

To achieve compatible behavior for cut(1) when an input file contains backspace characters, use other commands (for example, fold(1) or col(1)) to process backspaces, and pipe the output to cut(1).

date(1)

Summary of Changes

The date(1) command has new options. The additions include option C to print Century as a decimal number, option e to print day of the month as a decimal number in a two digit field with leading space fill, and the E and O modifier characters that indicate a different format for some of the other options. Options that can be modified with the E modifier are: c, C, x, y and Y. Options that can be modified with the O modifier are: d, e, H, I , m, M, S, U, w, W and y.

dd(1)

Features

The dd command handles the following new features in the HP-UX 9.0

■ Addition of some options (block, noblock, notrunc):

```
conv=block,conv=unblock,conv=notrunc
```

The following shows an example of executing dd on the command line with the new options:

```
dd if=INPUT-FILE of=OUTPUT-FILE cbs=NUMBER conv=block

dd if=INPUT-FILE of=OUTPUT-FILE cbs=NUMBER conv=unblock

dd if=INPUT-FILE of=OUTPUT-FILE conv=notrunc
```

■ Addition of code to check for a cascade type of input or output file.

Summary of Change

The dd command supports a conversion option to have ASCII, EBCDIC, IBM Sync outputs. New options (block, unblock, and notrunc) have been introduced for POSIX Conformance. The functions of each are as described below

Block. Treat the input as a sequence of
terminated or EOF terminated variable length records independent of the input block boundaries. Each record shall be converted to a record with a fixed length specified by the conversion block size (The parameter **cbs**)

Unblock. Convert fixed length records to variable length. Read a number of bytes equal to the conversion block size, delete all trailing spaces and append a newline.

Notrunc. Do not truncate the output file. Preserve blocks in the output file not explicitly written by this execution of the dd command.

Impact

Usage is not impacted.

Performance

There is no impact on performance

diskinfo(1M)

At 9.0, this command displays size in KBs to accommodate larger disks. Previously, the command displayed size in bytes.

ed(1)

Summary of Change

The following changes have been applied to HP-UX 9.0 ed:

- Line length support (maximum number of characters per line in a file) increases from 512 to LINE_MAX (2048, defined in `/usr/include/limits.h`).

- As a result of the new line length support, the possible value of the n suffix in a substitute command increases from the original value of 512 to LINE_MAX.

- The environmental variable SHELL determines the preferred command-line interpreter for use in all ! style commands. If this variable is null or not set, `sh(1)` is used.

- The l (ell) command lists most non-printable characters as one three-digit octal number with a preceding backslash. Exceptions are backslash, alert, back-space, form-feed, carriage-return, tab and vertical tab, which are listed respectively as:

    `\\, \a, \b, \f, \r, \t, and \v`

- The l (ell) command marks the end of each line with a $.

- Entering an end-of-file (Control-D) character in command mode cannot replace the q or Q command to quit an editing session.

- It is not allowed to enter a single period in input mode as backslash, period, or newline. The suggested method is to precede the period with another character and then use the substitute command to delete that character.

env(1)

Summary of Change

The syntax is:

```
env [-] [-i] [name = value] ...  [command [arguments ...]]
```

If the command is executed, the exit status of env becomes the exit status of command; otherwise, the **env** utility exits with one of the following values:

0   The env utility completed successfully.

1-125   An error occurred in the env utility

126   The command specified was found but could not be invoked.

127   The command specified could not be found.

fold(1)

Features

A new option:

```
     -b  (counts bytes instead of characters)
```

Summary of Change

The syntax is:

```
fold [ -bsw num ] input_file
```

Specifying **-b** on the command line causes the input to be folded every **num** bytes, rather than **num** columns. The value for **num** is still specified with the **-w** option. The default value for **num** is still 80.

Impact

Because some characters do not correlate well between bytes and columns, using the **-b** option can produce unexpected results. For example, it is possible for an 80 byte string to occupy as little as 1 column (all carriage returns) or as many as 640 columns (all tabs). If the input text contains tabs or other

multi-column characters (example: some NLS character sets), this option is probably not appropriate.

Performance

No performance impact.

Alternatives and Compatibility

No compatibility impact.

fsck(1M), mkfs(1M), newfs(1M)

Features

The default behaviors for fsck, mkfs, and newfs have been changed to provide higher protection for the user when the target device is in use (a mounted file system, or a swap device).

It is considered disastrous to perform newfs, mkfs, or fsck on a swap device. Thus, the commands stop when the target device is a swap device.

The fsck command now warns and prompts the user when the target device is a mounted file system but the -n or -F option has not been specified. However, this only happens when the init state is not in single user mode.

The mkfs and newfs commands stop to process when the target device is in use. If option -F is specified, the commands do not stop when the target device is a mounted file system.

Summary of Change

A new option, -F, was added to force fsck, mkfs, and newfs to continue to process, without prompting the user, when the target device is a mounted file system. The commands do not work on a swap device.

Impact

Existing fsck, mkfs, and newfs scripts need to specify option -F if they want to work on a mounted file system while they are not in init state s.

getconf(1)

Features

The following symbols, for which configuration values can be obtained, have
been added:

BC_STRING_MAX	COLL_WEIGHTS_MAX
POSIX2_BC_BASE_MAX	POSIX2_BC_DIM_MAX
POSIX2_BC_SCALE_MAX	POSIX2_BC_STRING_MAX
POSIX2_COLL_WEIGHTS_MAX	POSIX2_C_BIND
POSIX2_C_DEV	POSIX2_EXPR_NEST_MAX
POSIX2_FORT_DEV	POSIX2_FORT_RUN
POSIX2_LINE_MAX	POSIX2_LOCALEDEF
POSIX2_RE_DUP_MAX	POSIX2_SW_DEV
POSIX2_VERSION	_POSIX_ARG_MAX
_POSIX_CHILD_MAX	_POSIX_LINK_MAX
_POSIX_MAX_CANON	_POSIX_MAX_INPUT
_POSIX_NAME_MAX	_POSIX_NGROUPS_MAX
_POSIX_OPEN_MAX	_POSIX_PIPE_BUF

Summary of Changes

The following symbols have been removed:

SENDTO_MAX	COLL_ELEM_MAX
PASTE_FILES_MAX	SED_PATTERN_MAX

These were dropped from the IEEE P1003.2 standard.

grep(1)

Summary of Change

In `-e pattern_list`, the `pattern_list` can be a newline separated list of patterns. The `-e` and `-f` options can be used simultaneously.

id(1)

The `id` command now takes a `-G` option. The output format for the `-g` option was changed. The output format now has commas between supplementary group ids. This may affect any program interpreting the output of `id`.

ioscan(1M)

This new command generates a list of interface cards and disk and tape devices that are attached to a system. To get information, see the `ioscan(1M)` entry in the HP-UX Reference.

The `cs80` driver supports a maximum of 16 open devices at the same time. In the previous release, the `cs80` driver supported more than 16 devices for Series 300/400 computers and supported 16 devices for Series 700 computers.

In the 9.0 release, the `simon.h` header file is not included in the build environment because the driver was merged with the Series 700 HPIB interface driver called `hshpib`. Applications that included `simon.h` must now include `hshpib.h`. The source code for an application might also require changes because some constants have been changed in the `hshpib.h` header file. To do this, compare the `simon.h` file for 8.x and the `hshpib.h` file for 9.0.

ksh(1) - Korn Shell

Summary of Change

The `trap` command now also works with the symbolic names of the signals. The signal names can be obtained by the `kill -l` command.

Major Defect Fixes

- Before 9.0, the `ksh` command was unable to open some script files due to access failures; this problem was fixed.

- The shell builtin command `kill -CONT pid` no longer causes the login shell to exit when trying to awaken a suspended process.

- Sending a SIGSTOP to a child process no longer kills the non-interactive POSIX Shell.

- The scope of the typeset variables in a function is within the function call. The space used by these variables is now released once the function is finished.

- While creating symbolic links, Korn Shell no longer leaves a trailing + in the file name.

- The exit value is now 126 for files that are found but not executable, and 127 for files not found.

- The shell no longer leaves the PWD uninitialized, even if it was not previously set. The shell now initializes the PWD at every execution.

- POSIX Shell can now expand patterns such as `c+*` correctly in all cases.

- A `cd` to the current directory no longer fails if . is a stale NFS file handle. Now, this is detected and a `cd` to `full_path_name_of` . is done.

- Jobs numbers no longer truncate to 8 bits. Possible simultaneous jobs is no longer limited to 2. This limit was enhanced to CHILD_MAX, which is a configurable parameter, but is subject to system constraints like memory available.

Alternatives and Compatibility

The HPUX 9.0 Korn Shell has total compatibility with HPUX 8.0 Korn Shell. Changes have been introduced only in those areas which were causing severe defects in the Korn Shell functionalities.

Size Requirement

With the shared libraries linkage, the Korn Shell executable takes about 230 KBs of disk space.

ln(1)

Summary of Change

The ln(1) command no longer overwrites an existing file by default. The -f option must be used to overwrite an existing file.

Impact

Trying to link to an existing file with ln(1) fails if the -f option is not given. Previously, ln(1) would do the link if the destination file existed, even if the -f option was not specified.

Alternatives and Compatibility

Use the -f option, if a link to an existing file is desired.

lpstat(1M)

Features

The lpstat command now reports the remote machine and device for remote printers.

Summary of Change

The lpstat -v and lpstat -t commands report more information for remote printers

Impact

Users see more information on remote printers.

Performance

No impact on performance.

ls(1)

Summary of Changes

If the -F option is specified and if the file is a FIFO file, then a vertical bar is put after the file name.

If both options are specified in each of the following pair: -C and -l (ell), -m and -l (ell), -x and -l (ell), -C and -1 (one), -c and -u, the last option specified determines the output format.

Impact

The output format may look different if the file list contains a FIFO file and if the -F option is specified. If multiple options are specified, the output may be different, because of the differences in precedence rules.

mklp(1M)

Summary of Change

This command will be obsoleted for HP-UX 9.0

Alternatives

Use SAM or the lpadmin command to configure printers.

nice(1)

The HP-UX kernel was enhanced so that nice has a larger impact on non-CPU-bound processes. For example, batch processes that page often, have caused interactive performance problems. These problems have been solved in release 9.0. The general syntax is:

    nice -increment batch_process

The HP-UX Reference has more information.

In previous releases of HP-UX, the nice level of a process affected only CPU scheduling, which was largely irrelevant for non-CPU-bound processes. Release

9.0 includes changes that expand the impact of the **nice** level to the paging and swapping algorithms and to the disc I/O subsystem.

If a batch process is not started this way, and you wish to change its behavior without restarting it, see the command **renice**, or the system calls named **getpriority()** and **setpriority()**.

od(1)

Features

A new form of the command is now supported. The old form, although listed as obsolete, is still supported, but the new form should be preferred for use. The new form is:

```
od [ -v ]
   [ -A address_base ]
   [ -j skip ]
   [ -N count ]
   [ -t type_string ]
   [ file 0...]
```

Summary of Change

The **-v** option writes all data, without replacing matching output lines with an asterisk.

The **-A** option specifies the input offset base. **address_base** is a single character (d, o, x) that specifies the input offset base corresponding to decimal, octal, hexadecimal respectively. Use the character **n** to specify that the offset need not be written.

The **-j skip** option jumps over **skip** bytes from the beginning of the input.

The **-N count** option formats no more than **count** bytes of input. **count** is by default a decimal number unless it contains a leading **0x** or **0**, in which case it is treated, respectively, as a hexadecimal number or an octal number.

The **-t type_string** option specifies one or more output types. Use characters a, c, d, f, o, u and x, respectively, to specify **named character,** **character, signed decimal, floating point, octal, unsigned decimal** and **hexadecimal.** Characters d, o, u, x can be modified by an optional C, S, I

or L modifier, indicating that the conversion should be applied, respectively, to an item of type `character`, `short`, `integer` or `long`. The specifier `f` can be similarly modified by F, D or L, indicating that the conversion should be applied, respectively, to the type `float`, `double` or `long double`. In addition, you can append an unsigned integer to these specifiers to indicate the number of bytes of each instance of the output type to transform.

pr(1)

Summary of Changes

The `pr(1)` now accepts `-F` option as well as `-f`.

Impacts

Support continues for `-f` option; however, for portability, using `-F` is recommended. The functionality is equivalent.

printf(1)

Features

An additional conversion specifier b is supported.

Summary of Change

Use the b format specifier to print strings with `backslash-escape` sequences. b is like the s conversion specifier, except that the string may contain `backslash-escape` sequences that are converted to the characters they represent.

renice(1)

The nice level of an already running process can be changed with the new command named **renice**. For example, to nice a process already running with process id 197, execute:

```
renice -n -19 -p 197
```

Variations of this command can also change the priority of all processes for a user or all processes in a process group. See the **nice** section in these release notes for more information about how **renice** relates to interactive performance. See the HP-UX Reference for more information about **renice(1)**.

getpriority() and setpriority()

Use these new system calls to get and set the current nice level of a group of processes. This facilitates programmatic **renice**ing of a process.

rm(1)

Summary of Change

The syntax is:

```
rm [-f|-i] [-Rr] file ...
```

When removing a directory, if the user does not have write permission and the **stdin** is a tty, rm asks the **Remove?** question. rm also asks the **Remove?** question when the **-i** flag is set.

sed(1)

Summary of Change

The **sed(1)** command was changed to represent nonprintable characters by printing, with a preceding backslash, a three-digit octal number or the corresponding escape sequence.

The **s** command's n flag limit is now LINE_MAX (2048).

The hold space character limit was 4000 but is now 8192.

POSIX Shell; sh-posix(1)

Features

POSIX Shell (`/bin/posix/sh`) is very similar to Korn Shell. POSIX Shell is nearly compliant with the IEEE P1003.2 recommendations. POSIX recommendations will soon be adopted as the industry standard.

As an indication of HP's commitment to these evolving standards, the POSIX Shell is the default shell in the HP-UX 9.0 release. The files `/etc/profile`, `/etc/csh.login` and `/etc/d.login` are changed so the path `/bin/posix` is introduced ahead of `/bin` in the default PATH variable. Users who have set up their own PATH in `$HOME` or `$HOME/.cshrc` may not see the changes mentioned below.

What is different?

- The default PATH is set to:

  PATH=/bin/posix:/bin:/usr/bin:/usr/contrib/bin:/usr/local/bin

- When `sh` is typed at a shell prompt or when a shell escape like `!` is used from some utility, the POSIX Shell may be executed instead of the Bourne Shell.

- If `sh` passes as an argument to `execlp()` or `execvp()`, then it invokes the POSIX Shell.

- SAM now offers `/bin/posix/sh` as the default shell when setting up new user accounts or modifying existing ones.

- Shell scripts that do not use `#!shell path`, run under the POSIX Shell if the POSIX Shell was used to execute the shell script.

What stays the same?

- Login shells do not change automatically. If a user's login shell is `/bin/sh`, it stays the same.

- Shell scripts that use `#!shell_path` continue, as before, invoking the shell defined in the `shell_path`.

- Bourne Shell continues to be used when the system is in the single user mode.

Summary of Change

The following regular builtin utilities are now available as executable versions:

alias   Define or display aliases

bg   Run jobs in the background

cd   Change working directory

false   Return false value

fc   Process command history list

fg   Run jobs in the foreground

getopts   Parse utility options

jobs   Display status of jobs in the current session

kill   Terminate or signal processes

newgrp   Change to a new group

read   Read a line from standard input

true   Return true value

umask   Get or set file mode creation mask

unalias   Remove alias definitions

wait   Await process completion

New features of the POSIX Shell

`command` is now a builtin utility. There is also a separate, executable version of it. The `command` utility causes the shell to treat the arguments as a simple command, suppressing the shell function lookup. The shell by default does the command lookup in the order of:

    aliasexpansion::builtin_utilities::function_search::PATH search

A `trap` now also works with the symbolic names of the signals. The `kill -l` command retrieves the output of signal names. The format of the `trap` output is such that a shell script can use it to save, and then later reuse, a trap. For example:

        save_traps=$(trap) eval $save_traps

Added -a option to `unalias` builtin utility. This option can be used to remove all the alias definitions.

Added -p option to `export` builtin utility. When -p is specified, export writes to the standard output the names and values of all exported variables in the following format:

    export *variable=value*

The shell formats the output, including the proper use of quoting, so that it is suitable for re-input to the shell as commands that achieve the same exporting results. For example:

    export -p >temp_file
    temp_file   # restores variables.

Added -p option to `readonly` builtin utility. When -p is specified, `readonly` writes to the standard output, in the following format, the names and values of all read-only variables:

    readonly *constant=value*

The shell formats the output, including the proper use of quoting, so that it is suitable for re-input to the shell as commands that achieve the same attribute setting results.

Added -f and -v options to the `unset` builtin utility. If -v followed by a variable name is specified, the shell unsets the variable and removes it from the environment. If -f followed by a function name is specified, the shell unsets the function definition. If neither -f nor -v is specified, the shell checks first for a variable name, and if the variable is not found, the shell checks for the function definition.

Added -S option to the `umask` builtin utility to print the mask in symbolic mode. The default octal representation of the umask value is obsolete. For example:

    $ umask -S
      u=rwx,g=rwx,o=rx

Added -e option to the `test` builtin command. This option returns true if the file exists.

Removed `time` keyword from the POSIX Shell. The `/bin/time` utility is now called.

Changed the following messages

        Stopped (signal) changed to Stopped (SIGSTOP)

        Stopped (tty input) changed to Stopped (SIGTTIN)

        Stopped (tty output) changed to Stopped (SIGTTOU)

Added `-s` option to the builtin `kill` command. The new syntax is as follows:

        kill -s signal_name pid
        kill -l

The `kill -l` form can be used to get valid signal names. The following obsolescent versions are still supported:

        kill [-signal_name] pid
        kill [-signal_number] pid

Major Defect Fixes

See `ksh(1)` Major Defect Fixes.

Impact

The availability of executable versions of the builtin utilities create a user awareness of the existence of such functionalities in the shell. These features were previously documented only in the `ksh(1)` or `sh-posix(1)` entries in the HP-UX Reference. Now separate entries exist for each of the new utilities shown above.

