z/OS

IBM Ported Tools for z/OS: Perl for z/OS Feature User's Guide and Reference

SA23-1347-00

z/OS

IBM Ported Tools for z/OS: Perl for z/OS Feature User's Guide and Reference

SA23-1347-00

Note

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 227.

First Edition, June 2006

This edition applies to Version 1 Release 1 of IBM Ported Tools for z/OS: Perl for z/OS Feature (5655-M23) and to all subsequent releases and modifications until otherwise indicated in new editions.

IBM welcomes your comments. A form for readers' comments may be provided at the back of this document, or you may address your comments to the following address:

International Business Machines Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY 12601-5400
United States of America

FAX (United States & Canada): 1+845+432-9405
FAX (Other Countries):
Your International Access Code +1+845+432-9405

IBMLink (United States customers only): IBMUSM10(MHVRCFS)
Internet e-mail: mhvrcfs@us.ibm.com
World Wide Web: http://www.ibm.com/systems/z/os/zos/webqs.html

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this document
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Tables

About this document

This document presents the information you need to set up and use IBM Ported Tools for z/OS: Perl for z/OS Feature.

Who should use this document?

This document is for the system programmers who run a z/OS system with z/OS UNIX System Services (z/OS® UNIX), and for their users who use IBM Ported Tools for z/OS: Perl for z/OS Feature. On other open systems, some system programmer tasks may be done by an administrator.

This document assumes the readers are familiar with z/OS systems and with the information for z/OS and its accompanying products.

Where to find more information

Where necessary, this document references information in other documents about the elements and features of z/OS. For complete titles and order numbers for all z/OS documents, see *z/OS Information Roadmap*.

Direct your request for copies of any IBM publication to your IBM representative or to the IBM branch office serving your locality.

There is also a toll-free customer support number (1-800-879-2755) available Monday through Friday from 6:30 a.m. through 5:00 p.m. Mountain Time. You can use this number to:

- Order or inquire about IBM publications
- Resolve any software manufacturing or delivery concerns
- Activate the program reorder form to provide faster and more convenient ordering of software updates

Softcopy publications

The z/OS library is available on the *z/OS Collection Kit*, SK2T-6700. This softcopy collection contains a set of z/OS and related unlicensed product documents. The CD-ROM collection includes the IBM® Library Reader, a program that enables customers to read the softcopy documents.

Softcopy z/OS publications are available for web-browsing and PDF versions of the z/OS publications for viewing or printing using Adobe Acrobat Reader. Visit the z/OS library at http://www.ibm.com/systems/z/os/zos/bkserv/.

IBM Systems Center publications

IBM Systems Centers produce Redbooks that can be helpful in setting up and using z/OS. You can order these publications through normal channels, or you can view them with a Web browser. See the IBM Redbooks site at http://www.ibm.com/redbooks.

These documents have not been subjected to any formal review nor have they been checked for technical accuracy, but they represent current product understanding (at the time of their publication) and provide valuable information on a wide range of z/OS topics. You must order them separately.

Using LookAt to look up message explanations

LookAt is an online facility that lets you look up explanations for most of the IBM messages you encounter, as well as for some system abends and codes. Using LookAt to find information is faster than a conventional search because in most cases LookAt goes directly to the message explanation.

You can use LookAt from the following locations to find IBM message explanations for z/OS elements and features, z/VM®, VSE/ESA, and Clusters for AIX® and Linux:

- The Internet. You can access IBM message explanations directly from the LookAt Web site at http://www.ibm.com/eserver/zseries/zos/bkserv/lookat/.
- Your z/OS TSO/E host system. You can install code on your z/OS or z/OS.e systems to access IBM message explanations, using LookAt from a TSO/E command line (for example, TSO/E prompt, ISPF, or z/OS UNIX System Services).
- Your Microsoft Windows workstation. You can install code to access IBM message explanations on the *z/OS Collection* (SK3T-4269), using LookAt from a Microsoft Windows command prompt (also known as the DOS command line).
- Your wireless handheld device. You can use the LookAt Mobile Edition with a handheld device that has wireless access and an Internet browser (for example, Internet Explorer for Pocket PCs, Blazer, or Eudora for Palm OS, or Opera for Linux handheld devices). Link to the LookAt Mobile Edition from the LookAt Web site.

You can obtain code to install LookAt on your host system or Microsoft Windows workstation from a disk on your *z/OS Collection* (SK3T-4269), or from the LookAt Web site (click **Download**, and select the platform, release, collection, and location that suit your needs). More information is available in the LOOKAT.ME files available during the download process.

Using IBM Health Checker for z/OS

IBM Health Checker for z/OS is a z/OS component that installations can use to gather information about their system environment and system parameters to help identify potential configuration problems before they impact availability or cause outages. Individual products, z/OS components, or ISV software can provide checks that take advantage of the IBM Health Checker for z/OS framework. This book may refer to checks or messages associated with this component.

For additional information about checks and about IBM Health Checker for z/OS, see *IBM Health Checker for z/OS: User's Guide*. z/OS V1R4, V1R5, and V1R6 users can obtain the IBM Health Checker for z/OS from the z/OS Downloads page at http://http://www.ibm.com/systems/z/os/zos/downloads/.

SDSF also provides functions to simplify the management of checks. See *z/OS SDSF Operation and Customization* for additional information.

Chapter 1. Introduction

Perl, which stands for Practical Extraction and Report Language, is a general purpose, open source scripting language. Its powerful text manipulation functions allow file maintenance and automation of system administration. It is also popular for CGI programming on the web owing to its inherent ability to process text. It is highly adaptable and intended to be easy to use and efficient.

IBM Ported Tools for z/OS: Perl for z/OS Feature (Perl for z/OS) is a ported version of the open source application Perl (version 5.8.7) available from cpan.org. Perl for z/OS has been tested and packaged for use on z/OS.

This document is designed specifically for users of the Perl for z/OS product, and assumes the user has working knowledge of Perl. For more information on the IBM Ported Tools for z/OS product, go to the following site: `http://www.ibm.com/servers/eserver/zseries/zos/unix/port_tools.html`

Finding more information

Information specific to the z/OS UNIX implementation of Perl can be found in this document. For general information about the Perl language including links to additional reference sources, refer to the CPAN (Comprehensive Perl Archive Network) website:

`http://cpan.org`

The official online souce for Perl core user documentation can be found on the Perldoc website:

`http://perldoc.perl.org/`

Note: IBM is not responsible for content or examples provided by non-IBM Web sites or other non-IBM resources.

There are also several mailing lists that may prove useful. One such mailing list is perl-mvs, which provides a forum for people who are using perl on MVS platforms. Note that this mailing list is not provided or maintained by IBM. To subscribe to this mailing list, or to check out other Perl mailing lists that might be of interest, go to the following site:

`http://lists.cpan.org/`

And finally, there is the MVS-OE forum which provides general discussions on z/OS UNIX topics. Go here to subscribe:

`http://www-03.ibm.com/servers/eserver/zseries/zos/unix/bpxa1dis.html`

Supported utilities

In addition to the `perl` module, Perl for z/OS provides support for additional utilities that are included with the Perl application. The following is a list of utilities supported, along with a description of each:

a2p awk to Perl translator. a2p takes an awk script specified on the command line (or from standard input) and produces a comparable perl script on the standard output.

cppstdin
This utility invokes the C preprocessor on standard input and send the output to stdout. It is primarily used by other Configure units that ask about preprocessor symbols.

enc2xs
This utility builds a Perl extension for use by Encode from either Unicode Character Mapping files (.ucm) or Tcl Encoding Files (.enc). Besides being used internally during the build process of the Encode module, you can use enc2xs to add your own encoding to perl. No knowledge of XS is necessary.

find2perl
This utility acts as a translator used to convert **find** command lines to equivalent Perl code.

h2ph This utility converts any C header files specified to the corresponding Perl header file format.

h2xs This utility reads the .h file for a C library and then creates a skeleton for the .xs file (this is required to build a Perl module which links to the library). It also constructs a system of directories and makefiles in which to build and test the Perl module.

instmodsh
This utility provides an interactive shell type interface to query details of locally installed Perl modules.

libnetcfg
This utility is used to configure the libnet. Starting from Perl 5.8, libnet is part of the standard Perl distribution, but the libnetcfg can be used for any libnet installation. Also, without arguments, libnetcfg displays the current configuration.

psed This utility is a stream editor which reads the input stream consisting of specified files. It then processes the stream by applying edit commands contained in a script and finally writes the resulting lines to standard output.

s2p This utility produces verbose warning diagnostics.

splain This utility produces verbose warning diagnostics.

xsubpp
This utility is a compiler used to convert Perl XS code into C code.

Restrictions and limitations

The below restrictions and limitations apply to Perl for z/OS:

- The following encodings are not supported on EBCDIC:
  - Chinese
  - Japanese
  - Taiwanese
  - Korean

Chapter 2. Installing Perl for z/OS

This chapter should be read before installing Perl for z/OS. It consists of the following sections which provide pre-installation planning information and additional post-installation steps.

- **Pre-installation Planning.** This section provides information for file-system allocation and help on locating and relocating or removing open source versions of Perl that may reside on your system.
- **Post-installation setup.** This section provides required and optional post-installation steps such as providing symbolic links to Perl for z/OS,

Pre-installation planning

The following sections provide you with information that will be helpful in planning to install Perl for z/OS.

File-System Allocation

Perl for z/OS will installed into `/usr/lpp/perl`. The *Program Directory for IBM Ported Tools for z/OS* describes that an HFS file-system of 2400 tracks of 3390 DASD is required. This is approximately 136 M bytes (assuming 15 tracks/cyl 849,960 bytes/cylinder for a 3390 device).

Perl for z/OS supports the ability to be expanded dynamically with *user modules* (non-IBM supported modules which provide additional functionality, see Chapter 3, “Installing user modules,” on page 9 for more information). This is a very popular approach amongst Perl users and most system administrators will find themselves asked to consider adding user modules. While user modules can be installed outside of Perl's installation directory, there are times when you may find it beneficial to install it in Perl's installation directory. Because the default is to mount the `/usr/lpp/` file-system as read-only, a separate file-system for the installation of Perl should be mounted at `/usr/lpp/perl` to facilitate the addition of user modules.

Files and directory structure

Perl for z/OS will be installed into the following directories and files as follows:

```
/usr/lpp/perl/                  Parent directory for Perl for z/OS
/usr/lpp/perl/bin               Executeable modules
/usr/lpp/perl/bin/perl          Perl executeable
/usr/lpp/perl/lib               Parent directory of libraries
/usr/lpp/perl/lib/5.8.7         Parent directory of v 5.8.7 libraries
/usr/lpp/perl/man               Parent directory of man page
```

The following symbolic links are not automatically created by the installation process but should be created during post installation. (See “Post-installation setup” on page 6 for more information on defining these links):

```
/usr/bin/perl               -> /usr/lpp/perl/bin/perl
/usr/lib/libperl.so         -> /usr/lpp/perl/lib/5.8.7/os390-thread-multi/CORE/libperl.so
/usr/man/C/man1/hpeza200.book -> /usr/lpp/perl/man/C/man1/hpeza200.book
```

Migrating from open source versions of Perl

Before installing Perl for z/OS, you should check if a previous *open source* version of Perl acquired from another source has already been installed. Currently, open source versions of Perl (unsupported by IBM) which have been enabled for z/OS UNIX are available from CPAN.org and also the Tools and Toys webpage on the

z/OS UNIX System Services (z/OS UNIX) web site (http://www.ibm.com/systems/z/os/zos/features/unix/bpxa1toy.html). Because these versions are not supported, you may wish to remove them. However, Perl for z/OS can co-exist with previous versions, so if you choose to retain them, you will need to make sure that the location of the previous perl module and perl utilities does not conflict.

Although Perl for z/OS is functionally equivalent to other available versions of Perl at the same level or earlier, Perl itself has occasionally introduced changes between versions which are not compatible with previous versions and so your user community may need access to the previous version until those version-related migration issues are resolved. So, you should initially retain any previous versions until your user community has transitioned to Perl for z/OS .

The Perl application consists of both executeable modules and libraries. The executable modules and libraries for Perl for z/OS will be installed into directories in /usr/lpp/perl/. This is a unique directory specific to Perl for z/OS and so during the installation into these directories there should be no risk of conflict with open source versions of Perl.

The libraries for the open source versions of Perl are typically installed in `/usr/lib/perl5/` or `/usr/local/lib/perl5/` and as stated earlier, these will not conflict with the libraries for Perl for z/OS.

It is customary, however, for the executable modules (the module `perl` and Perl's utility programs) or links to them, to be located in `/usr/bin/` and/or `/usr/local/bin/` and sometimes `/bin/`. As an additional post-installation step, symbolic links to `perl` should be defined in one of these directories and this is the point at which a conflict with a previous version may occur. So, if you have a previous version, you will need to make sure to remove or rename these.

Tips for finding previous versions of Perl

As stated in “Migrating from open source versions of Perl” on page 3, the only conflict during the installation of Perl for z/OS and an open source version of Perl may be in the `/usr/bin/` and/or `/usr/local/bin/` directories where the executeables are typically installed. There should be no conflict with the installation of the libraries. So, this section will focus on providing tips for finding the executeables. Once the `perl` executeable is found, it can be used to determine the location of the libraries.

- **Use the `whence` command.** If perl is located in a directory which is defined in $PATH, then the `whence` command will indicate which directory it is in. From the UNIX command line, enter: `whence perl`
- **Check common directories**. Typically, the perl module, or a link to it, is installed in one or more of the following directories:

  ```
  /usr/bin
  /usr/local/bin
  /bin              (although not typical)
  ```

  Do an "`ls -ld perl*`" in each of these directories.
- **Use the find command**. The find command is the most thorough method as it performs a search on all directories. However, it can take several minutes to run on systems with heavily populated file-systems. Also, to check all paths, it will most likely need to be run from a superuser. You may wish to restrict the find to system directories such as `/usr` or `/bin`. Examples:

  ```
  find /usr -type f -name "perl*"
  find /bin -type f -name "perl*"
  ```

If you run this command from a non-superuser, you may wish to redirect all error messages for directories you do not have access to to `/dev/null`. To do so, add `" 2> /dev/null"` to the end of the above `find` commands.

- **Use the perl -V command to find libraries**. If you have located the perl module, you can use the -V option to identify the location of the libraries used by that version of perl:

  ```
  perl -V
  ```

  At the end of the output, you will find the location of the libraries. For example:

  ```
  @INC:
     /usr/local/lib/perl5/s390/5.00403
     /usr/local/lib/perl5
     /usr/local/lib/perl5/site_perl/s390
     /usr/local/lib/perl5/site_perl
     .
  ```

Removing or relocating previous versions of Perl

If you have determined that you have a previous version of Perl, you need to determine whether you should remove it or relocate it. You should initially **relocate** previous versions-- rather than remove them-- until you are satisfied that your user community has no dependencies on it.

Some other considerations:

- Perl executables from previous versions which have been installed in the common locations should be relocated or removed to avoid conflict with executables provided with Perl for z/OS. The complete list of these potential modules follows (however, not all of these are supported in Perl for z/OS):

  ```
  perl         perl5.8.*
  a2p          find2perl    pl2pm        podchecker   splain
  c2ph         h2ph         perlbug      pod2html     podselect    xsubpp
  cpan         h2xs         perlcc       pod2latex    prove
  cppstdin     instmodsh    perldoc      pod2man      psed
  dprofpp      libnetcfg    perlivp      pod2text     pstruct
  enc2xs       perl         piconv       pod2usage    s2p
  ```

- The location of the Perl for z/OS libraries will be different from the directories of the open source versions, so there is not a requirement to relocate the open source version of the libraries to install Perl for z/OS. However, if the previous versions are left installed in the customary locations, users not aware of the path of the libraries for Perl for z/OS may mistake those open source libraries for the supported libraries.
- The location of the libraries is hard-coded into the executables, so if the libraries are moved, the previous executables will not work unless the user defines the path to the libraries using the PERL5LIB environment variable. For example, presuming `/usr/bin/oldperl` is the new location of the Perl libraries:

  ```
  export PERL5LIB=/usr/bin/oldperl
  ```

Analyze existing perl scripts

If you have an open source version of Perl installed, then most likely there are Perl user's scripts on your system as well. While Perl does not guarantee that it is upwardly compatible, it is rare to encounter problems due to this. So, there is a high likelihood that existing scripts will work fine with Perl for z/OS without modification.

If you have Perl applications that are critical to your mission, however, you may wish to test them against Perl for z/OS before making Perl for z/OS the default version of Perl on your system. This can be done by installing Perl for z/OS but not creating the symbolic links in the standard directories such as `/usr/bin`. This way,

you can continue to use the previous version while directing specific test scripts to the new Perl for z/OS `perl` installed as `/usr/lpp/perl/bin/perl`.

Perl Script Analysis Tool. The Perl Script Analysis Tool (`psat.pl`) is a script available for download from the http://www.ibm.com/servers/eserver/zseries/zos/unix/perl/index.html. This is a tool that can be run against one or more scripts and which will generate a report of error messages and warnings for any known problems. This is an evolving tool which will be regularly updated as and should new potential pitfalls with existing scripts become known. You may wish to use this tool to perform a preliminary screening of existing Perl scripts before cutting over to Perl for z/OS

Post-installation setup

This section describes post-installation steps.

Add references to libperl.so

This step is required.. Perl for z/OS is a DLL (dynamic link library) application. It requires that the directory path of the primary DLL `libperl.so` be defined in the LIBPATH environment variable. (`libperl.so` is located in `/usr/lpp/perl/lib/5.8.7/os390-thread-multi/CORE/`).

If not defined, when `perl` is executed, the user will see the following error:

```
CEE3501S The module libperl.so was not found.
         The traceback information could not be determined.
```

and a `CEEDUMP` file may be created.

There are severals ways that LIBPATH can be updated and several places where LIBPATH should be updated. The following sections describe these methods and places.

Update LIBPATH environment

There are three approaches to adding `libperl.so` to the LIBPATH environment variable which are summarized as follows and described in more detail below:

1. to create a symbolic link to `libperl.so` in a directory which is already defined in the system default for LIBPATH;
2. to add the directory containing `libperl.so` to the system default setting of LIBPATH;
3. to require the user to add the directory containing `libperl.so` to their LIBPATH.

Create symbolic link to `libperl.so` in default `LIBPATH`. Unless your installation has changed or removed it, the default system profile (`/etc/profile` which is typically based on `/samples/profile`) will include the directory `/usr/lib`. For example, in `/etc/profile` you might see:

```
LIBPATH=/lib:/usr/lib:.
export LIBPATH
```

Presuming so, you should create a symbolic link to `libperl.so` in `/usr/lib` using the following shell commands (superuser authority will most likely be required):

```
cd /usr/lib
ln -s /usr/lpp/perl/lib/5.8.7/os390-thread-multi/CORE/libperl.so  libperl.so
```

Add directory of `libperl.so` to default LIBPATH. An alternative to creating a symbolic link in `/usr/lib` is to add the directory containing `libperl.so` to the default

LIBPATH. (This is not the preferred solution as it adds a directory to every user's LIBPATH which is not needed by every user). To implement this approach, edit the /etc/profile file and add the directory to the existing LIBPATH definition. For example, change it to:

```
LIBPATH=/lib:/usr/lib:/usr/lpp/perl/lib/5.8.7/os390-thread-multi/CORE:.
export LIBPATH
```

Require user to add the libperl.so directory to their LIBPATH. This approach requires that the user know to do this before running perl. This can be done by the user in two ways. The first would be to add the LIBPATH definition to their .profile in the same manner as shown for updating /etc/profile. The second approach would be to set it from the command line prior to executing perl or by preceeding the perl command or script with the modification as shown in the following example:

```
LIBPATH="/usr/lpp/perl/lib/5.8.7/os390-thread-multi/CORE:$LIBPATH"  perl hello.pl
```

This will cause LIBPATH to be changed only for the duration of the perl command.

Update Webserver configuration

In order to run Perl scripts through the IBM HTTP Server, the directory containing libperl.so needs to be added to the LIBPATH variable defined in httpd.envvars (which is typically found in the directory /etc). For example:

```
LIBPATH=/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/ldap/lib:/usr/lpp/perl/lib/5.8.7/os390-thread-multi/CORE
```

or if a link was created in /usr/lib to libperl.so as described earlier:

```
LIBPATH=/usr/lpp/internet/bin:/usr/lpp/internet/sbin:/usr/lpp/ldap/lib:/usr/lib
```

The 'PassEnv LIBPATH' statement may also need to be added to the httpd.conf file. Refer to the IBM HTTP Server User's Guide for more information on IBM HTTP Server configuration.

Update cron jobs

Note: If you have determined that there is not an open source version of Perl already installed on your system, then this step can be skipped as there will be not Perl scripts that are being run.

cron is a unix utility which allows programs to be scheduled and automatically run at preset times or intervals. crontab is the utility used to schedule these programs. Because LIBPATH is not an environment variable that is automatically set when a program is run under cron, any scheduled jobs that invoke perl need to have the LIBPATH variable prepended to the command. For example, if the command is myperlscript.pl, change it to:

```
LIBPATH=/usr/lpp/perl/lib/5.8.7/os390-thread-multi/CORE  myperlscript.pl
```

The programs that have been scheduled to run under cron can be found in the directory /usr/spool/cron/crontabs. Each file in this directory has a name which is the same as the user who scheduled the jobs and contains the jobs which they have scheduled. To find any perl programs which may be invoked, the grep utility can be used. grep is a command which will search a file or files for a specified string. A Perl program can be invoked by passing a perl script to perl as an argument (ex: perl myperlscript.pl) or by defining perl on the *shbang* statement on the first line of a script (ex: #!/usr/bin/perl). It is customary to name Perl programs with an extension of ".pl" or ".cgi", however, this is not a requirement, so there is no full-proof way to find all perl programs, however the following grep commands (when run by a superuser) will reveal most:

```
cd /usr/spool/cron/crontabs
grep -l perl *
grep -l "\.pl"
grep -l "\.cgi"
```

Should these commands reveal Perl programs, the `crontab -u user` command can be used to modify these files. However, it may be best to contact the owners and have them update their own crontab files as they will be most familiar with these commands.

Add symbolic link for man page

The Perl for z/OS man page file is `hpeza200.book` and is located in the directory `/usr/lpp/perl/man/C/man1/`. In order to easily view this with simply the command `man perl`, the directory `/usr/lpp/perl/man/C` (without the "/man1/") needs to be added to the default MANPATH environment variable defined in `/etc/profile`.

Alternately, you can create a symbolic link to `hpeza200.book` in the `/usr/man/C/man1/` directory which is typically defined as the default path of MANPATH in `/etc/profile`. To do this:

```
cd /usr/man/C/man1
ln -s  /usr/lpp/perl/man/C/man1/hpeza200.book  hpeza200.book
```

Add symbolic link to perl in /usr/bin

Perl is typically installed into `/usr/bin` (and sometimes also `/usr/local/bin` or `/bin` although these are less often done). Most perl scripts are not invoked by passing them as an argument to perl (as in: `perl myperlscript.pl`), but by defining perl as the interpreter on the shbang statement (ex: `#!/usr/bin/perl`. Because Perl for z/OS is installed as `/usr/lpp/perl/bin/perl`, in order for these scripts to run, the shbang statements would need to be changed. Therefore, a symbolic link for perl should be created in `/usr/bin` which points to `/usr/lpp/perl/bin/perl`. For example:

```
cd /usr/bin
ln -s  /usr/lpp/perl/bin/perl   perl
```

You may also wish to create a symbolic link in `/usr/local/bin` which is less typical than `/usr/bin` although reasonably common. It is not suggested that you create a link in `/bin` although that can be found on some systems.

Add path of perl to PATH environment variable

To facilitate running perl from the z/OS UNIX command line, update the PATH environment variable to include the path of perl. If a symbolic link to perl was created in the previous step in `/usr/bin`, then add `/usr/bin/` to the PATH environment variable. If not, then add the `/usr/lpp/perl/bin` to the PATH environment variable.

Changing the PATH variable for all users is typically done in the `/etc/profile` file. For example, in the `/etc/profile` file, find the last statement defining PATH. It will look something like:

```
PATH=/bin:/usr/lpp/java/J2/J1.3/bin:.
```

Add a new line after the last PATH statement to add the path of perl to the existing PATH definition:

```
PATH="$PATH:/usr/bin"
```

Chapter 3. Installing user modules

This section provides instructions on how to install user modules for your perl application. User modules are a self-contained piece of Perl code that can be used by a Perl program or by other Perl modules. It is conceptually similar to a C link library, or a C++ class. They are typically written by users of Perl to enhance the functionality of Perl and are made available to the Perl user community through repositories such as

`http://cpan.org`

.

Note: The **process** for adding user modules to the Perl for z/OS installation is supported by IBM, however, the **user modules** are not supported by IBM. Should you experience problems testing or running a user module, please contact the author of that module as IBM will not be able to provide service for it. Additionally, you may want to share your question with other Perl for z/OS users by posting your question to the `perl-mvs` mailing list which can be found at http://lists.cpan.org/.

Before you Begin

Insure you have the necessary access

The default process for installing user modules is to install them into Perl's installation directories. On most systems, this file-system will be mounted read-only and will require super-user or special authority.

If you do not have this access, you can install to a local directory that you control, however, additional steps will be required to perform the initial installation and also later to access the module from your Perl program.

Acquire required utilities

The installation process for most all user modules will require the use of the `gzip` and `gmake` utilities. These are not utilities provided with z/OS UNIX or supported by IBM. However, very usable versions can be downloaded from the IBM Tools and Toys Web site, which is a repository of helpful applications for z/OS UNIX which are not currently supported by IBM. These versions of `gzip` and `gmake` are not the latest version available, however, there have been no reports of users experiencing problems with them. For those who wish to have the latest and greatest versions of gzip and gmake, the open source code is available from http://gzip.org and http://www.gnu.org/software/make.

Note that while there are some similiarities between the z/OS UNIX `make` utility and `gmake`, there are enough differences that Perl user modules do not always build correctly when using the z/OS UNIX `make`. So, `gmake` should always be used to build user modules.

Install prerequisite modules

Some modules require that prerequisite modules be installed first. Review the `INSTALL`, `README`, or similar help file provided in the module package to determine this.

Steps for installing user modules

This section describes the typical steps required to install most user modules. Occassionally, some modules will use a modified process which should be described in an `INSTALL`, `README`, or similar help file provided in the module package.

1. **Acquire the user module package** (typically a "tar.gz" file-- a tar file which has been compressed with `gzip`). When transferring this file to your z/OS UNIX system, to prevent corruption of the file, be sure to use binary mode (also known as image mode in some file transfer applications) as opposed to "text" or "ascii" mode.
2. **Extract the files from the package.** Most all packages will be designed so that the extracted files are installed into a sub-directory (in the current directory) of the same name as the package. The file must first be uncompressed using gzip. For example:

   ```
   gzip -d module-2.0.tar.gz
   ```

   This will produce the tar file `module-2.0.tar`. The module file should next be extracted, or *unwound*, using the `pax` utility. (`tar` does not perform automatic codepage translation and has a few other traits that make it the less desired application for this). For example:

   ```
   pax -rvf module-2.0.tar  -o to=IBM-1047
   ```

   This will extract the files and also convert them from the ASCII code page (ISO8849-1) to the EBCDIC code page (IBM-1047) at the same time.

   You can perform the gzip and pax commands in one step by piping the output from gzip to pax as shown in the following example:

   ```
   gzip -cd module-2.0.tar.gz | pax -rv -oto=IBM-1047
   ```

   Note that if the package contains non-text files such as graphic files (jpg, gif, etc), they should be extracted in a second pass without the `-o to=IBM-1047`. For example:

   ```
   pax -rvf module-2.0.tar  *.jpg
   ```
3. **Set up the environment.** Define the following environment variables:

   ```
   export _C89_CCMODE=1
   export LIBPATH="$LIBPATH:/usr/lpp/perl/lib/5.8.7/os390-thread-multi/CORE"
   ```
4. **Build the module.**

   ```
   cd module-2.0
   perl Makefile.PL
   gmake
   ```
5. **Test the module.**

   ```
   gmake test
   ```
6. **Install the module.**

   ```
   gmake install
   ```
7. **You are done.** The module is now installed and ready for use.

Chapter 4. Considerations for porting or writing Perl scripts

The purpose of this chapter is to familiarize you with those aspects of Perl which may differ on z/OS UNIX as compared with other platforms. While every effort has been made to ensure that the behavior of Perl for z/OS is consistent with Perl on other platforms, each platform has its unique characteristics that can affect how Perl operates. When writing a new Perl script, particularly one that might be used on other platforms, or when porting a Perl script from another platform to z/OS UNIX, you should keep in mind the issues discussed in this chapter.

The following topics are discussed:

The path of perl on your system

It is standard practice to define the location of perl on the first line of a perl script. For example:

```
#!/usr/bin/perl
```

The default installation path of the `perl` module is `/usr/lpp/perl/bin/perl`, however, it is suggested as a post-installation step that `/usr/bin/perl` be created as a symbolic link to `/usr/lpp/perl/bin/perl` so that no changes are required to your perl scripts.

Your system administrator may have purposely or accidentally overlooked creating this link, so if your perl script fails, verify that the location of perl on your sytem matches that defined in your script.

Bootstrap statement not required

Prior to os/390 V2R8, the shell did not support the shbang ("#!...", "pound bang", or "Magic Number"), so the "`#!/usr/bin/perl`" statement was not recognized. In order to get perl scripts to run, you had to insert the following statement at the beginning of the script:

```
eval 'exec /usr/bin/perl -S $0 ${1+"$@"}'
if 0;
```

If you have these in your scripts, they should be removed and the proper shbang statement used. If you leave them in, your scripts will still work fine (presuming you've installed perl in the same location), but they will probably cause unneccessary confusion to new developers and also drag performance down a bit.

Calls to UNIX commands and utilities

When porting scripts from other platforms that contain calls to UNIX system commands or utilities, be aware that the syntax or behavior of these commands may vary. Two examples to illustrate this:

- The `sort` utility (not to be confused with the perl sort() function although this issue applies to it as well), can sort data in a different order depending on the platform. For more details, refer to “Sort order differences” on page 14.
- The `date` command, while a standard utility on UNIX systems, may vary from platform to platform and an option supported by `date` on other platforms may not be supported on z/OS UNIX, or vice-versa. The following examples show date commands that would not execute on other platforms:

```
z/OS:    date -c
linux:   date -d "12/31/03 23:59"  "+%s"
FreeBSD: date "-j -f "%Y%m%d_%H:%M" 20031231_23:59  "+%s"
```

Message numbers

To enhance serviceability, IBM assigns a unique message number to each warning and error message generated by Perl. These message numbers are of the form `HPE2`*xxxx* where *xxxx* is a unique four digit number.

If your script or process needs to compare these messages against an expected message, the presence of the message number can cause the message to not match. For this reason, message numbering can be disabled by setting the `_PERL_NOMSGID` environment variable to 1 prior to running the script. For example:

```
export _PERL_NOMSGID=1
```

File modes

The numeric value of file modes on z/OS UNIX differ from those in standard UNIX. The following example shows the difference between standard and z/OS UNIX using Tie with read only mode:

```
Standard UNIX
tie(@a, "Tie::File", $file, mode =>0, recsep => $RECSEP)

z/OS UNIX
tie(@a, "Tie::File", $file, mode =>2, recsep => $RECSEP)
or
tie(@a, "Tie::File", $file, mode =>O_RDONLY, recsep => $RECSEP)
```

Note: mode=>0 is for O_RDONLY on standard UNIX, but on z/OS UNIX it is mode=>2

To prevent this from occurring, use macros instead of numeric values for mode. The macros are listed below:

- O_RDONLY
- O_RDWR
- O_WRONLY

EBCDIC versus ASCII

EBCDIC, which stands for Extended Binary-Coded-Decimal Interchange Code, is the code set used on z/OS UNIX. By contrast, ASCII is the dominant code set on almost all other platforms. Scripts written for ASCII platforms without an awareness of code set independent coding practices, can run incorrectly on z/OS UNIX. The following sections discuss some of these issues.