One problem in having these utilities is trying to execute one in a shell that does not have that particular builtin utility. If you try to use the `fc` utility in the Bourne shell, the shell executes the executable version of `fc` from the `/bin` directory. There is no output from this `fc` because it is executed in a child sub-process. Users must be aware of the existence of these utilities. Similarly, the `/bin/cd` could be used as given in the following example, but this does not affect the working directory of the caller's environment.

```
find . -name name -exec cd
```

Performance

Alternatives and Compatibility

The POSIX Shell is to a great extent compatible with the HP-UX 9.0 Korn
Shell. So the existing users of the Korn Shell can switch between the HP-UX
9.0 Korn Shell and HP-UX 9.0 POSIX Shell with minimum impact.

Size Requirement

The size of POSIX Shell is comparable to that of the Korn Shell. With the
shared libraries linkage, POSIX Shell executable takes up around 230K bytes of
disk space.

sort(1)

Features

Digit-grouping separators are now handled in numeric sort keys specified with
the -n option. For example, commas in the string 1,234,567 are ignored and
thus the string is sorted based upon its arithmetic value.

Summary of Changes

The [+pos1 and [-pos2]] forms of restricted sort keys are listed as obsolete.
Their use is still supported, but the -k keydef option should be used instead.

Prior to release 9.0, the field_start and field_end parts of a restricted
sort key incorrectly referred to fields and positions with fields starting at 0.
Numbering now starts at 1.

The -c option no longer produces any output. The exit status is the only
indication that the input was correctly presorted or not.

Performance

The sort command uses up to 16 MBs of memory to sort larger data sets if
sufficient memory is available on the system. The prior limit was 1 MB.

su(1)

The su(1) command, when executed without the - argument, removes the HOME and ENV variables from the environment before executing a shell. This behavior is not backwards compatible with release 8.0, but it was required for security reasons.

swapinfo(1M)

For file system swap areas with non-zero RESERVE values, the swapinfo command indicates the units of the RESERVE value are KB, but actually they are file system blocks, as given to swapon(1M), which is usually 8 KBs.

tail(1)

Summary of Changes

Command line execution of tail(1) is completely changed to the following:

    tail [-f][-c number | -n number | -b number] [file]

Please refer to the HP-UX Reference for information. Obsolescent versions of the command are not affected.

Impacts

Existing scripts using the old syntax of tail(1) need to be modified to conform to the new one.

touch(1)

Summary of Changes

The touch(1) command now exits with status 0 when the -c option is used and the specified file does not exist. Previously, the same condition would result in a non-zero exit status.

Impacts

Existing scripts, relying on a successful (non-zero) exit status of touch(1) when -c option is used and the specified file does not exist, do not work properly.

tr(1)

Features

An additional feature to accept ctype and equiv classes with single brackets was added.

vi(1)

Summary of Change

The line length support (maximum number of characters per line in a file) was increased from 512 to LINE_MAX (2048, defined in /usr/include/limits.h).

Core Library Interfaces

Libcurses

Features

A default signal handler is added to the curses library to respond to the SIGWINCH signal generated as a result of any resizing operations on the window in which a curses application resides. Upon a window resize, the curses application immediately redraws the screen with the new window size. The environmental variables LINES and COLUMNS are also updated so that all the children processes of the application work with the correct window size.

In addition to the new signal handler, performance enhancements have been added to the curses library so that it is significantly faster.

Summary of Change

If there is a reduction to the window size, part of the application display is trimmed. Additionally, a snapshot of the trimmed portion is saved in the internal memory at the time of resize. This portion is not affected by the application as long as it stays invisible. If the cursor for the applications is trimmed, unexpected behavior results.

If a window is enlarged, any previously trimmed area is re-displayed (and re-activated). If the window is enlarged beyond its initial window size, the extra area is padded with blank spaces.

With the SIGWINCH support, curses applications are no longer limited to the initial window size. In other words, applications can specify coordinates beyond the initial window size at the startup time, but those extra areas are not visible (and active) until the window is enlarged.

Impact

To use the default libcurses SIGWINCH support, there is no extra action required. A user can choose to disable SIGWINCH support by installing a local SIGWINCH handler via the `sigvector(2)` command.

Performance

With the performance improvements, the new curses library consumes fewer CPU cycles and, by employing more intelligent screen-update algorithms, often causes fewer characters to be output to the terminal. None of the library interfaces have been changed, so applications do not need to be modified in any way.

All applications using the curses library should realize some performance improvement. The extent of the improvement is dependent not only on the nature of the application, but also on the type of terminals being used. Applications that make extensive use of highlighted display characters and/or output to HP terminals benefit from using the new library. However, the user might notice the screen being updated differently from before.

fnmatch(3C)

The syntax is:

```
int fnmatch(const char *pattern, const char *string, int flags);
```

Features

The FNM_NOESCAPE flag can be set as a part of the `flags` argument.

Summary of Change

If FNM_NOESCAPE is set in `flags`, a backslash character in `pattern` is treated as an ordinary character. If it is not set, then a backslash character in `pattern` followed by any other character matches that second character in `string`.

fprintf(3S)

Summary of Change

Two new conversion specifiers have been added: %C for wide characters (`type wchar_t`), and %S for wide character strings (`type wchar_t *`).

Impact

These additions are part of the X/Open Worldwide Portability Interface.

fscanf(3S)

Summary of Change

Two new conversion specifiers have been added:

- %C for wide characters (`type wchar_t`)
- %S for wide character strings (`type wchar_t *`).

Impact

These additions are part of the X/Open Worldwide Portability Interface.

getgrent(3C)

Summary of Change

The maximum number of users per group was increased to 220.

glob(3C)

Summary of Change

The syntax is:

```
int glob(const char *pattern, int flags,
    int (*errfunc)(const char *, int), glob_t *pglob);
```

The following items show the changes:

- Added GLOB_NOMATCH return value

- Added GLOB_ABORTED return value

- Added GLOB_NOESCAPE functionality to skip over escape in pattern

- Removed GLOB_ABEND definition

- Replaced GLOB_QUOTE with GLOB_NOESCAPE

popen(3S) and pclose(3S)

Summary of Change

The popen(3S) call now invokes the POSIX command interpreter,
/bin/posix/sh to execute the command string.

Impact

Although backward compatibility is assured in command syntax,
/bin/posix/sh exhibits slightly different semantic behavior from /bin/sh (the
interpreter that popen(3S) executed before 9.0) in the following areas:

- Trap handling

- Signal handling

- Set-user-id scripts

- Asynchronous job handling

- Different wording/spacing on some error messages

- The `/bin/posix/sh` command ignores the value of the environment variable `IFS` when executed.

- In a `chrooted` environment, `/bin/posix/sh` must be present.

A `pclose(3S)` now returns a value of 127 if `/bin/posix/sh` cannot be executed. Previously, a `pclose(3S)` returned 1 for this error condition.

regerror(3C)

Summary of Change

Added REG_ESPACE as return error for out of memory

regexec(3C)

Summary of Change

The syntax is:

```
int regexec(const regex_t *preg, const char *string,
    size_t nmatch, regmatch_t pmatch[], int eflags);
```

The following changes were made:

- Added new members (`rm_so, rm_eo`) to `regmatch_t`

- Obsoleted members (`rm_sp, rm_ep`) in `regmatch_t`

Impact

Applications that depend on these obsoleted fields no longer work. Shared library bound applications that use these structures must be recompiled.

rewind(3S)

Summary of Change

In conformance with ANSI, POSIX and X/Open standards, `rewind(3S)` no longer flushes the streams buffer.

Impact

This change may cause unexpected results in applications that expect the buffer to be flushed. Therefore, two versions of `rewind(3S)` are provided in release 9.0. Applications that were linked with shared libraries on HP-UX 8.x releases automatically link with the old version of `rewind(3S)`. Applications that are compiled on HP-UX release 9.0 automatically link to the new, conformant version of `rewind(3S)`.

sysconf(2)

Features

The following symbols, for which configuration values can be obtained, have been added:

        _SC_BC_STRING_MAX   _SC_COLL_WEIGHTS_MAX

        _SC_2_LOCALEDEF   _SC_SSIZE_MAX an

        _SC_2_FORT_RUN

Summary of Changes

The following symbols, which were dropped from the IEEE P1003.2 standard, have been removed:

        SENDTO_MAX   COLL_ELEM_MAX

        PASTE_FILES_MAX   SED_PATTERN_MAX

system(3S)

Summary of Change

A system(3S) now invokes the POSIX command interpreter, /bin/posix/sh, to execute the command string.

Impact

Although backward compatibility is assured in command syntax, /bin/posix/sh exhibits slightly different semantic behavior from /bin/sh (the interpreter system(3S) invoked before 9.0) in the following areas:

- Trap handling
- Signal handling
- Set-user-id scripts
- Asynchronous job handling
- Different wording/spacing on some error messages
- The /bin/posix/sh command ignores the value of the environment variable IFS when executed.
- In a chrooted environment, /bin/posix/sh must be present.

When system(3S) is passed a null pointer in place of a command string, it returns 1 if /bin/posix/sh is available, otherwise it returns 0f. Previously, system(3S) always returned 1 in this case.

telldir(3C) and seekdir(3C)

Summary of Change

The encoding of the current directory location that telldir() returns was changed.

Impact

Applications that depend on the old and undocumented encoding scheme no longer work properly. Such applications should be modified so that they do not

depend upon the encoding of values that `telldir()` returns. The encoding scheme was used, but never documented.

wordexp(3C)

Summary of Change

The syntax is:

```
int wordexp(const char *words, wordexp_t *pwordexp, int flags);
```

This new function performs word expansions and places the list of expanded words into the structure pointed to by **pwordexp**. The expansion would be the same as would be performed by the POSIX shell, if the words argument is the part of a command line representing the arguments to a utility.

wordfree(3C)

Summary of Change

The syntax is:

```
void wordfree(wordexp_t *pwordexp);
```

This new function frees any memory associated with **pwordexp** from a previous call to `wordexp()`.

limits.h and definitions of FLT_MIN, DBL_MIN

Summary of Change

The constants FLT_MIN and DBL_MIN in `limits.h` are now `#defined` only when the _XPG3 feature test macro is explicitly defined. The reason for this change is that starting with XPG4, FLT_MIN and DBL_MIN are defined in `float.h`.

Impact

In general, include `limitsf.h` and `float.h`. Then, regardless of which X/Open mode you compile in, you get definitions for FLT_MIN and DBL_MIN.

To enforce XPG4 mode, define _XOPEN_SOURCE and _XPG4, and include float.h.

To enforce XPG3 mode, define _XOPEN_SOURCE and _XPG3, and include limits.h.

Refer to stdsyms(5) in the HP-UX Reference for information on the _XOPEN_SOURCE, _XPG3, and _XPG4 feature test macros.

Communications

Communications is provided in HP-UX via several products. One or more products might be bundled with your system, and products can be purchased separately. To get more information, see products, such as the following ones, that appear in Chapter 4 of these notes.

```
ARPA/9000
BIND (namesaver)
LAN/9000
NCSNCK-RUN
NetLS
```

Documentation

New Documents

DTC Device File Access Utilities (DDFA), Part Number B1014-90012

This manual is part of the System Administration Documentation Set. It contains information on using the DTC Device File Access Utilities and the Telnet Port Identification features of ARPA/9000. Refer to HP ARPA SERVICES later in these release notes.

Online Help

If you use HP VUE or the X Window System, you can use the HP Help System. HP Help provides a wealth of help information that you can browse, search, display, and print. This information displays in a window called the Help Manager.

To start the Help Manager, click on the books and question mark icon on the front panel.

Help Manager includes the following topics (it might also include additional help, depending on what optional products you have purchased):

- HP Visual User Environment Help, Version 3.0—provides information on HP VUE. Refer to this help to learn about HP VUE and its features.

- HP-UX 9.0 Operating System Help—provides information on using the HP-UX operating system. Refer to this help to perform common HP-UX tasks, determine the cause of HP-UX errors, and locate information about HP-UX manuals.

- (If you have the Instant Ignition, pre-loaded, operating system) Welcome Messages—provides an introduction to your new workstation and the HP VUE interface. It also explains how to create user accounts and change passwords.

You can also get context-sensitive help by doing either of the following:

1. Move the cursor to the object or icon for which you want help, then press (F1), the help key.; or

2. Choose the up arrow above the Help control (the books and question mark) in the Front Panel. The Help subpanel appears.

        Choose {On Item} in the Help subpanel.

   The cursor changes to a question mark (?). Choose the object or icon for which you want help.

For more information about online help, refer to Chapter 3 of Using HP-UX.

Documentation Errata

acctcom(1M) Command

There is an error in the `acctcom(1M)` entry in the HP-UX Reference. In the
`acctcom(1M)` entry, the line:

```
The command name is preceded by a # if it was executed with super-user
privileges.
```

is incorrect. It should read:

```
The command name is preceded by a # if super-user privileges were
required to execute the command.
```

For example, a user logged in as root executes `date` to check the time. Since
this command does not require super-user privilege, it is shown by `acctcom`
without the # character on the line. On the other hand, executing `date`
0731180092 to set the time does require super-user privileges. So, it is shown
by `acctcom` with the # character on the line.

In the bugs section of the entry in the HP-UX Reference, the following
information should be added:

```
The accounting flag is not cleared when one process execs another, but
only when one process forks another.  One side-effect of this is that
some processes will be marked with #, when users do not expect this.
```

For example, the `login` program uses super-user privileges to assume the
identity of the user who is logging-in, setting the ASU bit (which ultimately
causes the # character in the `acctcom` output). After assuming the user's
identity, it execs the user's shell. Since the `exec` does not clear the ASU flag,
the shell inherits it and is marked with a # in the `acctcom` output.

Files and File Systems

/etc/group

Membership in each group is limited to 200 users. A workaround is possible if a system administrator creates multiple entries for a group. SAM supports this workaround if the entries are adjacent and if the first three fields that define the group are identical.

syncer(1M) and rtprio(1)

By default, the **/etc/rc** file is shipped so that HP-UX starts up the command named **syncer** with the **rtprio** command. This is no longer necessary on Series 300/400/700 because it magnifies the intrusion into the system when **syncer** sync's the disks. Optionally, you can delete the real-time priority part of the line in **/etc/rc**, changing it from:

        /usr/bin/rtprio 127 /etc/syncer

to just:

        /etc/syncer

Installing HP-UX

Series 300/400 systems can use a netdist server as a source of media for an installation. The `Installing and Updating HP-UX, Release 9.0` document has information about using a netdist server as a source of media.

Internationalization (Localization)

Localized Product Information for Series 300/400

The following localized products are provided for S300/400 HP-UX 9.0 release:

- Japanese HP-UX Runtime (Product number B2386A)
- Korean HP-UX Runtime (B2388A)
- Traditional Chinese HP-UX Runtime (B2389A)
- Simplified Chinese HP-UX Runtime (B2390A)
- German HP-UX Runtime (B2391A)
- French HP-UX Runtime (B2392A)

Each localized product consists of the HP-UX Runtime System and the Localized User Interface for HP-UX commands and HP VUE. NLIO is also included in the Asian products. In addition, the Japanese product has selected HP-UX Reference entries in Section 1 (about 50 files).

The Japanese localized user interface, including the localized reference entries, requires about 1.1 MBs of disk space. Other languages require about 0.7 MBs per language.

Using the Bourne Shell as the Default

In 9.0, the PATH variable is specified so the system finds the Posix Shell before it finds the Bourne Shell. If you want the Bourne Shell, you should move /bin/sh ahead of /bin/posix/sh in the PATH specification in your /etc/.profile and /etc/csh.login files.

Regardless of the specification of PATH, programmatic access to the shell through popen(3C) and system(3C) use the internationalized Bourne Shell when an Asian multi-byte language is used.

Kernel

Information related to the kernel appears throughout the release notes. See the following topics to get kernel-related information (for example, information about parameters).

- Core Commands
- Compatibility
- Memory
- Dynamic Buffer Cache
- Memory Mapped Files
- HP-UX Math Libraries
- Shared Libraries

Languages

With the 9.0 release, many languages are available as separate products. As such, the products have their own release notes. The release notes for the products are placed on the system during the update process. They appear, after the update, in the following directory:

        /etc/newconfig/90RelNotes

For example, this situation applies to the C, Pascal, and FORTRAN languages.

Some language products include a symbolic debugger, which is placed on the system during the update process. For more information, see the topic named Symbolic Debugger.

Linker

New Features

The following items describe the linker options added to HP-UX 9.0 since
the 8.0 release. To get more information, see the ld(1) entry in the HP-UX
Reference and the 9.0 Programming on HP-UX manual (P/N B2355-90026).

+s   indicate that the SHLIB_PATH environment variable
                    may be used by the shared library loader to locate shared
                    libraries at run-time.

+b pathlist   specify a list of directories that are to be searched at
                    run-time by the shared library loader to locate the shared
                    libraries needed by the program.

+e symbol   specify symbol to be exported from a shared library.

-h symbol   specify symbol not to be exported from a shared library.

+I symbol   specify an entry point (initializer) for a shared library to be
                    called when the library is loaded or unloaded.

-c file   use linker options from file.

-B nonfatal   indicate that unresolved symbols in shared libraries are not
                    to be considered fatal.

LP Spooler

Removed Features

The /etc/mklp script, which configures printers and the LP Spooler, and its
associated HP-UX Reference page was removed. HP recommends that people
use SAM to configure printers and the Spooler.

Memory

Dynamic Buffer Cache

The size of file system buffer cache can now grow or shrink as system needs dictate. In previous releases, the size of the buffer cache was determined at boot time. The configuration parameters NBUF and BUFPAGES set the size of the cache. If the size was not specified, the default size was fixed at 10 percent of physical memory.

The size of the buffer cache can still be fixed using the configuration parameters. If the size is not configured, however, the operating system implements a dynamic cache which can grow or shrink as needed to optimize system performance. In most applications the dynamic cache is desired, since it allows memory to be used more effectively.

Newly installed systems default to a dynamic file system buffer cache. Systems being updated also default to the dynamic cache. In this instance, the configuration parameters are removed (commented out) from /etc/conf/dfile.

Memory Mapped Files

A new feature of the 9.0 release for Series 300, 400 and 700 systems is the support for memory mapped files.

Memory mapped files provide a mechanism that allows a process to access files by **mapping** the file data into the process address space. Once a file has been mapped, the process directly accesses the file contents just as if it were memory. There are four basic primitives used to manipulate the mappings for memory mapped files:

mmap()	Map a file or portion of a file into a process' address space
mprotect()	Modify the protections of pages of a mapped file
munmap()	Unmap a file or portion of a file from a process' address space
msync()	Synchronize the contents of a mapped file with the file system

When a process maps a file into memory, it specifies one of two mapping modes:

+ `shared (MAP_SHARED)`

+ `private (MAP_PRIVATE)`

When a file is mapped or shared, modifications to the file are visible to other processes that also map the file. In addition, the modifications are retained (written back to the disk) when the process exits, unmaps the file, or uses `msync()` to synchronize the mapping with the file system. The current implementation of memory mapped files does not provide automatic coherency between file access using the `mmap()` interface and file access using `read()` and `write()`. The `msync()` system call may be used to synchronize the file contents between the two interfaces when coherency is necessary.

When a file is mapped private, modifications to the file are only visible to the process that created it. The modifications are only temporary; when the process exits or unmaps the files all changes are lost (i.e. they are not written back to the underlying file). When a file is mapped private, swap space resources are reserved to temporarily store modifications if the system must page portions of the file out of memory.

In addition to mapping files, the `mmap()` interface allows a process to create an anonymous (MAP_ANONYMOUS) mapping. An anonymous mapping creates a region of memory that is initialized to zero. Swap space resources are reserved to temporarily store modifications made to an anonymous region if the system must page portions of it out of memory.

An anonymous mapping may be created shared or private. Modifications made to a shared anonymous mapping are visible to a process and all of its descendents (child processes). Modifications made to a private anonymous mapping are only visible to the process that created the mapping.

The memory mapped files feature also provides a mechanism to modify the protections on pages within a region. Using the `mprotect()` system call, one or more pages may be marked read-only, read-write, read-write-executable, read-executable or non-accessible. If a process attempts to write a read-only page or to read a non-accessible page, it is sent a SIGBUS signal. The `mprotect()` system call may only be used on pages of memory mapped by the `mmap()` interface.

The madvise() system call permits a process to advise the system about
its future behavior in referencing a mapped file. The process may indicate
that it is performing mostly random references, sequential references or that
the system should treat the region normally. In addition, the process can
indicate that it will or will not need certain pages in the near future. The
current implementation of memory mapped files on HP-UX does not use the
information provided by madvise(), it is supplied for compatibility with other
vendor's implementations. Future versions of HP-UX may use the information
provided by madvise().

The final feature of memory mapped files for this release is a set of semaphore
functions for use with memory created using the mmap() interface. These
functions are part of the OSF Application Environment Specification. The
functions provided are:

msem_init()   initialize a semaphore in a mapped file

msem_lock()   lock a semaphore

msem_remove()   remove a semaphore

msem_unlock()   unlock a semaphore

For more information about memory mapped files, refer to the following entries
in the HP-UX Reference:

mmap(2), mprotect(2), msem_init(2), madvise(2) and mman(5).

Networking

The netmemmax Tunable Parameter

There is a new tunable parameter, netmemmax. This parameter defines the
size of the dynamically malloc'able memory for IP level reassembly of packet
fragments. Like other tunable parameters, netmemmax can be modified in the
gen file. You should modify it if the performance of the system degrades to a
point where it appears to hang.

Summary of Change

After stressful networking activity, mostly involving UDP applications, the system response may degrade to a point where it appears to hang. This can be the result of memory having been fragmented into small chunks that are made available to user-space applications by the garbage collection and compaction processes.

If this happens, you can tune HP-UX by modifying the value of `netmemmax`.

Impact

At the 9.0 release, the default value for `netmemmax` is 10% of the dynamic memory. This value should be sufficient for most user applications to work under normal conditions and network configurations. Previously, there was no limitation on memory usage.

Configuration

When netmemmax == 0 =>   The kernel may use at most 10% of dynamically malloc'able memory for IP level reassembly of packet fragmentation. (9.0 release default value)

When netmemmax == -1 =>   The kernel places no restriction on dynamically malloc'able memory for IP level reassembly of packet fragmentation. (8.0 release default value)

When netmemmax == X =>   The kernel may use at most X bytes of malloc'able memory for IP level reassembly of packet fragmentation.

Partitions (Includes Fileset|)

A release of HP-UX is distributed on media such as a CD-ROM or DDS cartridge tape. In some cases, you might be able to obtain media via a networked server. In either case, the media contains Partitions, which are collections of filesets. Filesets are collections of files that relate to a major functionality. You can get a complete list of available Partitions, and their Filesets, in the `Installing and Updating HP-UX` document.

New SLIP-RUN fileset

Summary of Change

The NETINET fileset does not include Serial Line IP (SLIP) feature. Instead, there is a new fileset, SLIP-RUN, that provides this feature.

Impact

To use the SLIP functionality, update your system by installing the SLIP-RUN fileset and, if necessary, the NETINET fileset.

XNS Fileset

The XNS fileset was removed from the 9.0 Release.

NETTRACELOG Fileset

Features

The NETTRACELOG fileset, consisting mainly of the `nettl` and `netfmt` commands, is essentially the same as for earlier releases. All options and behaviors have been preserved.

Summary of Change

The most significant change to the HP-UX 9.0 version of the formatter program, `netfmt`, is the use of shared libraries. Subsystems now provide subformatters to decode their tracing and logging information in the form of a shared library. This allows the `nettl` and `netfmt` commands to dynamically

include subsystems from any products installed on the system. There is no need to relink `netfmt` and `nettl` each time a new product is installed or removed, thus the commands are no longer recreated on the system. As a result the `nettlgen` command was obsoleted.

The NETTRACELOG configuration file, `/etc/conf/nettlgen.conf`, was revised. There is a new field, subsystem type, that determines whether the subsystem is from the kernel or user space. This information is necessary to support the various kernel space products that use tracing and logging.

The fields for the NS subsystems and the master formatter have been changed to use defaults. These changes are transparent to the user since the NETTRACELOG customize script converts the `nettlgen.conf` file from the old format to the new.

Another change to the `nettlgen.conf` file is the addition of the console option file that controls console logging. This gives a system administrator more control over the logging information, if any, that gets sent to the console. The console option file behaves just like the filter files that users should already be familiar with. A default file is created whenever no console option file is specified by the user.

Although you should never have to look into the configuration file, it is possible that you might. The file is self documenting, with each field's use and permissible values described at the beginning.

These new changes are further described in the entries in the HP-UX Reference for the product.

Impact

The user should not see any difference in the behavior of `nettl` or `netfmt` with the use of shared libraries. Only product developers need to be concerned with the shared library requirement of `nettl` and `netfmt`. Converting from archived to shared libraries is a matter of changing compile and linking options. No code changes should be necessary.

During the migration to this version of `netfmt`, it is possible that not all subsystems are updated at the same time. For example some filesets that are not bundled with the HP-UX 9.0 release may be updated at some time after the update to HP-UX 9.0 has been performed. This leads to a situation where the NETTRACELOG configuration file, `/etc/conf/nettlgen.conf`, contains

outdated information. In particular, references to archived libraries are left in the file.

The new version of the `netfmt` command are not able to use the NETTRACELOG configuration file if the command contains references to archived libraries. The `netfmt` command can link only with shared libraries. An attempt to load an archived library results in an error. The error can appear on the system console during a system boot or whenever `nettl` starts.

When the NETTRACELOG configuration file contains references to archived libraries the offending subsystem configuration line must be removed. This can be done using a text editor on the file `/etc/conf/nettlgen.conf`. The entire line can be deleted or commented out using the `#` character. It is assumed that the outdated subsystem automatically reconfigures itself when the proper version is installed on the system.

In general, this situation can only occur when filesets that are not bundled with the current release of the operating system are not updated.

The syntax change to `nettlgen.conf` should have no impact to the user because the `nettlconf` and customize scripts should do necessary conversions.

Performance

Using shared libraries may be slightly slower than the old method since the libraries have to be dynamically loaded each time a command is used. However, since `nettlgen` was obsoleted there is time saved in forgetting to re-run `nettlgen` and making other similar mistakes. The `nettlgen` command itself was extremely slow.

Alternatives and Compatibility

All trace/log files taken on older systems (8.0 and earlier) should be forward compatible and readable by `netfmt`. All trace/log files taken on 9.0 systems should be formatable on older versions of `netfmt` provided the individual product subsystem subformatters supports this.

Obsolescence

The `/etc/conf/nettlgen` command was replaced by the use of shared libraries. The `nettlgen` command took a long time to execute and was prone to errors.

Subsystem formatters in the form of archived libraries are no longer usable by the netfmt command. All subsystem formatters must be provided in the form of shared libraries so that they may be dynamically loaded by the netfmt command as needed.

XNS Fileset

The XNS fileset was removed from the 9.0 Release.

Shared Libraries

New Features

The following routines and options were added in the 9.0 release. For more information, see the shl_load(3X) entry in the HP-UX Reference and the 9.0 Programming on HP-UX manual.

New Options

SHLIB_PATH	An environment variable that can be used to specify directories to search at run-time to locate shared libraries needed by a program.
shl_load()	A new flags BIND_FIRST, BIND_NONFATAL, BIND_NOSTART, BIND_VERBOSE, DYNAMIC_PATH.

New Routines

shl_gethandle()	Gets information about the given handle.
shl_load()	New options BIND_FIRST, BIND_NONFATAL, BIND_NOSTART, BIND_VERBOSE.

Shared Library Debugging

The xdb(1) debugger now provides greater support of shared library debugging. Refer to HP-UX Symbolic Debugger User's Guide (P/N B2355-90044).

C Math Functions and Shared Libraries

For performance reasons, the PA-RISC 1.1 versions of the libm and libM math libraries are provided as archive libraries only. This can affect an application if C math functions are used in a shared library that is shl_loaded by another shared library (the parent library), and if the math functions are not used in the parent library. In this case, an application linked on HP-UX 9.0 with the parent library fails at runtime due to unresolved external references to the math functions if and when the other library is shl_loaded.

For more information about HP-UX math libraries at 9.0, see the Math Libraries topic in this file and the HP-UX Floating-Point Guide (P/N B2355-90024).

Standards

For the 9.0 release, HP-UX conforms to the following standards:

- OSF AES OS Component, Revision A (s300/s400/s700 only)
- X/Open Portability Guide Issue III, Base Branding, including the optional internationalization features.
- FIPS 151-1 and FIPS 151-2
- POSIX.1:1990 (IEEE Standard 1003.1:1990) (also known as ISO/IEC 9945-1: 1990)
- POSIX.2:1992 (IEEE Standard 1003.2:1992) (also known as ISO/IEC CD 9945-2.2)
- All options except the User Portability Utilities Option
- ANSI C (ANS X3.159-1989) (also known as ISO/IEC 9899: 1990)

- System V Interface Definition (SVID2)
  □ Base System
  □ All Extensions except: AS, NS (all platforms), and shmat(KE_OS) (s700/s800 only)
- OSF/Motif 1.2.1
- X Window System Version 11, Release 4
- X Window System Version 11, Release 5

Symbolic Debugger

With the 9.0 release, many languages are available as separate products. The symbolic debugger is included with some of these language products (for example, Pascal, FORTRAN, and C). These products have their own release notes including separate release notes for the symbolic debugger.

Release notes for the products are placed on the system during the update process. They appear, after the update, in the following directory:

```
/etc/newconfig/90RelNotes
```

System Administration Manager (SAM)

Features

The following items indicate the new or changed SAM functionality:

- OSF/Motif Graphical User Interface.
- Alpha-Terminal interface matching the graphical version.
- Object-action user interaction paradigm.
- User customizable object lists with user selectable sorting and filtering.
- Logging of SAM actions.

■ New hardware supported:

□ Models 890, 887 and 897

□ Magneto-Optical disks

□ Combination serial/LAN cards

□ Network printers

□ Compressed DDS and QIC tape devices

■ Modified SAM system areas:

□ Peripheral Devices

Improved configuration of serial and parallel printers. TCP/IP and NPX network printers. Cancel print jobs. Save and restore lp configuration. Remove terminal and modem configurations.

□ Disks and File Systems

Replaces file system management and disk configuration. Magneto-optical autochangers.

□ Network Configuration

Enhanced and reorganized for improved usability. Networking services status screen. Support for some NFS 4.1 features. X.25 configuration appears only when X.25 is installed.

□ Backup and Recovery

Makes remote backup device support more visible. Lists all backup devices on the system.

□ Kernel Configuration

Access to modification of all kernel parameters. Non-HP drivers are not removed by SAM during re-generation.

■ New SAM system areas:

□ Remote System Administration

Easy execution of SAM on another system.

□ Routine Tasks

Log file trimming. Large and core file removal. System shutdown.

- ☐ Process Management

  Process kill and renice. Performance monitoring. Job scheduling via cron.

- ■ Removed SAM system area:

  - ☐ Cluster configuration (not supported in 9.00 for the series 800). However, it remains available for series 300/400/700 systems.

Summary of Change

SAM now has a Motif Graphical User Interface (GUI). This change is also apparent in the character-based terminal user interface, which is nearly identical to the new GUI. Upon first entering SAM, users of previous versions are encouraged to explore the Help screens to understand the new user interface and character-based terminal keyboard traversal.

The user interaction model for SAM has changed. In past versions, the user traversed SAM menus knowing what action was desired. At the end of the traversal the user selected the object (or device) to perform the action on. Often the selection of the object required the user to use the help system to present the list of objects (such as disk drives) that SAM could find. With 9.0 SAM, the user selects an area of interest, such as user management, and they are presented with the list of valid objects, i.e., the list of current users. In this way the user sees the current system state up front. The user can sort or filter the list of objects as desired and perform actions on the objects. On selecting an action to perform, users are led to task oriented dialogs that guide them through the task.

The following items note the functional improvements and additions to SAM:

- ■ SAM now logs its actions to a file. The level of this logging is user selectable and the maximum size of the log file is user customizable.

- ■ SAM now performs more rigorous tests when a printer is added.

- ■ Network printers using the TCP or IP and NPX interfaces can now be configured into the spooling system using SAM.

- ■ Print jobs in the LP Spooler can be canceled.

- When a user successfully adds or removes a printer the new state of the printer spooler system is automatically saved. Later, if problems occur because the printer spooler state becomes corrupt, the user can restore the spooler to the last saved state. Users are also able to explicitly save the spooler's state at any time. The state is also automatically saved when updating an 8.0 system to 9.0.

- Terminals or modems currently configured can now be removed.

- The File System management area of SAM was replaced with a new area called Disks and File Systems. This new area incorporates disk configuration and File System Management.

- Magneto-optical disk devices, including optical disk autochangers can now be configured using the Disks and File Systems area of SAM. Also, these optical devices are valid backup devices in the Backup and Recovery area.

- The networking areas (NNC and X.25) have been reorganized. The new sub-areas are: Status, Network Interface Configuration, Remote System Connectivity, Device Connectivity, Network File System (NFS), and Security. The Status sub-area provides information regarding services installed in the system (i.e. Anonymous FTP, Sendmail, NFS, Bootp, NS-NFT, PAD Services and Network Computing System). The Local Location Broker of the NCS subsystem and the supported features of NFS 4.1 can now be enabled/disabled in this sub-area. The Network Interface Configuration section sub-area provides access to IEEE 802.3, Ethernet, Token Ring, FDDI, and X.25 network link information. Remote System Connectivity provides access to Internet Connectivity, Emulation (X.25), UUCP (X.25), and X.3 profile (X.25) configuration screens. PAD Server, PAD Printers, and X.3 Profiles can be configured in the Device Connectivity sub-area. The Network File System sub-area allows configuration of remote and local file systems. Individual directories, instead of whole file systems, can now be exported. The NFS anonymous user ID and permissions for file access can now be set.

  Note: The portions of SAM regarding X.25 are not present until the X.25 software product is installed on the users system.

- The Kernel Configuration area of SAM does allow the user to see kernel parameters, loaded drivers, and sub-systems. Users can modify all supported parameters (including the use of expressions) that are validated by SAM. Parameters unknown to SAM can be modified at the users own risk. Any

unknown drivers found by SAM are put back in the kernel when it is re-generated.

- The new Remote System Administration area of SAM allows the users to specify remote systems they want to administer. SAM then sets up the connection to these systems so that the user may easily select them at a later date.

- The new Routine Tasks area of SAM contains significant functionality to help the user with the ongoing maintenance of the system. Log file trimming allows the user to easily identify and trim ASCII and binary system log files. The user may also add application log files to the list. The large and core file removal sub-areas allow the user to easily find and remove unwanted core files, or files greater than a user specified size. Finally, system shutdown provides a simple interface to changing the user state or shutting down the system.

- The new process management area of SAM allows the user to easily view running processes and then renice or kill them if desired. Users can also schedule jobs via cron. Finally, users can start one of the following performance monitors:

        /usr/contrib/bin/monitor, /usr/bin/top, /usr/bin/sar

    The latter is Series 800 only, and only if it already exists on the system.

- Users now select a backup device from a list of valid backup devices on the system. The user is no longer required to specify a device file for a local backup device. SAM also supports backing up to and recovering from a remote backup device.

Cluster Configuration area of SAM is not present in the 9.0 for Series 800 computers because this functionality is not supported. However, it remains available for series 300/400/700 systems.

Impact

All users of SAM are impacted by these changes to SAM. There are two areas of potential user customization of SAM in previous releases: Task Customization and menu additions in Other Utilities.

Customized tasks that the user has placed in the User or Cluster configuration areas of SAM must be re-added when SAM is updated to 9.0.

Menu modifications the user has made in the other Utilities area for 8.0X SAM versions are not compatible with 9.0 SAM. When an 8.0X system is updated to 9.0, these menu customizations are automatically updated to the 9.0 format for the user.