It can be helpful to visualize the difference between the code pages and so graphical representations of the EBCDIC (IBM-1047) and ASCII (ISO8859-1) code pages have been provided in Appendix A, “Code pages,” on page 223.

For a general overview of character sets and code pages, refer to *National Language Support Reference Manual, Volume 2*, SE09-8002. For additional information on running Perl in EBCDIC platforms, see "Considerations for running Perl on EBCDIC platforms" in `http://perldoc.perl.org/perlebcdic.html`

Converting hex values to characters in the CGI input stream

A very common code phrase in Perl CGI scripts is the following:

```
$value =~ s/%([a-fA-F0-9]{2})/pack('C',chr(hex($1)))/ge;
```

A very similar variant of this which is sometimes seen with an early unofficial port of Perl to OS/390 was (note the 'E' versus 'C' in the above):

```
$value =~ s/%([a-fA-F0-9]{2})/pack('E',chr(hex($1)))/ge;
```

Neither will (or should) work with Perl for z/OS and will need to be replaced with statements shown later in this section.

The intent of both of these statements is to convert characters which are represented in hex in the CGI data stream into the equivalent characters on the local platform. For example, consider the following URL:

```
http://myurl.com/cgi-bin/display.pl?page=this is a test
```

To insure that the spaces are properly enterpreted as part of the value of page, they are translated into the characters "%20". 0x20 is the hexidecimal value of the code point for a space in the ASCII code page. The resulting data passed to the `display.pl` script is:

```
id=this%20is%20a%20test
```

In order to be interpreted correctly, these hex characters need to be translated back into their character representations. The first statement shown above is a very common code phrase used to do this.

The problem with the above first statement is that it converts 0x20 to the EBCDIC character which corresponds to 0x20 which is not a space. The problem with the above second statement is that the 'E' option which returns the correct EBCDIC character, however, it is not supported by Perl.

For this translation code to work correctly, it needs to be replaced with one of the following:

method 1:

```
$value =~ s/%([a-fA-F0-9]{2})/chr(utf8::unicode_to_native(hex($1)))/ge;
```

or

method 2:

```
$value =~ s/%([a-fA-F0-9]{2})/pack('U', hex($1))/ge;
```

The difference between these two is that in the method 2, the UTF8-flag for the variable is set on. This should not be an issue for most users, however, if you are not certain, use method 1. Note that the `utf8::unicode_to_native` and `utf8::native_to_unicode` functions are undocumented but supported.

Sort order differences

The default sort order differs between EBCDIC and ASCII platforms. When using the Perl sort() function, or calling the unix sort utility, you should be aware of these differences.

The following table demonstrates how the sort order varies. The "Orig" column show a random order of characters. The following columns show the different sort orders generated by the Perl sort() function and the unix sort utility on the z/OS UNIX and Linux platforms:

	z/OS		Linux	
Orig	sort()	sort	sort()	sort
=				
A	:	0	0	=
0	=	9	9	:
Z	a	:	:	0
:	z	=	=	9
	A	A	A	a
z	Z	Z	Z	A
9	0	a	a	z
a	9	z	z	Z

Characters versus code points

One of the most common problems when porting scripts from other platforms to z/OS is caused by the technique of referring to characters using their codepoint (ordinal) value rather than their symbolic value. For example, consider the following code:

```
$x=ord('A');
if ($x == 0x41)
  { print "x is 'A'     \n"; }
 else
  { print "x is not 'A' \n"; }
```

The if statement would only be true on an ASCII platform where the ordinal value of the character `A` is 0x41. If this code were run on z/OS UNIX, it would print `x is not 'A'` because the ordinal value of 'A' on an EBCDIC codepage is `0xC1`.

Newline ("\n")

The ordinal value of the newline ("\n") character is different between EBCDIC and ASCII. So, the symbolic representation of this character should always be used rather than the ordinal values. The following are incorrect and correct examples:

```
print "Hello World \012";   # incorrect
print "Hello World \n";     # correct
```

The following table shows the ordinal values of the newline and carriage return characters on EBCDIC and ASCII:

	Dec	Octal	Hex
EBCDIC	21	025	0x15
ASCII	10	012	0xA

Newlines with IPC (Socket related functions)

A common misconception in socket programming is that \n eq \012 everywhere. When using protocols such as common Internet protocols, \012 and \015 are called for specifically, and the values of the logical \n and \r (carriage return) are not reliable. The following is an example:

```
print SOCKET "Hi there, client!\r\n"; # Incorrect
print SOCKET "Hi there, client!\015\012"; # Correct
```

However, using \015\012 (or \cM\cJ, or \x0D\x0A) can be tedious and unsightly, as well as confusing to those maintaining the code. As such, the Socket module supplies a more convenient and user friendly method.

```
use Socket qw(:DEFAULT :crlf);
print SOCKET "Hi there, client!$CRLF" # Correct
```

When reading from a socket, remember that the default input record separator $/ is \n , but robust socket code will recognize it as either \012 or \015\012, which is end of line:

```
while (<SOCKET>) {
# ...
}
```

Because both CRLF and LF end in LF, the input record separator can be set to LF, and any CR stripped later. It is more effective to write it as follows:

```
use Socket qw(:DEFAULT :crlf);
local($/) = LF; # not needed if $/ is already \012

while (<SOCKET>) {
s/$CR?$LF/\n/; # not sure if socket uses LF or CRLF, OK
# s/\015?\012/\n/; # same thing
}
```

This example is preferred over the previous one, even for UNIX platforms, because now any \015's (\cM's) are stripped out.

Similarly, functions that return text data, such as a function that fetches a Web page, should translate newlines before returning the data, especially if they have not yet been translated to the local newline representation. A single line of code will often suffice:

```
$data =~ s/\015?\012/\n/g;
return $data;
```

For further reference, visit Writing portable Perl.

pack and unpack functions

When using the **pack** and **unpack** functions, keep in mind that the length value is taken from the character code point. Because of this, the behavior of these functions will differ when used on an EBCDIC platform as compared to an ASCII or other character set platform.

The following is an example code fragment containing the **unpack** function:

```
my $b = "X\t01234567\n" x 100;
my @a = unpack("(a1 c/a)*", $b);
```

In the above code fragment, the **unpack** function is attempting to unpack *$b* into an array with a resulting size of 200 with each value alternating between "X" and "01234567\n"; for example, ($a[0] = "X"; $a[1] = "01234567\n";), where *a1* refers to the *X* value (a one character long string item) and *c/a* refers to "01234567\n". However, this does not occur as expected on the EBCDIC system. The reason is in how "\" works in the **unpack** function. The value before the "\" is the length-item and the value after is the string-item. The length-item dictates the number of string-items to retrieve. In this case, the length-item is "\t" (tab) which on an ASCII system has the ordinal value of 9, indicating that a nine character long string item should to be retrieved next. (The ()* part indicates that this should be done repetitively until the end of *$b* is encountered.) However, on the EBCDIC system, the ordinal value for \t may not be 9. On some EBCDIC systems it is 5. Because of

this discrepancy in code points, the results of running the **pack** and **unpack** functions will differ if you are running in on EBCDIC platform as compared to an ASCII or other platform. Because the z/OS environment is EBCDIC, you will need to be aware of this when using the **pack** and **unpack** functions in this environment.

Non-contiguous character ranges

EBCDIC does not have the alphabet in a contiguous manner - there are gaps. For example, the character '>>' that lies between 'i' and 'j' will not be translated by the `tr/i-j/X`. This works in the other direction too, if either of the range endpoints is numeric, `tr/\x89-\x91/X` will translate `\x8b`, even though '\x89' is 'i' and '\x91' is 'j' and '\x8b' are gapped characters from the alphabetic viewpoint. Thus, knowledge of these gapped characters on the z/OS platform is required to write scripts.

Chapter 5. Perl command reference

Synopsis

perl [switches] [--] [programfile] [arguments]

Description

Perl, which stands for Practical Extraction and Report Language, is a general purpose, open source scripting language. Its powerful text-manipulation functions allow file maintenance and automation of system administration. It is also popular for CGI programming on the web owing to its inherent ability to process text. It is highly adaptable and intended to be easy to use and efficient.

For more information about Perl including links to additional reference sources, refer to the CPAN (Comprehensive Perl Archive Network) Web site:

`http://cpan.org`

Options

Perl accepts the following command line switches:

-0[octal]
This option specifies the record separator as an octal (\0 if no *octal* is present).

-a This option turns on autosplit mode when used with **-n** or **-p** . It splits $_ into the @F array, and inserts it as the first command inside the while loop created by **-n** or **-p**.

-C[number/list]
This option enables the listed Unicode features.

-c This option specifies "check syntax only mode" (runs BEGIN and CHECK blocks).

-d[:debugger]
This option specifies the program to run with the debugger.

-D[number/list]
This option specifies debugging flags (argument is a bit mask or alphabet characters).

-e program
This option enables a single line of the program to be entered (multiple **-e**'s are allowed, omit programfile).

-F/pattern/
This option specifies a *pattern* to split() on if the **-a** switch is also in effect. (//'s are optional).

-i[extension]
This option specifies that files processed by <> are to be edited in place. It will create a backup of the old file if an *extension* is supplied, by adding the *extension* to the name of the old file.

-Idirectory
This option specifies that the *directory* following **-I** are to be prefixed to @INC/#include (several **-I**'s are allowed).

-l[octal]
This option enables line ending processing.

-[mM][-]module
This option executes **use***module* before executing the program. If the first character after the **-M** or **-m** is a minus (-), then the **use** is replaced with **no**.

-n This option creates a 'while (<>) { ... }' loop around the program.

-p This option creates a 'while (<>) { ... }' loop around the program, and also prints out each line of the loop iteration.

-P This option specifies the program to be run through the C preprocessor before compilation by **perl**.

-s This option enables rudimentary parsing for switches on the command line after the program name but before any filename arguments or the -- switch terminator.

-S This option specifies that **perl** use the PATH environment variable to search for the program (unless the name of the program starts with a slash).

-t This option specifies "taint" warnings to be turned on.

-T This option specifies "taint" checks to be turned on.

-u This option specifies **perl** to dump core after parsing the program.

-U This option enables **perl** to perform unsafe operations.

-v This option prints the version and patch level of the Perl program.

-V[:variable]
This option prints a summary of the major Perl configuration values (or a single Config.pm variable).

-W This option enables all warnings.

-X This option disables all warnings.

Chapter 6. Perl for z/OS messages

HPE20001 Ambiguous range in transliteration operator

Explanation: You wrote something like tr/a-z-0//, which doesn't mean anything. To include a - character in a transliteration, put it either first or last. (In the past, tr/a-z-0// was synonymous with tr/a-y//, which was probably not what you would have expected.)

Example: HPE20001 Ambiguous range in transliteration operator

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Look at the place where the transliterate operator is used in the regular expression and check the syntax.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20002 *arg* argument is not a HASH or ARRAY element

Explanation: The argument to exists() must be a hash or array element, such as:

```
$foo{$bar}
     $ref->{"susie"}
```

In the message text:

arg
argument

Example: HPE20002 foo argument is not a HASH or ARRAY element

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number shown in the error message and make sure that the argument to exists() is a hash or array element.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20003 delete argument is not a HASH or ARRAY element or slice

Explanation: The argument to delete() must be either a hash or array element, such as

```
$foo{$bar}
  $ref->{"susie"}
```

Example: HPE20003 delete argument is not a HASH or ARRAY element or slice

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number shown in the error message and check to make sure that the argument to delete() is a hash or array element.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20004 exists argument is not a subroutine name

Explanation: The exists() function expects a function name as argument, not the function call. `exists &sub()` will generate this error.

Example: HPE20004 exists argument is not a subroutine name

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number shown in the error message and check whether the exists function

argument is a function name and not the function call.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20005 '!' allowed only after types *types* in *function*

Explanation: The '!' is allowed in pack() or unpack() only after certain types. See perlfunc/pack.

In the message text:

types
List of types, usually: "sSiIlLxX"

function
The function name "pack" or "unpack"

Example: HPE20005 '!' allowed only after types sSiIlLxX in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that you are not using '!' other than sSiIlLxX characters in the template section of the pack or unpack function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20006 Attempt to *string1 string2* a restricted hash

Explanation: The failing code tried to get or set a key that is not in the current set of allowed keys of a restricted hash.

In the message text:

string1
string Example: access disallowed

string2
string Example Key 'name of key'

Example: HPE20006 Attempt to access disallowed key 'maths' in a restricted hash

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether the key name you are using is correct. Is it a key of the hash you are using?

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20007 Attempt to delete readonly key '*string*' from a restricted hash

Explanation: The failing code attempted to delete a key whose value has been declared read-only from a restricted hash.

In the message text:

string
string Locked

Example: HPE20007 Attempt to delete read-only key `'sub'` from a restricted hash

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that you are not trying to delete the hash element whose value has been declared read-only.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20008 Args must match #! line

Explanation: The setuid emulator requires that the arguments used to invoke **perl** match the arguments specified on the #! line.

Example: HPE20008 Args must match #! line

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Because some systems impose a one-argument limit on the #! line, try combining switches. For example, change `w -U` to `-wU`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20009 Arg too short for msgsnd

Explanation: msgsnd() requires a string whose length is greater than or equal to sizeof(long).

Example: HPE20009 Arg too short for msgsnd

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the size of the argument sent to the msgsnd() function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20010 Assignment to both a list and a scalar

Explanation: If you assign to a conditional operator, the second and third arguments must either both be scalars or both be lists. Otherwise, Perl won't know which context to supply to the right side.

Example: HPE20010 Assignment to both a list and a scalar

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number displayed in the error message and determine whether the second and third arguments of the conditional operator should be either scalar or list.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20011 'I' and '>' may not both be specified on command line

Explanation: Perl does its own command line redirection and found that STDIN was a pipe, and that an attempt was made to redirect STDIN using '<'. You can use only one STDIN stream at a time.

Example: HPE20011 'I' and '>' may not both be specified on command line

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use I and > together when you are running a Perl script in a VMS system.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20012 Bad arg length for *function*, is *num*, should be *num*

Explanation: You passed a buffer of the wrong size to one of msgctl(), semctl() or shmctl().

In the message text:

function
function name Example: "Socket::inet_ntoa".

num
A number.

Example: HPE20012 Bad arg length for Socket::inet_ntoa, is 4, should be 6

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The arg length for the function you have used in the script is not correct. Correct it as shown in the error. For Example

Bad arg length for Socket::inet_ntoa, is 4, should be 6

Here inet_ntoa take arg length as 6 but 4 has been passed which is wrong.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20013 Bad arg length for *function*, length is *num*, should be *num*

Explanation: You passed a buffer of the wrong size to one of msgctl(), semctl() or shmctl().

In the message text:

function
function name Example: "Socket::inet_ntoa".

num
A number.

Example: HPE20013 Bad arg length for Socket::inet_ntoa, length is 4, should be 6

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The arg length for the function you have used in the script is not correct. Correct it as shown in the error. For Example

Bad arg length for Socket::inet_ntoa, is 4, should be 6

Here inet_ntoa take arg length as 6 but 4 has been passed which is wrong.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20014 Bad arg length for *function*, length is *num*, should be at least *num*

Explanation: You passed a buffer that was the wrong size.

In the message text:

function
function name Example: "Socket::inet_ntoa".

num
A number.

Example: HPE20014 Bad arg length for Socket::inet_ntoa length is 4, should be at least 6

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the buffer size.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20015 Attempt to bless into a reference

Explanation: The CLASSNAME argument to the bless() operator is expected to be the name of the package to bless the resulting object into.

Example: HPE20015 Attempt to bless into a reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: A reference was supplied, as in the following example:

```
bless $self, $proto;
```

Instead, do the following to avoid blessing into the reference:

```
bless $self, ref($proto) "" $proto;
```

System programmer response: No System Programmer response is required.

Problem determination: If you want to bless into the stringified version of the reference supplied, you need to stringify it yourself, as in the following example:

```
bless $self, "$proto";
```

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20016 Bad evalled substitution pattern

Explanation: You've used the `/e` switch to evaluate the replacement for a substitution, but perl found a syntax error in the code to evaluate.

Example: HPE20016 Bad evalled substitution pattern

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the syntax for an unexpected right brace (}) and correct it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20017 Bad filehandle: *filehandle*

Explanation: A symbol was passed to something (file handling function) wanting a filehandle, but the symbol did not have a filehandle associated with it.

In the message text:

filehandle
filehandle

Example: HPE20017 Bad filehandle: FH

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure open() is being called. Also, make sure the function call is made in correct package.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20018 Bad index while coercing array into hash

Explanation: The index looked up in the hash found as the 0'th element of a pseudo-hash is not legal.

Example: HPE20018 Bad index while coercing array into hash

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Index values must be at 1 or greater. See perlref.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20019 BEGIN failed--compilation aborted

Explanation: An untrapped exception was raised while a BEGIN subroutine was being executed. Compilation stops immediately and the interpreter is exited.

Example: HPE20019 BEGIN failed--compilation aborted

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct any errors in the the BEGIN block.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20020 Callback called exit

Explanation: A subroutine invoked from an external package via call_sv() exited by calling exit.

Example: HPE20020 Callback called exit

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether call_sv() is exited by calling exit, before calling call_sv().

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20021 Cannot compress integer in pack

Explanation: An argument to pack("w",...) was too large to compress. The BER compressed integer format can only be used with positive integers, and you attempted to compress Infinity or a very large number (> 1e308). See perlfunc/pack.

Example: HPE20021 Cannot compress integer in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you want to compress the integer, it should either be a positive integer or it should be less than 1e308.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20022 Cannot compress negative numbers in pack

Explanation: An argument to pack("w",...) was negative. The BER compressed integer format can only be used with positive integers.

Example: HPE20022 Cannot compress negative numbers in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Before you can compress the integer, it should either be a positive integer or it should be less than 1e308.

System programmer response: No System Programmer response is required.

Problem determination: For more information, see perlfunc/pack.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20023 Can only compress unsigned integers in pack

Explanation: An argument to pack("w",...) was not an integer. The BER compressed integer format can only be used with positive integers, but something else was compressed. See perlfunc/pack.

Example: HPE20023 Can only compress unsigned integers in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the argument passed for pack. If the template that was used is w, then it should have an unsigned integer as the argument.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20024 Can't bless non-reference value

Explanation: Only hard references can be blessed. This is how Perl "enforces" encapsulation of objects.

Example: HPE20024 Can't bless non-reference value

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check if you are trying to bless a non-reference value. This is not allowed.

System programmer response: No System Programmer response is required.

Problem determination: See perlobj.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20025 Can't chdir to *dir*

Explanation: You called `perl -x/foo/bar`, but `/foo/bar` is not a directory that you can chdir to, possibly because it doesn't exist.

In the message text:

dir
directory

Example: HPE20025 Can't chdir to foo/test

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: While running the perl program, check to make sure that the directory being given exists.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20026 Can't coerce array into hash

Explanation: An array was used where a hash was expected, but the array has no information about how to map from keys to array indices

Example: HPE20026 Can't coerce array into hash

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You cannot deal hash as an array . You can do that only with arrays that have a hash reference at index 0.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20027 Can't coerce *string1* **to integer in** *string2*

Explanation: Certain types of SVs (scalar values) , in particular real symbol table entries (typeglobs), cannot be forced to stop being what they are.

In the message text:

string1
string example: string,ref-to-glob etc

string2
string example: integer

Example: HPE20027 Can't coerce ARRAY to integer in array deref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You can not change the behaviour of globs. Please don't try to force the variable to change in that case.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20028 Can't coerce *string1* to number in *string2*

Explanation: Certain types of SVs (scalar values) , in particular real symbol table entries (typeglobs), cannot be forced to stop being what they are.

In the message text:

string
string example: string,ref-to-glob etc

string2
string example: integer

Example: HPE20028 Can't coerce %s to number in %s

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not try to convert the string to a number during arthematic operation.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20029 Can't coerce *string1* to string in *string2*

Explanation: Certain types of SVs (scalar values) , such as real symbol table entries (typeglobs), cannot be forced to stop being what they are.

In the message text:

string1
string example: unknown

string2
string example: string

Example: HPE20029 Can not coerce unknown to string in lc.

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not try to convert the *string* to a string.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20030 Can't call method "*function*" on an undefined value

Explanation: You used the syntax of a method call, but the slot filled by the object reference or package name contains an undefined value.

In the message text:

function
function or module name

Example: HPE20030 Can't call method "process" on an undefined value

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check if the package you are using to call the function is undefined.

Something like this will reproduce the error:

```
$BADREF = undef;
process $BADREF 1,2,3;
$BADREF->process(1,2,3);
```

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20031 Can't call method "*function*" on unblessed reference

Explanation: A method call must know in what package it is supposed to run. It ordinarily finds this information from the object reference supplied, but object references were not supplied in this case. A reference is not an object reference until it has been blessed.

In the message text:

function
function or module name

Example: HPE20031 Can't call method "process" on unblessed reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure the object reference being used is blessed.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20032 Can't declare class for non-scalar *non-scalar* in "*AccessSpecifier*"

Explanation: Currently, only scalar variables can be declared with a specific class qualifier in a "my" or "our" declaration. The semantics may be extended for other types of variables in the future.

In the message text:

non-scalar
non scalar can be reference

AccessSpecifier
Access specifier

Example: HPE20032 Can't declare class for non-scalar \@socks in "my"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure you are not declaring aclass for non-scalar variable

The following example is not a scalar variable:

```
'package Cat; my Cat \@socks; here \@socks
```

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20033 Can't declare *non-scalar* in "*AccessSpecifier*"

Explanation: Only scalar, array, and hash variables can be declared as "my" or "our" variables. They must have ordinary identifiers as names.

In the message text:

non-scalar
non scalar can be dereference

AccessSpecifier
Access specifier

Example: HPE20033 Can not declare scalar dereference in "our".

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Be sure that only dereference variables are declared as "my" or "our".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20034 Can't do inplace edit without backup

Explanation: The operating system cannot read from a deleted (but still opened) file.

Example: HPE20034 Can't do inplace edit without backup

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure the file is backed up before editing. To back up the file, do `-i.bak`, or a related instruction.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20035 Can't do setuid

Explanation: This typically means that ordinary perl tried to exec suidperl to do setuid emulation, but couldn't exec it. It looks for a name of the form sperl5.000 in the same directory that the perl executable resides under the name perl5.000, typically `/usr/local/bin` on UNIX machines.

Example: HPE20035 Can't do setuid

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If the file is there, check the execute permissions. If it is not there, ask your system administrator why the file was removed.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20036 Unable to do waitpid with flags

Explanation: This machine does not have either waitpid() or wait4(), so only waitpid() without flags is emulated.

Example: HPE20036 Unable to do waitpid with flags

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that this machine has wait4() and waitpid().

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20037 Can't emulate -*opt* on #! line

Explanation: The #! line specifies a switch that does not make sense at this point.

In the message text:

opt
option given with #! like -x -w.

Example: HPE20037 Can't emulate -x on #! line

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Remove all incorrect *opt* from the #! line. An example of an incorrect *opt* is **-x**.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20038 Can't exec *filename*

Explanation: Perl was trying to execute the indicated program because that's what the #! line said. If that is not wanted, "perl" may need to be mentioned on the #! line somewhere.

In the message text:

filename
file name

Example: HPE20038 Can't exec test.pl

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure the right file name is specified.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20039 Can't find an opnumber for "*keyword***"**

Explanation: A string of a form C<CORE::word> was given to prototype(), but there is no builtin with the name C<word>.

In the message text:

keyword
keyword Example: continue,connect etc

Example: HPE20039 Can't find an opnumber for "continue"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check that the prototype of the built-in is correct and that there are no mispellings.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20040 Can't find label *label*

Explanation: A goto was attempted to a label that is not mentioned anywhere that is possible to goto. See perlfunc/goto.

In the message text:

label
The label

Example: HPE20040 Can not find label jumphere.

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Define the label before using it

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20041 Can't find string terminator *var str var* **anywhere before EOF**

Explanation: Perl strings can stretch over multiple lines.This message means that the closing delimiter was omitted. Because bracketed quotes count nesting levels, the following is missing its final parenthesis: print q(The character '(' starts a side comment.);

In the message text:

var
It is a string type char variable, it can be:'"') ? '\" : '"'.

str
The String.

Example: HPE20041 Can't find string terminator ")" anywhere before EOF

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Because white space may have been included before or after the closing tag, check the syntax.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20042 Can't fork

Explanation: A fatal error occurred while trying to fork while opening a pipeline.

Example: HPE20042 Can't fork

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20043 Can't "goto" into the middle of a foreach loop

Explanation: A "goto" statement was executed to jump into the middle of a foreach loop. Jumping into the middle of foreach loop is not permissable. See perlfunc/goto.

Example: HPE20043 Can't "goto" into the middle of a foreach loop

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use "goto" to jump into any construct that requires initialization, such as a subroutine or a foreach loop.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20044 Can't "goto" out of a pseudo block

Explanation: A "goto" statement was executed to jump out of what might look like a block, except that it isn't a proper block. This usually occurs if a "goto" leaves from a sort() block or subroutine, which is not allowed. See perlfunc/goto.

Example: HPE20044 Can't "goto" out of a pseudo block

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use any of the loop control operators to exit out of a sort block or subroutine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20045 Can't goto subroutine from an eval-string

Explanation: The "goto subroutine" call can't be used to jump out of an eval "string". (It can be used to jump out of an eval {BLOCK}, but this is not advisable.)

Example: HPE20045 Can't goto subroutine from an eval-string

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use "goto" to enter a construct that is optimized away, or to get out of a block, eval-string or subroutine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20046 Can't goto subroutine outside a subroutine

Explanation: The "goto subroutine" call can only replace one subroutine call for another. See perlfunc/goto.

Example: HPE20046 Can't goto subroutine outside a subroutine

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use "goto subroutine" to get out of a subroutine. Try calling it out of an AUTOLOAD routine instead. Do not use it to go into any construct that requires initialization, such as a subroutine or a foreach loop. Also, do not use it to go into a construct that is optimized away, or to get out of a block or subroutine given to sort.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20047 Can't ignore signal CHLD, forcing to default

Explanation: Perl detected that it is being run with the SIGCHLD signal (sometimes known as SIGCLD) disabled. Because disabling this signal will interfere with proper determination of exit status of child processes, Perl has reset the signal to its default value. This situation typically indicates that the parent program under which Perl may be running (for example, cron) is being very careless.

Example: HPE20047 Can't ignore signal CHLD, forcing to default

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Make sure that SIG_DFL is used when USE_SIGCHLD is defined.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20048 Can't "last" outside a loop block

Explanation: A "last" statement was executed to break out of the current block, while there is no current block. Note that an "if" or "else" block does not count as a "loopish" block, as well as a block given to sort(), map() or grep(). By doubling the curlies, the same effect can be achieved, because the inner curlies will be considered a block that loops once. See perlfunc/last.

Example: HPE20048 Can't "last" outside a loop block

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use "last" to exit a block that returns a value. Because a block by itself is semantically identical to a loop that executes once, you can use "last" to effect an early exit out of such a block.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20049 Can't localize lexical variable *var*

Explanation: Local was used on a variable name that was previously declared as a lexical variable using "my". This is not allowed.

In the message text:

var
The Variable.

Example: HPE20049 Can't localize lexical variable `op_targ`

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: To localize a package variable of the same name, qualify it with the package name.

System programmer response: No System

Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20050 Can't localize pseudo-hash element

Explanation: The following was found: `C<< local $ar->{'key'} >>` , where $ar is a reference to a pseudo-hash. That has not been implemented yet.

Example: HPE20050 Can't localize pseudo-hash element

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You can achieve a similar effect by localizing the corresponding array element directly, such as `C<< local $ar->[$ar->[0]{'key'}] >>` .

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20051 Can't locate *filename*

Explanation: An attempt to `do` (or `require`, or `use`) a file failed, because that file could not be found. Perl looks for the file in all the locations mentioned in @INC, unless the file name included the full path to the file. See perlfunc/require.

In the message text:

filename
The filename.

Example: HPE20051 Can't locate SvPV_nolen

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Try one of the following: Set the PERL5LIB or PERL5OPT environment variable to point to the extra library, check to see if the script needs to add the library name to @INC, or check the spelling of the file.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20052 Can't locate object method "*method***" via package "***pkg***"**

Explanation: A method was called correctly, and it correctly indicated a package functioning as a class, but that package doesn't define that particular method, nor does any of its base classes. See perlobj.

In the message text:

method
The method name.

pkg
The package name.

Example: HPE20052 Can't locate object method "hello" via package "foo"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether the method is defined in this package. If it is not, then use the correct package.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20054 Can't modify *var1* **in** *var2*

Explanation: The item indicated cannot be assigned to or changed, such as with an auto-increment.

In the message text:

var1
The variable can be a constant/array/hash.

var2
The variable is scalar.

Example: HPE20054 Can't modify do block in local

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You cannot change the value of a constant or built-in-value in scalar.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20055 Can't modify non-lvalue subroutine call

Explanation: Subroutines meant to be used in lvalue context should be declared as such. Lvalue subroutines are still experimental and the implementation may change in future versions of Perl. See Lvalue subroutines.

Example: HPE20055 Can't modify non-lvalue subroutine call

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You can return a modifiable value from a subroutine by declaring the subroutine to return an lvalue.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20056 Can't "next" outside a loop block

Explanation: A "next" statement was executed to reiterate the current block, but there are no current blocks. An "if" or "else" block does not count as a "loopish" block, neither does a block given to sort(), map() or grep(). You can usually double the curlies to get the same effect though, because the inner curlies will be considered a block that loops once. See perlfunc/next.

Example: HPE20056 Can't "next" outside a loop block

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use "next" to exit a block that returns a value. Because a block by itself is semantically identical to a loop that executes once, "next" will exit such a block early.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20060 Can't open perl script "*filename***":** *reason*

Explanation: The script you specified cannot be opened for the indicated reason.

In the message text:

filename
file name

reason
Strings which gives the explanation for not opening the file.

Example: HPE20060 Can't open perl script "menu": EDC5129I No such file or directory.

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you are debugging a script that uses #! and normally rely on the shell's $PATH search, use the -S option. Perl will do the search so you don't have to type the path or `` `which $scriptname` ``.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20061 Can't redefine active sort subroutine *function*

Explanation: Perl optimizes the internal handling of sort subroutines and keeps pointers into them. An attempt was made to redefine one such sort subroutine when it was currently active, which is not allowed.

In the message text:

function
function name

Example: HPE20061 Can't redefine active sort subroutine func

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you want to redefine the sort routine, you should write `sort { &func } @x` instead of `sort func @x .`

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20062 Can't "redo" outside a loop block

Explanation: A "redo" statement was executed to restart the current block, but there are no current blocks.

Example: HPE20062 Can't "redo" outside a loop block

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You can usually double the curly brackets to get the same effect because the inner curly brackets will be considered a block that loops once. Note that an "if" or "else" block doesn't count as a "loopish" block; neither are blocks given to sort(), map() or grep().

System programmer response: No System Programmer response is required.

Problem determination: See redo.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code:

Automation: Not applicable.