Performance

The interactive performance of 9.0 SAM is generally similar to 8.0X versions. Entry into some of the SAM areas may take longer in the 9.0 version because more information is being presented to the user, and it can take significant amounts of time to collect this information. When significant amounts of time are required for any SAM operation, progress messages are presented to the user.

Alternatives/compatibility

If a user is dissatisfied with the new SAM, all the administration capabilities of SAM can be accomplished manually using techniques described in the HP-UX documentation.

Some Limitations in SAM

The following limitations might affect your use of SAM.

- The SAM kernel configuration area does not support reading in a new set of initial values from an existing kernel or configuration file (S800 or dfile). This functionality may be documented in the System Administration Tasks manual, but SAM does not support it.

- The SAM kernel configuration area does not support adding, removing, or modifying dump devices on Series 800 computers. Also, SAM does not support modifying swap devices on any Series. Both of these tasks are documented in the System Administration TAsks manual, but SAM does not support them. SAM still supports modifying the console on Series 800 computers.

- The SAM kernel configuration area does not support modifying cdfs or nfs in a clustered environment as documented in the Managing Clusters manual.

Instead of having SAM automatically generate all cnode kernels to match cdfs and nfs changes made to the server kernel, SAM prompts the user to make the same change on each of the cnodes prior to rebooting the server.

- SAM does not allow a user to regen the kernel and move it into place without rebooting the system.

Updating HP-UX

Added a -F option to /etc/update to force a noninteractive update to proceed despite errors that would be warnings if interactive. The update(1M) entry in the HP-UX Reference has information.

X11 Window System (See X Graphical User Environment)

X11 Windows Dev. Toolkit (See X Graphical User Environment)

X Graphical User Environment

HP-UX 9.0 contains X11R5, Motif 1.2 and VUE 3.0.

X11 R5 Server

X11 Server

The X11 server for the 9.0 release conforms to the X11 R5 release of the X Window System from the X Consortium.

The X11 R5 server is only supported on HP 9000 Series 300, 400, and 700 computer systems. The X11 R5 server is not supported on HP9000 Series 800 systems. The Series 800 HP-UX 8.0 X11 R4 server is not supported on HP-UX 9.0. Therefore, HP 9000 Series 800 users who require an X11 server cannot update to HP-UX 9.0.

X11 Server: Compatibility with previous releases

The X11 R5 server is fully backwards compatible at the X protocol level with the R4 servers in previous releases. Most existing X11 clients (R4 or earlier) that operate properly with the HP-UX 8.0 or 8.0* X11 R4 servers should operate properly with the HP-UX 9.0 X11 R5 server. A few old clients, however, may rely on defects in the R4 (or earlier) servers that have been fixed in the R5 implementation. See the Bug Compatibility Mode section below for a potential workaround.

X11 Server: New graphics device support

The HP-UX 9.0 X11 R5 server has support for the new A2091A CRX-48Z graphics device.

X11 Server: Bug compatibility mode

Like previous X11 servers, the X11 R5 server supports a bug compatibility mode of operation. When bug compatibility is enabled, certain innocuous defects that were in previous releases of the X11 server are re-enabled to permit old X clients that depend on the defects to operate with the R5 server. The default state for the HP-UX 9.0 R5 server is for bug compatibility mode to be disabled.

Bug compatibility mode can be enabled or disabled by the user in two ways. If the X Server is invoked with the command line option bc, bug compatibility mode is enabled. Alternatively, the xset client can be used to toggle the state of bug compatibility mode:

```
xset [-display host:dpy] -bc    (disables bug compatibility mode)

xset [-display host:dpy] bc     (enables bug compatibility mode)
```

X11 Server: Serial input device support

The X11 R5 server has the new capability to support serial input devices
(keyboard or pointing devices) through the dynamic loading of device drivers
for these devices at run-time. For more information on this facility, refer to the
file:

> /usr/contrib/X11drivers/input/serialdrv.doc

X11 Server: Keyboard keymaps

The HP-UX 9.0 X11 R5 server keymap file (XHPKeymaps) now includes
keymaps for many new localized Enhanced Vectra HIL keyboards.

X11 Server: New rgb.txt color name database files

Two new rgb.txt color name-to-RGB value database files are included with
the HP-UX 9.0 X11 R5 release. These files contain several new R5 color names.
Also, the RGB values in the files have been customized for particular HP
display model product series.

The files can be found in:

> /etc/newconfig/X11R5/rgb.txt.98xxx

and

> /etc/newconfig/X11R5/rgb.txt.A10xx

The rgt.txt.98xxx file was customized for color appearance on HP displays
with product numbers beginning with 98.

The rgb.txt.A10xx file was customized for color appearance on HP displays
with product numbers beginning with A10.

XhplibX11   XhpWindowToFile(), XhpFileToWindow(),
        XhpPixmaptoFile(), XhpFiletoPixmap(),
        XhpQueryImageFile() Library Functions

Support for the following HP-proprietary functions is being phased out
beginning with the HP-UX 9.0 X11 release.

> XHPWindowToFile()
> XHPPixmaptoFile()
> XHPFileToWindow()

```
XHPFiletoPixmap()
XHPQueryImageFile()
```

Documentation for these functions is being removed from the HP-UX 9.0 learning product materials.

X11 R5 Xlib (libX11): Device Independent Color Support

The new X Color Management System (`Xcms`) functions in R5 Xlib (`libX11`) support device-independent color spaces derivable from the CIE XYZ color space. This includes the CIE XYZ, xyY, L*u*v*, and L*a*b* color spaces as well as the TekHVC color space. In addition, linear RGB intensity value space was added as well as gamma correction for device RGB values. A uniform syntax was adopted for specifying colors in strings.

`Xlib` now supports client-side color name databases, and the existing functions that use color names (for example, `XLookupColor` and `XAllocNamedColor`) now handle all color spaces so that the contrivance of using `XParseColor` followed by `XAllocColor` is no longer necessary. `Xlib` now provides direct programming interfaces for dealing with color values in different spaces, and for converting between spaces. Also, new device-independent color spaces can be added.

The device-independent color model greatly expands the scope of visual attributes and specifications available to Xlib programs while maintaining compatibility with all existing Xlib programs.

Two sample device color characterization (`.dcc`) files are included with the HP-UX 9.0 X11 R5 release. These files can be used with the new `xcmsdb` utility, which provides a mechanism for placing screen color characterization data into root window properties. The files reside in:

```
/etc/newconfig/X11R5/98xxx.dcc
```

and

```
/etc/newconfig/X11R5/A10xx.dcc
```

The `98xxx.dcc` file was customized for color appearance on HP displays with product numbers beginning with `98`.

The `A10xx.dcc` file was customized for color appearance on HP displays with product numbers beginning with `A10`.

X11 R5 Font Server

Features

The font server is new for the X11 R5 release and HP-UX 9.0. It is in a new fileset called X11-FONTSRV. The font server provides networked font services to X11 display servers (either on workstations or X terminals) and X11 clients. Network font services include access to bitmap fonts, generation of bitmap fonts from scalable fonts, and management of font licensing. The HP-UX 9.0 font server includes scalable font rasterizers for Intellifont (Agfa/Compugraphic) and Type 1 (Adobe) scalable fonts, as well as a limited bitmap font scaling capability. Several sample Intellifont and Type 1 scalable outlines are included with the font server in the HP-UX 9.0 release.

Running the Font Server

The font server process is not automatically started on your system. Whether or not you need to start the font server depends on your font requirements and your network configuration (e.g., Is there another font server already available on the network for your X server to access?).

The font server may be started on a system with the command:

```
/usr/bin/X11/fs -daemon
```

If the font server is to be run on the same machine as the X server and if the font server is started before the X server, the font server does not need to be added to the font path of the X server. However, if the font server is started after the X server is already running, or if a font server is running on a different machine than the X server, then the font server must be added to the font path of the X server as follows:

```
xset +fp tcp/:7000
```

This adds the font server port to the front of the font path list searched by the X server. If you are adding a font server that is not running on the same machine as the X server, the xset command is extended to also specify the font server host:

```
xset +fp tcp/hostname:7000
```

Once the font server is running and has been added to the font path of an X
server, the server must rehash its fonts. This is also done with the xset client
via:

```
xset fp rehash
```

Summary of Change

With HP-UX 8.0, Intellifont scalable outlines were delivered in the
X11-FONTA fileset. With HP-UX 9.0, all scalable font outlines (Intellifont and
Type 1) are delivered in the new X11-FONTSRV fileset. The X11-FONTA
fileset still contains HP Roman8 bitmap fonts.

The X11 R5 release also has a new font format. X11 bitmap fonts are now
delivered in pcf format instead of snf. A new utility, bdftopcf, replaces the
old bdftosnf utility for converting fonts from standard bdf format to pcf
format. All fonts are shipped in compressed form, and font file names end with
a .pcf.Z suffix).

All existing TypeDirector/UX utilities are still supported on HP-UX 9.0.

Alternatives and Compatibility

The X11 R5 font server accommodates bitmap fonts in the new pcf format, the
old snf format, and standard bdf interchange format. Convert fonts from bdf
to pcf format with the included utility, bdftopcf.

All existing TypeDirector/UX utilities are still supported on HP-UX 9.0.

X11 R5 Clients

Features

This release updates the X client's executables to MIT's X11 R5 standard.
Since MIT is downgrading their support level for some clients, and has added
other new ones, there are some changes to the population of clients that we are
offering in this release.

Overall, the clients offered contain the general new features that come with
X11 R5 and are mentioned elsewhere in this report, namely:

- Application resources can now be set on a per-screen basis and resource files can now contain ? wildcard characters and can include references to other resource files.

- Internationalization of the clients uses the new comprehensive R5 standard for internationalizing X clients. Clients that were internationalized with 8.* are internationalized at 9.0, but those that were not, such as xterm, are not internationalized in 9.0.

- Perception-based device-independent color management as implemented in the R5 libraries is supported, which involves a new client that handles the color management database.

- Scalable fonts and font server protocol, to the degree that MIT clients take advantage of this.

- A new user authorization scheme was implemented for R5 as a step toward secure X. HP provides those clients that are necessary to support this scheme.

Hpterm now supports the use of up to 24 softkeys; see the descriptions of the softkeyInitializeAll and softkeyLast entries in the HP-UX Reference. The dc1Handshake resource enables the user to select how DC1 handshaking is controlled. The roman8 resource gives the user the ability to control whether hpterm uses roman8 encoding, or ISO-88591 by default. Support for Hebrew, Turkish, and Greek languages were added.

Two different flavors of hpterm are delivered with this release. The one built on X11R4 versions of Xlib, and Xt and Motif 1.1 is delivered by default. The version built on X11R5 versions of Xlib, and Xt and Motif 1.2 is delivered in /usr/bin/X11/hpterm.R5.sh.Z, a compressed shar file. To install this version, uncompress and unshar hpterm.R5.sh.Z. For more information on how to choose between the R4-based hpterm and the R5-based hpterm, see README.hpterm in hpterm.R5.sh.Z

The xterm terminal emulator has new features:

- Cuts of wrapped lines are now treated as a single line.

- Cuts of multi-page regions now work and highlight correctly.

- Pasting of large amounts of data into xterm now works.

- Various new resources have been added.

Summary of Change

The following clients are new to our product release:

xcmsdb   Manages screen color characterization data.

The following clients were supported in HP-UX 8.*, but are not included in 9.0.

xinit   This method of starting the X server was downgraded in
                favor of the **xdm** startup method, which also supports X
                terminals.

x11start   This HP-proprietary startup script is closely associated with
                xinit and is likewise not included in the 9.0 release.

xrefresh   A screen refresher made obsolete by the standard refresh
                functions of the VUE and Motif window managers.

xwininfo   A X window info utility having low support and usage.

xcutsel   Old-style cut selection method that is now obsolete.

bdftosnf   A R4 font conversion program; obsoleted in R5 by **bdftopcf**
                due to a generalization in the format of font files.

xfc   An old R3 font conversion program symbolically linked to
                bdftosnf in R4 for 8.0, which is now itself obsolete.

xfd   X Font displayer; low level of usage.

Impact

The primary impact is dropping of support for the **xinit** or **x11start**
Xstartup method. Instead, use the program **Vuelogin**, which is directly
based on MIT's startup mechanism, **xdm**. **Vuelogin** is not tied to the VUE
environment (and **xdm** is not) and can be used to start up a non-VUE standard
X session without using **xinit**. The main difference between the **xinit** startup
method and the **xdm** or **vuelogin** method is that the former starts the X
server, whereas the latter operates with the X server pre-started automatically.
This is consistent with X terminals. There are some differences in the startup
files associated with these two methods. Switching to the standard MIT
startup method is noticed primarily by those who have not used HP VUE or
xdm under 8.*.

Performance

The X11 R5 release offers performance gains, primarily resulting from the use of the R5 libraries. The increased size of the Xlib library, due to the additional localization features, causes that shared library to grow.

Alternatives/compatibility

Customers impacted by the loss of support for some clients that were supported in HP-UX 8.* should get the X11 R5 contrib distribution from the Interwork User Group.

X11 R5 Xlib

Features

New functionality was added to Xlib: device independent color, and internationalization (i18n).

Summary of Change

The Xcms (X Color Management System) functions in Xlib support device-independent color spaces derivable from the CIE XYZ color space. This includes the CIE XYZ, xyY, L*u*v*, and L*a*b* color spaces as well as the TekHVC color space. In addition, linear RGB intensity value space was added as well as gamma correction for device RGB values. A uniform syntax was adopted for specifying colors in strings.

Xlib now supports client-side color name databases, and the existing Xlib functions that use color names (for example, XLookupColor and XAllocNamedColor) now handle all color spaces so that the contrivance of using XParseColor followed by XAllocColor is no longer necessary. Xlib provides direct programming interfaces for dealing with color values in different spaces, and for converting between spaces.

New device-independent color spaces can be added. Internationalization, an internationalized application, is adaptable to the requirements of different native languages, local customs, and character string encodings. The process of adapting the operation to a particular native language, local custom, or string encoding is called localization. A goal of internationalization is to permit localization without program source modifications or recompilation.

Internationalization

Internationalization in `Xlib` is based on the concept of a `locale`. A locale defines the `localized` behavior of a program at run-time. Locales affect `Xlib` in the following ways:

- Encoding and processing of input method text.
- Encoding of resource files and values.
- Encoding and imaging of text strings.
- Encoding and decoding for inter-client text communication.

Xlib provides support for localized text imaging and text input. Sets of functions are provided for multibyte (`char&*`) text as well as wide character (`wchar_t`) text in the form supported by the host C language environment.

Controlling Keyboard Input Using HP's X Window System

The X Window System uses the concept of `keysyms` to control the mapping of keys into characters. The set of `keysyms` for a particular keyboard is organized into a table called the `keymap`. To get information about keyboard mapping or to set the keyboard mapping use the `xmodmap` command.

Mapping keyboard for both Extend-char and Meta

A common problem reported by people using HP's X Window System is the conflict between the use of the `extend-char` key to access the extended characters of `Roman8` or `Latin1` with HP's keyboards and the use of the `extend-char` key as a Meta key.

The default mapping is that both keys serve both purposes. However, with HP-UX 9.0 it is possible to configure the keyboard so that one key is used as the `extend-char` key and the other as the Meta key.

The `xmodmap` command can be used to inquire and set the mapping for keys on the keyboard. Run the following command.

```
xmodmap -pm
```

For a US or West European keyboard in the default state, this prints:

```
xmodmap:  up to 3 keys per modifier, (keycodes in parentheses):
```

```
shift       Shift_R (0xc),  Shift_L (0xd)
lock        Caps_Lock (0x37)
control     Control_L (0xe)
mod1        Meta_R (0xa),  Meta_L (0xb),  Mode_switch (0x36)
mod2
mod3
mod4
mod5
```

The mod1 modifier has entries for both Meta keysyms and for Mode_switch as well; and this creates a problem. The solution is to use mod2 for Mode_switch and change the Meta_L key into the Mode_switch key. To do this, use xmodmap and execute the following command:

```
xmodmap mods
```

where mods contains the following four lines:

```
remove Mod1 = Meta_L Mode_switch
keysym Mode_switch = NoSymbol
keysym Meta_L = Mode_switch
add Mod2 = Mode_switch
```

The entries in the file need to be in this order. Again, type:

```
xmodmap -pm
```

The results should be:

```
xmodmap:  up to 3 keys per modifier, (keycodes in parentheses):

shift       Shift_R (0xc),  Shift_L (0xd)
lock        Caps_Lock (0x37)
control     Control_L (0xe)
mod1        Meta_R (0xa)
mod2        Mode_switch (0xb)
mod3
mod4
mod5
```

The keyboard then uses the left extend-char key for extended characters and the right extend-char key for Meta. The client must be linked against R4 or R5 Xlib for this to work.

Dead Key Compose processing

HP's X Window System has supported dead key compose processing for HP workstations for some time. This capability is now supported for non-HP servers (workstations and X-terminals) connected to HP systems.

In this form of compose processing a mute (or dead) key is struck followed by a second key. The initial key is a diacritic and the second key is the ASCII character to which the diacritic is to be applied. The diacritic character must be a special muting `keysym` to initiate the dead-key compose processing. The list of `keysym` names and the diacritic character to which they apply follows.

```
keysym name                    diacritic character

hpmute_acute                          ´
hpmute_grave                          `
hpmute_asciicircum                    ^
hpmute_diaeresis                      ¨
hpmute_asciitilde                     ~
acute                                 ´
diaeresis                             ¨
```

To find out which muting diacritics are supported by a keyboard type:

```
xmodmap -pk
```

The entries in the third and fourth column of the keymap are the `extend` and `shift-extend` characters.