HPE20063 Can't return *illegalValue* from lvalue subroutine

Explanation: Perl detected an attempt to return illegal lvalues (such as temporary or readonly values) from a subroutine used as an lvalue. This is not allowed.

In the message text:

illegalValue
Illegal value (undef,temporary or readonly values)

Example: HPE20063 Can't return undef from lvalue subroutine

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not assign undef or readonly value to lvalue.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20064 Can't return outside a subroutine

Explanation: The return statement was executed in mainline code, that is, where there was no subroutine call to return out of.

Example: HPE20064 Can't return outside a subroutine

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not put the return statement outside the subroutine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20065 Can't take log of *num*

Explanation: For ordinary real numbers, you can't take the logarithm of a negative number or zero. Use the Math::Complex package that comes standard with Perl if you want to do that for the negative numbers.

In the message text:

num
number (negative number or zero)

Example: HPE20065 Can't take log of zero

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not take the log of a negative number or zero.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20066 Can't take sqrt of *num*

Explanation: For ordinary real numbers, you cannot take the square root of a negative number. Use the Math::Complex package that comes standard with Perl if you want to do that.

In the message text:

num
number (negative number)

Example: HPE20066 Can't take sqrt of -5

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not take the square root of a negative number.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20067 Can't undef active subroutine

Explanation: A routine that is currently running cannot be undefined. It can be redefined while it is running, though. You can even undefine the redefined subroutine while the old routine is running.

Example: HPE20067 Can't undef active subroutine

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether you are trying to undefine a routine that's currently running.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20068 Can't use bareword *word* **as** *DataType* **ref while "strict refs" in use**

Explanation: Only hard references are allowed by "strict refs". Symbolic references are not allowed.

In the message text:

word
Any word

DataType
Data type like ARRAY HASH etc.

Example: HPE20068 Can't use bareword ("results") as a HASH ref while "strict refs" in use

System action: Perl immediately stops interpreting the

file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you use use strict refs, be careful about the syntax while using the reference.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20069 Can't use %! because Errno.pm is not available

Explanation: The first time the %! hash is used, perl automatically loads the Errno.pm module. The Errno module is expected to tie the %! hash to provide symbolic names for `$!` errno values.

Example: HPE20069 Can't use %! because Errno.pm is not available

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you are using %!, make sure you have Errno.pm in your system.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20070 Can't use *data-type* for loop variable

Explanation: Only a simple scalar variable may be used as a loop variable on a foreach.

In the message text:

data-type
An invalid data-type.

Example: HPE20070 Can't use Non-scalar for loop variable

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use only a simple scalar variable as a loop variable on a foreach.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20071 Can't use global *magicVar* in "my"

Explanation: An attempt was made to declare a magical variable as a lexical variable. This is not allowed. The magic can be tied to only one location (namely the global variable), and it would be confusing to have variables that looked like magical variables but were not.

In the message text:

magicVar
Magical variable.

Example: HPE20071 Can't use global $1 in "my"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure magical variables are not declared as lexical variables.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20072 Can't use "my *var*" in sort comparison

Explanation: The global variables $a and $b are reserved for sort comparisons. You mentioned $a or $b in the same line as the <=> or cmp operator, and the variable had earlier been declared as a lexical variable.

In the message text:

var
variable name

Example: HPE20072 Can't use "my $a" in sort comparison

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Either qualify the sort variable with the package name, or rename the lexical variable.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20073 Can't weaken a nonreference

Explanation: You tried to weaken something that was not a reference.

Example: HPE20073 Can't weaken a nonreference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You can weaken references only, not non-references.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20074 Code missing after '/' in unpack

Explanation: You had a (sub-)template that ends with a '/'.

Example: HPE20074 Code missing after '/' in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: There must be another template code following the slash. See perlfunc/unpack.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20075 Code missing after '/' in pack

Explanation: You had a (sub-)template that ends with a '/'.

Example: HPE20075 Code missing after '/' in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: There must be another template code following the slash. See perlfunc/pack.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20076 *class* **Compilation failed in require**

Explanation: Compilation failed due to failing of BEGIN,CHECK or any other class.

In the message text:

class
class example BEGIN,CHECK etc

Example: HPE20076 BEGIN failed--Compilation failed in require

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check for any syntax errors or any

normal errors in classes included during the build process (including BEGIN,CHECK).

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20077 Constant(*str1*):*str2str3str4*

Explanation: The parser found inconsistencies either while attempting to define an overloaded constant, or when trying to find the character name specified in the `<\N{...}` escape. Maybe the corresponding `overload` or `charnames` pragma was not loaded.

In the message text:

str1
str1 is constant type.

st2
The string str 2 may be "$^H{" or "Call to &{$^H{" .

str3
str3 may be "(possibly a missing \"use charnames ...\")" or " ".

str4
str4 may be "} is not defined" or "}} did not return a defined value".

Example: HPE20077 Constant(overload):$^H{ } is not defined

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Load the correct pragma.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20078 Constant is not *const* reference

Explanation: A constant value (perhaps declared using the `use constant` pragma) is being dereferenced, but it amounts to the wrong type of reference. The message indicates the type of reference that was expected. This usually indicates a syntax error in dereferencing the constant value.

In the message text:

const
const name

Example: HPE20078 Constant is not PI reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the syntax error in deferencing the constant value. You may be trying to dereference the constant variable.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20079 Copy method did not return a reference

Explanation: The method which overloads "=" is buggy.

Example: HPE20079 Copy method did not return a reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: There may be a problem with overloading "=". Check whether you used the correct way to overload "=".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20080 CORE::*keyword* is not a keyword

Explanation: The CORE::namespace is reserved for Perl keywords.

In the message text:

keyword
reserve word for a language

Example: HPE20080 CORE::namespace is not a keyword

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You cannot use words that are reserved as keyword in the perl script. Check the error message , change the name for which you see the error, and then run the script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20081 Count after length/code in unpack

Explanation: You had an unpack template indicating a counted-length string, but you have also specified an explicit size for the string.

Example: HPE20081 Count after length/code in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not specify an explicit size for the string when using an unpack template that indicates a counted-length string.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20082 *arg* defines neither package nor VERSION--version check failed

Explanation: You referred to a particular module file (for example, "use Module 42") but that module file does not have either package declarations or a `$VERSION`.

In the message text:

arg
argument

Example: HPE20082 Argv defines neither package nor VERSION--version check failed

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the version you are using as the argument is defined in the respective package that you are providing with "use".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20083 Delimiter for here document is too long

Explanation: In a here document construct like `F00` , the label `F00` is too long for Perl to handle.

Example: HPE20083 Delimiter for here document is too long

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the label name is not too long for Perl to handle.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20084 DESTROY created new reference to dead object '*OBJ*'

Explanation: A DESTROY() method created a new reference to the object that is being destroyed. Perl is confused, and prefers to abort rather than create a dangling reference.

In the message text:

OBJ
oject name

Example: HPE20084 DESTROY created new reference to dead object 'CGI::Session'

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the line number mentioned in the error message, make sure the object used is active.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20085 *"filename"* did not return a true value

Explanation: A required (or used) file must return a true value to indicate that it compiled correctly and ran its initialization code correctly.

In the message text:

filename
script file name

Example: HPE20085 test.pm did not return a true value

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the files that you are giving with require ends with a "1;", so that they always return the true value.

System programmer response: No System Programmer response is required.

Problem determination: In perl for packages at the end it is always made to return true value.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20086 *arg* does not define $*module* ::VERSION--version check failed

Explanation: You specified a certain module (for example, "use Module 42") but that module did not define a `$VERSION`.

In the message text:

arg
Argument

module
module name

Example: HPE20086 Argv does not define $Module::VERSION--version check failed

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the $Module::VERSION variable is defined in the specified module. If it is, then you can "use" the version argument for the respective module.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20087 '/' does not take a repeat count in *function*

Explanation: You cannot put a repeat count of any kind right after the '/' code.

In the message text:

function
function name like pack or unpack here

Example: HPE20087 '/' does not take a repeat count in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the error line, check if you have used a repeat count of any kind right after the '/' code in either pack or unpack.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20088 entering effective uid failed

Explanation: While under the `use filetest` pragma, switching the real and effective UIDs failed.

Example: HPE20088 entering effective uid failed

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the error line, check if you are doing switching of real and effective UIDs properly.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20089 entering effective gid failed

Explanation: While under the `use filetest` pragma, switching the real and effective GIDs failed.

Example: HPE20089 entering effective gid failed

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the error line, check if you are doing switching of real and effective GIDs properly.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20090 Excessively long < > operator

Explanation: The contents of a < > operator cannot exceed the maximum size of a Perl identifier. If you're just trying to glob a long list of file names, try using the glob() operator, or put the file names into a variable and glob that.

Example: HPE20090 Excessively long < > operator

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Look to see if the perl identifier used in the error line is longer than Perl can handle. If you must use a long name, store it in a variable before using it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20091 exec? I'm not *that* kind of operating system

Explanation: The `exec` function is not implemented in MacPerl.

Example: HPE20091 exec? I'm not *that* kind of operating system

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You cannot use exec because it is not implemented in MacPerl.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20092 Execution of *filename* aborted due to compilation errors

Explanation: The final summary message when a Perl compilation fails.

In the message text:

filename
file name.

Example: HPE20092 Execution of test.pl aborted due to compilation errors

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Resolve the error messages, one by one, until you stop getting the error message.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20093 *class* failed--call queue aborted

Explanation: An untrapped exception was raised while executing a CHECK, INIT, or END subroutine. Processing of the remainder of the queue of such routines has been prematurely ended.

In the message text:

class
Class name

Example: HPE20093 END failed--call queue aborted

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Resolve the error messages, one by one, until you stop getting the error message.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20094 fcntl is not implemented

Explanation: The fcntl() function isn't available on your system.

Example: HPE20094 fcntl is not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check if fcntl has been implemented on your machine. Contact the system administrator if it hasn't.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20095 Final $ should be \$ or $name

Explanation: You have to decide whether the final $ in a string was meant to be a literal dollar sign, or was meant to introduce a variable name that happens to be missing. So you have to put either the backslash or the name.

Example: HPE20095 Final $ should be \$ or $name

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the error line, if you want to use $ as part of the string, you have to put the backslash (/) before $.

System programmer response: No System

Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20096 Format not terminated

Explanation: A Format must be terminated by a line with a solitary dot. Perl reached the end of your file without finding such a line.

Example: HPE20096 Format not terminated

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use solitary dots to end the Format in the script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20097 Global symbol "*name*" requires explicit package name

Explanation: You specified "use strict vars", which indicates that all variables must either be lexically scoped (using "my"), declared beforehand using "our", or explicitly qualified to say which package the global variable is in (using "::").

In the message text:

name
Symbol name

Example: HPE20097 Global symbol "$i" requires explicit package name

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you use Strict in the script, ensure that all the variables used in the script are lexical variables.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20098 Glob not terminated

Explanation: The lexer saw a left angle bracket in a place where it was expecting a term, so it's looking for the corresponding right angle bracket and not finding it. You may have omitted needed parentheses, and you really meant a "less than".

Example: HPE20098 Glob not terminated

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the glob function and check for matching right and left angle brackets.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20099 ()-group starts with a count in *function*

Explanation: A ()-group started with a count. A count is supposed to follow something: a template character or a ()-group.

In the message text:

function
function name like unpack or pack .

Example: HPE20099 ()-group starts with a count in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In pack or unpack, count should follow the template character or group.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20100 *filename* **had compilation errors.**

Explanation: The final summary message when a `perl -c` fails.

In the message text:

filename
File name

Example: HPE20100 test.pl had compilation errors.

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Because this is the final message, there are no actions that you can take. First run the script using perl -c which will list down the syntax errors and then debug the script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20101 *filename* **has too many errors**

Explanation: The parser stopped parsing the program after ten errors. Additional error messages would likely be uninformative.

In the message text:

filename
File name

Example: HPE20101 test.pl has too many errors

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the errors and continue.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20102 Illegal declaration of anonymous subroutine

Explanation: When using the `sub` keyword to construct an anonymous subroutine, you must always specify a block of code

Example: HPE20102 Illegal declaration of anonymous subroutine

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the Anonymous subroutine is followed by the block of code.

System programmer response: No System Programmer response is required.

Problem determination: See perlsub for further reference.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20103 Illegal declaration of subroutine *sub*

Explanation: A subroutine was not declared correctly. For more information, see perlsub.

In the message text:

sub
The subrotine name.

Example: HPE20103 Illegal declaration of subroutine foo

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the syntax of the declaring subroutine and make any corrections that are necessary.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20104 Illegal division by zero

Explanation: You tried to divide a number by 0. Either something was wrong in your logic, or you need to put a conditional in to guard against meaningless input.

Example: HPE20104 Illegal division by zero

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number mentioned in the error message and make sure that you are not dividing a number by 0; it is illegal.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20105 Illegal modulus zero

Explanation: You tried to divide a number by 0 to get the remainder.

Example: HPE20105 Illegal modulus zero

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number mentioned in the error message and check if you tried to divide a number by 0 to get the remainder.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20106 Illegal number of bits in vec

Explanation: The number of bits in vec() (the third argument) must be a power of two from 1 to 32 (or 64, if your platform supports that).

Example: HPE20106 Illegal number of bits in vec

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number mentioned in the error message and make sure that number of bits in vec() (the third argument) is a power of two from 1 to 32.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20107 In EBCDIC the v-string components cannot exceed 2147483647

Explanation: An error peculiar to EBCDIC. Internally, v-strings are stored as Unicode code points, and encoded in EBCDIC as UTF-EBCDIC. The UTF-EBCDIC encoding is limited to code points no larger than 2147483647 (0x7FFFFFFF).

Example: HPE20107 In EBCDIC the v-string components cannot exceed 2147483647

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number mentioned in the error message and make sure that the v-string size is below the limit in EBCDIC.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20108 Insecure directory in *env_var*

Explanation: You tried to do something that the tainting mechanism didn't like. The tainting mechanism is turned on when you're running setuid or setgid, or when you specify **-T** to turn it on explicitly. The tainting mechanism labels all data that's derived directly or indirectly from the user, who is considered to be unworthy of your trust. If any such data is used in a "dangerous" operation, you get this error.

In the message text:

env_var
environmental variable like $ENV{PATH}.

Example: HPE20108 Insecure directory in $ENV{PATH}.

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the directory is correct and secure.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20109 Invalid range "*BegChar-EndChar***" in transliteration operator**

Explanation: The range specified in the tr/// or y/// operator had a minimum character greater than the maximum character.

In the message text:

BegChar
First character in the specified range

EndChar
Last character in the specified range

Example: HPE20109 Invalid range "9-$" in transliteration operator

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The range you have specified for tr operator is not correct; make sure you mention the correct range. For reference, see perlop.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20110 Invalid separator character *code* **in attribute list**

Explanation: Something other than a colon or white space was seen between the elements of an attribute list. If the previous attribute had a parameter list in parentheses, perhaps that list was terminated too soon.

In the message text:

code
Codes like $,&

Example: HPE20110 Invalid separator character '$' in attribute list

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the attribute list has valid separator characters.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20111 ioctl is not implemented

Explanation: The ioctl() function is not supported.

Example: HPE20111 ioctl is not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In some machines, ioctl is not implemented. Contact your system administrator.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20112 IO layers (like '*layer*') unavailable

Explanation: Because your Perl has not been configured to support PerlIO, you cannot use IO layers.

In the message text:

layer
Perl IO layers

Example: HPE20112 IO layers (like 'APR') unavailable

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In order to use PerlIO, Perl must be configured with 'useperlio'. Contact your system administrator if you have questions.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20113 Label not found for "last *label*"

Explanation: You named a loop to break out of, but you are not currently in a loop of that name, not even if you count where you were called from.

In the message text:

label
Label name

Example: HPE20113 Label not found for "last LOOP"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the label name is in the script. Also see last.

System programmer response: No System Programmer response is required.

Problem determination: There should be a section of script by that name as shown in the following example:

```
LOOP:
          $a = foo{$a};
          .....
```

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20114 Label not found for "next *label*"

Explanation: You named a loop to continue, but you're not currently in a loop of that name, not even if you count where you were called from.

In the message text:

label
Label name

Example: HPE20114 Label not found for "next LOOP"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Be sure that the label name is the script. Also see perlfunc/next..

System programmer response: No System Programmer response is required.

Problem determination: There should be a section of script by that name as shown in the below example.

```
LOOP:
        $a = foo{$a};
        .....
```

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20115 Label not found for "redo *label*"

Explanation: You named a loop to restart, but you're not currently in a loop of that name, not even if you count where you were called from.

In the message text:

label
Label name

Example: HPE20115 Label not found for "redo LOOP"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the label name is in the script. Also see redo.

System programmer response: No System Programmer response is required.

Problem determination: There should be a section of script by that name as shown in the below example.

```
LOOP:
        $a = foo{$a};
        .....
```

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20116 length/code after end of string in unpack

Explanation: While unpacking, the string buffer was empty when an unpack length/code combination tried to obtain more data. This results in an undefined value for the length.

Example: HPE20116 length/code after end of string in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number mentioned in the error message and check for the unpack parameter. Also see perlfunc/pack .

System programmer response: No System Programmer response is required.

Problem determination: The length specified in the unpack function should be more than the string we use in the function.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20117 Malformed PERLLIB_PREFIX

Explanation: An error peculiar to OS/2. PERLLIB_PREFIX should be of the form prefix1;prefix2 or prefix1 prefix2 with nonempty prefix1 and prefix2. If C prefix1 is indeed a prefix of a built-in library search path, prefix2 is substituted.

Example: HPE20117 Malformed PERLLIB_PREFIX

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The error may appear if components are not found. Try to install the component.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20118 Malformed prototype for *function*:*str*

Explanation: You tried to use a function with a malformed prototype. The syntax of function prototypes is given a brief compile-time check for obvious errors like invalid characters. A more rigorous check is run when the function is called.

In the message text:

function
The function name

str
A string consisting argumnets to the function

Example: HPE20118 Malformed prototype for main::jk: $jk1,$jk2,$jk3

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the prototype of the function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20119 '%' may not be used in pack

Explanation: You can't pack a string by supplying a checksum, because the checksum process loses information, and you can't go the other way.

Example: HPE20119 '%' may not be used in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number mentioned in the error message and make sure that the pack or unpack function call is included. Also see perlfunc/unpack .

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20120 Missing argument to -*opt*

Explanation: The argument to the indicated command line switch must follow immediately after the switch, without intervening spaces.

In the message text:

opt
option here -c.

Example: HPE20120 Missing argument to -c

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check for the space in the command-line argument.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20121 Missing comma after first argument to *function* function

Explanation: While certain functions allow you to specify a filehandle or an "indirect object" before the argument list, this isn't one of them.

In the message text:

function
Function name.

Example: HPE20121 Missing comma after first argument to pack function

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that function arguments are separated by commas.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20122 Missing control char name in \c

Explanation: A string enclosed in double quotes ended with "\c", without the required control character name.

Example: HPE20122 Missing control char name in \c

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: \c is a control character. If you want to use \c , then you need escape it using \.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20123 Missing name in "my sub"

Explanation: The reserved syntax for lexically-scoped subroutines requires that they have a name with which they can be found.

Example: HPE20123 Missing name in "my sub"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Assign a name, followed by "my sub", to the lexically-scoped subroutines.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20124 Missing $ on loop variable

Explanation: Variables are always mentioned with the $ in Perl, unlike the shells, where it can vary from one line to the next.

Example: HPE20124 Missing $ on loop variable

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number mentioned in the error message and add the $ to the variable.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20125 Missing right curly or square bracket

Explanation: The lexer counted more opening braces ({) or brackets ([} than closing ones. As a general rule, you will find the missing one near the place you were last editing.

Example: HPE20125 Missing right curly or square bracket

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number mentioned in the error message and insert the missing closing brace (}) or bracket (]).

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20126 Module name must be constant

Explanation: Only a bare module name is allowed as the first argument to a "use".

Example: HPE20126 Module name must be constant

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Only package name to be given after "use" extension of the package should not be specified.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20127 Module name required with -opt option

Explanation: The `-M` or `-m` options say that Perl should load some module, but you omitted the name of the module.

In the message text:

opt
option like -m

Example: HPE20127 Module name required with -m option

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Consult perlrun for full details about `-M` and `-m`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20128 More than one argument to open

Explanation: The `open` function has been asked to open multiple files. This can happen if you are trying to open a pipe to a command that takes a list of arguments, but have forgotten to specify a piped open mode.

Example: HPE20128 More than one argument to open

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number listed in the error message where you have given more arguments in the open function . Also see perlfunc/open for details.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20129 '/' must be followed by 'a*', 'A*' or 'Z*' in pack

Explanation: You had a pack template indicating a counted-length string. Currently the only items that can have their length counted are a*, A* or Z*.

Example: HPE20129 '/' must be followed by 'a*', 'A*' or 'Z*' in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether '/' is followed by 'a*', 'A*' or 'Z*' in the pack function, if it does not, you need to add it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20130 '/' must follow a numeric type in unpack

Explanation: You had an unpack template that contained a '/', but this did not follow some unpack specification producing a numeric value. See perlfunc/pack.

Example: HPE20130 '/' must follow a numeric type in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the line number mentioned in the error message and check whether the unpack syntax is proper; that is, '/' must follow a numeric type in unpack.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20131 "my sub" not yet implemented

Explanation: Lexically-scoped subroutines have not yet been implemented.

Example: HPE20131 "my sub" not yet implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You cannot localize the subroutine because lexically-scoped subroutines have not yet been implemented.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20132 Negative '/' count in unpack

Explanation: The length count obtained from a length/code unpack operation was negative.

Example: HPE20132 Negative '/' count in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check in the error line for the unpack function whether the length count is negative. Also see perlfunc/pack .

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20133 Negative length

Explanation: You tried to do a read/write/send/recv operation with a buffer length that is less than 0.

Example: HPE20133 Negative length

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If the error is while using the reading of file, then try to read the data whose size is within the limit.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20134 Negative offset to vec in lvalue context

Explanation: When `vec` is called in an lvalue context, the second argument must be greater than or equal to zero.

Example: HPE20134 Negative offset to vec in lvalue context

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure, in lvalue context, that the second argument is greater than or equal to zero.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20135 No *opt* allowed while running setuid

Explanation: Certain operations are deemed to be too insecure for a setuid or setgid script to even be allowed to attempt. Generally speaking, there will be another way to do what you want that is, if not secure, at least securable.

In the message text:

opt
option Like -P

Example: HPE20135 No -P allowed while running setuid

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: When you are using setuid, do not use certain options such as -P. For more details, see Perl security.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20136 No comma allowed after *filehandle*

Explanation: A list operator that has a filehandle or "indirect object" is not allowed to have a comma between that and the following arguments. Otherwise it would just be another one of the arguments. One possible cause for this is that you expected to have imported a constant to your name space with **use** or **import**. Howerver, the import did not take place. It may, for example, be that your operating system does not support that particular constant.

In the message text:

filehandle
file handles

Example: HPE20136 No comma allowed after filehandle

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: A comma is not allowed after filehandle or indirect object, for example

```
print FOOBAR, "This text is color FOOBAR\n";
```

The above print statement is incorrect; you need to take off the comma after FOOBAR.

System programmer response: No System Programmer response is required.

Problem determination: While an explicit import list would probably have caught this error earlier, it naturally does not remedy the fact that your operating system still does not support that constant. Maybe you have a typo in the constants of the symbol import list of **use** or **import** or in the constant name at the line where this error was triggered.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20138 No DB::DB routine defined

Explanation: The currently executing code was compiled with the **-d** switch, but for some reason the current debugger (for example, **perl5db.pl** or a `Devel::module` did not define a routine to be called at the beginning of each statement.

Example: HPE20138 No DB::DB routine defined

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check with the system administrator whether perl5db.pl is available on your machine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20139 No DBsub routine

Explanation: The executing code was compiled with the **d** switch, but for some reason the current debugger (for example, **perl5db.pl** or a `Devel::module`) didn't define a `DB::sub` routine to be called at the beginning. of each ordinary subroutine call.

Example: HPE20139 No DBsub routine

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether the `DB::sub` routine to be called at the beginning of each ordinary subroutine call is defined.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20140 No group ending character '*code*' found in template

Explanation: A pack or unpack template has an opening '(' or '[' without its matching counterpart.

In the message text:

code
Character codes.

Example: HPE20140 No group ending character ')' found in template

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the error line, check the syntax of pack or unpack. A closing bracket may be missing.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20142 No #! line

Explanation: The setuid emulator requires that scripts have a well-formed #! line even on machines that do not support the #! construct.

Example: HPE20142 No #! line

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Some machines do not support the #! construct. For those machines, do not use #!.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20143 "no" not allowed in expression

Explanation: The "no" keyword is recognized and executed at compile time, and does not return a useful value.

Example: HPE20143 "no" not allowed in expression

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use keywords. For more information, refer to Perl modules.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20146 No package name allowed for variable *var* in "our"

Explanation: Fully qualified variable names are not allowed in "our" declarations. Such syntax is reserved for future extensions.

In the message text:

var
variable name

Example: HPE20146 No package name allowed for variable $foo in "our"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You must have given something like this our $foo::bar. That is not allowed. Because you cannot localize the variable of a package, remove "our" from the decleration.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20147 No Perl script found in input

Explanation: You called `perl -x`, but a line beginning with #! and containing the word "perl" was not found in the file.

Example: HPE20147 No Perl script found in input

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to see if the first line #! contains the word "perl".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20148 No setregid available

Explanation: Configure did not find anything resembling the setregid() call for your system.

Example: HPE20148 No setregid available

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that setregid() is set up on your system.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20149 No setreuid available

Explanation: Configure did not find anything resembling the setreuid() call for your system.

Example: HPE20149 No setreuid available

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that setreuid() is set up on your system.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20150 No such class *name*

Explanation: You provided a class qualifier in a "my" or "our" declaration, but this class does not exist at this point in your program.

In the message text:

name
class name.

Example: HPE20150 No such class MyClass

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the class name you are using in the error line is present.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20151 No such pseudo-hash field "*fieldname***"**

Explanation: You tried to access an array as a hash, but the field name used is not defined. The hash at index 0 should map all valid field names to array indices for that to work.

In the message text:

fieldname
pseudo hash field name.

Example: HPE20151 No such pseudo-hash field "maths"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You can use the array as a hash (in this case, it is called a 'pseudo hash'), but the hash index must be properly mapped.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20152 No such pseudo-hash field *str1* in variable *str2* of type *pkg*

Explanation: You tried to access a field of a typed variable where the type does not know about the field name. The field names are looked up in the %FIELDS hash in the type package at compile time. The %FIELDS hash is %usually set up with the 'fields' pragma.

In the message text:

str1
The str1 is the field name.

str2
The str2 is the variable.

pkg
The package name

Example: HPE20152 No such pseudo-hash field "notthere" in variable \$obj3 of type D3

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check that the field is present in the package and if it isn't, do not try to access it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20153 Not a CODE reference

Explanation: Perl was trying to evaluate a reference to a code value (that is, a subroutine), but found a reference to something else instead.

Example: HPE20153 Not a CODE reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You can use the ref() function to find out what kind of ref it really was. See also perlref .

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20154 Not a format reference

Explanation: I am not sure how you managed to generate a reference to an anonymous format, but this indicates you did, and that it did not exist.

Example: HPE20154 Not a format reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: A reference to anonymous format does not exist. Make sure you are using the reference for existing format.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20155 Not a GLOB reference

Explanation: Perl was trying to evaluate a reference to a "typeglob" (that is, a symbol table entry that looks like `*foo`), but found a reference to something else instead. You can use the ref() function to find out what kind of ref it really was.

Example: HPE20155 Not a GLOB reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: See perlref for more information on references and try to correct the error.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20156 Not a HASH reference

Explanation: Perl was trying to evaluate a reference to a hash value, but found a reference to something else instead. You can use the ref() function to find out what kind of ref it really was.

Example: HPE20156 Not a HASH reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You are trying to deference because you think that the reference is a hash reference, which it is not. Use ref() to find the reference type and then try using it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20157 Not an ARRAY reference

Explanation: Perl was trying to evaluate a reference to an array value, but found a reference to something else instead. You can use the ref() function to find out what kind of ref it really was.

Example: HPE20157 Not an ARRAY reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You are trying to deference thinking that reference is an array reference but it is something else. Use ref() to check the reference type and then try using it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20158 Not a perl script

Explanation: The setuid emulator requires that scripts have a well-formed #! line even on machines that don't support the #! construct. The line must mention perl.

Example: HPE20158 Not a perl script

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the #! construct is present and includes perl.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20159 Not a SCALAR reference

Explanation: Perl was trying to evaluate a reference to a scalar value, but found a reference to something else instead. You can use the ref() function to find out what kind of ref it really was.

Example: HPE20159 Not a SCALAR reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You are trying to deference because you think that the reference is scalar reference, which it is not. Use ref() to find the reference type and then try using it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20160 Not a subroutine reference

Explanation: Perl was trying to evaluate a reference to a code value (that is, a subroutine), but found a reference to something else instead. You can use the ref() function to find out what kind of ref it really was.

Example: HPE20160 Not a subroutine reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You are trying to deference thinking that reference is subroutine but it is something else. Check the reference type using ref() and then try using it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20161 Not enough arguments for *function*

Explanation: The function requires more arguments than you specified.

In the message text:

function
Function name

Example: HPE20161 Not enough arguments for Tree

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the function indicated by the error message has enough arguments.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20162 Null filename used

Explanation: You can't require the full file name, because on many machines that means the current directory.

Example: HPE20162 Null filename used

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the file name used with require is not a full file name. For more information, see perlfunc/require.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20163 Null picture in formline

Explanation: The first argument to formline must be a valid format picture specification. It was found to be empty, which probably means you specified an uninitialized value.

Example: HPE20163 Null picture in formline

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the syntax for formline. For more information, see perlform.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20164 Offset outside string

Explanation: You tried to do a read/write/send/recv operation with an offset pointing outside the buffer. The sole exception to this is that `sysread()`ing past the buffer will extend the buffer and zero pad the new area.

Example: HPE20164 Offset outside string

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the line number mentioned in the error message, cross-check for read/write/send/recv operation with buffer.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20165 Out of memory during "large" request for *num* bytes, total sbrk() is *num* bytes

Explanation: The malloc() function returned 0, indicating there was insufficient remaining memory (or virtual memory) to satisfy the request. However, the request was judged large enough (compile-time default is 64K), so a possibility to shut down by trapping this error is granted.

In the message text:

num
A number of bytes.

Example: HPE20165 Out of memory during "large" request for 134221824 bytes, total sbrk() is 64000 bytes

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: When you request for memory, make sure it does not exceed 64k bytes.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20166 Out of memory during request for *num* bytes, total sbrk() is *num* bytes

Explanation: The malloc() function returned 0, indicating there was insufficient remaining memory (or virtual memory) to satisfy the request.

In the message text:

num
A number of bytes.

Example: HPE20166 Out of memory during request for 1016 bytes, total sbrk() is 1000 bytes

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The request was judged to be small, so the possibility to trap it depends on the way perl was compiled. By default it is not trappable. However, if compiled for this, Perl may use the contents of `$^M` as an emergency pool after die()ing with this message. In this case the error is trappable once, and the error message will include the line and file where the failed request happened.

System programmer response: No System Programmer response is required.

Problem determination: Error message will include the line and file where the failed request happened based on this correct the allocation of memory.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20167 Out of memory during request for *num* bytes, total sbrk() is *num* bytes!

Explanation: The malloc() function returned 0, indicating there was insufficient remaining memory (or virtual memory) to satisfy the request.

In the message text:

num
A number of bytes.

Example: HPE20167 Out of memory during request for 1016 bytes, total sbrk() is 1000 bytes

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The request was judged to be small, so the possibility to trap it depends on the way perl was compiled. By default it is not trappable. However, if compiled for this, Perl may use the contents of `$^M` as an emergency pool after die()ing with this message. In this case the error is trappable I<once>, and the error message will include the line and file where the failed request happened.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20168 Out of memory during ridiculously large request

Explanation: You can't allocate more than 2^31+"small amount" bytes.

Example: HPE20168 Out of memory during ridiculously large request

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the line mentioned in the error message, check if there are any errors in the Perl program. For example, `$arr[time]` instead of `$arr[$time]`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20169 pack/unpack repeat count overflow

Explanation: You can't specify a repeat count so large that it overflows your signed integers.

Example: HPE20169 pack/unpack repeat count overflow

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the error line, change the repeat count to an acceptable number. See perlfunc/pack .

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20170 panic: kid popen errno read

Explanation: forked child returned an incomprehensible message about its errno. This is an internal error that cannot be resolved by the user.

Example: HPE20170 panic: kid popen errno read

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Go to the error line and determine whether forking is happening properly.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20171 Can't locate object method "*function*" via package "*package*"

Explanation: You called a method correctly, and it correctly indicated a package functioning as a class, but that package doesn't define that particular method, nor does any of its base classes.

In the message text:

function
The function name.

package
The package name.

Example: HPE20171 Can't locate object method "func" via package "packa"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check for the function definition in the particular package given in the error message.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20172 Permission denied

Explanation: The setuid emulator in suidperl decided you were up to no good.

Example: HPE20172 Permission denied

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the file permissions for the setuid emulator.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20173 'P' must have an explicit size in unpack

Explanation: The unpack format P must have an explicit size, not "*".

Example: HPE20173 'P' must have an explicit size in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: When you use P in unpack, you must specify the exact length. For more information, see perlfunc/unpack.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20174 -P not allowed for setuid/setgid script

Explanation: The script would have to be opened by the C preprocessor by name, which provides a race condition that breaks security.

Example: HPE20174 -P not allowed for setuid/setgid script

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the -P option if you are using setuid or setgid in script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20175 POSIX getpgrp cannot take an argument

Explanation: Your system has POSIX getpgrp(), which takes no argument, unlike the BSD version, which takes a pid.

Example: HPE20175 POSIX getpgrp can't take an argument

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use arguments with getpgrp().

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20176 Prototype not terminated

Explanation: You have omitted the closing parenthesis in a function prototype definition.

Example: HPE20176 Prototype not terminated

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the script to see whether you have omitted the closing parenthesis in a function prototype for any functions.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20177 Quantifier in {,} bigger than *num*

Explanation: There is currently a limit to the size of the min and max values of the {min,max} construct. The <-- HERE shows in the regular expression about where the problem was discovered.

In the message text:

num
number (here Quantifier max limit.)

Example: HPE20177 Quantifier in {,} bigger than 32766

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Quantifier (min and max) in {,} regular expression you have given is greater than limited value.

Try providing less than the maximum value.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20178 Quantifier follows nothing

Explanation: You started a regular expression with a quantifier. Backslash it if you meant it literally. The <-- HERE shows in the regular expression about where the problem was discovered. See perlref.

Example: HPE20178 Quantifier follows nothing

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you use Quantifier, then it cannot be empty. Check to see whether the regular expression that uses the quantifier is empty or not.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20179 Range iterator outside integer range

Explanation: One (or both) of the numeric arguments to the range operator ".." are outside the range which can be represented by integers internally. One possible workaround is to force Perl to use magical string increment by prepending "0" to your numbers.

Example: HPE20179 Range iterator outside integer range

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the arguments used for range operator is an integer.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20180 Reallocation too large: *datasize*

Explanation: You cannot allocate more than 64K on an MS-DOS machine.

In the message text:

datasize
data size.

Example: HPE20180 Reallocation too large: 700000

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you are using MS-DOS, you cannot allocate more than 64K. Try to reduce the size of data.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20181 Recompile perl with -DDEBUGGING to use -D switch (did you mean -d ?)

Explanation: You cannot use the **-D** ion unless the code to produce the desired output is compiled into Perl, which entails some overhead, which is why it is currently left out of your copy.

Example: HPE20181 Recompile perl with -DDEBUGGING to use -D switch (did you mean -d ?)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the -D option because it requires extra support.

System programmer response: No System

Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20182 Recursive inheritance detected in package '*package*'

Explanation: More than 100 levels of inheritance were used. Probably indicates an unintended loop in your inheritance hierarchy.

In the message text:

package
Package name.

Example: HPE20182 Recursive inheritance detected in package 'IN::PACKAGE"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure you have not used more than limited levels of inheritance in script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20183 Recursive inheritance detected while looking for method '*function*' in package '*package*'

Explanation: More than 100 levels of inheritance were encountered while invoking a method.

In the message text:

function
Function name

package
Package name

Example: HPE20183 Recursive inheritance detected while looking for method 'circle' in package 'drawing'

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Probably indicates an unintended loop in your inheritance hierarchy. Check the script to find out if there is any looping in your inheritance.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20184 Script is not setuid/setgid in suidperl

Explanation: The suidperl program was invoked on a script without a setuid or setgid bit set.

Example: HPE20184 Script is not setuid/setgid in suidperl

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether the setuid or setgid bit is set.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20185 Search pattern not terminated

Explanation: The lexer could not find the final delimiter of a // or m{} construct. Remember that bracketing delimiters count nesting level. Missing the leading `$` from a variable `$m` may cause this error.

Example: HPE20185 Search pattern not terminated

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response:

```
Example print "$X{What ho?}";
```

Here perl sees the ? and interprets it as the beginning of a pattern match operator, similar to /.

If you use ? then make sure you you escape it and use.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20186 select not implemented

Explanation: This machine does not implement the select() system call.

Example: HPE20186 select not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Machine may not have select call. Contact system administrator.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20187 Self-ties of arrays and hashes are not supported

Explanation: Self-ties of arrays and hashes are not supported in the current implementation.

Example: HPE20187 Self-ties of arrays and hashes are not supported

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Self-ties are of arrays and hashes are not supported in the current implementation, so you cannot use it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20188 Sequence (? incomplete

Explanation: A regular expression ended with an incomplete extension "(?". The <-- HERE shows in the regular expression about where the problem was discovered.

Example: HPE20188 Sequence (? incomplete

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Regular expression is ending with incomplete Sequence (? , which leads to the error. Therefore, make sure you end the sequence.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20189 Sequence (*code*...) not recognized

Explanation: You used a regular expression extension that does not make sense. The <-- HERE shows in the regular expression about where the problem was discovered.

In the message text:

code
codes(some character sequence in regular expression.)

Example: HPE20189 Sequence (?<;...) not recognized

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Regular expression has Sequence

((?<; ...) , which is not recognized. This leads to error, and therefore you can not use such sequence.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20190 Sequence (?#... not terminated

Explanation: A regular expression comment must be terminated by a closing parenthesis. Embedded parentheses are not allowed. The <-- HERE shows in the regular expression about where the problem was discovered.

Example: HPE20190 Sequence (?#... not terminated

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Regular expression in which Sequence (?#... not terminated, which leads to error. Make sure you terminate it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20191 Sequence (?*code*...) not implemented

Explanation: A proposed regular expression extension has the character reserved but has not yet been written. The <-- HERE shows in the regular expression about where the problem was discovered.

In the message text:

code
codes (some character sequence in regular expression.)

Example: HPE20191 Sequence (?<...) not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Sequence (?<...) not implemented, so you can not use it in a regular expression.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20192 Sequence (?{...}) not terminated or not {}-balanced

Explanation: If the contents of a (?{...}) clause contains braces, they must balance in order for Perl to properly detect the end of the clause.

Example: HPE20192 Sequence (?{...}) not terminated or not {}-balanced

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the regular expression please make sure the brackets are balanced properly.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20193 setegid() not implemented

Explanation: You tried to assign to $), and your operating system doesn't support the setegid() system call (or equivalent), or at least Configure didn't think so.

Example: HPE20193 setegid() not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether your system supports the setegid() call.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20194 seteuid() not implemented

Explanation: You tried to assign to < $> >, and your operating system doesn't support the seteuid() system call (or equivalent), or at least Configure didn't think so.

Example: HPE20194 seteuid() not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether your system supports the seteuid() call.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20195 setpgrp can't take arguments

Explanation: Your system has the setpgrp() from BSD 4.2, which takes no arguments, unlike POSIX setpgid(), which takes a process ID and process group ID.

Example: HPE20195 setpgrp can't take arguments

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the setpgrp function call at the error line does not pass any argument.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20196 setrgid() not implemented

Explanation: You tried to assign to value to $>, and your operating system doesn't support the setrgid() system call (or equivalent), or at least Configure didn't think so.

Example: HPE20196 setrgid() not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether your system supports setrgid().

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20197 setruid() not implemented

Explanation: You tried to assign to value $< , and your operating system doesn't support the setruid() system call (or equivalent), or at least Configure didn't think so.

Example: HPE20197 setruid() not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The version of Perl that you are using does not support setruid.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20198 Setuid/gid script is writable by world

Explanation: The setuid emulator won't run a script that is writable by the world, because the world might have written on it already.

Example: HPE20198 Setuid/gid script is writable by world

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The file name you mention with require should be within quotes.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20199 <> should be quotes

Explanation: You wrote `require file` when you should have written `require 'file'` .

Example: HPE20199 <> should be quotes

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The file name you mention with require should be within quotes.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20200 Your random numbers are not that random

Explanation: When trying to initialize the random seed for hashes, Perl could not get any randomness out of your system.

Example: HPE20200 Your random numbers are not that random

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Because Perl could not get any randomness out of your system, this indicates that something is wrong. Check the hash syntax.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20201 You need to quote *function*

Explanation: You assigned a bareword as a signal handler name. Unfortunately, you already have a subroutine of that name declared, which means that Perl 5 will try to call the subroutine when the assignment is executed, which is probably not what you want.

In the message text:

function
The function name Example: pack

Example: HPE20201 You need to quote "pack"

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If it is what you want, put an & in front.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code:

Automation: Not applicable.

HPE20202 YOU HAVEN'T DISABLED SET-ID SCRIPTS IN THE KERNEL YET! FIX YOUR KERNEL, OR PUT A C WRAPPER AROUND THIS SCRIPT!

Explanation: You may not be able to disable set-id scripts in the kernel because you may not have the sources to your kernel.

Example: HPE20202 YOU HAVEN'T DISABLED SET-ID SCRIPTS IN THE KERNEL YET!\n\ FIX YOUR KERNEL, OR PUT A C WRAPPER AROUND THIS SCRIPT!\n

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Either fix the kernel or put a C wrapper around the script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20203 'x' outside of string in unpack

Explanation: You had a pack template that specified a relative position after the end of the string being unpacked. See perlfunc/pack.

Example: HPE20203 'x' outside of string in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the template by providing the position inside the string.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20204 'X' outside of string in unpack

Explanation: You had a (un)pack template that specified a relative position before the beginning of the string being (un)packed. See perlfunc/pack.

Example: HPE20204 'X' outside of string in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the template by providing the position inside the string.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20205 '@' outside of string in unpack

Explanation: You had a template that specified an absolute position outside the string being unpacked. For more information, see perlfunc/unpack.

Example: HPE20205 '@' outside of string in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the template by providing the position inside the string.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20206 Within []-length '*' not allowed in *function*

Explanation: The count in the (un)pack template may be replaced by `[TEMPLATE]` only if `TEMPLATE` always matches the same amount of packed bytes that can be

determined from the template alone. This is not possible if it contains * -length.

In the message text:

function
The function will be pack or unpack.

Example: HPE20206 Within []-length '*' not allowed in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the template by removing * from the template.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20207 Within []-length '*code*' not allowed in *function*

Explanation: The count in the (un)pack template may be replaced by `[TEMPLATE]` only if `TEMPLATE` always matches the same amount of packed bytes that can be determined from the template alone. This is not possible if it contains any of the xphs @, /, U, u, w or a *-length.

In the message text:

function
The function will be pack or unpack.

code
The codes will be one of these @, /, U, u, w or a *.

Example: HPE20207 Within []-length '/' not allowed in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If the following codes are present in the template, remove them: @, /, U, u, w or a *-,

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20208 Wide character in *function*

Explanation: Perl encountered a wide character (greater than 255) when it wasn't expecting one. This warning is on by default for functions. For more information, see perlfunc/binmode and perlfunc/open.

In the message text:

function
The function name.

Example: HPE20208 Wide character in print

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: The easiest way to turn off the warning is to add the `:utf8` layer to the output, `binmode STDOUT, ':utf8'`. Another way is to add `no warnings 'utf8';` but this is not often suggested. In general, you should explicitly mark the filehandle with an encoding.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20209 Wide character

Explanation: Perl encountered a wide character (>255) when it wasn't expecting one. For more information, see perlfunc/binmode and perlfunc/open.

Example: HPE20209 Wide character

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: The easiest way to turn off this warning is to add the `:utf8` layer to the output, `binmode STDOUT, ':utf8'`. Another way is to add `no warnings 'utf8';` but this is not often suggested. In general, you should explicitly mark the filehandle with an encoding.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20210 Wide character in $/

Explanation: Perl encountered a wide character (greater than 255) when it wasn't expecting one. For more information, see perlfunc/binmode and perlfunc/open.

Example: HPE20210 Wide character in $/

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: The easiest way to turn off this warning is to add the `:utf8` layer to the output, `binmode STDOUT, ':utf8'`. Another way is to add `no warnings 'utf8';` but this is not often suggested. In general, you should explicitly mark the filehandle with an encoding.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20211 Warning: Use of "*op*" without parentheses is ambiguous

Explanation: You wrote a unary operator followed by something that looks like a binary operator that could also have been interpreted as a term or unary operator.

In the message text:

op operator.

Example: HPE20211 Use of "rand" without parentheses is ambiguous

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use parentheses as appropriate. For instance, if you know that the rand function has a default argument of 1.0, and you write rand + 5;you may think that you wrote the same thing as rand() + 5. However, you got rand(+5). Using parenthese would have prevented that situation.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20212 Warning: something's wrong

Explanation: You passed warn() an empty string (the equivalent of `warn ""`) or you called it with no args and $_ was empty

Example: HPE20212 Warning: something's wrong

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Provide the warn function with arguments.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20213 Version number must be constant number

Explanation: The attempt to translate a use Module n.n LIST statement into its equivalent BEGIN block found an internal inconsistency with the version number. This is an internal error that cannot be resolved by the user.

Example: HPE20213 Version number must be constant number

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20214 Variable "*var*" will not stay shared

Explanation: An inner (nested) named subroutine is referencing a lexical variable defined in an outer subroutine. When the inner subroutine is called, it will probably see the value of the outer subroutine's variable as it was before and during the *first* call to the outer subroutine. In this case, after the first call to the outer subroutine is complete, the inner and outer subroutines will no longer share a common value for the variable. In other words, the variable will no longer be shared.Furthermore, if the outer subroutine is anonymous and references a lexical variable outside itself, then the outer and inner subroutineswill never share the given variable.

In the message text:

var
The variable name.

Example: HPE20214 Variable "a" will not stay shared

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Make the inner subroutine anonymous, using the `sub {}` syntax. When inner anonymous subs that reference variables in outer subroutines are called or referenced, they are automatically rebound to the current values of such variables.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20215 Variable "*var*" may be unavailable

Explanation: An inner (nested) anonymous subroutine is inside a named subroutine, and outside that is another subroutine; and the anonymous (innermost) subroutine is referencing a lexical variable defined in the outermost subroutine.

In the message text:

var
The variable name.

Example: HPE20215 Variable "v" may be unavailable

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Make the middle subroutine anonymous, using the `sub {}` syntax. Perl has specific support for shared variables in nested anonymous subroutines; a named subroutine in between interferes with this feature.

System programmer response: No System Programmer response is required.

Problem determination: `sub outermost { my $a; sub middle { sub { $a } } }` If the anonymous subroutine is called or referenced (directly or indirectly) from the outermost subroutine, it will share the variable as you would expect. But if the anonymous subroutine is called or referenced when the outermost subroutine is not active, it will see the value of the shared variable as it was before and during the *first* call to the outermost subroutine, which is probably not what you want.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20216 Can't locate object method "*method*" via package "*pkg*" (perhaps you forgot to load "*pkg*"?)"

Explanation: You called a method correctly, and it correctly indicated a package functioning as a class, but that package doesn't define that particular method, and neither does any of its base classes. For more information, see perlobj.

In the message text:

method
The method name.

pkg
The package name.

Example: HPE20216 Can't locate object method

"hello" via package "foo" (perhaps you forgot to load "foo"?)"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether the method is defined in this package. If the method isn't defined, use the correct package.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20217 "*keyword*" variable *var* masks earlier declaration in same *str*

Explanation: A "my" or "our" variable has been redeclared in the current scope or statement, effectively eliminating all access to the previous instance.

In the message text:

k The Keyword will be my or our.

var
The variable name.

str
The string will be scope or statment.

Example: HPE20217 "my" variable ab masks earlier declaration in same scope

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check to be sure that the earlier variable will still exist until the end of the scope or until all closure referents to it are destroyed.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20218 Variable length lookbehind not implemented

Explanation: Lookbehind is allowed only for subexpressions whose length is fixed and known at compile time. For more information, see perlre.

Example: HPE20218 Variable length lookbehind not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use fixed length for lookbehind In the subexpression.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20219 Lookbehind longer than "*length*" not implemented

Explanation: There is a limit on the length of string that lookbehind can handle.

In the message text:

length
The maximum length of the string.

Example: HPE20219 Lookbehind longer than "255" not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the length of the string with the maximum limit.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

**HPE20220 Unknown switch condition (?(*str*

Explanation: The condition part of a (?(condition)if-clause"else-clause) construct is not known. The condition may be lookahead or lookbehind (the condition is true if the lookahead or lookbehind is true), a (?{...}) construct (the condition is true if the xph evaluates to a true value), or a number (the condition is true if the set of capturing parentheses named by the number matched).

In the message text:

str
The condition part (?(condition)if-clause"else-clause) of string.

Example: HPE20220 Unknown switch condition (?(x) in regex

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the regex, correct the (?(condition)if-clause"else-clause) construct with proper condition. For more information, see perlre.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20221 Unmatched (

Explanation: Unbackslashed parentheses must always be balanced in regular expressions. For more information, see perlre.

Example: HPE20221 Unmatched (

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Balance the regular expression with the correct number of parentheses.

System programmer response: No System Programmer response is required.

Problem determination: If you're a vi user, the % key is valuable for finding the matching parenthesis.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20222 Unmatched)

Explanation: Unbackslashed parentheses must always be balanced in regular expressions. For more information, see perlre.

Example: HPE20222 Unmatched)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Balance the regular expression with the correct number of parentheses.

System programmer response: No System Programmer response is required.

Problem determination: If you're a vi user, the % key is valuable for finding the matching parenthesis.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20223 Can't do {n,m} with n > m

Explanation: Minima must be less than or equal to maxima. For more information, see perlre.

Example: HPE20223 Can't do {n,m} with n > m

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the regular expression with the minima and maxima properly defined.

System programmer response: No System Programmer response is required.

Problem determination: If you want your regexp to match something 0 times, just put {0}.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20224 Regexp *+ operand could be empty

Explanation: The part of the regexp subject to either the * or + quantifier could match an empty string

Example: HPE20224 Regexp *+ operand could be empty

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the regexp.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20225 Nested quantifiers

Explanation: You can't quantify a quantifier without intervening parentheses. Things like ** or +* or ?* are illegal. Note that the minimal matching quantifiers, C<*?>, C<+?>, and C appear to be nested quantifiers, but aren't. For more information, see perlre.

Example: HPE20225 Nested quantifiers

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Put parentheses between the quantifiers.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20226 Unmatched [

Explanation: The brackets around a character class must match. For more information, see perlre.

Example: HPE20226 Unmatched [

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the regex by matching the brackets. If you want to include a closing bracket in a character class, backslash it or put it first.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20227 Internal urp

Explanation: Something went awry in the regular expression parser. This is an internal error occurred that cannot be resolved by the user.

Example: HPE20227 Internal urp

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: Trappable errors may be trapped using the eval operator

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20228 Missing right brace on *char*{}

Explanation: Missing right brace in `\p{...}` or `\P{...}`.

In the message text:

char
The char will be p or P.

Example: HPE20228 Missing right brace on \p{}

System action: Perl immediately stops interpreting the

file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the missing right brace.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20229 Missing right brace on \\x{}

Explanation: A right brace is missing in the \\x{}.

Example: HPE20229 Missing right brace on \\x{}

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the right brace.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20230 /*str*/ should probably be written as "*str*"

Explanation: You used a pattern where Perl expected to find a string, as in the first argument to `join`. Perl will treat the true or false result of matching the pattern against $_ as the string, which may not be what you had in mind.

In the message text:

str
Any string.

Example: HPE20230 /hello/ should probably be written as "hello"

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Provide the argument as string instead of a pattern.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20231 sort is now a reserved word

Explanation: Before sort was a keyword, people sometimes used it as a filehandle.

Example: HPE20231 sort is now a reserved word

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you are using sort as a filehandle or something else, change it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20232 Sort subroutine didn't return a numeric value

Explanation: A sort comparison routine must return a number. For more information, see perlfunc/sort.

Example: HPE20232 Sort subroutine didn't return a numeric value

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use <=> or `cmp` in sort to return a numetric value.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20233 Sort subroutine didn't return single value

Explanation: A sort comparison subroutine cannot return a list value with more or less than one element. For more information, see perlfunc/sort.

Example: HPE20233 Sort subroutine didn't return single value

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the proper number of elements.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20234 Substitution pattern not terminated

Explanation: The lexer couldn't find the interior delimiter of an s/// or s{}{} construct. Remember that bracketing delimiters count nesting level. Missing the leading $ from variable `$s` may cause this error

Example: HPE20234 Substitution pattern not terminated

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the substitution with a interior delimiter, so that the pattern is properly terminated.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20235 Substitution replacement not terminated

Explanation: The lexer couldn't find the final delimiter of an s/// or s{}{} construct. Remember that bracketing delimiters count nesting level. Missing the leading $ from variable `$s` may cause this error.

Example: HPE20235 Substitution replacement not terminated

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the substitution with a final delimiter, so that the pattern is properly terminated.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20236 Switch (?(condition)... contains too many branches

Explanation: A (?(condition)if-clause"else-clause) construct can have at most two branches (the if-clause and the else-clause). For more information, see perlre.

Example: HPE20236 Switch (?(condition)... contains too many branches

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you want one or both to contain alternation, such as using `this"that"other`, enclose it in clustering parentheses:
(?(condition)(?:this"that"other)"else-clause)

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20237 Switch condition not recognized

Explanation: If the argument to the (?(...)if-clause"else-clause) construct is a number, it can be only a number. For more information, see perlre.

Example: HPE20237 Switch condition not recognized

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the construct by providing a number as a argument.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20238 switching effective uid is not implemented

Explanation: While under the `use filetest` pragma, we cannot switch the real and effective UIDs or GIDs.

Example: HPE20238 switching effective uid is not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not attempt to switch the effective UID.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20239 switching effective gid is not implemented

Explanation: While under the `use filetest` pragma, we cannot switch the real and effective UIDs or GIDs.

Example: HPE20239 switching effective gid is not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not attempt to switch the effective GID.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20241 *filename* **syntax OK**

Explanation: The syntax of this particular file is fine. It is displayed when you use it as perl -c filename

Example: HPE20241 test.pl syntax OK

System action: It will compile the perl file and check for syntax.

Operator response: No System Operator response is required.

User response: If you see this message, you can assume that the syntax of the file is correct. You can go ahead and execute it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code:

Automation: Not applicable.

HPE20242 syntax error

Explanation: Probably means you had a syntax error. Common reasons include: keyword is misspelled, semicolon is missing, comma is missing, an opening or closing parenthesis is missing, an opening or closing

brace is missing, a closing quote is missing.

Example: HPE20242 syntax error

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the syntax and correct it.

System programmer response: No System Programmer response is required.

Problem determination: The only way to figure out what's triggering the error is to call `perl -c` repeatedly, deleting half the program each time to see if the error goes away.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20243 System V IPC is not implemented on this machine

Explanation: The System V interprocess communication mechanisms like semaphores, message queues and shared memory, are not implemented on your machine.

Example: HPE20243 System V IPC is not implemented on this machine

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use functions related to System V IPC.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20244 System V msgxxx is not implemented on this machine

Explanation: System V msgrcv, msgctl and other functions are not implemented on your machine.

Example: HPE20244 System V msgxxx is not implemented on this machine

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the message queue functions provided by System V.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20245 System V semxxx is not implemented on this machine

Explanation: System V semget, semctl and other functions are not implemented on your machine.

Example: HPE20245 System V semxxx is not implemented on this machine

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the semaphore functions provided by System V.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20246 -T and -B not implemented on filehandles

Explanation: Perl can't peek at the stdio buffer of filehandles when it doesn't know about your kind of stdio. You'll have to use a filename instead.

Example: HPE20246 -T and -B not implemented on filehandles

System action: Perl immediately stops interpreting the

file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use -T and -B on filename instead of filehandle

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20247 Target of goto is too deeply nested

Explanation: You tried to use `goto` to reach a label that was too deeply nested for Perl to reach.

Example: HPE20247 Target of goto is too deeply nested

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Avoid using a deeply nested goto, so that Perl can easily resolve it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20248 That use of $[is unsupported

Explanation: Assignment to `$[` is now strictly circumscribed, and interpreted as a compiler directive.This is to prevent the problem of one module changing the array base out from under another module inadvertently. For more information, see perlvar/$[.

Example: HPE20248 That use of $[is unsupported

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You can say only one of $[= 0; $[= 1; ... local $[= 0; local $[= 1; ...

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20249 The crypt() function is unimplemented due to excessive paranoia.

Explanation: Configure couldn't find the crypt() function on your machine, probably because your vendor didn't supply it.

Example: HPE20249 The crypt() function is unimplemented due to excessive paranoia.

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the crypt() function if it is not implemented on your machine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20250 The stat preceding -l _ wasn't an lstat

Explanation: It does not make sense to test the current stat buffer for symbolic linkhood if the last stat that wrote to the stat buffer already went past the symlink to get to the real file.

Example: HPE20250 The stat preceding -l _ wasn't an lstat

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use an actual file name instead.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20251 The stat preceding lstat() wasn't an lstat

Explanation: It does not make sense to test the current stat buffer for symbolic linkhood if the last stat that wrote to the stat buffer already went past the symlink to get to the real file.

Example: HPE20251 The stat preceding lstat() wasn't an lstat

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use an actual file name instead.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20252 The 'unique' attribute may only be applied to 'our' variables

Explanation: Currently this attribute is not supported on `my` or `sub` declarations. For more information, see perlfunc/our.

Example: HPE20252 The 'unique' attribute may only be applied to 'our' variables

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the 'unique' attribute on `my` and `sub`. Use it only on `our`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20254 thread failed to start:*str*

Explanation: The entry point function of threads->create() failed for some reason

In the message text:

str
The str explains the reason, as why the thread failed to start.

Example: HPE20254 thread failed to start: Usage: Thread::Signal::_threadpid()

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use this particular function in the format as specified in the Usage part of the error.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20255 times not implemented

Explanation: Your version of the C library apparently doesn't do times()

Example: HPE20255 times not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Don't use the times function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20256 Too deeply nested ()-groups in *function*

Explanation: Your pack or unpack template contains ()-groups with a very deep nesting level.

In the message text:

function
The function will be pack or unpack.

Example: HPE20256 Too deeply nested ()-groups in %s

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the template with proper nesting.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20257 Too few args to syscall

Explanation: There has to be at least one argument to syscall() to specify the system call to call.

Example: HPE20257 Too few args to syscall

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the proper number of arguments to syscall().

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20258 Too many args to syscal

Explanation: Perl supports a maximum of 14 arguments to syscall().

Example: HPE20258 Too many args to syscal

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Limit the number of arguments to 14 to syscall().

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20259 Too many arguments for *function*

Explanation: The function requires fewer arguments than you specified.

In the message text:

function
The function name.

Example: HPE20259 Too many arguments for pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the limit of arguments for the function and provide only that number of arguments.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20260 Transliteration pattern not terminated

Explanation: The lexer couldn't find the interior delimiter of a tr/// or tr[][] or y/// or y[][] construct.

Example: HPE20260 Transliteration pattern not terminated

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the tranliteration by providing correct interior delimiter.

System programmer response: No System Programmer response is required.

Problem determination: Missing the leading $ from variables `$tr` or `$y` may cause this error.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20261 Transliteration replacement not terminated

Explanation: The lexer couldn't find the final delimiter of a tr///, tr[][], y/// or y[][] construct.

Example: HPE20261 Transliteration replacement not terminated

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the transliteration by providing the final delimiter.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20262 '*descriptor*' trapped by operation mask

Explanation: You tried to use an operator from a Safe compartment in which it's disallowed. For more information, see Safe.

In the message text:

descriptor
opcode descriptor.

Example: HPE20262 'pushmark' trapped by operation mask

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Avoid using an operator in the Safe Compartment.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20263 truncate not implemented

Explanation: Your machine doesn't implement a file truncation mechanism that Configure knows about.

Example: HPE20263 truncate not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the truncate function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20264 Type of arg *num* to *function* must be *var* (not *descriptor*)

Explanation: This function requires the argument in that position to be of a certain type. Arrays must be @NAME or `@{EXPR}`. Hashes must be %NAME or `%{EXPR}`. For more information, see perlref.

In the message text:

num
Argument Position

function
Any function which takes array or hash as argument.

var
A var will be a hash or an array.

descriptor
Opcode Descriptor

Example: HPE20264 Type of arg 1 to push must be array (not scalar dereference)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: No implicit dereferencing is allowed--use the {EXPR} forms as an explicit dereference.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20265 umask not implemented

Explanation: Your machine doesn't implement the umask function and you tried to use it to restrict permissions for yourself (EXPR & 0700).

Example: HPE20265 umask not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the umask function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20266 Unable to create sub named "*sub*"

Explanation: You attempted to create or access a subroutine with an illegal name.

In the message text:

sub
The subroutine name.

Example: HPE20266 Unable to create sub named "$foo"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the name of the subroutine and correct it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20267 Undefined format "*str*"called

Explanation: The format indicated doesn't seem to exist. For more information, see perlform.

In the message text:

str
The illegal format.

Example: HPE20267 Undefined format "STDOUT" called

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the format with the correct field definitions.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20268 Undefined sort subroutine "*sub*" called

Explanation: The sort comparison routine specified doesn't seem to exist. For more information, see perlfunc/sort.

In the message text:

sub
A Subroutine

Example: HPE20268 Undefined sort subroutine "main::langcmp" called

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the way that the sort subroutine was defined.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20269 Undefined subroutine &"*sub*" called

Explanation: The subroutine indicated hasn't been defined, or if it was, it has since been undefined

In the message text:

sub
The subroutine name.

Example: HPE20269 Undefined subroutine &main::max called

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the definition for the subroutine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20270 Undefined subroutine called

Explanation: The anonymous subroutine you're trying to call hasn't been defined, or if it was, it has since been undefined.

Example: HPE20270 Undefined subroutine called

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the definition for the subroutine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20271 Undefined subroutine in sort

Explanation: The sort comparison routine specified is declared but doesn't seem to have been defined yet. For more information, see perlfunc/sort.

Example: HPE20271 Undefined subroutine in sort

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the definition for the declared sort subroutine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20272 Undefined top format "*str*" called

Explanation: The format indicated doesn't seem to exist. For more information, see perlform.