Multi-key Compose processing

For release 9.0, HP's X Window System supports a form of compose processing that can be done using only ASCII characters. To use this form of compose processing, set a `keysym` to the `Multi_key` `keysym`. For example the `Enter/Print` key on an ITF keyboard could be used as the `Multi_key`. To do this, execute the following command:

```
xmodmap -e keysym Execute = Multi_key
```

Then, compose processing can be done by typing the Multi_key (`Print`) followed by two other keys. One key should be the ASCII key that corresponds to one of the diacritic symbols and the other key should be the ASCII character to which the diacritic should be applied. The two keys can be typed

in any order. For example, typing `Print ' e` generates a null character. The
table of ASCII characters and the diacritics they are used for follows

```
ASCII character     diacritic character

    '                        ´

    `                        `

    ^                        ^

    :                        ¨

    |                        ¨

    ~                        ~
```

Impact

The device-independent color model greatly expands the scope of visual
attributes and specifications available to `Xlib` programs.

The internationalization support provides developers with a industry-standard
method of creating multi-locale applications.

Performance

Databases should be significantly smaller in memory, and loading and
parsing resources should be faster. Clients that are built archived have an
increase in code size due to the support for device-independent color and
internationalization.

Alternatives and Compatibility

The device-independent color model is compatible with all existing `Xlib`
programs. The internationalization support does not affect existing
applications.

Obsolescence

Fortran bindings are not shipped with HP-UX 9.0. The support for the
bindings is now delivered with the Fortran language.

X11 R5 Input Extension

There is an error in the XInput.h header file. User programs that include XInput.h and compile with the -Aa option get a compilation error. The problem is due to ANSI prototypes which were added to XInput.h for the 9.0 release. Two workarounds exist for the problem.

1. Add the following lines to the application program ahead of the include for XInput.h:

   #define NEED_EVENTS #include <X11/Xproto.h> or,

2. Delete the ANSI prototype for XInputEventToWire from XInput.h. (This routine is called by XSendExtensionEvent. Clients should never call it directly. The ANSI prototype should therefore never be needed and can therefore be removed).

X11 R5 Xt

Features

Most of the major changes made to the Intrinsics relate to improved performance and customization. At the data-structure level, Release 5 retains complete binary compatibility with Release 4.

Summary of Change

The Release 5 Intrinsics have implemented memory-saving measures that include the reorganization of string constants into a single array, a rewrite of the translation manager, and the use of permanently allocated strings where possible.

Base Translations Resource. A new pseudo-resource, XtNbaseTranslations, permits application developers to specify translation tables in application defaults files while still giving end users the ability to augment or override individual event sequences. This change affects only those applications that wish to take advantage of the new functionality, or those widgets that may have previously defined a resource named baseTranslations.

Resource File Search Path. The current specification allows implementations greater flexibility in defining the directory structure used to hold the application class and per-user application defaults files. Previous specifications required the substitution strings to appear in the default path in a certain order, preventing sites from collecting all the files for a specific application together in one directory. The Release 5 specification allows the default path to specify the substitution strings in any order within a single path entry. Users need to pay close attention to the documentation for the specific implementation to know where to find these files and how to specify their own XFILESEARCHPATH and XUSERFILESEARCHPATH values when overriding the system defaults.

Customization Resource. XtResolvePathname supports a new substitution string, %C, for specifying separate application class resource files according to arbitrary user-specified categories. This provides separate monochrome and color application class defaults files. The substitution value is obtained by querying the current resource database for the application resource name, customization, and class, Customization. Any application that previously used this resource name and class must account for the possibly conflicting semantics.

Per-Screen Resource Database. To allow a user to specify separate preferences for each screen of a display, a per-screen resource specification string was added, and multiple resource databases were created; one for each screen. This affects any application that modified the (formerly unique) resource database associated with the display subsequent to the Intrinsics database initialization. Such applications need to be aware of the particular screen on which each shell widget is to be created.

Impact

The R5 Intrinsics should not significantly impact developers.

Performance

Most applications see a decrease in memory usage. Clients that are built archived see an increase in code size. In a few cases, execution time is faster.

Alternatives/compatibility

The R5 Intrinsics are binary and API-compatible with the R4 Intrinsics.

HP Motif 1.2

Features

The OSF/Motif 1.2 release contains new R5-based internationalization capability, drag-and-drop data interchange, tear-off menus, CUA Style Guide compliance changes, general enhancements, and improved performance relative to the Motif 1.1 release. Minor enhancements were made in mwm and UIL.

Summary of Change

Many of the changes to Motif for Revision 1.2 were made to enable the development of internationalized applications. Most dramatic were the changes to the Text and the TextField widgets, including interfaces that support both fixed-width wide characters (wchar_t) and variable-width multibyte characters (char *). The Compound String routines were modified to handle locale-encoded strings and to integrate X11 R5 fontset capability.

X11R5 uses the concept of an input method (IM) that provides a layer of mapping between the keys typed by users and the corresponding characters passed to an application. The Toolkit was enhanced to support the transparent connection to an input method for those languages that require an input method. The VendorShell widget was modified to support the geometry management of an Under-the-Spot input method.

The XmDisplay widget was added to support internationalization. The XmScreen widget was added to support the specifications of resources on a per-screen basis.

Drag and Drop Capability

In compliance with the OSF/Motif Style Guide, Revision 1.2 supports the drag and drop metaphor for data interchange. The Drag and Drop specification was fully implemented. Refer to the OSF/Motif Programmer's Guide and the HP-UX Reference for information on the drag and drop interface.

Tear-off Menus

When Tear-off Menus are enabled by the application, the user can retain menus on the display for subsequent selections. Each tearable menupane has a tearoff button. When the mouse drag button is pressed on the tearoff button, the pane tears off and can be dragged and then placed by releasing the mouse drag button. The window manager surrounds the tearoff menupane with a menu button and a title. Shifting focus to a tearoff windowpane follows standard window manager policy.

Performance

The `XmText` widget now uses a line table rather than a one-dimensional array. Interactive performance is improved relative to Motif 1.1.

Insensitive Visuals

Visual indications are provided to indicate whether a component can respond to input from users. Labels and buttons have had this behavior in previous releases. In Release 1.2, the behavior is extended to the `Text`, `TextField`, `ScrollBar`, and `ArrowButton` widgets.

Titles for Frames

In Revision 1.2, title labels can be displayed in a `Frame` widget. New resources were added to specify the position of the title in the frame.

Audible Warning

A new resource, `XmNaudibleWarning`, was added to the `VendorShell` widget to specify whether an audible cue should accompany a warning message. `Text` widgets determine the value for this resource from the value of `XmNaudibleWarning`.

Color Enhancements

Users can specify default background color and thresholds for shadow calculation on a per screen basis with these 3 new resources on the `XmScreen` widget:

■ `XmNlightThreshold`

- XmNdarkThreshold

- XmNforegroundThreshold

A new function, XmChangeColor, changes the background and other colors for a specified widget.

Baseline Alignment

A new function, XmWidgetGetBaselines, determines the position of the widget's text baseline. Another function, XmWidgetGetDisplayRect, determines the size and position of the character cell.

2-D Menu Traversal

The left, right, up, and down traversal arrows navigate within a menupane. The up and down arrows wrap between columns, and the right and left arrows post the previous or next menupane.

Input Focus

A new resource, XmNinitialFocus, was added to the Manager class to specify the first widget to receive input focus. This resource specifies a widget and is ignored for all popup menus, menubars, option menus, and pulldown menus.

Traversal Access Functions

To support better interaction with keyboard traversal, Revision 1.2 includes these new functions:

- XmGetFocusWidget

- XmGetTabGroup

- XmIsTraversable

- XmIsVisible

- XmGetVisibility

The XmTrackingLocate function includes the following modifications:

- The function fields all events, not just the button press.

- The function returns on any keystroke or a button press.

- The function can be called for non-sensitive widgets.

A new function, `XmTrackingEvent`, was added, which is just like `XmTrackingLocate`, except that it returns a pointer to the X event.

Virtual Keys

A new function, `XmTranslateKey`, provides a mechanism for applications to define their own `XtKeyProc`, which turns off virtual key processing. Also, a database is looked up in `/usr/lib/Xm/bindings`.

Resource Management

Several new functions were added for managing representation types, including:

- `XmRepTypeRegister`
- `XmRepTypeAddReverse`
- `XmRepTypeValidValue`
- `XmRepTypeGetRegistered`
- `XmRepTypeGetId`
- `XmRepTypeGetNameList`
- `XmRepTypeGetRecord`

Changes for CUA Windows Compliance

In Motif Release 1.2, `KActivate` only activates the default action in Menus and DialogBoxes. In `XmBulletinBoard`, only action buttons (PushButtons and subclasses) can be the default when the user presses `KActivate`. Traversing from one action button to another makes the currently selected button become the default.

Specific Widget Changes and Enhancements

Several minor enhancements were made to the following widgets:

- `XmBulletinBoard`
- `XmCommand`
- `XmList`

- XmScrolledWindow

- XmRowColumn

- XmScrollBar

- XmSelectionBox

- XmText

- XmTextField

Refer to the Motif documentation for descriptions of specific changes.

MWM Enhancements

The following changes were incorporated into the Motif Window Manager (mwm):

- An internationalized .mwmrc file.

- Support for the Rtn continuation character in the .mwmrc file.

- Search XBMLANGPATH for bitmap files.

- Support for tight icon placement.

- Support for pop-down and replay event behavior in mwm menus.

- Support scrolled window traversal to scrolled-off children in icon box.

- Support for geometry of the feedback window with the geometry resource.

- Support for the SHAPE non-rectangular window extension.

The window manager includes these new resources:

stackGrouping	Determines whether a window family should be stacked as a group with its parent window.
transientStackingPolicy	Determines whether all application windows are lowered when a secondary window is lowered.
moveOpaque	Controls whether an image of the window, or just an outline, is moved.

The window manager includes these new and enhanced functions:

f.restore	Restores a window to its previous state. Double clicking on an icon is bound to this function rather than to f.normalize.
f.lower	Includes a **within** argument to move the window within the application stacking order, but retains the parent window below the children rule. Also, includes a **freeFamily** argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.
f.raise	Includes a within argument to move the window within the application stacking order, but retains the parent window below the children rule. Also, includes a **freeFamily** argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.
f.raise_lower	Includes a **within** argument to move the window within the application stacking order, but retains the parent window below the children rule. Also, includes a **freeFamily** argument to move the window absolutely without regard to its local family stack. Both modifiers move the window within the local family stack, but do not move the family stack.

There is one new **mwm** action:

    Alt Esc

It behaves similarly to the **f.next_key**, except that the window is always raised regardless of the value of **focusAutoRaise**.

UIL Changes

The following changes were made to UIL:

- A new command flag, **-s**, was added to enable the use of **setlocale**.

- New UIL syntax to support font sets and font tables.

- New UIL syntax to support wide character strings.

- Syntax changes to WML to allow for the definition of automatically created children of composite widgets.

- UIL can read binary databases (WMD files) containing WML information with the **wmd** option.

- **Mrm** includes one new function, `MrmOpenHierarchyPerDisplay`, which is the same as MrmOpenHierarchy, except that display is passed as an explicit argument. This function replaces `MrmOpenHierarchy`.

UIL Installation Note

Installation of the X11R5 Developer's Environment destructively replaces the Motif 1.1 UIL compiler with the Motif 1.2 UIL compiler. The Motif 1.2 UIL compiler cannot be used to compile UIL source files for Motif 1.1 applications. If additional UIL development for Motif 1.1 applications is expected, then the Motif 1.1 UIL compiler (`/usr/bin/X11/uil`) should be given another name before proceeding with the installation of the X11R5 Developer's Environment.

Impact

Applications should be able to relink or recompile with no problem. In addition, numerous improvements were made in the quality of the widgets, so that applications should benefit from this as well.

Performance

Interactive text performance is much improved, especially in the area of scrolling. Memory-saving changes were implemented, but some of the savings were offset by space taken by new functionality. Most applications should see a minor decrease in memory usage. Clients that are built archived have an increase in code size.

Alternatives and Compatibility

Motif 1.2 is source and binary compatible with Motif 1.1. However, there are a few minor changes that developers should note.

Include File Changes

In the Motif 1.1 release, links to the AtomMgr.h, Protocols.h, ProtocolsP.h, and MwmUtil.h header files were placed in X11 and Xm in the make includes and make install procedures. This was for backwards compatibility with the 1.0 release. The make includes and make install procedures no longer create these links. The files should be included as Xm/ ... instead of X11/ ... .

The following header files were in Release 1.1 of Motif, but are now obsolete in Release 1.2:

- ExtObject.h
- Traversal.h
- VaSimple.h
- VendorE.h
- VendorEP.h

The files are maintained for source compatibility.

Other Visual Changes

The following items describe minor visual differences with respect to Motif 1.1:

- Several pixels have changed in the 3-D beveled look.
- Revision 1.2 color generation routines include minor changes.
- There are minor layout differences due to bug fixes in geometry management.

File Set Name Change

X11-SHLIBS from the previous release of HP-UX was renamed X11R4-SHLIBS. Update issues a warning if you try to load an application that depends on X11-SHLIBS. The application still works if you load X11R4-SHLIBS from the OS media.

HP VUE 3.0

VUE 3.0 runs on 9.0 systems. There are differences between VUE 2.01, which ran on previous releases, and VUE 3.0 in the areas of installation, product structure, and component characteristics. The following sections list the differences.

Installation

VUE 2.01 files are installed in many places throughout the file system. VUE 3.0 files are mostly installed in the **/usr/vue** branch of the file system. This includes, **app-defaults** files, icons, executables, and configuration files. Help files are installed in the **/usr/vhelp** directory, facilitating easier removal in a help file server configuration.

Dr_Vue: HP VUE Diagnostic Tool

Dr_Vue is an HP VUE 3.0 diagnostic tool located in the **/usr/contrib/bin/X11** directory. The purpose of **Dr_Vue** is two-fold:

1. To check on the status of the HP VUE fileset, and

2. To check on the runtime configuration of the system.

Dr_Vue does not initiate any fixes directly. Instead, **Dr_Vue** diagnoses and report problems.

If HP VUE is currently running, **Dr_Vue** is initiated by selecting the **General** toolbox inside the Toolboxes subpanel and double clicking on the **Unsupported** toolbox. Once inside that toolbox, double click on the **Dr_Vue** icon. An action associated with the **Dr_Vue** icon creates an **hpterm** window and runs **Dr_Vue**.

Dr_Vue can also be run from the command line by typing the following:

        /usr/contrib/bin/X11/dr_vue | more

Dr_Vue issues INFO, WARNING, or ERROR statements with information that should help expose problems that exist with the installation or configuration of HP VUE. There is one WARNING statement that should be ignored; it states that the **/usr/vue/nls** directory is missing. This directory should not be present.

Backward Compatibility

VUE 3.0 is functionally compatible with VUE 2.01. User customizations of VUE that affect functionality are supported (e.g., action customizations, controls added to the front panel). Visual customizations are generally supported, but, given the extensive new VUE 3.0 visuals, they often have to be redone to be attractive.

This summary addresses the impact of moving from HP VUE 2.01 to 3.0. Examples of backwards compatibility include:

- Workspace Manager reads the user's 2.01 front panel configuration if one exists; otherwise, the new 3.0 front panel is used.

- A 2.01 to 3.0 front panel conversion program (`vuefp2to3`) is available.

- VUE 2.01 help is preserved and is available from the front panel slideup along with the new 3.0 help volumes.

- Customized 2.01 action and filetype specifications are used

- A 2.01 to 3.0 action/filetype conversion script is available.

- The location of 3.0 toolboxes has changed from 2.01. The 2.01 toolboxes are available within the 3.0 toolboxes.

- 2.01 display dependent sessions are used. Online step-by-step help exists which explains how to move to display independent sessions.

- The `Moving To HP VUE 3.0 from HP VUE 2.01 or X11` online help is available which provides detailed information and step-by-step instructions.

The end user's 2.01 environment is carried across to 3.0. User noticeable affects of updating are described below.

System Configuration Files

Most HP VUE 3.0 files are installed in the `/usr/vue` directory instead of scattered throughout the file system as they were in 2.01. The ramifications of this change on system configuration files are described below.

The set of configuration files used by Login Manager and Session Manager to provide default system startup are customizable by system administrators. These files are moved to `/usr/vue/VUE2.0config`. Therefore, customizations to these files are temporarily lost until the system administrator merges the

customizations with the new default configuration files. These include X server startup (`Xservers`), the appearance of the login screen (`Xresources`), the system default front panel (`sys.vuewmrc`), the system default session (`sys.session`), the default session startup script (`Xsession`), `Xstartup`, `Xreset` and `Xconfig`.

The impact varies depending on the extent of system level customization. The auto-customization portion of the `Xsession` script is maintained from 2.01 to 3.0; it is copied into the 3.0 `Xsession` script.

The 3.0 installation process removes files delivered with 2.01, including, VUE specific directories (as long as they contain only files delivered with 2.01). Files not delivered with 2.01 that are in VUE specific directories are not removed.

Mixed Networks

Interoperability problems exist with the File Manager for networks running both 2.01 and 3.0 with nfs-mounted home directories. In general, problems can occur when a user is manipulating a single home directory concurrently from different workstations.

User Sessions

The user's 2.01 session is moved to 3.0 intact with two exceptions. Hpterms to remote systems running 3.0 come up, but in the current workspace rather than the specified workspace. This is due to `hpterm`'s change in location from `/usr/bin/X11` to `/usr/vue/bin` and session information containing path information. To recover, users must move their `hpterm` to the correct workspace and save their session.

2.01 users logging into 3.0 the first time see the default 3.0 palette. Any palettes the user has customized are kept intact; the session palette setting is lost. To recover, the user must select the palette and save the session.

Backdrops

A few of the backdrops were obsoleted to save disk space. There is a compressed archive of these obsoleted backdrops and step-by-step online help instructions for recovering them (`/usr/vue/examples/vuebackd.sh.Z`).

Resources

System level customizations to the app-defaults files are lost. This is standard procedure for applications updating app-defaults files. Note that user level resources are preserved.

Front Panel

A few front panel resources have been obsoleted. These resources are for specifying geometries for the front panel itself, and the workspace switch control. This is handled automatically in 3.0. If specified, these resources have no effect.