In the message text:

str
The illegal top format.

Example: HPE20272 Undefined top format "STDOUT_TOP" called

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the top format.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20273 Unknown BYTEORDER

Explanation: There are no byte-swapping functions for a machine with this byte order.

Example: HPE20273 Unknown BYTEORDER

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Don't use byte orders other than Little Endian and Big Endian.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20274 Unknown open() mode '*mode*'

Explanation: The second argument of 3-argument open() is not among the list of valid modes: < , > , >> , +< , +> , +>> , -" , "- , <& , < >& .

In the message text:

mode
Invalid mode

Example: HPE20274 Unknown open() mode '<<-'

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the correct file opening modes as specified.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20275 Bad name after *sym str*

Explanation: You started to name a symbol by using a package prefix, and then didn't finish the symbol.

In the message text:

sym
The symbol name.

str
The str may be " ' " or "::" .

Example: HPE20275 Bad name after mypack::

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You cannot interpolate outside of quotes, so $var = 'myvar'; $sym = mypack::$var; is not the same as $var = 'myvar'; $sym = "mypack::$var"; If you want to name a symbol with a package prefix, it should be in " ".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20276 Bareword "*str*" not allowed while "strict subs" in use

Explanation: With "strict subs" in use, a bareword is only allowed as a subroutine identifier, in curly brackets or to the left of the "=>" symbol.

In the message text:

str
The string is bareword.

Example: HPE20276 Bareword "foo" not allowed while "strict subs" in use

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Avoid using bareword when using strict subs.You need to predeclare a subroutine if the bareword is subroutine name.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20277 Can't find Unicode property definition "*str*"

Explanation: The Unicode property you used is not among the list of Unicode properties.

In the message text:

str
The definition of unicode property.

Example: HPE20277 Can't find Unicode property definition "Ltter"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you did mean to use a Unicode property, refer to perlunicode for the list of known properties. If you didn't mean to use a Unicode property, escape the `\p`, either by `\\p` (just the `\p`) or by `\Q\p` (the rest of the string, until possible `\E`).

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20278 Can't locate object method "*method*" via package "*pkg*"

Explanation: You called a method correctly, and it correctly indicated a package functioning as a class, but that package doesn't define that particular method, nor does any of its base classes. For more information, see perlobj.

In the message text:

method
The method name.

pkg
The package name.

Example: HPE20278 Can't locate object method "hello" via package "foo"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether the method is defined in this package. If it is not, then use the correct package.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20279 Can't return array to lvalue scalar context

Explanation: You tried to return a complete array from an lvalue subroutine, but you called the subroutine in a way that made Perl think you meant to return only one value.

Example: HPE20279 Can't return array to lvalue scalar context

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You probably meant to write parentheses around the call to the subroutine, which tell Perl that the call should be in list context.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20280 Can't return hash to lvalue scalar context

Explanation: You tried to return a complete hash from an lvalue subroutine, but you called the subroutine in a way that made Perl think you meant to return only one value.

Example: HPE20280 Can't return hash to lvalue scalar context

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You probably meant to include parentheses around the call to the subroutine, which tells Perl that the call should be in list context.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20281 goto must have label

Explanation: Unlike with "next" or "last", you're not allowed to goto an unspecified destination. For more information, see perlfunc/goto.

Example: HPE20281 goto must have label

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the proper label for goto.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20282 Identifier too long

Explanation: Perl limits identifiers (names of things such as variables and functions) to about 250 characters for simple names, and somewhat more for compound names (like `$A::B`). You've exceeded Perl's limits.

Example: HPE20282 Identifier too long

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check whether your identifier is more than 250 characters. If it is, then reduce the number of characters so that it meets the limits.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20283 Illegal binary digit '*num*'

Explanation: You used a digit other than 0 or 1 in a binary number.

In the message text:

num
It can be any other digit than 0 or 1.

Example: HPE20283 Illegal binary digit '2'

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: For binary numbers, use 0 or 1.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20284 Illegal binary digit '*num*' ignored

Explanation: You may have tried to use a digit other than 0 or 1 in a binary number. Interpretation of the binary number stopped before the offending digit.

In the message text:

num
It may be any other digit than 0 or 1.

Example: HPE20284 Illegal binary digit '3' ignored

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: For binary numbers, use 0 or 1.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20285 Illegal character *char*(carriage return)

Explanation: Perl normally treats carriage returns in the program text as it would any other white space, which means you should never see this error when Perl was built using standard options.

In the message text:

char
The char is octal value of carriage return.

Example: HPE20285 Illegal character \015(carriage return)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Your version of Perl appears to have been built without this support. Talk to your Perl administrator.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20286 Illegal octal digit '*digit*' ignored

Explanation: You may have tried to use an 8 or 9 in an octal number. Interpretation of the octal number stopped before the 8 or 9.

In the message text:

digit
The digit will be 8 or 9.

Example: HPE20286 Illegal octal digit '8' ignored

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use use 8 or 9 digit in an octal number

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20287 Illegal octal digit '*digit*'

Explanation: You used an 8 or 9 in an octal number.

In the message text:

digit
The digit will be 8 or 9.

Example: HPE20287 Illegal octal digit '9'

System action: Perl immediately stops interpreting the

file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use a 8 or 9 digit in an octal number.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20288 Invalid type '*char*' in *function*

Explanation: The given character is not a valid pack or unpack type. For more information, see perlfunc/pack.

In the message text:

char
Any invalid character.

function
The function will be pack or unpack.

Example: HPE20288 Invalid type 'z' in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use characters that are not valid when using the pack or unpack function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20289 Invalid type ',' in *function*

Explanation: The ',' is not a valid pack or unpack type but used to be ignored.

In the message text:

funtion
The function will be pack or unpack.

Example: HPE20289 Invalid type ',' in pack.

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Remove the ',' in the pack or unpack function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20290 Invalid type '*char*' in unpack

Explanation: The given character is not a valid unpack type. For more information, see perlfunc/pack.

In the message text:

char
Any invalid character.

Example: HPE20290 Invalid type 'k' in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use a valid character for the unpack type.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20291 Invalid type '*char*' in pack

Explanation: The given character is not a valid pack type. For more information, see perlfunc/pack.

In the message text:

character
Any invalid character.

Example: HPE20291 Invalid type 'm' in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use a valid character in pack.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20292 BEGIN not safe after errors--compilation aborted

Explanation: Perl found a `BEGIN {}` subroutine (or a `use` directive, which implies a `BEGIN {}`) after one or more compilation errors had already occurred. Since the intended environment for the `BEGIN {}` could not be guaranteed (due to the errors), and since subsequent code likely depends on its correct operation, Perl stopped compiling.

Example: HPE20292 BEGIN not safe after errors--compilation aborted

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Remove the compilation errors and then run the script again.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20293 Socket::*function* not implemented on this architecture

Explanation: This socket function is not available on your machine.

In the message text:

function
The function will be pack_sockaddr_un or unpack_sockaddr_un.

Example: HPE20293 Socket::pack_sockaddr_un not implemented on this architecture

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the socket function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20294 leaving effective uid failed

Explanation: While under the `use filetest` pragma, switching the real and effective UIDs failed.

Example: HPE20294 leaving effective uid failed

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not switch the real and effective UIDs.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20295 leaving effective gid failed

Explanation: While under the `use filetest` pragma, switching the real and effective GIDs failed.

Example: HPE20295 leaving effective gid failed

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not switch real and effective GIDs.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20296 Malformed integer in [] in *function*

Explanation: Only digits are permitted between the brackets enclosing a numeric repeat count. For more information, see perlfunc/pack.

Example: HPE20296 Malformed integer in [] in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use only numeric count.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20298 Number too long

Explanation: Perl limits the representation of decimal numbers in programs to about 250 characters. You've exceeded that length.

Example: HPE20298 Number too long

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Try using scientific notation: "`1e6`" instead of "`1_000_000`".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20299 Operation `*str1***': no method found,***str2* **argument** *str3str4str5str6*

Explanation: An attempt was made to perform an overloaded operation for which no handler was defined. While some handlers can be autogenerated in terms of other handlers, there is no default handler for any operation, unless the `fallback` overloading key is specified to be true. For more information, see overload.

In the message text:

str1
The operation.

str2
The str2 may be " " or "left ".

str3
The str3 may be "in overloaded package" or "has no overloaded magic".

str4
The str4 is empty or package name.

str5
The str5 may be " " or 'right argument has no overloaded magic magic" or "right argument in overloaded package".

str6
The str6 is empty or package name.

Example: HPE20299 Operation `cmp': no method found, left argument in overloaded package Graph::Edge, right argument in overloaded package Graph::Edge

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Define the handler for the overloaded function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20300 Empty *char*{}

Explanation: \p and \P are used to introduce a named Unicode property. You used \p or \P in a regular expression without specifying the property name.

In the message text:

char
The character p or P.

Example: HPE20300 Empty \p{}

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide a Unicode property in the regex like \p{Letter}. Do not keep it empty.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20301 Reference to nonexistent group

Explanation: You used something like \7 in your regular expression, but there are not at least seven sets of capturing parentheses in the expression. If you wanted to have the character with value 7 inserted into the regular expression, prepend a zero to make the number at least two digits: \07.

Example: HPE20301 Reference to nonexistent group

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the regular expression with the existing group.

System programmer response: No System Programmer response is required.

Problem determination: Trappable errors may be trapped using the eval operator.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20302 Internal disaster

Explanation: Something went wrong in the regular expression parser.This is an internal error that cannot be resolved by the user.

Example: HPE20302 Internal disaster

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Because Perl is not able to determine the regular expression, you should verify the regular expression and rewrite it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20303 Variable "*var*" is not imported

Explanation: While "use strict" was in effect, you referred to a global variable that you apparently thought was imported from another module, because something else of the same name (usually a subroutine) is exported by that module. It usually means you put the wrong funny character on the front of your variable.

In the message text:

var
The variable name.

Example: HPE20303 Variable "@a" is not imported

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use unique variable names. To prevent ambuguity errors, do not use duplicate variable names in the same module even if you are importing from another module.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20304 (Did you mean &*sub* instead?)

Explanation: You probably referred to an imported subroutine &FOO as $FOO or some such.

In the message text:

sub
The subroutine name.

Example: HPE20304 (Did you mean $FOO instead?)

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use & when you are referring to a subroutine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20305 Had to create *sym* unexpectedly

Explanation: A routine asked for a symbol from a symbol table; however, the symbol was not defined in the table so it had to be created on an emergency basis to prevent a core dump.

In the message text:

sym
The symbol.

Example: HPE20305 Had to create len unexpectedly

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Define the symbol.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20307 Value of *construct* can be "0"; test with defined()

Explanation: In a conditional expression, you used `HANDLE`, <*> (glob),`each()`, or `readdir()` as a boolean value. Each of these constructs can return a value of "0", which would make the conditional expression false. That is probably not what you intended.

In the message text:

construct
The Construct.

Example: HPE20307 Value of `HANDLE` can be "0"; test with defined()

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: When using these constructs in conditional expressions, test their values with the `defined` operator.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20309 Eval-group in insecure regular expression

Explanation: Perl detected tainted data when trying to compile a regular expression that contains the `(?{ ... })` zero-width assertion, which is not safe. See perlre and perlsec for more information.

Example: HPE20309 Eval-group in insecure regular expression

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use a secure regular expression.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20310 Junk on end of regexp

Explanation: The regular expression parser is confused.This is an internal error that cannot be resolved by the user.

Example: HPE20310 Junk on end of regexp

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Delete any junk data that may be included with the regular expression.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20311 Eval-group not allowed at runtime, use re 'eval'

Explanation: Perl tried to compile a regular expression containing the `(?{ ... })` zero-width assertion at run time, as it would when the pattern contains interpolated values.

Example: HPE20311 Eval-group not allowed at runtime, use re 'eval'

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use zero-width assertion, even if you have to use another pattern like eval().

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20312 UTF-16 surrogate 0x*unicode*

Explanation: You tried to generate half of an UTF-16 surrogate by requesting a Unicode character between the code points 0xD800 and 0xDFFF (inclusive). That range is reserved exclusively for the use of UTF-16 encoding (by having two 16-bit UCS-2 characters); but Perl encodes its characters in UTF-8, so what you got is an illegal character.

In the message text:

unicode
The Unicode character

Example: HPE20312 UTF-16 surrogate 0xDFFE.

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If you know what you are doing, you can turn off the warning by `no warnings 'utf8';`. You can also use the Unicode character in the range.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20313 Malformed UTF-16 surrogate

Explanation: Perl thought it was reading UTF-16 encoded character data but encountered a malformed Unicode surrogate.

Example: HPE20313 Malformed UTF-16 surrogate

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Provide the correct UTF-16 character.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20314 SWASHNEW didn't return an HV ref

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20314 SWASHNEW didn't return an HV ref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20315 SWASHGET didn't return result of proper length

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20315 SWASHGET didn't return result of proper length

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20316 panic: swash_fetch

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20316 panic: swash_fetch

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20317 panic: utf16_to_utf8: odd bytelen *num*

Explanation: Something tried to call utf16_to_utf8 with an odd (as opposed to even) byte length. This is an internal error that cannot be resolved by the user.

In the message text:

num
The length

Example: HPE20317 panic: utf16_to_utf8: odd bytelen 3.

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use even byte lengths.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20318 Unicode character 0x*unicode* is illegal

Explanation: Certain Unicode characters have been designated off-limits by the Unicode standard and should not be generated. If you know what you are doing, you can turn off this warning by `no warnings 'utf8';` .

In the message text:

unicode
The unicode character.

Example: HPE20318 Unicode character 0xDFFF is illegal

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use characters that are allowed by the Unicode standard.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20319 Using an array as a reference is deprecated

Explanation: You tried to use an array as a reference, as in `<< @foo->[23] >>` or `<< @$ref->[99] >>`. Versions of perl <= 5.6.1 used to allow this syntax, but shouldn't have. It is now deprecated, and will be removed in a future version.

Example: HPE20319 Using an array as a reference is deprecated

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use arrays as a reference. Instead, use arrays such as @foo [19].

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: D- A deprecation (optional).

Automation: Not applicable.

HPE20320 Using a hash as a reference is deprecated

Explanation: You tried to use a hash as a reference, as in `<< %foo->{"bar"} >>` or `<< %$ref->{"hello"} >>`. Versions of perl <= 5.6.1 used to allow this syntax, but shouldn't have. It is now deprecated, and will be removed in a future version.

Example: HPE20320 Using a hash as a reference is deprecated

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use hash as a reference. Instead, you can use hash as %foo{"bar"}.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: D- A deprecation (optional).

Automation: Not applicable.

HPE20321 NOT IMPL LINE *num*

Explanation: The target line was not implemented.

In the message text:

num
Line number

Example: HPE20321 NOT IMPL LINE %d

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Implement the target line.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code:

Automation: Not applicable.

HPE20322 Possible precedence problem on bitwise *operator* operator

Explanation: Your program uses a bitwise logical operator in conjunction with a numeric comparison operator, like this : if ($x & $y == 0) { ... } This expression is actually equivalent to `$x & ($y == 0)`, due to the higher precedence of `==`. This is probably not what you want.

In the message text:

operator
The operator

Example: HPE20322 Possible precedence problem on bitwise & operator

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If you really meant to write, disable the warning, or, better yet, put the parentheses explicitly and write `$x & ($y == 0))`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20323 Useless use of *function* with no values

Explanation: You used the push() or unshift() function with no arguments apart from the array, like `push(@x)` or `unshift(@foo)`. That won't usually have any effect on the array, so is completely useless.

In the message text:

function
The function

Example: HPE20323 Useless use of push() with no values

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: It is possible that push(@tied_array) could have some effect if the array is tied to a class that implements a PUSH method. If so, you can write it as `push(@tied_array,())` to avoid this warning.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20324 defined(@array) is deprecated

Explanation: defined() is not usually useful on arrays because it checks for an undefined scalar value.

Example: HPE20324 defined(@array) is deprecated

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use defined() with arrays. If you want to see if the array is empty, use `if (@array) { # not empty }` for example.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: D- A deprecation (optional).

Automation: Not applicable.

HPE20325 (Maybe you should just omit the defined()?)

Explanation: defined() is not always useful because it checks for an undefined scalar value.

Example: HPE20325 (Maybe you should just omit the defined()?)

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not not use defined(). If you want to obtain results similar to what defined() provides, use another function, especially in the case of array or hash.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20326 defined(%hash) is deprecated

Explanation: defined() is not usually useful on hashes because it checks for an undefined scalar value.

Example: HPE20326 defined(%hash) is deprecated

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use defined() in the case of hash. If you want to see if the hash is empty, just use `if (%hash) { # not empty }` for example.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: D- A deprecation (optional).

Automation: Not applicable.

HPE20327 panic: unexpected lvalue entersub args: type/targ *optype*:*oparg*

Explanation: This is an internal error that cannot be resolved by the user.

In the message text:

optype
op type

oparg
op argument

Example: HPE20327 panic: unexpected lvalue entersub args: type/targ READ:fd

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20328 panic: unexpected lvalue entersub entry via type/targ *optype*:*oparg*

Explanation: This is an internal error that cannot be resolved by the user.

In the message text:

optype
op type

oparg
op argument

Example: HPE20328 panic: unexpected lvalue entersub entry via type/targ READ:fd

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20329 Unexpected constant lvalue entersub entry via type/targ *optype*:*oparg*

Explanation: The lvalue entered is not correct.

In the message text:

optype
op type

oparg
op argument

Example: HPE20329 Unexpected constant lvalue entersub entry via type/targ READ:fd

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the syntax while using the lvalue with sub.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20330 internal error: obsolete function save_hints() called

Explanation: Perl tried to compile a regular expression containing the `(?{ ... })` zero-width assertion at run time, as it would when the pattern contains interpolated values. Because that is a security risk, it is not allowed. See perlre for more information.

Example: HPE20330 internal error: obsolete function save_hints() called

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: It is not permitted, but if you still want to do this, you can explicitly build the pattern from an interpolated string at run time and use that in an eval().

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20331 Can't define subroutine *function* (GV is unique)

Explanation: Each function should have a unique name.

In the message text:

function
function name

Example: HPE20331 Can't define subroutine func (GV is unique)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that all function names are unique.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20332 Can't redefine unique subroutine *function*

Explanation: Every function should have unique name.

In the message text:

function
function name

Example: HPE20332 Can't redefine unique subroutine Store.

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use a different name for the function, because you cannot reuse function names.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20333 panic: ck_grep

Explanation: The program failed an internal consistency check trying to compile a grep.This is an internal error that cannot be resolved by the user.

Example: HPE20333 panic: ck_grep

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20334 panic: ck_split

Explanation: Failed an internal consistency check trying to compile a split. This is an internal error that cannot be resolved by the user.

Example: HPE20334 panic: ck_split

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20335 Found = in conditional, should be ==

Explanation: You used = instead of == in conditional.

Example: HPE20335 Found = in conditional, should be ==

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: You said if ($foo = 123) when you meant if ($foo == 123) (or something like that).

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20336 Useless use of sort in scalar context

Explanation: You used sort in scalar context, as in : my $x = sort @y; This is not very useful, and perl currently optimizes this away.

Example: HPE20336 Useless use of sort in scalar context

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use sort on scalars.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20337 Useless use of *var* in void context

Explanation: You did something without a side effect in a context that does nothing with the return value, such as a statement that doesn't return a value from a block, or the left side of a scalar comma operator. Very often, this points to a failure of Perl to parse your program the way you thought it would. For example, you'd get this if you mixed up your C precedence with Python precedence and said $one, $two = 1, 2; when you meant to say ($one, $two) = (1, 2); Another common error is to use ordinary parentheses to construct a list reference when you should be using brackets or curlies. For example, if you say $array = (1,2); when you should have said $array = [1,2]; The brackets and curlies explicitly turn a list value into a scalar value, while parentheses do not. So when a parenthesized list is evaluated in a scalar context, the comma is treated like the comma operator in C; that is, the left argument is thrown away, which is not what you want. See perlre for more information. This warning will not be issued for numerical constants equal to 0 or 1 since they are often used in statements like 1 while sub_with_side_effects() ; String constants that would normally evaluate to 0 or 1 are warned about.

In the message text:

var
The variable.

Example: HPE20337 Useless use of %s in void context

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check the variable that was used.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20338 Applying *regex* to *var1* will act on scalar(*var2*)

Explanation: The pattern match (<CODE>//<\code>), substitution (<CODE>s///<\code>), and transliteration (<CODE>tr///<\code>) operators work on scalar values. If you apply one of them to an array or a hash, it will convert the array or hash to a scalar value -- the length of an array, or the population info of a hash -- and then work on that scalar value. This is probably not what you meant to do. See perlfunc/grep and perlfunc/map for alternatives.

In the message text:

regex
The Regular expression can be pattern match, substitution or transliteration.

var1
The variable can be array or hash.

var2
The variable can be scalar.

Example: HPE20338 Applying substitution to array will act on scalar(@array).

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check whether you are using (pattern match, substitution or transliteration) regular expression on scalar values. If you are, then do not use it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20339 Parentheses missing around "*var*" list

Explanation: You said something like my $foo, $bar = @_; when you meant my ($foo, $bar) = @_; Remember that "my", "our", and "local" bind tighter than comma.

In the message text:

var
The variable.

Example: HPE20339 Parentheses missing around "$foo" list .

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use parentheses when using list because the comma operator takes precedence.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20340 Bareword found in conditional

Explanation: The compiler found a bareword where it expected a conditional, which often indicates that an "" or && was parsed as part of the last argument of the previous construct; for example: open FOO "" die. It may also indicate a misspelled constant that has been interpreted as a bareword: use constant TYPO => 1; if (TYOP) { print "foo" }

Example: HPE20340 Bareword found in conditional

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check the expression before compiling. Use strict pragma to avoid such errors.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20341 Runaway prototype

Explanation: Subroutine complains if it isn't supplied with parameters matching some pre-specified template.

Example: HPE20341 Runaway prototype

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check the parameters to the subroutine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20342 Constant subroutine *sub* redefined

Explanation: You redefined a subroutine which had previously been eligible for inlining. See perlsub/"Constant Functions" for commentary and workarounds.

In the message text:

sub
The subroutine.

Example: HPE20342 Constant subroutine sub() redefined

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Previously compiled invocations of the function will still be using the old value of the function. If you need to be able to redefine the subroutine, you need to ensure that it isn't inlined, either by dropping the () prototype (which changes calling semantics, so beware) or by thwarting the inlining mechanism in some other way, such as sub not_inlined () { 23 if $]; }.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20343 Subroutine *sub* redefined

Explanation: You redefined a subroutine.

In the message text:

sub
The Subroutine.

Example: HPE20343 Subroutine sub() redefined

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: To suppress this warning, say

```
{
       no warnings 'redefine';
    eval "sub name { ... }";
     }
```

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20344 Use of -l on filehandle *filehandle*

Explanation: A filehandle represents an opened file, and when you opened the file it already went past any symlink you are presumably trying to look for. The operation returned *undef*.

In the message text:

filehandle
The filehandle.

Example: HPE20344 Use of -l on filehandle FOO

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use a filename instead.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20345 panic: sysopen with multiple args

Explanation: This is an internal error that cannot be resolved by the user.

Example: HPE20345 panic: sysopen with multiple args

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20346 More than one argument to '*mode*&' open

Explanation: The `open` function has been asked to open multiple files. This can happen if you are trying to open a pipe to a command that takes a list of arguments, but have forgotten to specify a piped open mode. See perlfun for more information.

In the message text:

mode
The mode.

Example: HPE20346 More than one argument to `<&' open

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the modes argument add the piped open mode.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20347 Unrecognized signal name "*signal*"

Explanation: You specified a signal name to the kill() function that was not recognized. Issue `kill -l` in your shell to see the valid signal names on your system.

In the message text:

signal
Signal name which is not recognized from the system.

Example: HPE20347 Unrecognized signal name "sigtra"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use a signal name that has already been defined.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20348 More than one argument to '>*mode*' open

Explanation: The `open` function has been asked to open multiple files. This can happen if you are trying to open a pipe to a command that takes a list of arguments, but have forgotten to specify a piped open mode. See perlfunc/open for more information.

Example: HPE20348 More than one argument to '>-' open

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the modes argument add the piped open mode.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20349 More than one argument to '<*mode*' open

Explanation: The `open` function has been asked to open multiple files. This can happen if you are trying to open a pipe to a command that takes a list of arguments, but have forgotten to specify a piped open mode. See perlfunc/open for more information.

Example: HPE20349 More than one argument to '<-' open

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In the modes argument add the piped open mode.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20350 *function* not implemented

Explanation: You tried to do something with a function beginning with "sem", "shm", or "msg" but System V IPC is not implemented in your machine.

In the message text:

`function`
The function.

Example: HPE20350 sem not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In some machines, the functionality may exist but not be configured, so configure the machine before processing. For help, consult your system support.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20351 No code specified for -e

Explanation: The -e option was specified, but there was no code for it to execute.

Example: HPE20351 No code specified for -e

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Since the -e option stands for execution, it should follow the expression to execute. Therefore, make sure you have provided an expression followed by -e option.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20352 No directory specified for -I

Explanation: Directory not mention for -I option.

Example: HPE20352 No directory specified for -I

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: When using the -I option, specify a directory.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20353 *package* **version** *vers1* **required--this is only version** *vers2*

Explanation: You are trying to include a package that is not allowed in the version of Perl you are using.

In the message text:

package
package name

vers1
Required version number

vers2
Current version number

Example: HPE20353 File::Spec version 0.08 required--this is only version 0.87

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use the version of Perl that allows the package you want to include.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20354 POSIX class [:*char_seq***:] unknown**

Explanation: The class in the character class [: :] syntax is unknown. See perlre for more information.

In the message text:

char_seq
The character sequence.

Example: HPE20354 POSIX class [: :] unknown

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The POSIX character classes do `not` have the `is` prefix the corresponding C interfaces have: in other words, it's `[[:print:]]`,not `isprint`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20355 POSIX syntax [*char_seq***] is reserved for future extensions**

Explanation: Within regular expression character classes ([]) the syntax beginning with "[%c" and ending with "%c]" is reserved for future extensions. See perlre for more information.

In the message text:

char_seq
The character sequence.

Example: HPE20355 POSIX syntax [%c %c] is reserved for future extensions .

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you need to represent those character sequences inside a regular expression character class, just quote the square brackets with the backslash: "\[%c" and "%c\]".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20356 POSIX syntax [*char_seq*] belongs inside character classes

Explanation: The character class constructs [: :], [= =], and [. .] go inside character classes, the [] are part of the construct, for example:/[012[:alpha:]345]/. Note that [= =] and [. .] are not currently implemented; they are simply placeholders for future extensions and will cause fatal errors. See perlre for more information.

In the message text:

char_seq
The character sequence.

Example: HPE20356 POSIX syntax [%c %c] belongs inside character classes

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use character class constructs [: :], [= =], and [. .] outside a character class.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20357 shm I/O not implemented

Explanation: You don't have System V shared memory IPC on your system.

Example: HPE20357 shm I/O not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The functionality may exist but not be configured. Consult your system support.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20358 suidperl is no longer needed since the kernel can now execute setuid perl scripts securely.

Explanation: Suidperl is not required to run the perl. Normal user can only run it SUID is been disabled.

Example: HPE20358 suidperl is no longer needed since the kernel can now execute setuid perl scripts securely.

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you get this error then run the perl normally no need of using suidperl.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

**HPE20359 Trailing **

Explanation: The regular expression ends with an unbackslashed backslash. See perlre for more information.

Example: HPE20359 Trailing \

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the regular expression and add backslash, if one is needed.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20360 Allocation too large:*datasize*

Explanation: You can't allocate more than 64K on an MS-DOS machine.

In the message text:

datasize
Size of data in bytes.

Example: HPE20360 Allocation too large:0xfffe

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the system configuration and change the allocation.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: X- A very fatal error (non-trappable).

Automation: Not applicable.

HPE20361 Allocation too large:*datasize*

Explanation: You can't allocate more than 64K on an MS-DOS machine.

In the message text:

datasize
Size of data in bytes.

Example: HPE20361 Allocation too large:0xffff

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response:

User response: Check the system configuration and change the allocation.

System programmer response:

Problem determination: No additional information.

Source: Perl

Module:

Routing code:

Descriptor code: X- A very fatal error (non-trappable).

Automation:

HPE20362 Ambiguous call resolved as CORE::*sub()***,** *keyword*

Explanation: A subroutine that you declared has the same name as a Perl keyword, and you have used the name without qualification for calling one or the other. Perl decided to call the builtin because the subroutine is not imported. To force interpretation as a subroutine call, either put an ampersand before the subroutine name, or qualify the name with its package. Alternatively, you can import the subroutine (or pretend that it's imported with the `use subs,/code> pragma). To silently interpret it as the Perl operator, use the ,code>CORE::` prefix on the operator (e.g. `CORE::log($x)`) or declare the subroutine to be an object method. (See perlsub/"Subroutine Attributes" for more information.)

In the message text:

sub
Subroutine

keyword
Keyword

Example: HPE20362 Ambiguous call resolved as CORE::sub(), sub

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Because the subroutine name was the same as the keyword name, either use a different name for the subroutine or put an ampersand in front of it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20363 Ambiguous use of -*sub* **resolved as -&***sub***()**

Explanation: You said something that may not be interpreted the way you thought.

In the message text:

sub
Subroutine.

Example: HPE20363 Ambiguous use of -FOO resolved as -&FOO()

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Make the use of *sub* more clear by supplying a missing quote, operator, parenthesis pair or declaration.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20364 Ambiguous use of *char* resolved as operator *char*

Explanation: You said something that may not be interpreted the way you thought.

In the message text:

char
Character.

Example: HPE20364 Ambiguous use of * resolved as operator *

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Make the use of *char* clear by supplying a missing quote, operator, parenthesis pair or declaration.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20365 Ambiguous use of *char1*{*subchar2*} resolved to *char1subchar2*

Explanation: Within regular expression character classes ([]), the syntax beginning with "[%c" and ending with "%c]" is reserved for future extensions. See perlre for more information.

In the message text:

char1
Funny Character

sub
Subroutine

char2
Constant Character

Example: HPE20365 Ambiguous use of #{sub[} resolved to #sub[

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If you need to represent those character sequences inside a regular expression character class, just quote the square brackets with the backslash: "\[%c" and "%c\]".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20366 Ambiguous use of *char*{*sub*} resolved to *charsub*

Explanation: Within regular expression character classes ([]) the syntax beginning with "[%c" and ending with "%c]" is reserved for future extensions. The <-- HERE shows in the regular expression about where the problem was discovered. See perlre for more information.

In the message text:

char
Funny Character

sub
Subroutine

Example: HPE20366 Ambiguous use of #{sub} resolved to #sub

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If you need to represent those character sequences inside a regular expression character class, just quote the square brackets with the backslash: "\[%c" and "%c\]".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20367 Argument "*str*" isn't numeric in *str1*

Explanation: The indicated string was fed as an argument to an operator that expected a numeric value instead.

In the message text:

str
String

str1
String

Example: HPE20367 Argument "foo" isn't numeric in foo

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use a numeric value.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20368 Argument "*str*" isn't numeric

Explanation: The indicated string was fed as an argument to an operator that expected a numeric value instead.

In the message text:

str
String

Example: HPE20368 Argument "bar" isn't numeric

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Give a numeric value to the argument.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20369 Argument list not closed for PerlIO layer "*str*"

Explanation: When pushing a layer with arguments onto the Perl I/O system you forgot the) that closes the argument list. (Layers take care of transforming data between external and internal representations.) Perl stopped parsing the layer list at this point and did not attempt to push this layer.

In the message text:

str
String will be a layer.

Example: HPE20369 Argument list not closed for PerlIO layer "stdio"

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use ')' and close the list. It may be the result of the value of the environment variable PERLIO.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20370 Array @*var* missing the @ in argument *num* of *descriptor()*

Explanation: In previous versions of Perl, you were able to omit the @ on array names in some spots. However, this feature is now deprecated.

In the message text:

var
Array type variable

num
Number of Arguments

descriptor
Descriptor type

Example: HPE20370 Array @ash missing the @ in argument 3 of null operation()

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use @ when referring to arrays.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: D- A deprecation (optional).

Automation: Not applicable.

HPE20371 Hash %*var* missing the % in argument *num* of *descriptor()*

Explanation: Previous versions of Perl let you omit the % on hash names in some spots. This feature is now deprecated.

In the message text:

var
Hash type variable

num
Number of Arguments

descriptor
Descriptor type

Example: HPE20371 Hash %ash missing the % in argument 3 of null operation()

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use %% when referring to arrays.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: D- A deprecation (optional).

Automation: Not applicable.

HPE20372 assertion botched (*str1*?): *str2* *filename*:*n*

Explanation: The malloc package that comes with Perl had an internal failure. This is an internal error that cannot be resolved by the user.

In the message text:

str1
str1 can be dignostics message

str2
The code which generated dignostics message

filename
file name

n line number

Example: HPE20372 assertion botched (chunk's tail overwrite?): ((unsigned int *)((caddr_t)ovp + nb))[-1] == 0x55555555 (malloc.c:2226)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting a problem to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20373 assertion botched (

Explanation: The malloc package that comes with Perl had an internal failure. This is an internal error that cannot be resolved by the user.

Example: HPE20373 assertion botched (

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20374 A thread exited while *num* threads were running

Explanation: When using threaded Perl, a thread (not necessarily the main thread) exited while there were still other threads running. See perl/threads for more information.

In the message text:

num
Number of threads

Example: HPE20374 A thread exited while 3 threads were running

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Collect the return values of the created threads by joining them before exiting from the main thread.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20375 Attempt to free unreferenced glob pointers

Explanation: The reference counts got confused on symbol aliases. This is an internal error that cannot be resolved by the user.

Example: HPE20375 Attempt to free unreferenced glob pointers

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20376 Attempt to free non-existent shared string '*str1*'*str2*

Explanation: Perl maintains a reference counted internal table of strings to optimize the storage and access of hash keys and other strings. This is an internal error that cannot be resolved by the user.

In the message text:

str1
String will be Hex Key.

str2
String will be UTF or empty string

Example: HPE20376 Attempt to free non-existent shared string 'str'" "

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not decrement the reference count of a string that is not in the table.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20377 Attempt to free non-arena SV: 0x*addr*

Explanation: All SV objects are supposed to be allocated from arenas that will be garbage-collected on exit. An SV was discovered to be outside any of those arenas. "This is an internal error that cannot be resolved by the user.

In the message text:

addr
Memory address.

Example: HPE20377 Attempt to free non-arena SV: 0x123456

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20378 Attempt to free unreferenced scalar: SV 0x*addr*

Explanation: Perl went to decrement the reference count of a scalar to see if it would go to 0, and discovered that it had already gone to 0 earlier, and should have been freed, and in fact, probably was freed. This could indicate that SvREFCNT_dec() was called too many times, or that SvREFCNT_inc() was called too few times, or that the SV was mortalized when it shouldn't have been, or that memory has been corrupted.

In the message text:

addr
Memory address.

Example: HPE20378 Attempt to free unreferenced scalar: SV 0x247458

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not free scalar whose reference count is zero.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20379 Attempt to free temp prematurely: SV 0x*addr*

Explanation: Mortalized values are supposed to be freed by the free_tmps() routine. It indicates that something else is freeing the SV before the free_tmps() routine gets a chance, which means that the free_tmps() routine will be freeing an unreferenced scalar when it does try to free it.

In the message text:

addr
Memory address.

Example: HPE20379 Attempt to free temp prematurely: SV 0x567894

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not free temp before free_tmps() routine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20380 Attempt to pack pointer to temporary value

Explanation: You tried to pass a temporary value (like the result of a function, or a computed expression) to the "p" pack() template. This means the result contains a pointer to a location that could become invalid anytime, even before the end of the current statement.

Example: HPE20380 Attempt to pack pointer to temporary value

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: To avoid this warning, use literals or global values as arguments to the "p" pack() template to avoid this warning.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20381 Attempt to use reference as lvalue in substr

Explanation: You supplied a reference as the first argument to substr() used as an lvalue. See perlfunc/substr for more information.

Example: HPE20381 Attempt to use reference as lvalue in substr

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Before supplying a reference to substr(), you need to dereference it first.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20382 Bad filehandle: *sym*

Explanation: A symbol was passed to something wanting a filehandle, but the symbol has no filehandle associated with it. You may not have done an open(); if you did, you may have done it in another package.

In the message text:

sym
Symbol.

Example: HPE20382 Bad filehandle: io

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check that the filehandle is valid before using it in other functions.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20383 Bad free() ignored

Explanation: An internal routine called free() on something that had never been malloc()ed in the first place.

Example: HPE20383 Bad free() ignored

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Set the environment variable `PERL_BADFREE` to 0.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20384 Bad hash

Explanation: One of the internal hash routines was passed a null HV pointer. "This is an internal error that cannot be resolved by the user.

Example: HPE20384 Bad hash

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20385 Bad realloc() ignored

Explanation: An internal routine called realloc() on something that had never been malloc()ed in the first place.

Example: HPE20385 Bad realloc() ignored

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Set environment variable `PERL_BADFREE` to 1.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20386 Bad symbol for array

Explanation: An internal request asked to add an array entry to something that wasn't a symbol table entry. This is an internal error that cannot be resolved by the user.

Example: HPE20386 Bad symbol for array

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Add a symbol in a symbol table for array.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20387 Bad symbol for hash

Explanation: An internal request asked to add a hash entry to something that wasn't a symbol table entry. This is an internal error that cannot be resolved by the user.

Example: HPE20387 Bad symbol for hash

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Add a symbol in a symbol table for hash. Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20388 Bad symbol for filehandle

Explanation: An internal request asked to add a filehandle entry to something that wasn't a symbol table entry. This is an internal error that cannot be resolved by the user.

Example: HPE20388 Bad symbol for filehandle

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Add a symbol in a symbol table for filehandle. Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20389 Bad symbol for filehandle (GV is unique)

Explanation: An internal request asked to add a filehandle entry to something that wasn't a symbol table entry. This is an internal error that cannot be resolved by the user.

Example: HPE20389 Bad symbol for filehandle (GV is unique)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Add a symbol in a symbol table for filehandle. Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20390 Bareword "*package*" refers to nonexistent package

Explanation: You used a qualified bareword of the form `Foo::`, but the compiler saw no other uses of that namespace before that point.

In the message text:

package
Package name

Example: HPE20390 Bareword "Foo::" refers to nonexistent package

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: You need to predeclare a package.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20391 *char* better written as $*char*

Explanation: Outside of patterns, backreferences live on as variables. The use of backslashes is grandfathered on the right-hand side of a substitution, but stylistically it is better to use the variable form because other Perl programmers will expect it.

In the message text:

char
Charcter

Example: HPE20391 \1 better written as $1

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Variable form works better if there are more than 9 backreferences.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20392 Scalar value *var* better written as $*var*

Explanation: You've used an array/hash slice (indicated by @) to select a single element of an array/hash. Generally it's better to ask for a scalar value (indicated by $). The difference is that `$foo[&bar]` always behaves like a scalar, both when assigning to it and when evaluating its argument, while `@foo[&bar]` behaves like a list when you assign to it, and provides a list context to its subscript, which can do weird things if you're expecting only one subscript.

In the message text:

var
The variable name.

Example: HPE20392 Scalar value @foo[&bar] better written as $foo[&bar]

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use $ when you reference to an array or hash.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20393 dump() better written as CORE::dump()

Explanation: You used the obsolescent `dump()` built-in function, without fully qualifying it as `CORE::dump()`. Maybe it's a typo. See Perl functions for more information.

Example: HPE20393 dump() better written as CORE::dump()

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: `dump()` function is now largely obsolete, partly because it's very hard to convert a core file into an executable, and because the real compiler backends for generating portable bytecode and compilable C code have superseded it. That's why you should now invoke it as `CORE::dump()`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20394 Binary number > 0b11111111111111111111111111111111 non-portable

Explanation: The binary number you specified is larger than 2**32-1 (4294967295) and therefore is not portable between systems. See perlport for more iinformation about portability concerns.

Example: HPE20394 Binary number > 0b11111111111111111111111111111111 non-portable

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Change the binary number to be less than 2**32-1 (4294967295).

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20395 Bit vector size > 32 non-portable

Explanation: Bit vector sizes larger than 32 is not portable between systems.

Example: HPE20395 Bit vector size > 32 non-portable

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Change the size of the bit vector to be equal to or less than 32.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20396 Bizarre copy of *str1* in *str2*

Explanation: Perl detected an attempt to copy an internal value that is not copyable. This is an internal error that cannot be resolved by the user.

In the message text:

str1
: string

str2
string

Example: HPE20396 Bizarre copy of sting in lc

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting a problem to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20397 Bizarre copy of *str*

Explanation: Perl detected an attempt to copy an internal value that is not copyable. This is an internal error that cannot be resolved by the user.

In the message text:

str
string

Example: HPE20397 Bizarre copy of string

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting a problem to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20400 msgsnd not implemented

Explanation: Send an ipc message (m!c) is not implemented.

Example: HPE20400 msgsnd not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether msgsnd has been implemented in your machine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20401 msgrcv not implemented

Explanation: The msgrcv perl routine, which calls the system msgrcv function that is used to read a message from a message queue, is not implemented.

Example: HPE20401 msgrcv not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the msgrcv routine in the message queue code.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20402 semop not implemented

Explanation: The semop perl routine, which calls the System V IPC function semop to perform semaphore operations such as signaling and waiting, is not implemented.

Example: HPE20402 semop not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the semop routine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20403 Can't open a reference

Explanation: An attempt was made to open a scalar reference for reading or writing, using the 3-arg open() syntax : open FILEH, '>', $ref; but the version of Perl is compiled without perlio, and this form of open is not supported. This warning is related to IO.

Example: HPE20403 Can't open a reference

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use the 3-arg open syntax to do a file open after compiling perl with perlio.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20404 Missing command in piped open

Explanation: An attempt was made to use the `open(FH, "" command")` or `open(FH, "command "")` construction, but the command was missing or blank. This warning is related to pipes.

Example: HPE20404 Missing command in piped open

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: The open(FH, ""command") construction should be used with the `command` filled in appropriately.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20405 Can't open bidirectional pipe

Explanation: An attempt was made to say `open(CMD, ""cmd"")`, which is not supported. This warning is related to pipes.

Example: HPE20405 Can't open bidirectional pipe

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: You can try several modules in the Perl library to open a birectional pipe, such as IPC::Open2. You could also try directing the the pipe's output to a file using ">", and then have it read under a different filehandle.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20406 Filehandle STD*str* reopened as *filehandle* only for input

Explanation: A filehandle was opened for reading which has the same ID as STDOUT or STDERR. The STDOUT or STDERR was closed previously. The warning is related to IO.

In the message text:

str
The string could be OUT or ERR

filehandle
The filehandle

Example: HPE20406 Filehandle STDOUT reopened as FOO only for input

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not open a filehandle having the same ID as STDOUT or STDERR.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20407 Filehandle STDIN reopened as *filehandle* **only for output**

Explanation: A filehandle was opened for writing that has the same filehandle ID as STDIN. This happened since STDIN was closed previously. This warning is related to IO.

In the message text:

filehandle
The filehandle

Example: HPE20407 Filehandle STDIN reopened as FOO only for output

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not open a filehandle that has the same ID as STDIN.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20408 Can't do inplace edit: *filename* **is not a regular file**

Explanation: An attempt was made to use the **-i** switch on a special file in /dev or a FIFO. The file was ignored. This severe warning is of type inplace. This belongs to the inplace category.

In the message text:

filename
The file name

Example: HPE20408 Can't do inplace edit: /dev/fd0 is not a regular file

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not attempt to inplace edit a special file such as a file in /dev or a FIFO.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20409 Can't do inplace edit: *filename* **would not be unique**

Explanation: The file system does not support file names longer than 14 characters and Perl was unable to create a unique file name during inplace editing with the **-i** switch. The file was ignored. This belongs to the inplace category.

In the message text:

filename
The filename

Example: HPE20409 Can't do inplace edit: filename1234567 would not be unique

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use a file name that is not unique while doing a inplace edit.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20410 Can't rename *filename1* to *filename2*: *errnodisplay*, skipping file

Explanation: The rename done by the **-i** switch failed for some reason, probably because you don't have write permission to the directory.

In the message text:

filename1
The original file name

filename2
The new file name

errnodisplay
The string representation of the errno

Example: HPE20410 Can't rename /usr1/file1 to /usr1/file1bak: EDC5111I Permission denied, skipping file

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check whether you have permission to write to the appropriate directory and then attempt a inplace edit.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20411 Can't remove *filename*: *errnodisplay*, skipping file

Explanation: An inplace edit was requested without creating a backup file. Perl was unable to remove the original file to replace it with the modified file. The file was left unmodified.

In the message text:

filename
The filename.

errnodisplay
The string representation of the errno.

Example: HPE20411 Can't remove file1: EDC5133I: No space left on device, skipping file

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: A backup file should always be created first before doing an inplace edit.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20412 Can't do inplace edit on *filename*: *errnodisplay*

Explanation: The file was not created due to the indicated reason.

In the message text:

filename
The filename.

errnodisplay
The string representation of the errno.

Example: HPE20412 Can't do inplace edit on file1: EDC5129I No such file or directory.

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: One of the reasons for a failure to do in-place edit is there is no such file or directory on which the edit is attempted. Verify that the mentioned file exists and then try the edit again.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20413 Wide character in print

Explanation: Unicode strings were output to a stream without a I/O layer. The raw bytes used internally will be used (these could either be the native character set or UTF-8 as the case may be). The warning is issued if the string contains a character beyond 0x00FF.

Example: HPE20413 Wide character in print

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Prepend binmode(STDOUT, ":utf8") to the script to avoid this warning.

System programmer response: No System Programmer response is required.

Problem determination: perl -e 'print "\x{DF}\n", "\x{0100}\x{DF}\n"'; This generates the warning and produces a fairly useless mixture of bytes. This is because a character > 0x00FF was used. You can avoid this situation by prepending binmode(STDOUT, ":utf8") to the print call.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20414 Use of *construct* is deprecated

Explanation: The construct indicated is no longer suggested for use, generally because there's a better way to do it, and also because the old way has bad side effects.

In the message text:

construct
The construct name for example "package" with no arguments. If the package is called without any arguments, then this message displays

Example: HPE20414 Use of "package" with no arguments is deprecated

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use the appropriate construct in place of the deprecated construct.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: D- A deprecation (optional).

Automation: Not applicable.

HPE20415 (Missing semicolon on previous line?)

Explanation: This message is in conjunction with the message "HPE20697 *item* found where operator expected".

Example: HPE20415 (Missing semicolon on previous line?)

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check whether the syntax and punctuation of the code is correct. However, do not automatically put a semicolon on the previous line just because you saw this message.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20416 (Do you need to predeclare *str*?)

Explanation: This message in conjunction with the message "HPE20697 *item* found where operator expected" often means a subroutine or module name is being referenced that hasn't been declared yet. This may be because of ordering problems in your file, or because of a missing "sub", "package", "require", or "use" statement.

In the message text:

str
The subroutine name or package name or module name

Example: HPE20416 (Do you need to predeclare foo?)

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If something that isn't defined yet is being referenced, you do not have to define the subroutine or package before the current location. You can use an empty "sub foo;" or "package FOO;" to enter a "forward" declaration.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20417 (Missing operator before *str*)

Explanation: This message is in conjunction with the message "HPE20697 *item* found where operator expected". Often the missing operator is a comma. This belongs to the syntax category. The operator could be missing before the end of line or before a construct or a variable or a part of a statement.

In the message text:

str
The string 'end of line' or a part of a statement or a variable or a construct

Example: HPE20417 (Missing operator before $var?)"

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check whether a comma (or some other operator) is missing before the end of line if the operator is expected just before the end of line. If not, look for a missing operator before a part of a statement or a construct or a variable.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20418 panic: constant overflowed allocated space

Explanation: The space required to store a constant value is greater than the space allocated for it. This is an internal error that cannot be resolved by the user

Example: HPE20418 panic: constant overflowed allocated space

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20419 panic: YYMAXLEVEL

Explanation: This is in the context of the parser. The yyactlevel variable has reached a threshold value which is defined as a macro, YYMAXLEVEL. The yyactlevel variable should have a value less than YYMAXLEVEL. This is an internal error that cannot be resolved by the user

Example: HPE20419 panic: YYMAXLEVEL

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20420 panic: INTERPCASEMOD

Explanation: The lexer got into a bad state at a case modifier. This is an internal error that cannot be resolved by the user.

Example: HPE20420 panic: INTERPCASEMOD

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20421 panic: yylex

Explanation: The lexer got into a bad state while processing a case modifier. This is an internal error that cannot be resolved by the user.

Example: HPE20421 panic: yylex

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20422 panic: INTERPCONCAT

Explanation: The lexer got into a bad state parsing a string with brackets. This is an internal error that cannot be resolved by the user.

Example: HPE20422 panic: INTERPCONCAT

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20423 Unrecognized character \x*hexchar*

Explanation: The Perl parser has no idea what to do with the specified character in the Perl script (or eval). Perhaps an attempt was made to run a compressed script, a binary program, or a directory as a Perl program.

In the message text:

hexchar
The unrecognized 2 byte hex character

Example: HPE20423 Unrecognized character \x03

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Avoid running a script or a program which contains the above mentioned characters.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20424 (Maybe you didn't strip carriage returns after a network transfer?)

Explanation: This message appears in conjunction with "HPE20285 Illegal character *char*(carriage return)". Typically, literal carriage returns within program text are treated as white space. A preprocessor symbol was defined that treats carriage returns within program text more strictly. Maybe a network transfer was completed but carriage returns were not stripped from the text.

Example: HPE20424 (Maybe you didn't strip carriage returns after a network transfer?)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use carriage returns after a network transfer.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20425 panic: input overflow

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20425 panic: input overflow

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20426 Unterminated <> operator

Explanation: The lexer found a left angle bracket in a place where it was expecting a term, so it's looking for the corresponding right angle bracket, and not finding it. Chances are some needed parentheses were left out earlier in the line, and the intended code was really a "less than".

Example: HPE20426 Unterminated <> operator

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Ensure that all parentheses are closed before a left angled bracket is used alone to mean a 'less-than' sign.

System programmer response: No System Programmer response is required.

Problem determination: Trappable errors may be trapped using the eval operator

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20427 panic: scan_num

Explanation: scan_num() got called on something that wasn't a number. This is an internal error that cannot be resolved by the user.

Example: HPE20427 panic: scan_num

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20428 *msg program_name* **has too many errors.**

Explanation: The parser has given up trying to parse the program after ten errors. Further error messages would likely be uninformative. This error is fatal.

In the message text:

msg
The error message which may contain: (Missing semicolon on previous line?) or (Missing operator before %.*s?)

program_name
The program name

Example: HPE20428 Syntax error in file ./a.t at line 258, next 2 tokens "my (" Syntax error in file ./a.t at line 272, next 2 tokens "my $hnrcIsNew " Syntax error in file ./a.t at line 580, next 2 tokens "my $email" Syntax error in file ./a.t at line 618, next 2 tokens "local $debug " Syntax error in file ./a.t at line 651, next 2 tokens "my(" Syntax error in file ./a.t at line 724, next token "}" Syntax error in file ./a.t at line 785, next token "}" Syntax error in file ./a.t at line 792, next 2 tokens "}" Syntax error in file ./a.t at line 801, next 2 tokens "}" Syntax error in file ./a.t at line 808, next 2 tokens "}" a.t has too many errors.\n

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Minimize the number of errors in the Perl program

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20429 Unsupported script encoding UTF32-LE

Explanation: The program file begins with a Byte Order Mark which marks it in a Unicode encoding that Perl cannot read.

Example: HPE20429 Unsupported script encoding UTF32-LE

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the utf32-le encoding if it is not supported on the current version of perl.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20430 Unsupported script encoding UTF16-LE

Explanation: The program file begins with a Byte Order Mark which marks it in a Unicode encoding that Perl cannot read.

Example: HPE20430 Unsupported script encoding UTF16-LE

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the utf16-le encoding if it is not supported on the current version of Perl.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20431 Unsupported script encoding UTF16-BE

Explanation: The program file begins with a Byte Order Mark which marks it in a Unicode encoding that Perl cannot read.

Example: HPE20431 Unsupported script encoding UTF16-BE

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the utf16-be encoding if it is not supported on the current version of Perl.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20432 Unsupported script encoding UTF32-BE

Explanation: The program file begins with a Byte Order Mark which marks it in a Unicode encoding that Perl cannot read.

Example: HPE20432 Unsupported script encoding UTF32-BE

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the utf32-be encoding if it is not supported on the current version of Perl.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20433 Reversed *symbol1symbol2* operator

Explanation: The assignment operator was written backwards. The = must always come last, to avoid ambiguity with subsequent unary operators.

In the message text:

symbol1
The '+' or '-' symbol

symbol2
The symbol =

Example: HPE20433 Reversed += operator

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Always place the = sign last to avoid the warning.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20434 Multidimensional syntax *construct* not supported

Explanation: Multidimensional arrays aren't written like `$foo[1,2,3]`.

In the message text:

construct
The multidimensional array representation

Example: HPE20434 Multidimensional syntax $foo[1,2,3] not supported

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Represent multi-dimensional arrays as `$foo[1][2][3]`, as in C.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20435 Can't use *num* to mean *symbol num* in expression

Explanation: In an ordinary expression, backslash is a unary operator that creates a reference to its argument. A backslash can be used to indicate a backreference to a matched substring only as part of a regular expression pattern. Trying to do this in ordinary Perl code produces a value that prints out looking like SCALAR(0xdecaf).

In the message text:

num
The number indicating the matched variable

symbol
The $ symbol

Example: HPE20435 Can't use \1 to mean $1 in expression

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use the $ form instead of the backslash form.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20436 Operator or semicolon missing before *symbol1var2*

Explanation: A variable or subroutine call was used where the parser was expecting an operator. The parser assumed that an operator was really intended to be used. However, that assumption could be incorrect. This severe warning belongs to the ambiguous category.

In the message text:

symbol1
The symbol which could be a '*'

var2
The pointer variable

Example: HPE20436 Operator or semicolon missing before *foo at - line 8.

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use a variable or subroutine call where an operator is expected.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20437 Precedence problem: open *filehandle* should be open(*filehandle*)

Explanation: The old irregular construct open FH "" die; is now misinterpreted as open(FH "" die); because of the strict regularization of Perl 5's grammar into unary and list operators. (The old open was a little of both.) Parentheses should be placed around the filehandle, or the new "or" operator should be used instead of """". This severe warning belongs to the precedence category.

In the message text:

filehandle
The filehandle

Example: HPE20437 Precedence problem: open FOO should be open(FOO)

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use parentheses around the filehandle when calling an open of the form : open filehandle "" die;

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20438 Possible attempt to separate words with commas

Explanation: qw() lists contain items separated by whitespace; therefore commas aren't needed to separate the items. (You may have used different delimiters than the parentheses shown here; braces are also frequently used.) This warning belongs to the qw category.

Example: HPE20438 Possible attempt to separate words with commas

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If you have something like the following: qw! a, b, c !; Then that should should be written as : qw! a b c !;

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20439 Possible attempt to put comments in qw() list

Explanation: qw() lists contain items separated by white space; as with literal strings, comment characters are not ignored, but are instead treated as literal data. (delimiters different from the ones shown here might have been used; braces are also frequently used.)

Example: HPE20439 Possible attempt to put comments in qw() list

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: For comments, the qw() list should be built with quotes and commas: @list = ('a', # a comment 'b', # another comment);

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20440 Illegal character in prototype for *sub* : *var*

Explanation: An illegal character was found in a prototype declaration. Legal characters in prototypes are $, @, %, *, ;, [,], &, and \. This warning belongs to the syntax category.

In the message text:

sub
The subroutine name

var
The malloced string

Example: HPE20440 Illegal character in prototype for main::badproto : \@bar

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use legal characters for the subroutine prototype.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20441 Possible unintended interpolation of *var* in string

Explanation: Something like `@foo' was indicated in a double-quoted string but there was no array `@foo` in scope at the time. To indicate a literal @foo, it should be written as \@foo; otherwise you need to find out about the array in question. This warning belongs to the ambiguous category.

In the message text:

var
The pointer variable

Example: HPE20441 Possible unintended interpolation of @arr in string

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: An array should be indicated in a double=quoted string as \@foo

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20442 elseif should be elsif

Explanation: The code containing elseif will be interpreted as an attempt to call a method named 'elseif' for the class returned by the following block. This severe warning belongs to the syntax category.

Example: HPE20442 elseif should be elsif

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use elsif instead of elseif.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20443 *op*(...) interpreted as function

Explanation: If a list operator is followed by a left parenthesis, the list operator is treated as a function with all operators and arguments within the pair of parentheses. This warning belongs to the syntax category.

In the message text:

op The list operator

Example: HPE20443 sort (...) interpreted as function

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use parentheses when calling a list operator to prevent it from being treated as a function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20444 Misplaced _ in number

Explanation: An underscore (underbar) character in a numeric constant did not separate two digits.

Example: HPE20444 Misplaced _ in number

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: The underscore (underbar) character should be used appropriately.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20445 Integer overflow in *numformat* number

Explanation: The hexadecimal, octal or binary number which has been specified either as a literal or as an argument to hex() or oct() is too big for the architecture, and has been converted to a floating point number. On a 32-bit architecture, the largest hexadecimal, octal or binary number representable without overflow is 0xFFFFFFFF, 037777777777, or 0b11111111111111111111111111111111 respectively. Note that Perl transparently promotes all numbers to a floating point representation internally--subject to loss of precision errors in subsequent operations. This warning belongs to the overflow category.

In the message text:

numformat
The numeric format. It can have values : decimal, hexadecimal, octal or binary.

Example: HPE20445 Integer overflow in binary number

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Make sure that the numbers used are within range.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20446 *numformat* **number >** *num* **non-portable**

Explanation: Because the specified number is larger than the permissible value, it is not portable between systems. The largest permissible value is 2**32 - 1. This warning belongs to the portable category.

See perlport for more on portability concerns.

In the message text:

numformat
The numeric format which could be hexadecimal, octal or binary

num
The number

Example: HPE20446 Hexadecimal number > 0xffffffff non-portable

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use a number less than or equal to 2**32 -1 to make it portable between systems.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20447 Integer overflow in decimal number

Explanation: Because the decimal number, which has been specified either as a literal or as an argument to hex() or oct(), is too big for the architecture, it has been converted to a floating point number. On a 32-bit architecture, the largest binary number representable without overflow is 0b11111111111111111111111111111111. Note that Perl transparently promotes all numbers to a floating point representation internally--subject to loss of precision errors in subsequent operations. This warning belongs to the overflow category.

Example: HPE20447 Integer overflow in decimal number

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use decimal numbers within range.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20448 panic: paren_elems_to_push < 0

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20448 panic: paren_elems_to_push < 0

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedure for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20449 panic: end_shift

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20449 panic: end_shift

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System

Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20450 panic: unknown regstclass *num*

Explanation: The starting class for the regular expression was not known.

In the message text:

num
numbers

Example: HPE20450 panic: unknown regstclass 10

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use the correct class in the regular expression.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20452 *function***() called too early to check prototype**

Explanation: A function was called that has a prototype before the parser saw a definition or declaration for it, and Perl could not check that the call conforms to the prototype. For more information, see perlsub.

In the message text:

function
The function name.

Example: HPE20452 foo() called too early to check prototype

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Either an early prototype declaration for the subroutine in question needs to be added, or the subroutine definition needs to be moved ahead of the call to get proper prototype checking. Alternatively, if you are sure that the function is being called correctly, an ampersand before the name would help avoid the warning.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20453 Can't call method "*method***"** *str*

Explanation: A method was not called because a package or object was not referenced, or a value was not specified.

In the message text:

method
The method name.

str
The reason string. It could be either "without a package or object reference" or "on an undefined value"

Example: HPE20453 Can't call method "meth" on an undefined value at - line 1

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that either an object or package is referenced, or a value is defined, before a method is called.

System programmer response: No System Programmer response is required.

Problem determination: A Fatal error may be trapped using the eval operator.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20454 Can't check filesystem of script "*scriptname*" for nosuid/noexec

Explanation: For some reason, the file system of the script cannot be checked for nosuid or noexec. This is an internal error which should not be seen.

In the message text:

scriptname
The script name.

Example: HPE20454 Can't check filesystem of script a.pl for nosuid/noexec

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20455 Setuid script "*scriptname*" on nosuid filesystem

Explanation: An attempt was made to run a setuid script on a file system that does not support the setuid bit.

In the message text:

scriptname
The script name.

Example: HPE20455 Setuid script a.pl on nosuid filesystem

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether a script has the setuid bit set before attempting it on a nosuid filesystem. It should show permissions like : "-rwsr--r--". The 's' in the permissions indicates that the setuid bit is set.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20456 Setuid script "*scriptname*" on noexec filesystem

Explanation: A script, which has the setuid bit set, was attempted on a file system where running a script is not allowed because the noexec permission was set.

In the message text:

scriptname
The script name.

Example: HPE20456 Setuid script b.pl on noexec filesystem

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Verify if the script has the setuid bit set. Also, verify whether the file system has the noexec permission set before you attempt to run the script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20457 Setuid script changed

Explanation: The permissions of the script have been changed so as to set the setuid bit. However, if the shell does not support setuid scripts, the script will not run.

Example: HPE20457 Setuid script changed

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: While changing a script to enable the setuid bit, ensure that the shell on which you intend to run the script supports setuid scripts.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20458 Setuid script on nosuid or noexec filesystem

Explanation: An attempt was made to run a setuid script on a file system that either is a noexec file system or does not support the setuid bit.

Example: HPE20458 Setuid script on nosuid or noexec filesystem

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether a script has the setuid bit set before attempting it on a nosuid file system. It should show permissions like : "-rwsr--r--". The 's' in the permissions indicates that the setuid bit is set. Also verify whether the filesystem allows setting the setuid bit or if it is a noexec filesystem (may be verified by the mount options).

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20459 Setuid script not plain file

Explanation: If a script is read from a socket, a pipe or another device, the setuid emulator won't run that script.

Example: HPE20459 Setuid script not plain file

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not run a perl script from a socket, a pipe or any other device.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20460 Setuid script name may not begin with dash

Explanation: The name of the setuid script began with a dash, which is not allowed.

Example: HPE20460 Setuid script name may not begin with dash

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: cond_signal should be called on a variable that is shared.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20461 No setuid script name

Explanation: There is no name for the setuid script

Example: HPE20461 No setuid script name

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Assign a name to the setuid script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20462 Can't do setegid!

Explanation: The setegid() call failed within the setuid emulator of suidperl.

Example: HPE20462 Can't do setegid!

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether the calling process has appropriate privileges before calling setegid. Also, the setegid should be attempted only if the gid is equal to the real group id or the saved set-group-id.

System programmer response: No System Programmer response is required.