If there is a customized 2.01 front panel specified in the `.vue` directory it is used with new 3.0 visuals (`vuewm` automatically runs the `vuefp2to3` conversion utility). The 2.01 front panel specification is left in place. To change the 2.01 specification to a 3.0 specification the `vuefp2to3` utility should be used. To get new 3.0 functionality the customized 2.01 specification should be removed or parts of the `system.vuewmrc` specification merged in (from `/usr/vue/config/system.vuewmrc`). Customized bitmaps referenced by the front panel specification are displayed with two colors and without a mask. The 3.0 color icon editor can be used to convert the bitmap to a multicolored and masked icon (the transparent color indicates the mask).

Front Panel Mail Indication

The mail indicator on the front panel may periodically revert from the **new mail** state with `elm` running. If `elm` is running when new mail arrives, the mail control indicates new mail for a period of time, then reverts to the normal mail icon. This is because `elm` periodically resynchronizes (wakes up to read the mail file), and then the mail control notices that the mail file has been read. To turn off resynchronization in `elm`, set timeout to zero in the `elmrc` file (`$HOME/.elm/elmrc`).

For correct behavior, mail controls in the front panel must be of type `mail`, not `monitor_file`. The front panel converter may insert the wrong control type when converting a custom VUE 2.0 front panel if the name of the control is not `Mail`.

Window Management

Some users might notice new window or workspace behavior on some applications running in VUE 3.0. The new behavior you might see is that a window occupies more than one workspace over time as you move among workspaces.

This change is part of a defect fix to allow windows that are not in the current workspace to get into the current workspace by simply mapping themselves again. The old behavior of vuewm broke some applications by confining some of their dialogs to other workspaces and fragmenting their user interface.

AbsentMapBehavior is a client-specific resource, so this behavior can be easily tuned. It is documented in the vuewm(1X) entry in the HP-UX Reference.

VUE Lite

There are two primary configurations of VUE: 1) full VUE, and 2) VUE Lite. Users can select (on a display by display basis) between full VUE and VUE Lite sessions. VUE Lite is a reduced version of VUE that has performance characteristics similar to a non-VUE (x11start plus mwm) configuration. VUE Lite includes:

■ vuelogin

■ vuewm (workspaces, front panel, action invocation)

■ vuestyle (with reduced functionality in color, font, background and window manager customization)

■ rudimentary session management

VUE Lite does NOT include:

■ vuefile

■ drag and drop

■ full vuestyle customization (mouse settings, keyboard settings, etc.)

■ vuesession (save clients, mouse/keyboard settings, etc.)

■ screen locking

General Appearance And Behavior Changes

- Motif Style Guide Compliance

- New help presentation and information for all components

- Multicolored icons

- Icon library

New VUE Components

- Graphical editor, `vuepad`.

- Color icon editor.

- File annotator (including text and audio annotations)

New Courtesy Clients

These clients are not tightly integrated into the VUE environment and are provided with minimal support.

- Financial calculator

- `xcolumns` game

- Action diagnostic tool

VUE Library

The VUE 2.01 library (`libXv.a`) is shipped with VUE 3.0 to support existing non-VUE clients that used the VUE API (for on-line help access and session management). The on-line help API is in a separate library for VUE 3.0 (see On-Line Help below) and a new session management API is NOT being provided. ICCCM mechanisms are sufficient to provide application session management.

On-Line Help

A new help library and client is delivered with VUE 3.0. New features and characteristics of the library include:

- No dependencies on the VUE product.

- Works with Motif 1.1 and 1.2.

- Provides a standard look and feel interface for displaying, accessing and processing help information.

- Help information can incorporate multiple fonts and graphics.

- Hypertext and hypergraphics links.

- Standard Motif cut and paste.

- Localizable (including 16 bit languages).

- Help information can be printed (including graphics and 16 bit fonts).

- Authoring markup based on the SGML standard.

A new help browsing client (**helpview**) based on the new help library is delivered with VUE 3.0. The VUE 2.01 help client (**vuehelp**) is shipped with VUE 3.0 to handle non-VUE help text written with the VUE 2.01 **vuehelp** format.

Login Manager (vuelogin)

The Login Manager contains the following new features:

- System hostname displayed on login screen.

- Kerberos authentication is available in addition to standard authentication (subject to approval by government regulatory agencies).

- Customizable languages menu.

- VUE Lite/Full session can be selected at login time.

- The copyright screen after login has cosmetic changes.

Session Manager (vuesession)

The Session Manager manages and reduces contention between clients started up as part of a session (improves login performance on low memory systems). The Session Manager contains the following new functionality:

- Interactive selection of full screen cover when locking.

- A saved session can be used on more than one display.

- A saved session can more easily be transferred between systems.

- Optional locking of screen on timeout.

- Fonts selected on a per-language basis.

- Support for system-wide enabling of access to a display.

- Support for C2 secure system screen locking.

Window/Workspace Manager (vuewm)

The vuewm client was changed in two key areas:

1. general window manager functionality, and

2. front panel appearance, behavior and functionality.

General window manager changes include:

- vuewm and vuesession cooperate to improve login performance.

- Key resources for the window manager are easier to change (through a vuestyle GUI)

- vuewm and vuestyle communication performance enhancements.

- Help available through the [F1] key.

Front panel changes include:

- New visual design (smooth look).

- Support for multi-colored icons with masks.

- New multi-colored icons for the front panel controls.

- Slideup sub-panels.

- Support for animation on pushing a control.

- Support for front panel keyboard traversal.

- Clearer and more robust syntax for configuring the front panel.

- Cleanup of user changeable front panel configuration resources (more robust, consistent and manageable).

VUE 2.01 front panel specifications are automatically converted by the VUE 3.0 vuewm. Existing front panel customizations are functionally supported, but there may be some visual degradation.

Style Manager

The following are the new Style Manager features.

- Style management options can be constrained

- Easy selection of key window manager resources (keyboard focus policy, use of icon box, and so on).

- Easy selection of key session manager resources (screen lock on timeout).

- Support for static color displays.

- Easy configuration of VUE color use.

- Improved backdrop selection.

File Manager

Changes were made to the File Manager (`vuefile`) for VUE 3.0. These changes include:

- Use root window as a desktop for depositing frequently used objects for easy access.

- Multicolor icons.

- Click and type file renaming.

- Quick access to a terminal opened on the current working directory.

- Quick help on icons.

- Arbitrary placement of icons.

- Improved Find File and Filter File dialogs.

- Full support for keyboard traversal.

- Support for mouse button 1 drag and drop.

- Support for quick changes to an known directory.

- File content based file typing.

The VUE 2.01 Drag-and-Drop Manager was merged into File Manager process (for enhanced performance on startup and drag-and-drop interactions).

Action and File Type Databases

The action and file type database syntax was redesigned to be easier to specify and more robust. VUE 2.01 databases are automatically converted at run-time by VUE 3.0 components.

Command Invoker

The VUE 2.01 command invoker process is used to activate local and remote commands and applications. For VUE 3.0 the process was eliminated and turned into a library used by **vuewm** and **vuefile**. The cache of **hpterm**'s was eliminated (to improve login performance and cut down on system resource usage at the cost of run-time performance of actions that use **hpterms**). The command invoker library also does automatic xhosting to allow remote command access to the user's display.

Toolbox

The organization of VUE toolboxes (application folders) was changed for VUE 3.0. Pushing the toolbox button on the front panel brings up the user's toolbox. The system toolbox and a catalog of applications (that the user can order) are accessible through the toolbox slideup.

Graphical Editor

The **vuepad** graphical editor is based on the Motif text edit widget and supports standard cut and paste, word wrap, drag-and-drop and printing. For VUE 3.0, vuepad is capable of handling 16 bit languages.

Color Icon Editor

An editor to create and modify colored icons is shipped with VUE 3.0. This editor can be used to produce icons that tie into VUE's color schemes and dynamic colors.

File Annotator

The file annotator is new for VUE 3.0 and allows the user to attach audio and text annotations to a file. It also can be used to view or file annotations.

4

Other Topics

This part contains information for topics related to specific products you purchase (for example, Starbase, C++, and NetLS).

ANSI/C Compiler

With the 9.0 release, many languages are available as separate products. These products have their own release notes, which are placed on the system during the update process. After the update, such release notes appear in the following directory:

    /etc/newconfig/90RelNotes

This situation applies to the ANSI/C Compiler.

C++

The 9.0 release includes a new release of C++ called HP C++ 3.0. The new release includes new enhancements as follows:

- Support for templates (parameterized types) and exception handling. HP C++ 3.0 is based on USL C++ 3.0.

- As a part of special bundling of C++, HP C++ also includes Standard Components, which is a class library of general purpose programming routines and tools.

The part numbers for HP C++ products are B2400A & B2401A (S400 HP-UX), B2402A & B2403A (S700) and B2404A & B2405A (S800). For more

information, see datasheet #5091-5375E, which is available from the Literature Distribution Center.

Features and Benefits of HP C++ 3.0

```
Feature                          Benefit
-------------------------------------------------------------------------
C++ Compiler                     Generates efficient code;
                                 Faster Compilation

Object Oriented Debug            Locate errors quicker;

Standard Components and other    Off the shelf software;
Class Libraries                  Reduces code that needs to be
                                 written

Tools and Header files extended  Enable use of existing software
to work with C++                 directly from C++

Templates or parameterized types Promotes Software Reuse

Exception Handling               Provides a convenient mechanism for
                                 handling error conditions

Conforms to C++ ANSI base        Standards compliant;
document                         Investment protection
```

GKS

The HP Graphical Kernel System (GKS) was removed from the Corporate Price List in March 1992. To continue HP's commitment to graphics standards, HP has developed a partnership with Advanced Technology Center (ATC) to supply ATC-GKS. This product will be marketed, supported, maintained, and packaged as an HP product.

Network File System (NFS)

Features

At 9.0 NFS has the following NFS 4.1 (from Sun Microsystems) features:

- exportfs(1M)

  ```
  export of root
  directory exports
  read-only exports
  ```

- automount(1M)

- New options to the following commands:

  ```
  mount(1M)
  rpcinfo(1M)
  ypbind(1M) {NIS or Yellow Pages}
  ```

New Command: exportfs(1M)

The `exportfs(1M)` command provides the following new functionality:

- Export of Root

  Allows the NFS server to export `root` capability to NFS clients.

- Directory Exports

  Allows the NFS server to export individual directories and subdirectories to a client, instead of forcing exports to take place at the file system level only.

- Export Read-Only

  Allows the NFS server to export read-only capability on a file or directory to the client.

The `exportfs` command reads `/etc/exports` and manipulates the `/etc/xtab` file. It calls the system routine `exportfs` to keep the kernel up to date with the `/etc/xtab` file. The `exportfs` command uses the `libc/exportent` routines to manage the `/etc/xtab` file. See the `exportent(3)` HP-UX Reference entry for more information.

New Command: automount(1M)

The `automount` command allows NFS mounts to occur dynamically. When the client (user or application) attempts to access a remote NFS file, the file system mount is automatically and transparently established. After a period of inactivity (default is 5 minutes) automount unmounts the mounted file system.

The `automount` command was available on Series 700 computers running the 8.07 release.

Concerning `automount`, see the later section called "Corrections to Installing and Administering NFS Services" for information about corrections to a document.

Enhanced Functionality: mount(1M)

The following options are new for the `mount` command:

noac	Suppress fresh client attributes when opening a file.
nocto	Suppress client attribute and name (lookup) caching.
acregmin=n	Hold cached attributes for at least **n** seconds after file modification.
acregmax=n	Hold cached attributes for no more than **n** seconds after file modification.
acdirmin=n	Hold cached attributes for at least **n** seconds after directory update.
acdirmax=n	Hold cached attributes for no more than **n** seconds after directory update.
actimeo=n	Set **min** and **max** times for regular files and directories to **n** seconds.

For more information, see the `mount(1M)` entry in the HP-UX Reference.

Enhanced Functionality for Several Commands

The `rpcinfo(1M)` command was updated to include a new option when listing registered RPC programs, a new option to discover any hosts running a specified program and version, and a new option to delete RPC registration for the specified program and version. See the `rpcinfo(1M)` entry in the HP-UX Reference.

The `rpc.mountd` command has a new option for tracing. See the `mountd(1M)` entry in the HP-UX Reference.

The `ypbind(1M)` command has a new option for fixing a security hole. This option is provided to allow backward compatibility for the `ypset(1M)` command. The `-ypset` option allows the `ypset(1M)` command to be executed to change the binding. For maximum security, this option should be used for debugging the network from a remote machine.

The `/bin/passwd` file and `ypmake(1M)` command now support NIS (YP) password aging for both local `/etc/passwd` files and NIS password maps. See Appendix A in the Installing and Administering NFS Services document.

Increase in NFS Limit

The NFS limit for user membership in groups has increased 8 groups to 16 groups (the HP-UX limit is 20). See the `group(4)` entry in the HP-UX Reference.

Any NIS operation accessing the host maps is now case insensitive. See `gethostbyname(3N)` in the HP-UX Reference.

Impact of New Features

exportfs(1M)

The only enhancement that requires your attention for backward compatibility is the inclusion of the new `/usr/etc/exportfs` command. It is NECESSARY to run `exportfs` after changing the `/etc/exports` file directly. The command is:

```
/usr/etc/exportfs -a    (all)
```

It is necessary to run **/usr/etc/exportfs** with the **-a** option to have the new **/etc/exports** changes take affect. See the **exportfs(1M)** and **exports(4)** entries in the HP-UX Reference for more information.

The **/etc/netnfsrc** script was modified to execute **exportfs** at boot time. Therefore, administrators only need to run **exportfs** explicitly when there is a change to the **/etc/exports** file.

Only one entry per directory or file is valid in **/etc/exports**. In previous versions of NFS, the ability to have multiple entries for a file system in **/etc/exports** was an undocumented feature that no longer exists.

You can now export on a directory by directory basis. See the **exportfs(1M)** entry in the HP-UX Reference for more information.

You cannot export a directory that resides within the same file system and is either a parent or sub-directory of a directory that is currently exported. For example, **/usr** and **/usr/local** cannot both be exported if they reside in the same disk partition.

mountd(1M)

The **rpc.mountd** deamon should now be started from **/etc/netnfsrc**. Take the following actions:

1. Compare your **/etc/inetd.conf** file with the version in **/etc/newconfig**. Make edits as necessary and make sure the **rpc.mountd** entry is removed.

2. Edit the **/etc/netnfsrc** file by setting the START_MOUNTD variable to 1.

Corrections to Installing and Administrating NFS Services (P/N B1013-90009)

On page 9-9, under the description for indirect and direct map specification in the master map, it says,

    If the map name if prefaced with a plus (+), automount searches
    for an NIS map.

This is incorrect.

By default, **automount** tries to resolve the direct/indirect map names with NIS. If **automount** cannot find an NIS map by that name, it searches the current

directory for a file by that name. To avoid ambiguity, use the full path name when specifying a file for a direct or indirect map. Do not use a plus (+) when specifying an NIS map for a direct or indirect map in the master map.

As described on page 9-11, the syntax **mapname** is used to include the contents of another map within a map.

On page 9-13, delete the following line:

```
To specify an NIS map, preface the map name with a plus (+).
```

On page 9-13, replace the line:

```
/tools     +auto.toolfiles     -rw,intr
```

with:

```
/tools     auto.toolfiles     -rw,intr
```

HP ARPA Services: gated(1M)

Features

The `gated(1M)` command incorporates three independent programs: Berkeley's `routed`, `egpup`, and `hello` programs. It also provides a common configuration file. The 2.1 version includes support for multiple timers and tasks to deal with events. It also provides an enhanced configuration-file format, making configuration and support easier. The following items note the 2.1 improvements for `gated(1M)`.

■ Improved configuration.

The configuration file, `grammer`, was redefined and is now more consistent across all three protocols.

■ Enhanced route control policy.

The route control policy was enhanced to provide more flexible route control. Specifically, the following features were added.

□ Routes can now be listened to and/or propagated by protocol (for example, RIP), autonomous system, interface (for example `lan0`), source gateway.

□ Preferences can now be used to order routes.

□ Masks can now be used when specifying routes.

■ Border Gateway Protocol (BGP)

The Border Gateway Protocol was added and implemented according to RFC-1105 with minor exceptions.

■ Enhanced Exterior Gateway Protocol (EGP)

The Exterior Gateway Protocol was rewritten to be fully conformant with RFC-904. All modes are supported.

■ Enhanced Routing Information Protocol (RIP)

The Routing Information Protocol is now compliant with RFC-1058, and now correctly supports flash updates.

■ Online Configuration Changes

You are now able to disable or enable any gated feature, including complete protocols by signal handling (without killing and restarting `gated(1M)`).

■ Increased control over ICMP redirects.

You can now specify additional options to control ICMP redirects providing more flexible handling of these packets.

■ Improved Tracing

The tracing facility was expanded by providing tracing options with detailed output available for monitoring packets and state machines.

■ New State Table Diagnostic Tool

State dumps are now possible while `gated` is running (via `signal`).

Impact

Those who use the previous version of `gated` must convert their old configuration file to the new configuration file format. A conversion tool is provided that facilitates the conversion process. For more information on converting the previous configuration file for `gated` to the new format, see the `README.conv` file in the `/etc/newconfig/gated` directory.

Computers Affected

The `gated(1M)` command is supported on all HP 9000 systems.

HP ARPA Services: DTC Utilities

Dedicated Port Configuration File

The services include a file named `/etc/dp_file`, which contains Data Communications and Terminal Controller (DTC) port information. This file is used by the system administrator to create mappings between DTC ports and pty device special file names. This information is used for two purposes:

■ Login access via Telnet

■ Outgoing connections to DTC ports via Telnet

The following sections describe the functionality associated with the Dedicated Port Configuration File.

Login Access Via Telnet

Telnet Option Port ID is a new feature of ARPA/9000 that facilitates login access to HP 9000 systems from HP Data Communications and Terminal Controllers (DTCs). Telnet Option Port ID allows the system administrator to configure a correspondence between pty device special file names and DTC ports so that the source of inbound Telnet sessions from DTC ports can be determined and the connection can be assigned to a specific pty device special

file. Previously, all Telnet connections were assigned pty device special files randomly from a pool.

Telnet Option Port ID manages incoming connections from the DTC to the system.