Problem determination: Check whether the calling process has appropriate privileges to call setegid or if the gid is not equal to the real -group id or saved set-group-id.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20463 Can't do seteuid!

Explanation: The function to set the effective user ID of the calling process failed. The value of the UID argument may not be valid and is not supported by the implementation. It is also possible that the process does not have appropriate privileges and UID does not match the real user ID or the saved set-user-ID.

Example: HPE20463 Can't do seteuid!

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use a valid UID argument to the seteuid call and also ensure that the process that is calling seteuid has appropriate privileges.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20464 Can't exec "*filename***": *errnodisplay***

Explanation: A system(), exec(), or piped open call could not execute the named program for the indicated reason. This warning belongs to the exec category.

In the message text:

filename
The filename.

errnodisplay
The string form of the errno

Example: HPE20464 Can't exec /u/isldev/file1: EDC5129I No such file or directory. at b.t line 2.

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Verify that the file has correct permissions, it is present in `$ENV{PATH}`, the executable has been compiled for the same architecture that it is being attempted to run on, and that the #! line in the script points to the Perl interpreter which has been built.

System programmer response: No System Programmer response is required.

Problem determination: Typical reasons for the failure include: the permissions were wrong on the file, the file wasn't found in `$ENV{PATH}`, the executable in question was compiled for another architecture, or the #! line in a script points to an interpreter that can't be run for similar reasons. (Or maybe your system doesn't support #! at all.)

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20470 Can't locate package *variable* for @*package*::ISA

Explanation: The @ISA array contained the name of another package that does not seem to exist.

In the message text:

variable
The pointer variable corresponding to the package in question

package
The package name

Example: HPE20470 Can't locate package Foo for @main::ISA

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Verify that the @ISA array for a package contains names of existing packages.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20471 Cannot modify non-existent substring

Explanation: A nonexistent string was passed to the internal function that does assignment in a substr() routine. The internal routine attempts to replace an existing substring with the new specified substring but fails because the original string is missing.

Example: HPE20471 Cannot modify non-existent substring

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Call the substr() routine in such a way that a valid substring is generated and passed to the internal assignment function.

System programmer response: No System Programmer response is required.

Problem determination: For example:

```
$a = "this is a sample string";
$b = substr, $a, 5, 6, "replace";
print "a : $a\n";

a: thissamplesample string
```

The example just shown is not the actual scenario. During the course of the above, if the original string goes missing while calling the internal routine which does an insert (replace), the message is generated.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20472 Can't open *filename*: *errnodisplay*

Explanation: The implicit opening of a file through use of the <> filehandle, either implicitly under the `-n` or `-p` command-line switches, or explicitly, failed for the indicated reason. Usually this is because the read permission for the file named on the command line is not available. This severe warning belongs to the inplace category.

In the message text:

filename
The filename.

errnodisplay
The string form of the errno

Example: HPE20472 Can't open file1: EDC5111I Permission denied.

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Verify that appropriate read permissions are available for the file named on the command line.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20476 Checking overloading of *operator* in package *package*

Explanation: The operator that might have been overloaded could be one of the following:

1. conversion operators (bool, +0, "")
2. arithmetic operators (+, -, *,/, %, **, x, . , neg)
3. logical operators (!)
4. bitwise operators (&,", ~, ^, !, <<, >>)
5. assignment operators (+=, -=, *=, /=, %=, **=,x=, .=, <<=, >>=, ++, --)

In the message text:

operator
The operator

package
The package name

Example: HPE20476 Checking overloading of `DESTROY' in package `IO::Handle'

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use the overload pragma to overload an operator appropriately if the overloading check above indicates an error.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20477 *str* **method `***method***' overloading `***operator***' "\"in package `***package***'**

Explanation: Either the method overloading an operator in a package could not be resolved or a stub was found while resolving the method overloading an operator in a package.

In the message text:

str
The string which could be either of : 1) Stub found while resolving, 2) Can't resolve.

method
The method name

operator
The operator name

package
The package name

Example: HPE20477 Stub found while resolving method foo overloading '+' in package packfoo

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use the overload pragma correctly to define a method overloading an operator.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20479 Can't return array to lvalue scalar context

Explanation: You tried to return a complete array from an lvalue subroutine, but you called the subroutine in a way that made Perl think you meant to return only one value.

Example: HPE20479 Can't return array to lvalue scalar context

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether parentheses are needed around around the call to the subroutine. The parentheseses tell Perl that the call should be in list context.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20480 Can't return hash to lvalue scalar context

Explanation: You tried to return a complete hash from an lvalue subroutine, but you called the subroutine in a way that made Perl think you meant to return only one value.

Example: HPE20480 Can't return hash to lvalue scalar context

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Write parentheses around the call to the subroutine, which tell Perl that the call should be in list context.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20481 Can't stat script *scriptname*

Explanation: The fstat call fails on a script which is open already. This is an internal error that cannot be resolved by the user

In the message text:

scriptname
The script name.

Example: HPE20481 Can't stat script foo.pl

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: Check if a normal stat call works. If it doesnt, its likely that the stat call has a problem on the platform. If it works, then, the stat call specifically has a problem with opening the filehandle and not a normal filename.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20484 Can't upgrade that kind of scalar

Explanation: A scalar value is generally represented using a struct SV. When the scalar value is upgraded to a more complex form by adding more 'members' to the structure, the upgradation takes place based on the value of the **flags** member of the struct SV. If the value of the **flags** does not match any of the predefined values, it means that the structure (scalar value) cannot be upgraded. It also indicates that an inter-conversion was attempted between some of the top SV types (corresponding to some of the defined **flags** values).This is an internal error that cannot be resolved by the user.

Example: HPE20484 Can't upgrade that kind of scalar

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20485 Can't upgrade to undef

Explanation: A struct SV may be upgraded to more complex forms like IV, NV, RV, PV and so on. The undefined SV is at the bottom of these forms, in the scheme of upgradability. Upgrading to undef indicates an error in the code calling sv_upgrade. This is an internal error that cannot be resolved by the user.

Example: HPE20485 Can't upgrade to undef

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20486 Can't upgrade filter_add data to SVt_PVIO

Explanation: The filter_add function which associates the filter object and the source stream attempts to upgrade a new struct SV to SVt_PVIO (which is a complex struct SV form) and fails. This is prior to assigning the filter object to the successfully upgraded SV struct. This is an internal error that cannot be resolved by the user.

Example: HPE20486 Can't upgrade filter_add data to SVt_PVIO

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20488 "use" not allowed in expression

Explanation: The "use" keyword is recognized and executed at compile time, and returns no useful value.

Example: HPE20488 "use" not allowed in expression

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use the 'use' keyword in an expression.

System programmer response: No System Programmer response is required.

Problem determination: For more information, see perlmod.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20489 Character in 'C' format wrapped in pack

Explanation: This error is generated when `"C"` is used to format unicode characters.The `"C"` format is only for encoding native operating system characters (ASCII, EBCDIC,and so on) and not for Unicode characters.

Example: HPE20489 Character in 'C' format wrapped in pack

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use the `pack("C",$x)` syntax only if `0<$x<255`. If you actually want to pack Unicode codepoints, use the `"U"` format instead.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20490 Character in 'c' format wrapped in pack

Explanation: Pack was called in the following format : `pack("c", $x)` where $x is either less than -128 or more than 127. The `"c"` format is only for encoding native operating system characters (ASCII, EBCDIC, and so on) and not for Unicode characters.

Example: HPE20490 Character in 'c' format wrapped in pack

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use the format `pack("c", $x)` only if `-128<$x<127`. If you wanted to pack Unicode codepoints, you can use the `"U"` format instead.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20491 *error_msg* **Compilation failed in regexp**

Explanation: Perl could not compile the regular expression. This is a fatal error.

In the message text:

error_msg
An error message providing additional information.

Example: HPE20491 Global symbol "$RE_PAREN" requires explicit package name at (re_eval 2) line 2. Compilation failed in regexp

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Ensure that global symbols have been declared using explicit package names. Other errors which show up in the **string** as in the explanation field above need to be avoided appropriately.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20493 Complex regular subexpression recursion

Explanation: The regular expression engine uses recursion in complex situations where back-tracking is required. Recursion depth is limited to 32766, or perhaps less in architectures where the stack cannot grow arbitrarily. ("Simple" and "medium" situations are handled without recursion and are not subject to a limit.)

Example: HPE20493 Complex regular subexpression recursion

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Try shortening the string under examination; looping in Perl code (for example, with `while`) rather than in the regular expression engine. Or try rewriting the regular expression so that it is simpler or backtracks less. (See perlfaq2 for information about Mastering Regular Expressions.)

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20494 cond_signal() called on unlocked variable

Explanation: Cond_signal is used to wake up another thread that is inside a cond_wait call. To ensure that the signallng thread does not call cond_signal before the signaled thread has had a chance to wait, cond_signal is called on a locked variable that can be locked only after the waiting thread has relinquished the lock (which is after it has entered cond_wait). This warning belongs to the threads category.

Example: HPE20494 cond_signal() called on unlocked variable

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Ensure that a variable is locked before calling cond_signal on it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20495 cond_broadcast() called on unlocked variable

Explanation: Within a thread-enabled program, an attempt was made to call cond_broadcast on a variable that was not locked. The cond_broadcast function is used to wake up a different waiting thread which relinquished the lock after calling cond_wait. The cond_broadcast function should be called after locking the variable. This warning belongs to the threads category.

Example: HPE20495 cond_broadcast() called on unlocked variable

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Call cond_broadcast on a variable that is locked.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20496 cond_signal can only be used on shared values

Explanation: cond_signal was used on a variable that was not shared. This means if cond_signal is called on a variable within a main thread, it will not be able to wake up a different thread where the variable cannot be accessed.

Example: HPE20496 cond_signal can only be used on shared values

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Ensure that cond_signal is called on a variable that is shared.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20498 Argument to cond_broadcast needs to be passed as ref

Explanation: The argument to the cond_broadcast function was not passed as a reference. This happens specifically when the argument was passed but it was not passed as a reference.

Example: HPE20498 Argument to cond_broadcast needs to be passed as ref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The argument to the cond_broadcast function should be passed as : cond_broadcast_enabled(ref), where ref is the reference.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20499 Semicolon seems to be missing

Explanation: A syntax error was probably caused by a missing semicolon, or possibly by some other missing operator, such as a comma.

Example: HPE20499 Semicolon seems to be missing

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check the syntax. You can use the -c option to check the syntax.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20500 Use of uninitialized value *str function*

Explanation: An undefined value was used as if it were already defined. It was interpreted as a "" or a 0, but maybe it was a mistake.

In the message text:

str
: string

function
: function name

Example: HPE20500 Use of uninitialized value in concatenation

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: You can use the undefined value to perform certain operation. To suppress the warning message, assign a defined value to your variables.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20501 Unquoted string "*str*" may clash with future reserved word

Explanation: You used a bareword that might someday be claimed as a reserved word.

In the message text:

str
string

Example: HPE20501 Unquoted string "string" may clash with future reserved word

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If you are using bareword, put it in quotes. It is best to put such a word in quotes, or capitalize it somehow, or insert an underbar into it. You might also declare it as a subroutine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20502 Unsuccessful *fileopr* on filename containing newline

Explanation: A file operation was attempted on a file name, but failed. Perhaps it did not succeed because the file name contained a newline, which can occur if you did not use chomp() to remove trailing newlines.

In the message text:

fileopr
file operations

Example: HPE20502 Unsuccessful write on filename containing newline

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Perform chomp() before using the file in the script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20503 Cannot use *datatype* ref as *datatype* ref

Explanation: You have mixed up your reference types. You have to dereference a reference of the type needed.

In the message text:

datatype
datatype

Example: HPE20503 Can't use ARRAY ref as HASH ref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: While deferencing, you must make sure that the type of reference the variable has uses ref() function. You can use the ref() function to test the type of the reference, if need be.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20504 Cannot use string ("*str*") as *datatype* ref while "strict refs" in use

Explanation: Only hard references are allowed by "strict refs". Symbolic references are disallowed.

In the message text:

str
string

datatype
datatype

Example: HPE20504 Cannot use string ("STRING") as ARRAY ref while "strict refs" in use

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you are using strict ref, then you need to make sure you are referencing the symbolic constant.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20505 Cannot use an undefined value as *datatype* reference

Explanation: A value used as either a hard reference or a symbolic reference must be a defined value.

In the message text:

datatype
datatype

Example: HPE20505 Cannot use an undefined value as ARRAY reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that you are using the defined value for symbolic references.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20506 Modification of non-creatable array value attempted, subscript *num*

Explanation: You tried to make an array value spring into existence, and the subscript was probably negative, even counting from end of the array backwards.

In the message text:

num
number

Example: HPE20506 Modification of non-creatable array value attempted, subscript -1

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure you are using the right array index.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20507 Modification of non-creatable hash value attempted, subscript "*str*"

Explanation: You tried to create a hash value, but it could not be created.

In the message text:

str
string

Example: HPE20507 Modification of non-creatable hash value attempted, subscript "default"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Change the initial value of the default.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20508 Modification of a read-only value attempted

Explanation: You tried, directly or indirectly, to change the value of a constant. You did not, of course, try "2 = 1", because the compiler catches that.

Example: HPE20508 Modification of a read-only value attempted

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: But an easy way to do the same thing is:

```
sub mod { $_[0] = 1 }
mod(2);
```

Another way is to assign to a substr() that's off the end of the string.

Yet another way is to assign to a C<foreach> loop I<VAR> when I<VAR> is aliased to a constant in the look I<LIST>:

```
 $x = 1;
foreach my $n ($x, 2) {
    $n *= 2; # modifies the $x,
             # but fails on
             # attempt to modify
             # the 2
 }
```

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20509 Out of memory!

Explanation: The malloc() function returned 0, indicating there was not enough remaining memory (or virtual memory) to satisfy the request. Perl has no option but to exit immediately.

Example: HPE20509 Out of memory!

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to see if you are using more than 64k bytes of memory.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: X- A very fatal error (non-trappable).

Automation: Not applicable.

HPE20510 Insecure dependency in *function state*

Explanation: You tried to do something that the tainting mechanism did not like. The tainting mechanism is turned on when you are running setuid or setgid, or when you specify **-T** to turn it on explicitly.

In the message text:

function
A function

state
The state Perl was in which contributed to the problem.

Example: HPE20510 Insecure dependency in glob while running with -T

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The tainting mechanism labels all data that is derived directly or indirectly from the user. If any such data is used in a "dangerous" operation, you get this error.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20511 Unsupported socket function "*function*" called

Explanation: Your machine does not support the Berkeley socket mechanism, or at least that is what Configure thought.

In the message text:

function
functions (socket functions)

Example: HPE20511 Unsupported socket function "recv" called

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether your machine supports Berkeley socket or has been configured to use it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20512 Unsupported directory function "*function*" called

Explanation: Your machine does not support opendir() and readdir().

In the message text:

function
functions (directory function)

Example: HPE20512 Unsupported directory function "readdir" called

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether your machine supports opendir() and readdir().

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20513 The *function* function is unimplemented

Explanation: The function indicated is not implemented on this architecture, according to the probings of Configure.

In the message text:

function
The unimplemented function

Example: HPE20513 The pipe function is unimplemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Function is not implemented in your system. Contact system administrator.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20514 "my" variable *var* can't be in a package

Explanation: Lexically scoped variables aren't in a package, so it doesn't make sense to try to declare one with a package qualifier on the front.

In the message text:

var
variable name

Example: HPE20514 "my" variable fp can't be in a package

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use local() if you want to localize a package variable.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20515 Can't localize through a reference

Explanation: You said something like `local $$ref` , which Perl can't currently handle. When it goes to restore the old value of whatever $ref pointed to after the scope of the local() is finished, it can't be sure that $ref will still be a reference.

Example: HPE20515 Can't localize through a reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Be sure that you are not localizing the reference.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20516 panic: memory wrap

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20516 panic: memory wrap

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20517 Goto undefined subroutine &*package*

Explanation: This is displaced when you try to debug the executable file of error script.

In the message text:

package
package name

Example: HPE20517 Goto undefined subroutine &Apache::Constants::SERVER_ERROR

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you are running a error script you will get this error. Correct the error in the script and then run the script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20518 Goto undefined subroutine

Explanation: The subroutine specified on the goto statement is not defined.

Example: HPE20518 Goto undefined subroutine

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Verify that the name has not been misspelled and that the subroutine has been included into your program.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20519 *str* **returned from lvalue subroutine in scalar context**

Explanation: In scalar array is returned by lvalue subroutine.

In the message text:

str
string

Example: HPE20519 Array returned from lvalue subroutine in scalar context

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You are trying to something like this.

```
sub lv0 : lvalue { };
  lv0 = (2,3);
```

In second line left hand side is in scalar context and left hand side in list context to fix this you can give something like this.

```
sub lv0 : lvalue { };
  (lv0) = (2,3);
```

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20520 no argument for locked method call

Explanation: The lock method was called without any argument.

Example: HPE20520 no argument for locked method call

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: lock method call expect argument.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20521 bad top format reference

Explanation: top format reference is not correct.

Example: HPE20521 bad top format reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Change the top format reference, using the correct syntax.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20522 Possible memory corruption: *function* **overflowed 3rd argument**

Explanation: An ioctl() or fcntl() returned more than Perl was bargaining for. Perl guesses a reasonable buffer size, but puts a sentinel byte at the end of the buffer just in case. This sentinel byte got clobbered, and Perl assumes that memory is now corrupted.

In the message text:

function
function

Example: HPE20522 Possible memory corruption: ioctl overflowed 3rd argument

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure you have passed the correct number of argument for the ioctl or fcntl functions.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20523 Argument to lock needs to be passed as ref

Explanation: lock function is passed with non-reference argument. The argument to lock has to be reference variable.

Example: HPE20523 Argument to lock needs to be passed as ref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the argument you are passing to the lock function is a reference variable.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20524 Argument to share needs to be passed as ref

Explanation: share function take only reference as the argument.

Example: HPE20524 Argument to share needs to be passed as ref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that you are passing a reference to the share function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20525 panic: destruct destroyed thread *thread* **(***string***)**

Explanation: An internal error occurred that cannot be resolved by the user.

In the message text:

thread
thread

string
reason

Example: HPE20525 panic: destruct destroyed thread new (somereason)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20526 panic: cannot find thread data

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20526 panic: cannot find thread data

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20527 Cannot join a detached thread

Explanation: After a thread has been detached, it runs until it is finished. Perl will then automatically clean up after it.

Example: HPE20527 Cannot join a detached thread

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Try to check whether the thread is still alive before joining it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20528 Thread already joined

Explanation: You cannot join a thread that has already been joined. If you try to do that, an exception will be raised.

Example: HPE20528 Thread already joined

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you want to join a thread that has already been joined, you must free the thread first before joining it again.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20529 destruct *tid* **@** *string* **by** *thread* **now** *known_threads*

Explanation: Clean up after thread is done with thread_destruct is called. This is an information-only message.

In the message text:

num
number.

thread
thread

string
reason

known_thread
thread which does not have thread id

Example: HPE20529 destruct 45 @ tinterp by thread now knownthread.

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: No action needs to be taken because this is an information-only message.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20530 CLONE *obj*

Explanation: Joining the thread, this code needs to take the return value from the call_sv and send it back. If the object being cloned is not valid, this error will occur.

Example: HPE20530 CLONE SocketPac::unpack.

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If cloning is attempted on a non valid object, this error will occur. Determine the cause and try re-running the script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20531 Argument to cond_wait needs to be passed as ref

Explanation: Arugument to cond_wait is not a reference.

Example: HPE20531 Argument to cond_wait needs to be passed as ref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that you are passing the right argument to cond_wait.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20532 cond_wait lock needs to be passed as ref

Explanation: reference argument is expected at the error line.

Example: HPE20532 cond_wait lock needs to be passed as ref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Pass the refrence argument for cond_wait.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20533 Argument to cond_timedwait needs to be passed as ref

Explanation: Reference argument is expected at the error line.

Example: HPE20533 Argument to cond_timedwait needs to be passed as ref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check whether the argument at the error line is a reference.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20534 cond_timedwait lock needs to be passed as ref

Explanation: reference argument is expected at the error line.

Example: HPE20534 cond_timedwait lock needs to be passed as ref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure cond_timedwait lock has only reference argument.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20535 Argument to cond_signal needs to be passed as ref

Explanation: cond_signal function expect reference argument.

Example: HPE20535 Argument to cond_signal needs to be passed as ref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure you are passing a reference argument for cond_signal.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20536 Cannot share globs yet

Explanation: sharing of perl globs is not allowed.

Example: HPE20536 Cannot share globs yet

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not share perl globs.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20537 Cannot share subs yet

Explanation: sharing of perl subs is not allowed.

Example: HPE20537 Cannot share subs yet

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not share perl subs.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20538 cond_timedwait not supported on this platform

Explanation: cond_timedwait not supported in this platform.

Example: HPE20538 cond_timedwait not supported on this platform

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You cannot use cond_timedwait because this funciton is not supported in this platform.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20539 panic: cond_timedwait (*num*)

Explanation: An internal error occurred that cannot be resolved by the user.

In the message text:

num
number

Example: HPE20539 panic: cond_timedwait (100)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20540 panic: cond_timedwait

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20540 panic: cond_timedwait

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20541 Invalid value for shared scalar

Explanation: This occurs when you use the shared attribute on a blessed object.

Example: HPE20541 Invalid value for shared scalar

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use shared attributes for blessed objects.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20542 *thread* **waiting - owned by** *fileowner filename*:*linenum*

Explanation: When DEBUG_LOCKS is set, one can lock the particular line of the file. If another person trys to access it, this error will be showed.

In the message text:

thread
thread

fileowner
owner of file

filename
Name of file.

linenum
line number which is locked.

Example: HPE20542 process1 waiting - owned by ISLDEV test.pm:1000

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Warning message showed you need to wait until lock is released,

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20543 *refvar* **s=***var* **has no shared SV**

Explanation: Not able to find the pointer for the shared variable.

In the message text:

var
Shared variable

refvar
The variable that has the pointer.

Example: HPE20543 vai s=var has no shared SV

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure the syntax for sharing variables is correct.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20544 variable is not shared

Explanation: Not able to find the shared variable.

In the message text:

variable
variable

Example: HPE20544 $hashref is not shared

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that you are searching for the correct shared variable.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20545 Explicit blessing to '' (assuming package main)"

Explanation: You are blessing a reference to a zero length string. This has the effect of blessing the reference into the package main.

Example: HPE20545 Explicit blessing to '' (assuming package main)"

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: You may want to provide a default target package. For example: bless($ref, $p "" 'MyPackage');

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20546 panic: cond_timedwait-reset

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20546 panic: cond_timedwait-reset

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20547 panic: join with a thread with strange ordinal *num*

Explanation: An internal error occurred that cannot be resolved by the user.

In the message text:

num
Number.

Example: HPE20547 panic: join with a thread with strange ordinal 4

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20548 panic: join with a thread which could not start

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20548 panic: join with a thread which could not start

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20549 Use of /c modifier is meaningless without /g

Explanation: (W regexp) You used the /c modifier with a regex operand, but didn't use the /g modifier.

Example: HPE20549 Use of /c modifier is meaningless without /g

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use /g if you want to use the /c modifier. (This may change in the future.)

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20550 Use of /c modifier is meaningless in s///

Explanation: You used the /c modifier in a substitution. This has no value.

Example: HPE20550 Use of /c modifier is meaningless in s///

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use the /c modifier in substitutions because it will not work.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20559 Too late to run INIT block

Explanation: A INIT block is being defined during run time proper, when the opportunity to run them has already passed.

Example: HPE20559 Too late to run INIT block

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Perhaps you are loading a file with `require` or `do` when you should be using `use` instead. Or perhaps you should put the `require` or `do` inside a BEGIN block

System programmer response: No System

Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20560 panic: cannot push :perlio for *PerlIO*

Explanation: An internal error occurred that cannot be resolved by the user.

In the message text:

PerlIO
Perl io

Example: HPE20560 panic: cannot push :perlio for utf8

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20561 panic: decode did not return a value

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20561 panic: decode did not return a value

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20562 panic: decode did not return UTF-8 '*string***'**

Explanation: This is an internal error that cannot be resolved by the user.

In the message text:

string
string

Example: HPE20562 panic: decode did not return UTF-8 'HYPHEN'

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Please follow your local procedures for reporting a problem to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20563 panic: encode did not return a value

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20563 panic: encode did not return a value

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20564 panic: opcode "*string*" value *num* is invalid

Explanation: An internal error occurred that cannot be resolved by the user.

In the message text:

string
operation name

num
operation code

Example: HPE20564 panic: opcode "READ" value 64 is invalid

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20565 panic: invalid bitspec for "*string*" (type *bitspec*)

Explanation: This is an internal error that cannot be resolved by the user.

In the message text:

string
operation name

bitspec
bits specification

Example: HPE20565 panic: invalid bitspec for "shift" (type 0x77ef)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Please follow your local procedures for reporting a problem to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20566 panic: opcode *num* (*string*) out of range

Explanation: An internal occur occurred that cannot be resolved by the user.

In the message text:

string
operation name

num
operation code

Example: HPE20566 panic: opcode 68 (read) out of range

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20567 panic: unexpected simple REx opcode *num*

Explanation: An internal error occurred that cannot be resolved by the user.

In the message text:

num
operation code

Example: HPE20567 panic: unexpected simple REx opcode 68

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20568 panic: pregfree comppad

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20568 panic: pregfree comppad

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20569 panic: regfree data code '*code*'

Explanation: An internal error occurred that cannot be resolved by the user.

In the message text:

code
codes

Example: HPE20569 panic: regfree data code 'd'

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20570 panic: Devel::DProf inconsistent subroutine return

Explanation: Devel::DProf called a subroutine that exited using goto(LABEL), last(LABEL) or next(LABEL). Leaving that way a subroutine called from an XSUB will likely lead to a crash of the interpreter. This is a bug that may be fixed in the future.This is an internal error that cannot be resolved by the user.

Example: HPE20570 panic: Devel::DProf inconsistent subroutine return

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20571 panic: top_env

Explanation: The compiler attempted to do a goto, or something similar. This is an internal error that cannot be resolved by the user.

Example: HPE20571 panic: top_env

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20572 panic: restartop

Explanation: An internal routine requested a goto (or something like it), and didn't supply the destination. When you are using goto, you have to mention the destination label. This is an internal error that cannot be resolved by the user.

Example: HPE20572 panic: restartop

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20573 panic: pp_split

Explanation: Something terrible went wrong in setting up for the split. This is an internal error that cannot be resolved by the user.

Example: HPE20573 panic: pp_split

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting a problem to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20574 panic: unlock_condpair unlocking non-mutex

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20574 panic: unlock_condpair unlocking non-mutex

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20575 panic: unlock_condpair unlocking mutex that we don't own

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20575 panic: unlock_condpair unlocking mutex that we don't own

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20576 panic: mapstart

Explanation: The compiler is confused with respect to the map() function. This is an internal error that cannot be resolved by the user.

Example: HPE20576 panic: mapstart

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20577 panic: bad gimme: *num*

Explanation: An internal error occurred that cannot be resolved by the user.

In the message text:

num
number

Example: HPE20577 panic: bad gimme: 5

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20578 panic: return

Explanation: We popped the context stack to a subroutine or eval context, and then discovered it wasn't a subroutine or eval context. This is an internal error that cannot be resolved by the user.

Example: HPE20578 panic: return

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20579 panic: last

Explanation: We popped the context stack to a block context, and then discovered it wasn't a block context. This is an internal error that cannot be resolved by the user.

Example: HPE20579 panic: last

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20580 panic: goto

Explanation: We popped the context stack to a context with the specified label, and then discovered it wasn't a context we know how to do a goto in. This is an internal error that cannot be resolved by the user.

Example: HPE20580 panic: goto

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20581 panic: die

Explanation: We popped the context stack to an eval context, and then discovered it wasn't an eval context. This is an internal error that cannot be resolved by the user.

Example: HPE20581 panic: die

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20582 panic: pp_match

Explanation: The internal pp_match() routine was called with invalid operational data. This is an internal error that cannot be resolved by the user.

Example: HPE20582 panic: pp_match

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20583 panic: pp_match start/end pointers

Explanation: The internal pp_match() routine was called with invalid operational data. This is an internal error that cannot be resolved by the user.

Example: HPE20583 panic: pp_match start/end pointers

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20584 panic: pp_iter

Explanation: The foreach iterator got called in a non-loop context frame. This is an internal error that cannot be resolved by the user.

Example: HPE20584 panic: pp_iter

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20585 panic: pp_subst

Explanation: The internal pp_subst() routine was called with invalid operational data. This is an internal error that cannot be resolved by the user.

Example: HPE20585 panic: pp_subst

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20586 panic: sv_setpvn called with negative strlen

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20586 panic: sv_setpvn called with negative strlen

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20587 panic: del_backref

Explanation: Failed an internal consistency check while trying to reset a weak reference. This is an internal error that cannot be resolved by the user.

Example: HPE20587 panic: del_backref

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20588 panic: sv_insert

Explanation: The sv_insert() routine was told to remove more string than there was string. This is an internal error that cannot be resolved by the user.

Example: HPE20588 panic: sv_insert

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20589 panic: sv_pos_b2u: bad byte offset

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20589 panic: sv_pos_b2u: bad byte offset

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20590 semi-panic: attempt to dup freed string

Explanation: The internal newSVsv() routine was called to duplicate a scalar that had previously been marked as free. This is an internal error that cannot be resolved by the user.

Example: HPE20590 semi-panic: attempt to dup freed string

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20591 panic: frexp

Explanation: The library function frexp() failed, making printf("%f") impossible. This is an internal error that cannot be resolved by the user.

Example: HPE20591 panic: frexp

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20592 panic: ss_dup inconsistency

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20592 panic: ss_dup inconsistency

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20593 panic: malloc

Explanation: Something requested a negative number of bytes of malloc. This is an internal error that cannot be resolved by the user.

Example: HPE20593 panic: malloc

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Increase the size of bucket because it requires a positive number of bytes. "1 if does not fit, -1 if easily fits in a smaller bucket, otherwise 0".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20594 panic: realloc

Explanation: Something requested a negative number of bytes of realloc. This is an internal error that cannot be resolved by the user.

Example: HPE20594 panic: realloc

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Increase the size of bucket because it requires a positive number of bytes. "1 if does not fit, -1 if easily fits in a smaller bucket, otherwise 0".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20595 panic: calloc

Explanation: Something requested a negative number of bytes of realloc. This is an internal error that cannot be resolved by the user.

Example: HPE20595 panic: calloc

System action: Perl immediately stops interpreting the

file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20596 panic: pthread_getspecific

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20596 panic: pthread_getspecific

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20597 panic: pthread_setspecific

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20597 panic: pthread_setspecific

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20598 panic: perl_cond_wait called by last runnable thread

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20598 panic: perl_cond_wait called by last runnable thread

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20599 panic: no strftime

Explanation: An internal error occurred that cannot be resolved by the user.

Example: HPE20599 panic: no strftime

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20600 NULL regexp parameter

Explanation: The internal pattern matching routines are out of control. This is an internal error that cannot be resolved by the user.

Example: HPE20600 NULL regexp parameter

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Parameter regular expression should not be Null. Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20601 corrupted regexp program

Explanation: The regular expression engine got passed a regexp program without a valid magic number. This is an internal error that cannot be resolved by the user.

Example: HPE20601 corrupted regexp program

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20602 regexp memory corruption

Explanation: The regular expression engine got confused by what the regular expression compiler gave it. This is an internal error that cannot be resolved by the user.