Telnet Option Port ID depends upon a configuration file, /etc/dp_file. This file contains mappings between DTC IP address/board/port and pty names.

Telnet Option Port ID uses the IAC DO ENVIRON Telnet command to initiate an exchange of information between the DTC and the host. When a DTC requests a Telnet session with the ARPA/9000 host, the host sends an IAC DO ENVIRON command to the DTC, which responds with IAC WILL ENVIRON followed by the board and port number of the calling port on the DTC.

The DDFA software creates information used by telnetd to determine if incoming Telnet connections are coming from DTC ports that have been defined in the dp_file. If an incoming Telnet connection is from a DTC port defined in the dp_file, then the well-known pty defined there is used to service the connection. If the incoming connection is not from a DTC port defined in the dp_file, then the pty is assigned in the traditional manner (randomly from a pool).

DTC Device File Access (DDFA) Utilities Software Component

The DTC Device File Access (DDFA) software allows access from HP-UX systems and user-written applications to HP DTCs using standard HP-UX structures. DDFA provides an interface to remote (LAN-connected) DTC ports, which is similar to the interface for local MUX ports. DDFA is a group of configuration files, executable files, and a daemon.

DDFA is documented in the DTC Device File Access Utilities Manual and in the following entries in the HP-UX Reference:

ddfa(7)	DTC Device File Access Utilities description
dp(4)	Dedicated port file
dpp(1m)	Dedicated port file parser
ocd(1m)	Outbound connection daemon

ocdebug(1m)   Debug version of ocd

pcf(4)   Port configuration file

DDFA Utilities are automatically installed with ARPA Services. Refer to the Administering ARPA/9000 Services document.

DDFA is intended for DTC users who need access to DTC devices by using pty device files names. Without DDFA, pty device file names are assigned randomly to devices on DTCs. This randomness makes it difficult to associate a specific pty device file with a specific device connected to a DTC.

DDFA allows the system or network administrator to configure a correspondence between the DTC IP address, board, and port with specified pty device file names. Then, devices on the DTC can be referred to by their pty device file names. For example, these pty device file names can be used by the application programmer with standard `read()`, `write()`, `open()`, `close()`, and `ioctl()` calls. Another example is using the HP-UX spooler to identify a DTC printer. Previously, because DTC devices were assigned random pty device file names, a DTC device could not be specified with the HP-UX spooler. With DDFA configuration, a pty device file name for a DTC printer can be specified with the HP-UX spooler.

Dedicated Port Configuration File

The correspondence, or mapping, between the DTC and pty device file names is created in a configuration file, `/etc/newconfig/ddfa/dp`, which is called the `dp file`. See the `dp(4)` entry in the HP-UX Reference. This is a master template file that you should copy to `/etc/ddfa/dp`.

The dp file is part of DDFA, but it is also used by Telnet to identify incoming connections. This feature is known as `Telnet Port Identification` or `DTC Port Identification via Telnet`. An incoming connection is one that originates from the DTC.

The dp file is parsed by the dedicated port parser program, `/etc/dpp`, which spawns an Outbound Connection Daemon, `/etc/ocd`. The ocd daemon manages the connection and data transfer to each DTC port. The debug version of `/etc/ocd` is `/etc/ocdebug`.

In addition, there is a DDFA port configuration file, /etc/newconfig/ddfa/pcf. This is a master template file that the user should copy to /etc/ddfa/pcf. This file contains timer, connection, and data information for each output device specified in the dp file. Therefore, the pcf file is required only for outgoing connections (connections initiated from the host to the DTC). The ocd daemon uses this information to manage the Telnet connection from the pty device file to the DTC port.

Telnet Port Identification

Communication between the DTC and the host is done via Telnet. Previously, all Telnet connections were assigned pty device files randomly from a pool of available ptys. Enhancements to the Telnet daemon, telnetd, enable Telnet to use the DTC board and port and pty device file configuration information set up by DDFA in the dp file. The dp file is parsed by the /etc/dpp program , which is also part of DDFA. Telnet uses a binary lookup file created by /etc/dpp.

If an incoming Telnet connection is from a DTC port defined in the dp file, then the well-known pty device file defined there is used to service the connection. If the incoming connection is not from a DTC port defined in the dp file, then the pty is assigned in the traditional manner, randomly from a pool of available ptys.

Impact

There is no impact on ARPA/9000 users or administrators who are not specifically interested in using these features. For more information, see the DTC Device File Access Utilities document and the Administering ARPA/9000 Services document.

Computer Affected

All HP 9000 Series 300, 400, 700, and 800 computers are affected.

FORTRAN/9000 (f77)

Series 300/400 Computers

New Product Features

The new features available with the 9.0 release of the Series 300/400
FORTRAN 77 language are:

- BSD compatibility library (libU77).
- C preprocessor support.
- Selected FORTRAN 90 Features.

For a detail listing of the new features for the 9.0 release of the Series 300/400
FORTRAN 77 language, see the HP-UX FORTRAN/9000 9.0 Release Notes
document and the file Fortran in the directory: /etc/newconfig/90RelNotes.

Related Documentation

The f77(1) command in the HP-UX Reference provides a summary reference
to this command.

Other related documentation:

- HP-UX FORTRAN/9000 Programmer's Reference (B2408-90010)
- HP-UX FORTRAN/9000 Programmer's Guide (B2408-90009)
- Programming on HP-UX (B2355-90026)
- HP-UX Portability Guide (B2355-90025)

Instant Ignition

The following sections note what is new, changed, or removed from the INSTANT IGNITION functionality.

The IGNITION Fileset

Boot-up Checklist

The boot-up checklist now displays busy, OK, and FAIL in yellow, green and red, respectively, to indicate the status of initiating various pieces of functionality.

Utilities

- The Instant Ignition utilities were moved from the miscellaneous directory of the application manager to the Office_Apps directory of the General toolbox.

- The XHPCALC action was renamed Calculator.

- The DATEBOOK action was renamed Datebook.

- The XCAL action was renamed Calendar.

Peripherals

An action called Peripherals was added under the System_Info directory of the General toolbox. This action displays a list of the peripherals currently attached to the system.

System Administration

Actions for Updist, UpdateFilesets, RemoveFilesets, RebootSystem and SingleUser were added for 9.0.

The Instant Ignition system administration actions were moved from the system_apps/sys_admin directory of the application manager to the System_Admin directory of the General toolbox.

The HALTSYS action, which called /usr/lib/X11/iiapps/haltsys.sh, was replaced with the HaltSystem action, which calls /usr/lib/X11/ignition/shutdown.ksh.

Printer Configuration

To configure a printer, SAM calls an Instant Ignition routine that allows the printer to be added to the HP VUE printer subpanel.

No-windows Login

The prompts for screen-size parameters that previously occurred during login to no-windows mode were replaced by a routine that automatically sizes HP terminals.

Configuring Your System

Although the IGNITION Utilities are accessible as soon as the fileset is loaded, the Boot-up Checklist and No-Windows Login features must be configured into your system. To do this, login as root, change directory to /, and execute the script /etc/newconfig/Ignition/configure.sh.

The EXPLORE Program

The demo actions that were shipped with the HP-UX 8.07 EXPLORE program are not supported on 9.0. You may remove these actions by executing rmfn -L EXPLORE as root or you may manually modify the action definitions to match the VUE 3.0 format.

Diagnostics

Diagnostic daemons on an 9.0 Instant Ignition system can be enabled or disabled by modifying a single line in the /etc/rc file. To run diagnostics or SupportWave, ENABLE the daemons by ensuring that the following line in /etc/rc does not begin with a # character.

```
add_list start_diags Starting diagnostics
```

To improve performance, especially on low-end systems, DISABLE the daemons by adding a # character (if necessary) to the beginning of the line.

```
# add_list start_diags       Starting diagnostics
```

The IGNITION-HELP fileset

Help Screens

A sequence of help screens explaining the features of an Instant Ignition system are provided with the new IGNITION-HELP fileset. These screens have textual and graphical hyperlinks.

Initial Login Welcome

The default VUE session was modified so that a help window is automatically opened for each user at their initial login.

Default Welcome Message

The Welcome button on the HP VUE Help subpanel is preconfigured by the IGNITION-HELP customize script to determine if you are root. Then you get an appropriate welcome message.

Configuring Your System

If you have already run the IGNITION configuration script as described above, then the IGNITION-HELP features have already been installed. The help functionality can also be configured separately by becoming a system administrator and executing /etc/newconfig/Ignition/config_help.sh from the / directory.

Terminal I/O

There were several changes in the area terminal I/O for HP-UX 9.0.

New Features

- SIGWINCH and the associated 'ioctl' calls (TIOCSWINSZ and TIOCGWINSZ) are now supported on all HP-UX platforms. Applications supporting the SIGWINCH signal is now notified when their window is resized.

The default action for the SIGWINCH signal is to ignore the signal. Applications that do not support SIGWINCH do not see any difference from previous releases.

- New fields were added to the device file minor number for the built-in serial interfaces on Series 700 systems and the enhanced built-in serial interface on some Series 400 and 300 systems. The `termio(7)` entry in the HP-UX Reference explains the model numbers affected in the SYSTEM DEPENDENCIES section.

  Briefly, the new fields let you control how the serial driver uses the `receive` and `transmit fifos` on the UART. By adjusting the transmit and receive fifo use, you can effect a trade-off between performance and flow control responsiveness appropriate to the application of the serial port. The `termio(7)` entry in the HP-UX Reference explains the new interface and has a discussion of advantages and disadvantages of different settings in the SYSTEM DEPENDENCIES section.

- The implementation of RTS/CTS hardware flow control was enhanced on the serial ports referenced above. The new implementation allows the use of hardware flow control to pace inbound serial traffic. Previously, a combination of flow control methods was necessary (hardware flow control to protect hardware fifos and XON/XOFF to protect inbound data buffers in the operating system). The new approach allows flow control of bi-directional binary data traffic through the use of the RTS and CTS signals.

- The interface for enabling RTS/CTS hardware flow control was modified to be consistent across HP-UX platforms. For Series 300, 400, and 700 systems, a bit in the device file minor number can still be used to enable hardware flow control on the serial port. Additionally, the TERMIOX interface documented in the `termiox(7)` entry in the HP-UX Reference explains how to enable and disable hardware flow control settings. This setting overrides the one set through the device file minor number.

- STTY was modified to set and report the status of RTS/CTS hardware flow control. The new fields are `ctsxon` and `rtsxoff`. The fields are also reported in the `stty -g` option. They are located at the end of the data structure so that output is compatible with earlier versions of `stty`.

- Support for additional baud rates was added to STTY. The new baud rates are 57600, 115200, 230400, and 460800 baud. These new settings apply only to serial interfaces supporting the higher line speeds.

Performance

The character handling code in the Line Discipline on Series 700 systems was modified to reduce the overhead on multi-byte reads and writes. The reduction in overhead is dependent on the number of bytes read or written in a single system call. Path length measurements have shown a reduction of about 70% in the TIO path for 100 byte reads and writes. Single byte reads and writes were measured at about a 3% reduction. These measurements are only an indication of the potential improvement. As with all performance measurements, actual CPU overhead reduction is application and system dependent.

LSSERVER

LSSERVER is the run-time component of the NetLS product. LSSERVER provides an environment that allows you to install license passwords and run products that are licensed with NetLS. For information on how this version of LSSERVER interacts with LSLOCK (the NetLS application developer's package), see the section on NetLS.

The 9.0 release of LSSERVER includes changes to the `ls_targetid` utility and corrections for reported inadequacies.

Enhancements to the ls_targetid utility

The `ls_targetid` utility displays output in a more functional manner. The new output shows the node's target ID, now called a permanent target ID, as follows:

        Permanent Target ID: (target id displayed here)

The Permanent Target ID is derived from a permanent hardware source on the system from which the `ls_targetid` program is executed. The current sources for a permanent target id are:

- SPU ID number

- LAN link level address (LLA)

If a system has an external, removable hardware device, such as an HP HIL (pod) device, the `ls_targetid` output displays the computer's permanent target ID, plus a second ID, which is called a portable target ID.

The portable ID is derived from the external device. You can use either the permanent or portable target ID for license generation, depending on your usage needs and your product's sales agreement. In most cases, the permanent target ID is the correct ID to use for generating a license. However, you can use the portable target ID to generate a password for a nodelocked license that can be transferred to a different machine by moving the external device.

The `ls_targetid` command has the following new options:

-h or -usage	Gives usage information.
-q	Runs in quiet mode and omits the Permanent/Portable headings.
-v	Runs in verbose mode and displays descriptive information about permanent and portable target IDs, in addition to displaying the IDs.

Corrections

The product has a new version of `stcode.db`, which contains the NetLS and NCS error codes. The new `stcode.db` allows error text to be displayed when an error occurs.

The LSSERVER installation process was changed to include creation of the `/usr/netls` directory. (To be able to run, the `netlsd` daemon requires this directory.) This corrects a condition in which the `netlsd` daemon would not start on systems that did not have LSLOCK installed.

The `netlsd` daemon startup process was changed to test for the existence of an active `llbd` daemon rather than checking for the existence of the file `llbdbase.dat`. To run, the `netlsd` daemon requires the `llbd` daemon. Previously, the `netlsd` startup checked for the existence of the `llbdbase.dat` file, instead of checking for the actual `llbd` process. This caused the `netlsd` startup to fail at times when it should have succeeded.

There is new algorithm for calculating the space needed for the current license database file (/usr/netls/cur_db) and for storing information within the file. The algorithm reduces the size of the cur_db file.

The ls_targetid utility contains corrections for the following conditions:

- Permissions for the /dev/ls_targetid file were changed to allow access by everyone.

- The ls_admin utility no longer allows expired compound passwords to be installed, and you can now add a license while a license is checked out without corrupting the current license database file (/usr/netls/cur_db).

NCS 1.5.1

HP-UX 9.0 includes the Network Computing Kernel (NCK) portion of the Network Computing System (NCS) 1.5.1 product. The following sections discuss changes to NCS 1.5.1 at HP-UX 9.0.

Additions to NCS 1.5.1

This section describes features of the HP-UX release of NCS 1.5.1 that were added since the NCS 1.5.1 manuals were printed.

- The status code file stcode.db now contains status codes for both the NCS and NetLS products.

- When you run the what command on NCS utilities, the output is nck version 1.5.1 (92/07/17).

- The local location broker daemon, llbd, can now be started and stopped using the SAM utility.

- The ppfm.h and idl/c/*.h header files in NCS/NCK have been updated for better compatibility with C++. The header files now match the ones released with the 91/10/31 version of NCS/NIDL (fixes APR 7D982895/SR 5003042317; also shipped as patches PHSS-0973 (300 series), PHSS-0974 (700 series), and PHSS_0975 (800 series.))

NCS 1.5.1 Limitations

■ NCS and software that depends on NCS does not work on standalone systems whose only network interface is a loopback interface (for example, lo0). To operate, NCS requires a true network interface (for example, Ethernet lan0) that provides a valid IP address.

■ On any host that uses NCS, a network interface that is configured for use with IP must have its address and netmask assigned correctly. Otherwise, NCS may fail to operate. A typical symptom of an incorrect network address is that the lb_admin command cannot contact the local llbd daemon and reports communications failures. A typical symptom of an incorrect netmask is that the corresponding broadcast address is also incorrect, causing lb_admin to be unable to contact the Global Location Broker and to report communications failures.

Using SAM to Start llbd

You can use SAM to start, stop, and restart (stop followed immediately by start) the llbd daemon. Select the:

        Networking/Communications ->

line from the main menu and then select:

        Services: Enable/Disable

For the daemons that appear in the popped up screen, select the line that begins:

        NCS LLBD

Enabling llbd

Pull down the Actions menu and select Enable. SAM performs two actions:

■ Starts an llbd daemon.

■ Ensures that llbd starts automatically during future system starts.

If you select Enable when an llbd is already running, SAM does not start a second llbd.

Disabling llbd

Pull down the **Actions** menu and select **Disable**. SAM performs two actions:

- Stops the **llbd** daemon.

- Ensures that **llbd** does not start during future system starts.

SAM does not stop the **llbd** when no **llbd** is running.

Restarting llbd

Pull down the **Actions** menu and select **Restart**. SAM stops the **llbd** that is currently running and starts a new daemon.

When you complete the changes, pull down the **View** menu and select **Exit**.

Kernel Regeneration Messages

After you manipulate **llbd**, you may see messages about the kernel needing to be regenerated. If you did nothing but enable, disable, or restart LLBD during a SAM session, you should cancel the kernel regeneration. Running **llbd** does not affect the kernel.

NCS 1.5.1 Enhancements and Corrections

- The **lb_admin** command **clean** correctly handles servers that register non-nil objects; in previous releases, this command incorrectly offered to delete registrations for such servers. (Fixes APR 7D988743 /SR 1653002014.)

- The global location broker daemon, **glbd**, now handles diskless clusters properly and starts up properly in heterogeneous clusters. (Fixes APRs 7D98AC33, 7D98BF2E, 7D98E828, and 7D98F5E9, corresponding to SRs 5000658369, 5000655969, 5000656009, and 5000655977. These corrections were also shipped as the following patch to HP-UX 8.0: PHSS-0415 (300 series.))

- The library, **libnck.a**, was updated so that **rpc_$bind** verifies that the specified network family is valid. (This correction was shipped as the following patches to HP-UX 8.05: PHSS-0769 (300 series), PHSS-0770 (800 series), PHSS-0771 (700 series.))

- The status code file, **stcode.db**, contains the **socket_$** status codes, which were inadvertently omitted from previous releases.

- At HP-UX 8.x, if a client made a remote procedure call to a server and that server in turn made a remote procedure call to another server, the initial call from the client would sometimes time-out prematurely. The NCK runtime library in HP-UX 9.0 corrects this condition. (Corrections APR 7D98F62E /SR 5003074906.)

- At HP-UX 8.x, the NCS utilities and NCS-based applications did not function correctly on hosts having kernels that contained the `atalk` AppleTalk driver. The NCK runtime library and utilities in HP-UX 9.0 corrects this condition. (Corrections APR 44032598 /SR J600629857.)

Documentation Clarifications

This section contains errata for the already printed NCS documentation.

glbd Replication

For `glbd` replication to succeed, all replicas of the Global Location Broker Daemon must propagate via the same protocol family. This means that all replica addresses must be listed under the same protocol family in all replica lists. See Chapter 4 of `Managing NCS Software` (Order Number D-11895-E) for a discussion of starting `glbd` replicas.

Using rrpc_$ Calls

The NCK runtime library does not accept `rrpc_$` requests that specify non-nil objects. Clients that make `rrpc_$` calls should always supply `uuid_$nil` as the object UUID in the RPC handle.

NetLS

The Network License System (NetLS) product allows software vendors to create licenses for their products in order to manage, enforce, and monitor software usage and usage policies. Once a product has been licensed with NetLS, customers use the NetLS runtime environment to manage their licensed products. NetLS consists of two components:

■ LSSERVER provides the environment that allows users to install license passwords and run products that are licensed with NetLS.

■ LSLOCK provides a set of tools that allow software developers to license their products and generate license passwords.

LSSERVER 2.01.08 is bundled with the 9.0 release of HP-UX. However, LSLOCK 2.01.08 is available as a separately orderable product. Applications that are licensed with LSLOCK version 2.01 or later runs on HP-UX 9.0 systems. In addition, you can install LSLOCK version 2.01 or later on an HP-UX 9.0 system.

For information about the new features and changes to LSSERVER version 2.01.08 (which is bundled with this release of HP-UX 9.0), see the section on LSSERVER. To take advantage of these changes to LSSERVER, applications must be licensed by the 2.01.08 version of LSLOCK.

NFS/9000

NFS/9000 is bundled with the HP-UX operating system software.

Native Language Support (NLS)

Summary of New Features

The following items highlight the features:

- Worldwide Portability Interface (WPI)
- New `locale(1M)` and `localedef(1M)`
- Removal of `getmsg(3C)` from `libc`
- Provide `getmsg.o` for backwards compatibility
- Support five new locales:

      arabic.iso88596, greek.iso88597,
      hebrew.iso88598, turkish.iso88599, thai

- Deliver 64 new `iconv` data files.
- Miscellaneous system interface changes (`setlocale`, `getlocale`, `nl_langinfo`, `catopen`)
- New filesets were added for the X11 windowing system.

      JAPANESE-X11   0.3 MBs
      KOREAN-X11   0.3 MBs
      CHINESET-X11   0.3 MBs
      CHINESES-X11   0.3 MBs

WPI Routines

Many of the existing XPG library interfaces work with byte values, which limits their support of multi-byte codesets. X/Open could not modify all of the library interfaces to work with wide characters, because that would break backwards compatibility with previous systems.

Instead, X/Open has a parallel set of interfaces that are functionally equivalent to the current single byte interfaces, but work with wide characters. These interfaces, collectively know as the Worldwide Portability Interfaces, are recommended for use by character-based portable applications.

Character Classification

- `wctype()`: determine whether property is valid for current locale.
- `walnum()`: is character alphanumeric? alpha?
- `iswcntrl()`: is character cont* rol?
- `iswdigit()`: is character a digit?
- `iswgraph()`: is character a graphic?
- `iswlower()`: is character lower case?
- `iswprint()`: is character printable?
- `iswpunct()`: is character punctuation?
- `iswspace()`: is character whitespace?
- `iswupper()`: is character upper case?
- `iswxdigit()`: is character hex digit?
- `iswctype()`: determines whether character has given property

File Input/Output

- `fgetwc()`: get wide character from a file
- `fputwc()`: write wide character to a file
- `getwc()`: get wide character from a file
- `getwchar()`: get wide character from stdin
- `putwc()`: write character to a file
- `putwchar()`: write wide character to stdout
- `ungetwc()`: unget wide character
- `fgetw()`: get wide character string from a file
- `fputws()`: write wide character string to a file
- `getws()`: get wide character string from file
- `putws()`: write wide character string to file

- `fprintf()`, `printf()`, `sprintf()`: %C, %S conversion specifiers added for wide character or wide string formatted output

- `fscanf()`, `scanf()`, `sscanf()`: %C, %S conversion specifiers added for wide character or wide string formatted input

Case Conversion

- `towlower()`: converts wide character to lower case

- `towupper()`: converts wide character to upper case

String Manipulation

- `wcscat()`: concatenate two wide character strings

- `wcschr()`: find character in wide character string

- `wcscmp()`: compare two wide character strings

- `wcscpy()`: copy one wide character string to another

- `wcscspn()`: finds section of wide character string not in set

- `wcslen()`: determine the length of wide character string

- `wcsncat()`: concatenate wide character strings up to length

- `wcsncmp()`: compare wide character strings up to length

- `wcsncpy()`: copy wide character strings up to length

- `wcspbrk()`: locate first occurrence of wide character in wide char string

- `wcsrchr()`: find character starting from end of wide character string

- `wcsspn()`: finds section of wide character string in set

- `wcstok()`: find next token in wide character string

- `wcswcs()`: find one wide character string within another

- Locale Specific String Conversions

- `wcstod()`: convert wide character string to double-precision number

- `wcstol()`: convert wide character string to long

- `wcstoul()`: convert wide character string to unsigned long

Sorting and Collation

- `wcscoll()`: wide character string collate

- `wcsxfrm()`: wide character string transform

- `wcsftime()`: convert date and time to wide character string

- `wcswidth()`: number of column positions of a wide character string

- `wcwidth()`: number of column positions of a wide character code

Impact

Applications now have a set of XPG4 portable tools to enable their application to be character set independent and operate on other XPG4 conformant systems.

Alternatives and Compatibility

Although the wide character interfaces adhere to XPG4 standards and are preferred for portability, multibyte interfaces in `nl_tools_16(3C)` are available only on HP-UX systems. Interfaces in `multibyte(3C)` allow for converting between multibyte and wide character.

Performance

No performance degradation of applications using the WPI routines is expected when compared to the same applications using the existing multi-byte routines.

Size requirement

The WPI routines are available in the archived library, `/usr/lib/libwpi.a`, and the shared library, `/usr/lib/libwpi.sl`, in the NLS-CORE fileset, which is currently 472 KBs. Combined, the WPI libraries occupy about 120 KBs of disk space.

The memory usage of an application using WPI routines cannot be predicted, as it is entirely application dependent. However, a specific statement can be made regarding the size of a character: a normal character is one byte; whereas, a WPI character is four bytes.

New Locale(1M) and Localedef(1M)

To conform to POSIX standards, `locale(1M)` and `localedef(1M)` commands now conform to the specifications dictated by the POSIX 1003.2 standard.

The `localedef(1M)` and `locale(1M)` commands are in `/usr/bin`. The new compatible locale files are built with the `localedef(1M)` utility and named `localef` in the respective language directories of `/usr/lib/nls`.

Impact

The format of the scripts accepted by the new `localedef(1M)` and the old `buildlang(1M)` is very different. Thus, the locale input scripts that work with the old `buildlang(1M)` utility are not compatible with the new `localedef(1M)` and vice versa. To maintain backwards compatibility with existing executables, the existing `locale.def` files is also delivered in release 9.0 in addition to the new `locale.inf` files.

Applications compiled with archived `libc` prior to HP-UX release 9.0 access the old `locale.def`. Those compiled with shared libraries or with HP-UX 9.0 access the new `locale.inf` files.

Changes to the POSIX definition of LC_CTYPE character classification for alpha, print, and graph in `localedef` input scripts are only realized in applications that are linked with the shared library version of `libc` or recompiled with the 9.0 version of `isalpha()`, `isprint()`, and `isgraph()`.

Alternatives and Compatibility

`Buildlang(1M)` and its compatible locale files still remain on the 9.0 release for backwards compatibility.

Performance

The performance of `localedef` is not expected to degrade other than the extra time necessary to read in the few new `LC_*` data items that did not exist in the old locale input scripts. Nevertheless, `localedef(1M)` is executed only once per locale, so the performance of the `localedef(1M)` command is not critical.

Size requirement

The new locale files, `locale.inf`, need to be included on the disk along with the existing locale files, `locale.def`, for backwards compatibility with existing applications. The `locale.inf` files exist in the same directory as the existing `locale.def` files. Each existing locale file is about 3-4 KBs and each new locale file is about 4-5 KBs. Although there are currently 49 locales on HP-UX and this number will double. The impact on the disk is minimal because only a few locales are generally loaded on any one system.

The size of `localedef(1M)` is estimated to be 85 KBs and the size of `locale(1M)` is 54 KBs. Both commands are delivered in the NLS-CORE fileset, which is currently 472 KBs.

Memory size is not an issue for the operation of `locale(1M)`, `localedef(1M)`, or the utilities that load the new locale data.

Obsolescence

`Buildlang(1M)` and its `locale.def` files will not be supported or delivered in the next release as `localedef(1M)` and its `locale.inf` files are the preferred and standard locale tools.

LC_ALL category items, YESSTR and NOSTR, were removed in favor of the X/Open preferred LC_MESSAGES items named YESEXPR and NOEXPR.

Removal of getmsg(3C)

The obsolete NLS call, `getmsg(3C)`, was removed from `libc`.

Background

The HP NLS `getmsg(3C)` programmatic call in `libc` conflicts with the AT&T Streams programmatic call, `getmsg(2)` in `/usr/lib/libstr.a`, which was added to the 9.0 release. The name of the Streams `getmsg(2)` call cannot be changed because it is an industry-wide standard. However, the impending obsolescence of the HP NLS `getmsg(3C)` call has been advertised in the `getmsg(3C)` entry in the HP-UX Reference for the past 2 releases. Since an equivalent (but not portable) call can be made to `catgetmsg(3C)` or to the portable and recommended `catgets(3C)`, HP proceeded with the obsolescence of the NLS `getmsg(3C)` call.

Getmsg(3C) exists in /lib/libc.a, /lib/libc.sl, and /lib/libp/libc.a, but libc is used to generically refer to all three libraries rather than to mention all three libc files each time.

Change

Getmsg(3C) was removed from libc and /usr/old/getmsg.o is provided for applications requiring backwards compatibility. Applications can then link to the /usr/old/getmsg.o if the getmsg(3C) calls have not been converted into the portable catgets(3C) call. Applications that use the streams library should not use /usr/old/getmsg.o.

Impact

Applications that were linked with the getmsg(3C) in the shared library, /lib/libc.sl, in the 8.0 HP-UX release and did not modify the source code to use catgets(3C) need to be re-linked to the /lib/getmsg.o file to resolve the link to the getmsg(3C) shared library call.

All HP-UX libraries that call getmsg(3C) must replace the call with calls to catgets(3C) or catgetmsg(3C).

Applications compiled with the shared library version of NLS getmsg(3C) are not binary compatible with the HP-UX 9.0 system. This applies to 8.0 based releases (8._/IF_/MCF) of the S700 and the Series 800 systems.

Alternatives and Compatibility

Usage of the standards conformant catgets(3C) call is recommended as a replacement to getmsg(3C).

Size Requirement

No impact foreseen.

New Locales

Six new locales are provided. Thai is treated as a new language.

/usr/lib/arabic/iso88596/locale.inf

/usr/lib/greek/iso88597/locale.inf

/usr/lib/hebrew/iso88598/locale.inf

/usr/lib/turkish/iso88599/locale.inf

/usr/lib/thai/locale.inf

/usr/lib/chinese-t/big5/locale.inf

Impact

No impact to the application developer.

Alternatives and Compatibility

None.

Performance data

N/A

Size requirement

■ 3 KBs for each optional locale loaded on a system

■ 3 KBs for the optional THAI fileset overhead

Locales Modified: katakana, japanese, japanese.euc

Characters in the range 0xA1-0xDF were deleted from the upper, lower, alpha, and alnum class in LC_CTYPE and placed in the print class in LC_CTYPE.

New iconv Data Files

To enable interoperability of data with different character encoding schemes, 64 new iconv files are provided:

iso8858/1 <=> ebcdic conversion (14 west European languages, 28 files)

```
/usr/lib/iconv/direct/iso81}amere    /usr/lib/iconv/direct/amere}iso81
/usr/lib/iconv/direct/iso81}c-fre    /usr/lib/iconv/direct/c-fre}iso81
/usr/lib/iconv/direct/iso81}danie    /usr/lib/iconv/direct/danie}iso81
/usr/lib/iconv/direct/iso81}dutce    /usr/lib/iconv/direct/dutce}iso81
/usr/lib/iconv/direct/iso81}engle    /usr/lib/iconv/direct/engle}iso81
/usr/lib/iconv/direct/iso81}finne    /usr/lib/iconv/direct/finne}iso81
/usr/lib/iconv/direct/iso81}frene    /usr/lib/iconv/direct/frene}iso81
/usr/lib/iconv/direct/iso81}germe    /usr/lib/iconv/direct/germe}iso81
/usr/lib/iconv/direct/iso81}icele    /usr/lib/iconv/direct/icele}iso81
/usr/lib/iconv/direct/iso81}itale    /usr/lib/iconv/direct/itale}iso81
/usr/lib/iconv/direct/iso81}norwe    /usr/lib/iconv/direct/norwe}iso81
/usr/lib/iconv/direct/iso81}porte    /usr/lib/iconv/direct/porte}iso81
/usr/lib/iconv/direct/iso81}spane    /usr/lib/iconv/direct/spane}iso81
/usr/lib/iconv/direct/iso81}swede    /usr/lib/iconv/direct/swede}iso81
```

iso8859/2 <=> ebcdic conversion (6 east European languages, 12 files)

```
/usr/lib/iconv/direct/iso82}czece    /usr/lib/iconv/direct/czece}iso82
/usr/lib/iconv/direct/iso82}hunge    /usr/lib/iconv/direct/hunge}iso82
/usr/lib/iconv/direct/iso82}polie    /usr/lib/iconv/direct/polie}iso82
/usr/lib/iconv/direct/iso82}rumae    /usr/lib/iconv/direct/rumae}iso82
/usr/lib/iconv/direct/iso82}serbe    /usr/lib/iconv/direct/serbe}iso82
/usr/lib/iconv/direct/iso82}slove    /usr/lib/iconv/direct/slove}iso82
```

iso8859/5 <=> ebcdic conversion (2 Cyrillic languages, 4 files)

```
/usr/lib/iconv/direct/iso85}russe    /usr/lib/iconv/direct/russe}iso86
/usr/lib/iconv/direct/iso85}bulge    /usr/lib/iconv/direct/bulge}iso86
```

iso8859/6 <=> ebcdic conversion (Arabic, 2 files)

```
/usr/lib/iconv/direct/iso86}arabe    /usr/lib/iconv/direct/arabe}iso86
```

iso8859/7 <=> ebcdic conversion (Greek, 2 files)

```
/usr/lib/iconv/direct/iso87}greee    /usr/lib/iconv/direct/greee}iso87
```

iso8859/8 <=> ebcdic conversion (Hebrew, 2 files)

/usr/lib/iconv/direct/iso88}hebre   /usr/lib/iconv/direct/hebre}iso88

iso8859/9 <=> ebcdic conversion (Turkish, 2 files)

/usr/lib/iconv/direct/iso89}turke   /usr/lib/iconv/direct/turke}iso89

arabic8 <=> iso8859/6 conversion (Arabic, 2 files)

/usr/lib/iconv/direct/arab8}iso86   /usr/lib/iconv/direct/iso86}arab8

greek8 <=> iso8859/7 conversion (Greek, 2 files)

/usr/lib/iconv/direct/gree8}iso87   /usr/lib/iconv/direct/iso87}gree8

hebrew8 <=> iso8859/8 conversion (Hebrew, 2 files)

/usr/lib/iconv/direct/hebr8}iso88   /usr/lib/iconv/direct/iso88}hebr8

turkish8 <=> iso8859/9 conversion (Turkish, 2 files)

/usr/lib/iconv/direct/turk8}iso89   /usr/lib/iconv/direct/iso89}turk8

thai8 <=> ebcdic conversion (Thai, 2 files)

/usr/lib/iconv/direct/thai8}thaie   /usr/lib/iconv/direct/thaie}thai8

big5 <=> roc15 conversion (Chinese, 2 files)

/usr/lib/iconv/direct/big5}roc15   /usr/lib/iconv/direct/roc15}big5

Impact

No impact to the application developer.

Alternatives and Compatibility

None.

Performance data

N/A

Size requirement

An additional 20 KBs for the 64 `iconv` tables in the optional fileset
NLS-CORE (currently 472 KBs)

Pascal

With the 9.0 release, many languages are available as separate products. As
such, the products have their own release notes. The release notes for the
products are placed on the system during the update process. They appear,
after the update, in the following directory:

        /etc/newconfig/90RelNotes

This situation applies to the Pascal language.

Starbase

Starbase and the 9.0 Math Libraries (Series 300/400/700/800)

Starbase applications compiled or recompiled on HP-UX 9.0 or later MUST
link the math library, -lm, at compile time. This is true for both shared and
archive libraries.

Starbase applications compiled on HP-UX versions 8.07 or previous versions do
not need to link in the math library unless the application specifically uses it.

The 9.0 version of the Starbase Device Drivers Manual, Pub.No. B2355-90019,
provides an example of how to link a Starbase graphics program to display
output in an X11 window for each device driver. Each example lists the -lm
except for CADplt. The -lm math library should be added to the CADplt
linking example as well.

HP-UX Symbolic Debugger 9.0 Release (xdb)

New Product Features

The new features available with the 9.0 release of the HP-UX symbolic debugger are:

- Additional support for debugging shared libraries, including symbolic debugging.
- Support for C++ 3.0.
- Support for screen mode on additional terminal types and X Window resizing.
- Enhanced stack/local variable viewing.
- Enhanced source file location handling.

For a detail listing of the new features for the 9.0 release of the HP-UX symbolic debugger, see the HP-UX Symbolic Debugger 9.0 Release Notes document shipped with your particular language and the file Debugger in the /etc/newconfig/90RelNotes directory.

Related Documentation

The xdb(1) and cdb(1) commands in the HP-UX Reference provide information about these commands.

The HP-UX Symbolic Debugger User's Guide (B2355-90044) describes the features of the symbolic debugger.

Index