Example: HPE20602 regexp memory corruption

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the regular expression.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20603 corrupted regexp pointers

Explanation: The regular expression engine got confused by what the regular expression compiler gave it. This is an internal error that cannot be resolved by the user.

Example: HPE20603 corrupted regexp pointers

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the regular expression.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20604 regexp unwind memory corruption

Explanation: The regular expression engine got confused by what the regular expression compiler gave it. This is an internal error that cannot be resolved by the user.

Example: HPE20604 regexp unwind memory corruption

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20605 List form of piped open not implemented

Explanation: The list form of `piped open` is not implemented for perl version 5.007003 and below.

Example: HPE20605 List form of `piped open` not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you type the following command line syntax while using perl version 5.007003 and below, the subsequent error will occur:

```
perl -le "print $]; open(P, '-"', $^X, '-e', 1)
or die $!"
```

5.007003

List form of **piped open** not implemented at -e line 1.

%SYSTEM-F-ABORT, abort

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20606 Unknown Unicode option letter *opt*

Explanation: You specified an unknown Unicode option. See Running the Perl interpreter documentation of the `-C` switch for a list of known options.

In the message text:

opt
The Option.

Example: HPE20606 Unknown Unicode option letter -d

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the correct letter of the Unicode option.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20607 Unknown Unicode option value *opt*

Explanation: You specified an unknown Unicode option. See Running the Perl interpreter documentation of the `-C` switch for the list of known options.

In the message text:

opt
The Option.

Example: HPE20607 Unknown Unicode option value -C7

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Provide the correct Unicode option value.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20608 Filehandle *filehandle* opened only for *str* put

Explanation: A file is opened in read mode and you are trying to write to it. Or, a file is opened in write mode and you are trying to read it.

In the message text:

filehandle
file handler.

str
string

Example: HPE20608 Filehandle FH opened only for input

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If a file is opened in read mode, you cannot write to it. Similarly, if a file is opened in write mode, you cannot read it. Make sure to only write to files in write mode, and read from files in read mode.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20609 Filehandle opened only for *str* put

Explanation: A file is opened in read mode, and you are trying to write to it. Or, a file is opened in write mode, and you are trying to read from it.

In the message text:

str
string (in or out)

Example: HPE20609 Filehandle opened only for input

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If a file is opened in read mode, you can only read the file. Similarly, if a file is opened in write mode, you can only write to the file.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20610 (Are you trying to call *functionpars* on dirhandle *filehandle*?)

Explanation: You could flock on a normal file that is uniquely associated to each directory (like "${dir}.lock". Since flock is advisory only, you cannot rely on it outside your application.

In the message text:

function
function name

pars
parantheses

filehandle
file handler

Example: HPE20610 \t(Are you trying to call flock() on dirhandle SOMEDIR?)\n

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not depend on flock outside your application.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20611 (Are you trying to call *function pars* on dirhandle?)

Explanation: You can flock on a normal file that is uniquely associated to each directory (like "${dir}.lock". Since flock is advisory only, you cannot depend on it outside your application.

In the message text:

function
function name

pars
parantheses

Example: HPE20611 (Are you trying to call %s%s on dirhandle?)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not rely on flock outside your application.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20612 A variable may not be unshared

Explanation: An attempt was made to unshare a variable; this is not allowed.

Example: HPE20612 A variable may not be unshared

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: There are certain shared variables that cannot be unshared. Try to avoid unsharing these variables.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20613 Usage: attributes::boots trap $module

Explanation: A usage error occurred while using the `bootstrap` function. It may be a result of a mismatched argument.

Example: HPE20613 Usage: attributes::boots trap $module

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure the `bootstrap` function is being used correctly.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20614 Usage: attributes::_modify_attrs $reference, @attributes

Explanation: A usage error occured while using the `attributes::_modify_attrs` function. It may be a result of a mismatched argument.

Example: HPE20614 Usage: attributes::_modify_attrs $reference, @attributes

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure the `attributes__modify_attrs` function is being used correctly.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20615 Usage: attributes::_fetch_attrs $reference

Explanation: A usage error occured while using the `attributes::_fetch_attrs` function. It may be a result of a mismatched argument.

Example: HPE20615 Usage: attributes::_fetch_attrs $reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure the `attributes::_fetch_attrs` function is being used correctly.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20616 Usage: attributes::_guess_stash $reference

Explanation: A usage error occured while using the `attributes::_guess_stash` function. It may be a result of a mismatched argument.

Example: HPE20616 Usage: attributes::_guess_stash $reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure the `attributes__guess_stash` function is being used correctly.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20617 Usage: attributes::reftype $reference

Explanation: A usage error occured while using the `attributes::_reftype` function. It may be a result of mismatched arguments.

Example: HPE20617 Usage: attributes::reftype $reference

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure the `attributes::_reftype` function is being used correctly.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20618 Usage: attributes::_warn_reserved ()

Explanation: A usage error occured while using the `attributes::_warn_reserved` function. It may be because of a mismatch of arguments.

Example: HPE20618 Usage: attributes::_warn_reserved ()

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure the `attributes__warn_reserved` function is being used correctly.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20619 Size magic not implemented

Explanation: The size of the sv magic variable is not implemented.

Example: HPE20619 Size magic not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The size of the magic variable is not implemented. Therefore, you can not use the `mg_size` function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20620 No such hook: *hook*

Explanation: Internal hook set using the %SIG hash key, which was passed for the %SIG, is not correct.

In the message text:

hook
hook name (eg. __DIE__)

Example: HPE20620 No such hook: ALRM

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure the hook being used is correct.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20621 SIG*Signal* **handler "***signalhandler***" not defined.**

Explanation: The signal handler named in %SIG doesn't, in fact, exist. Perhaps you put it into the wrong package.

In the message text:

Signal
Signal Example INT,IO etc

signalhandler
signal handler basically function

Example: HPE20621 SIGINT handler "signal_inite" not defined.

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check if the signal handler name is correct or you are using the right package,

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20622 No such signal: SIG*signal*

Explanation: You specified a signal name as a subscript to %SIG that was not recognized.

In the message text:

signal
Signal name which is not recognized from the system.

Example: HPE20622 No such signal: SIGABT

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Try using `kill -l` in your shell session to see a list of valid signal names on your system.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20623 Do not know how to handle magic of type *var*

Explanation: The internal handling of magical variables has been cursed. This is an internal error that cannot be resolved by the user.

In the message text:

var
The variable name.

Example: HPE20623 Do not know how to handle magic of type \how

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20624 Bizarre SvTYPE [*num*]

Explanation: On some platforms such as Windows where the fork() system call is not available, Perl can be built to emulate fork() at the interpreter level.

In the message text:

num
number

Example: HPE20624 Bizarre SvTYPE [139]

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The best solution is to get a version of Perl that works properly with fork() commands or find another method for running the scripts.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20625 Cloning substitution context is unimplemented

Explanation: The context type is cloning substitution, but it is not implemented.

Example: HPE20625 Cloning substitution context is unimplemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Cloning substitution context is not implemented. Check to make sure the context type of the cloning substitution is implemented.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20626 Invalid argument to sv_cat_decode

Explanation: Out of 5 arguments, there may be invalid argument passed to sv_cat_decode.

Example: HPE20626 Invalid argument to sv_cat_decode

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure to pass the appropriate argument, correct the error, and then re-run the script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20627 Undefined value assigned to typeglob

Explanation: An undefined value was assigned to a typeglob, a `*foo = undef`. This does nothing.

Example: HPE20627 Undefined value assigned to typeglob

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Make sure to use the correct syntax; the syntax should be `undef *foo`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20628 Reference is already weak

Explanation: You have attempted to weaken a reference that is already weak.

Example: HPE20628 Reference is already weak

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check if a reference has already been weakened or not by using isweak() before weakening a reference.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20629 Reference miscount in sv_replace()

Explanation: The internal `sv_replace()` function was handed a new SV with a reference count of other than 1.

Example: HPE20629 Reference miscount in sv_replace()

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Reference count should be 1 in `sv_replace()`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20630 Newline in left-justified string for *x*printf

Explanation: There is a newline in a string to be left justified by `printf` or `sprintf`. The padding spaces will appear after the newline.

In the message text:

x will be "s" if the function is "sprintf" or blank if "printf".

Example: HPE20630 Newline in left-justified string for sprintf

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Remove the newline from the string and put formatting characters in the `sprintf` format.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20631 Cannot create %s::ISA

Explanation: Cannot create the package to place in ISA.

In the message text:

package
The package name

Example: HPE20631 Cannot create main::ISA

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the syntax of the package.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20632 Use of inherited AUTOLOAD for non-method *method*::*fun()* is deprecated

Explanation: As an accidental feature, `AUTOLOAD` subroutines are looked up as methods (using the `@ISA` hierarchy) even when the subroutines to be autoloaded were called as plain functions (e.g.`Foo::bar()`), not as methods (e.g. `< Foo->bar() >` or `<$obj->bar() >`). Perl issues an optional warning when non-methods use inherited `AUTOLOAD`s. The simple rule is: inheritance will not work when autoloading non-methods.

In the message text:

method
The method name.

function
The function name.

Example: HPE20632 Use of inherited AUTOLOAD for non-method Foo::bar() is deprecated

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: In any module that used to depend on inheriting `AUTOLOAD` for non-methods from a base class named `BaseClass`, execute `*AUTOLOAD = \&BaseClass::AUTOLOAD` during startup. In code that says `use AutoLoader; @ISA = qw(AutoLoader);` you should remove AutoLoader from @ISA and change `use AutoLoader;` to `use AutoLoader 'AUTOLOAD';`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: D- A deprecation (optional).

Automation: Not applicable.

HPE20633 Use of $*str* is deprecated

Explanation: If you are using $# , it was an attempt to emulate a poorly defined **awk** feature. If you are using $* , this variable turned on multi-line pattern matching, both for you and any subroutines that you happen to call.

In the message text:

str
The string may be # or *. .

Example: HPE20633 Use of $# is deprecated

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use an explicit `printf()` or `sprintf()` instead for $# . For $* , you should use the `//m` and `//s` modifiers to do that, without the dangerous action-at-a-distance effects of `$*`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: D- A deprecation (optional).

Automation: Not applicable.

HPE20634 Name "*str1*::*str2*" used only once: possible typo

Explanation: Typographical errors often show up as unique variable names. If you had a good reason for having a unique name, then just mention it again somehow to suppress the message.

In the message text:

str1
The string.

str2
The string.

Example: HPE20634 Name "main::files" used only once: possible typo

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: The `our` declaration is provided to supress the message purpose.

NOTE: This warning detects symbols that have been used only once, so $c, @c, %c, *c, &c, sub c{}, c(), and c (the filehandle or format) are considered the same; if a program uses $c only once but also uses any of the others, it will not trigger this warning.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20635 substr outside of string

Explanation: You tried to reference a substr() that pointed outside of a string. That is, the absolute value of the offset was larger than the length of the string. See substr. This warning is fatal if substr is used in an lvalue context (as the left hand side of an assignment or as a subroutine argument, for example).

Example: HPE20635 substr outside of string

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Substring should be within the range.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20636 Constant subroutine *sub* undefined

Explanation: You undefined a subroutine which had previously been eligible for inlining. See Perl subroutines for commentary and workarounds.

In the message text:

sub
The subroutine.

Example: HPE20636 Constant subroutine subPI() undefined

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Previously compiled invocations of the function will still be using the old value of the function. If you need to be able to redefine the subroutine, you need to ensure that it is not inlined, either by dropping the () prototype (which changes calling semantics, so beware) or by thwarting the inlining mechanism in some other way, such as

```
sub not_inlined () {
  23 if $];
}
```

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20637 Odd number of elements in anonymous hash

Explanation: You specified an odd number of elements to initialize a hash, which is unusual, because hashes come in key/value pairs.

Example: HPE20637 Odd number of elements in anonymous hash

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Select an even number of elements to initialize a hash.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20638 splice() offset past end of array

Explanation: You attempted to specify an offset that was past the end of the array passed to splice(). Splicing will instead commence at the end of the array, rather than past it. See splice.

Example: HPE20638 splice() offset past end of array

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Try explicitly pre-extending the array by assigning `$#array = $offset`.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20639 Use of freed value in iteration

Explanation: Perhaps you modified the iterated array within the loop? This error is typically caused by code like the following:

```
@a = (3,4);
@a = () for (1,2,@a);
```

For speed and efficiency reasons, Perl internally does not do full reference-counting of iterated items. Hence, deleting such an item in the middle of an iteration causes Perl to see a freed value.

Example: HPE20639 Use of freed value in iteration

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure not to modify arrays while they are being iterated over.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20640 Reference found where even-sized list expected

Explanation: You gave a single reference where Perl was expecting a list with an even number of elements (for assignment to a hash).

Example: HPE20640 Reference found where even-sized list expected

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: This usually means that you used the anon hash constructor when you meant to use parens. In any case, a hash requires key/value **pairs**.

%hash = { one => 1, two => 2, }; # WRONG

%hash = [qw/ an anon array /]; # WRONG

%hash = (one => 1, two => 2,); # right

%hash = qw(one 1 two 2); # also fine

Use even numbered lists only.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20641 Odd number of elements in hash assignment

Explanation: You specified an odd number of elements to initialize a hash, which is unusual, because hashes come in key/value pairs.

Example: HPE20641 Odd number of elements in hash assignment

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use even number of elements to initialize a hash.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20642 glob failed (can't start child: *errnodisplay*)

Explanation: Something went wrong with the external program(s) used for `glob` and < <*.c> >. Usually, this means that you supplied a `glob` pattern that caused the external program to fail and exit with a nonzero status. If the message indicates that the abnormal exit resulted in a coredump, this may also mean that your csh (C shell) is broken.

In the message text:

errnodisplay
Error number with its description.

Example: HPE20642 glob failed (can't start child: 3362 no such file)

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Change all of the csh-related variables in config.sh: If you have tcsh, make the variables refer to it as if it were csh (e.g. `full_csh='/usr/bin/tcsh'`); otherwise, make them all empty (except that `d_csh` should be `'undef'`) so that Perl will think csh is missing. In either case, after editing config.sh, run `./Configure -S` and rebuild Perl.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20643 glob failed (child exited with status *num errnodisplay*)

Explanation: Something went wrong with the external program(s) used for *glob* and *< <*.c> >*. Usually, this means that you supplied a *glob* pattern that caused the external program to fail and exit with a nonzero status. If the message indicates that the abnormal exit resulted in a coredump, this may also mean that your csh (C shell) is broken.

In the message text:

num
Number.

errnodisplay
Error number with its description.

Example: HPE20643 glob failed (child exited with status 1,coredump)

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: You should change all of the csh-related variables in config.sh: If you have tcsh, make the variables refer to it as if it were csh (e.g. *full_csh='/usr/bin/tcsh'*); otherwise, make them all empty (except that *d_csh* should be *'undef'*) so that Perl will think csh is missing. In either case, after editing config.sh, run *./Configure -S* and rebuild Perl.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20644 utf8 "\x*unicode*" does not map to Unicode

Explanation: When reading in different encodings, Perl tries to map everything into Unicode characters. The bytes you read in are not legal in this encoding, for example utf8 "\xE4" does not map to Unicode if you try to read in the a-diaereses Latin-1 as UTF-8.

In the message text:

unicode
The unicode character.

Example: HPE20644 utf8 "\xE4" does not map to Unicode

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check the Unicode character to determine whether it is present in that encoding.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20645 Deep recursion on anonymous subroutine

Explanation: This subroutine has called itself indirectly 100 times more than it has returned.

Example: HPE20645 Deep recursion on anonymous subroutine

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: This probably indicates an infinite recursion, unless you're writing unusual benchmark programs, in which case it indicates something else.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20646 Deep recursion on subroutine "*sub*"

Explanation: This subroutine has called itself (directly or indirectly) 100 times more than it has returned.

In the message text:

sub
The subroutine name.

Example: HPE20646 Deep recursion on subroutine "main::sum"

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: This probably indicates an infinite recursion, unless you are writing unusual benchmark programs, in which case it indicates something else.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20647 Use of reference "*str*" as array index

Explanation: You tried to use a reference as an array index; this is probably not what you meant, because references in numerical contexts tend to be huge numbers, and therefore usually indicates programmer error.

In the message text:

str
The string may be a refrence.

Example: HPE20647 Use of reference "ref" as array index

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If you intended to use the reference as an array index, explicitly numify your reference, like so: `$array[0+$ref]`. This warning is not given for overloaded objects, because you can overload the numification and stringification operators.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20648 Possible Y2K bug: *str*

Explanation: You are concatenating the number 19 with another number, which could be a potential Year 2000 problem.

In the message text:

str
The string.

Example: HPE20648 Possible Y2K bug: 00

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: This warning can only be seen when specified by -DPERL_Y2KWARN configuration option. Normally, this warning message is not seen as Perl is Y2K complaint.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20649 About to append an integer to '19'

Explanation: This message is displayed with HPE20648 and clarifies the reason for the Y2K warning.

Example: HPE20648 Possible Y2K bug: HPE20649 about to append an integer to '19'

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure not to append an integer to '19'.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20650 panic: unknown argument type *num*, **line** *num*

Explanation: This is an internal error that cannot be resolved by the user.

In the message text:

num
number

Example: HPE20650 panic: unknown argument type 10, line 20

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20651 panic: unknown argument type *num*, **arg** *num*, **line** *num*

Explanation: This is an internal error that cannot be resolved by the user.

In the message text:

num
number

Example: HPE20651 panic: unknown argument type 4, arg 5, line 11

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20652 panic: aryrefarg *num*, line *num*

Explanation: This is an internal error that cannot be resolved by the user.

In the message text:

num
number

Example: HPE20652 panic: aryrefarg 6, line 11

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20653 panic: PerlIO layer array corrupt

Explanation: If the PerlIO layer is mentioned but not defined, this error will occur.

Example: HPE20653 panic: PerlIO layer array corrupt

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20654 panic: sysconf: *string*

Explanation: This is an internal error that cannot be resolved by the user.

In the message text:

string
string

Example: HPE20654 panic: sysconf: pagesize not known

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20655 panic: sysconf: pagesize unknown

Explanation: This is an internal error that cannot be resolved by the user.

Example: HPE20655 panic: sysconf: pagesize unknown

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20656 panic: bad pagesize *num*

Explanation: This is an internal error that cannot be resolved by the user.

In the message text:

num
number

Example: HPE20656 panic: bad pagesize = -1064

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20657 panic: sprintf overflow - memory corrupted!

Explanation: This is an internal error that cannot be resolved by the user.

Example: HPE20657 panic: sprintf overflow - memory corrupted!

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20658 panic: Off-page sbrk

Explanation: This is an internal error that cannot be resolved by the user.

Example: HPE20658 panic: Off-page sbrk

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20659 panic: do_trans_simple line *n*

Explanation: The internal do_trans routines were called with incorrect operational data. This is an internal error that cannot be resolved by the user.

In the message text:

n The line number.

Example: HPE20659 panic: do_trans_simple line 9

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20660 panic: do_trans_count line *n*

Explanation: The internal do_trans routines were called with incorrect operational data. This is an internal error that cannot be resolved by the user.

In the message text:

n The line number.

Example: HPE20660 panic: do_trans_count line 9

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20661 panic: do_trans_complex line *n*

Explanation: The internal do_trans routines were called with incorrect operational data. This is an internal error that cannot be resolved by the user.

In the message text:

n The line number.

Example: HPE20661 panic: do_trans_complex line 7

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20662 panic: do_trans_simple_utf8 line *n*

Explanation: The internal do_trans routines were called with incorrect operational data. This is an internal error that cannot be resolved by the user.

In the message text:

n The line number.

Example: HPE20662 panic: do_trans_simple_utf8 line 8

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20663 panic: do_trans_complex_utf8 line *n*

Explanation: The internal do_trans routines were called with incorrect operational data. This is an internal error that cannot be resolved by the user.

In the message text:

n The line number.

Example: HPE20663 panic: do_trans_complex_utf8 line 5

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20664 panic: magic_len: *num*

Explanation: This is an internal error that cannot be resolved by the user.

In the message text:

num
number

Example: HPE20664 panic: magic_len: 150

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20665 panic: magic_killbackrefs

Explanation: Failed an internal consistency check while trying to reset all weak references to an object. This is an internal error that cannot be resolved by the user.

Example: HPE20665 panic: magic_killbackrefs

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20666 panic: magic_mutexfree

Explanation: This is an internal error that cannot be resolved by the user.

Example: HPE20666 panic: magic_mutexfree

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20667 panic: pad_alloc

Explanation: The compiler got confused about which scratch pad it was allocating, as well as freeing, temporaries and lexicals from. This is an internal error that cannot be resolved by the user.

Example: HPE20667 panic: pad_alloc

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20668 panic: pad_sv po

Explanation: An invalid scratch pad offset was detected internally. This is an internal error that cannot be resolved by the user.

Example: HPE20668 panic: pad_sv po

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20669 panic: pad_swipe curpad

Explanation: The compiler got confused about which scratch pad it was allocating, as well as freeing, temporaries and lexicals from. This is an internal error that cannot be resolved by the user.

Example: HPE20669 panic: pad_swipe curpad

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20670 panic: pad_swipe po

Explanation: An invalid scratch pad offset was detected internally. This is an internal error that cannot be resolved by the user.

Example: HPE20670 panic: pad_swipe po

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20671 panic: pad_reset curpad

Explanation: The compiler got confused about which scratch pad it was allocating and freeing temporaries and lexicals from. This is an internal error that cannot be resolved by the user.

Example: HPE20671 panic: pad_reset curpad

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20672 panic: pad_free curpad

Explanation: The compiler got confused about which scratch pad it was allocating and freeing temporaries and lexicals from. This is an internal error that cannot be resolved by the user.

Example: HPE20672 panic: pad_free curpad

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20673 panic: pad_free po

Explanation: An invalid scratch pad offset was detected internally. This is an internal error that cannot be resolved by the user.

Example: HPE20673 panic: pad_free po

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20674 panic: cv_clone: *string*

Explanation: This is an internal error that cannot be resolved by the user.

In the message text:

string
string

Example: HPE20674 panic: cv_clone: namesv

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20675 panic: save_threadsv called in non-threaded perl

Explanation: This is an internal error that cannot be resolved by the user.

Example: HPE20675 panic: save_threadsv called in non-threaded perl

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20676 panic: corrupt saved stack index

Explanation: The savestack was requested to restore more localized values than there are in the savestack. This is an internal error that cannot be resolved by the user.

Example: HPE20676 panic: corrupt saved stack index

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20677 panic: leave_scope pad code

Explanation: This is an internal error that cannot be resolved by the user.

Example: HPE20677 panic: leave_scope pad code

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20678 panic: leave_scope inconsistency

Explanation: The savestack probably got out of sync. At least, there was an invalid enum on the top of it. This is an internal error that cannot be resolved by the user.

Example: HPE20678 panic: leave_scope inconsistency

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20679 Constant(*string*) unknown: *string*

Explanation: A new constant which is properly not known.

In the message text:

string
string

Example: HPE20679 Constant(undef) unknown: possibly a missing "use charnames ..."

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the syntax of the constant and correct any errors.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20680 Constant subroutine *package*::*function* redefined

Explanation: You redefined a subroutine which had previously been eligible for inlining.

In the message text:

function
function name

package
package name

Example: HPE20680 Constant subroutine Constant subroutine Compress::Zlib redefined

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: See perlsub "Constant Functions" for commentary and workarounds.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20681 Subroutine *package*::*function* redefined

Explanation: You redefined a subroutine.

In the message text:

function
function name

package
package name

Example: HPE20681 Subroutine compress::zip redefined

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: To suppress this warning, say { no warnings 'redefine'; eval "sub name { ... }"; }

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20682 (?p{}) is deprecated - use (??{})

Explanation: This error will occur if you are using the regular expression similar to m/(?p{ 'a' })/.

Example: HPE20682 (?p{}) is deprecated - use (??{})

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: ? is used mostly to indicate the ending match, so either escape the ? character or do not use it at the begining.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20683 Invalid attribute name *attr*

Explanation: The attribute name used has either a special character or $ in the begining.

In the message text:

attr
attribute name

Example: HPE20683 invalid attribute name j$u

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the syntax and correct any errors.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20684 Invalid subroutine reference or name

Explanation: Reference to subroutine is not correct.

Example: HPE20684 Invalid subroutine reference or name

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the syntax of the subroutine reference.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20685 (Did you mean "local" instead of "our"?)

Explanation: "our" does not localize the declared global variable. You have declared it again in the same lexical scope, which seems superfluous.

Example: HPE20685 (Did you mean "local" instead of "our"?)

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use "local" to localize the variable.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20686 "our" variable *var* redeclared

Explanation: You seem to have already declared the same global once before in the current lexical scope.

var
The Variable.

Example: HPE20686 "our" variable foo redeclared

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not declare a variable which has already been declared.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20687 Died

Explanation: You passed an empty string to `die()` (the equivalent of `die ""`), or you called it with no args and both `$@` and `$_` were empty.

Example: HPE20687 Died

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: When you are using the `die()` function, make sure to pass arguments to it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20688 do_study: out of memory

Explanation: This should have been caught by `safemalloc()` instead. This is an internal error that cannot be resolved by the user.

Example: HPE20688 do_study: out of memory

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20689 Exiting *str1* via *str2*

Explanation: You are exiting an eval/format/special block construct (like a sort block or subroutine)/subroutine/substitution by unconventional means, such as with a goto or a loop control statement.

In the message text:

str1
The string may be eval/format/special block construct.

str2
The string may be goto, or a loop control statement.

Example: HPE20689 Exiting eval via goto

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: You cannot use goto or loop to exit from a subroutine or block.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20690 (Skipping label #*num label*)

Explanation: If the label which you encounter is not the label your searching for, this error message will appear.

In the message text:

num
number

lable
lable name

Example: HPE20690 (Skipping label #2 lable)

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: While searching for the right label, this message will be displayed until the label is found.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20691 (Found label #*num label*)

Explanation: While searching for the label, if you hit the correct one, this message will be displayed.

In the message text:

num
number

label
lable name

Example: HPE20691 (Found label #2 LEVEL)

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: When the label is found, this message will be displayed.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20692 (Found loop #%ld)

Explanation: Control ops (cops) are one of the three ops OP_NEXTSTATE, OP_DBSTATE, and OP_SETSTATE that are separate statements. They hold information important for lexical state and error reporting. At run time, PL_curcop is set to point to the most recently executed cop, and thus can be used to determine our current state. This error generated when the current state gets into a loop. This is an internal error that cannot be resolved by the user.

In the message text:

num
A long number

Example: HPE20692 (Found loop #20394)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20693 Octal number > 037777777777 non-portable

Explanation: The octal number you specified is larger than 2**32-1 (4294967295) and therefore non-portable between systems. See perlport for more on portability concerns.

Example: HPE20693 Octal number > 037777777777 non-portable

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Make sure the octal number is within the range.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20696 Format *str* redefined

Explanation: You redefined a format.

In the message text:

str
The string.

Example: HPE20696 Format STDOUT redefined

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: To suppress this warning, say

```
{
 no warnings 'redefine';
 eval "format NAME =...";
    }
```

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20697 *str* found where operator expected

Explanation: The Perl lexer knows whether to expect a term or an operator. If it sees what it knows to be a term when it was expecting to see an operator, it gives you this warning.

In the message text:

str
The string can be a term.

Example: HPE20697 Bareword found where operator expected

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Replace the bareword with an operator or delimiter, such as a semicolon.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20698 sdbm store returned *num*, errno *num*, key "*string*"

Explanation: This warning is emmitted when you try to store a key or a value that is too long. It means that the change was not recorded in the database.

In the message text:

string
string

num
number

Example: HPE20698 sdbm store returned -1, errno 22, key "..."

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: There are a number of limits on the size of the data that you can store in the SDBM file. The most important is that the length of a key, plus the length of its associated value, may not exceed 1008 bytes.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20699 No write permission to sdbm file

Explanation: You can read the file but you cannot write to it.

Example: HPE20699 No write permission to sdbm file

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the permissions of the file and change it to write mode.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20700 Unrecognized escape *code* passed through

Explanation: You used a backslash-character combination which is not recognized by Perl as having any special meaning. *code* will be displayed as is.

In the message text:

code
A character

Example: HPE20700 Unrecognized escape \J. passed through

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: It is not an error to use this escape character sequence, however, since it is not recognized by Perl, verify that you wish to use that character sequence as is instead of replacing it with an escape sequence which is recognized by Perl.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20701 db is not of type SDBM_File

Explanation: db is not connected to the SDBM_File, but you are trying to perform the SDBM_File operation.

Example: HPE20701 db is not of type SDBM_File

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure you have db connected to the SDBM_File before performing such operation.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20702 Cannot reswap uid and euid

Explanation: The setreuid() call failed for some reason in the setuid emulator of suidperl.

Example: HPE20702 Cannot reswap uid and euid

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Examine the call to setreuid and correct any errors.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20703 Cannot apply "*layer*" in non-PerlIO perl

Explanation: While providing a new interface to the USE_PERLIO implementation, if the layer is other than :raw, this error will occur.

In the message text:

layer
layers like :raw and :bytes

Example: HPE20703 Cannot apply ":bytes" in non-PerlIO perl

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Please confirm which layer you are using.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20704 Recursive call to Perl_load_module in PerlIO_find_layer

Explanation: This is an internal error occurred that cannot be resolved by the user.

Example: HPE20704 Recursive call to Perl_load_module in PerlIO_find_layer

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20705 Layer does not match this version of Perl

Explanation: The layer you are using does not match the version of Perl that is currently active.

Example: HPE20705 Layer does not match this version of Perl

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure the version of Perl currently in use supports the layer.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20706 Don't know how to get file name

Explanation: Failed to get the name of the perlIO.

Example: HPE20706 Don't know how to get file name

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the passed parameter to make sure its correct.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20707 'X' outside of string in *function*

Explanation: You had a pack() template that specified a relative position after the end of the string being unpacked or packed. For more information, see perlfunc/pack

In the message text:

f The function name.

Example: HPE20707 'X' outside of string in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Correct the relative position so that it is before the end of the string.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20708 '*code***' allowed only after types** *string* **in** *function*

Explanation: The '!' is allowed in pack() or unpack(), only after certain types.

In the message text:

code
codes

string
string

function
function like pack or unpack

Example: HPE20708 '!' allowed only after types sSiIlLxX in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: See perl functions for more information on pack() or unpack().

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20709 Cannot use both '<' and '>' after type '%c' in %s

Explanation: You cannot use < or > after type 'I' in pack or unpack.

In the message text:

code
codes

function
function like pack or unpack

Example: HPE20709 Can't use both '<' and '>' after type 'I' in pack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure you are not using < or > next to I in pack or unpack.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20710 Cannot use '*code*' in a group with different byte-order in *function*

Explanation: You cannot use the < or > in a group with a different byte-order in the pack or unpack function.

In the message text:

code
codes

function
function like pack or unpack

Example: HPE20710 Cannot use '<' in a group with different byte-order in unpack

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure you are not using < or > in a group in pack or unpack.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20711 Duplicate modifier '*char*' after '*char*' in *function*

Explanation: You have applied the same modifier more than once after a type in a pack template. For more information, see perlfunc/pack

In the message text:

char
A modifier.

funtion
The funtion may be pack or unpack.

Example: HPE20711 Duplicate modifier '>' after 'I' in pack

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check your template. If you are using the same modifier, remove it from the template.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20712 Attempt to tie unique GV

Explanation: Attempting to tie a unique GV (gobal variable) to a variable.

Example: HPE20712 Attempt to tie unique GV

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that you are not trying to tie a unique GV to a normal variable.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20713 Value of node is *num* in Offset macro

Explanation: If the value of node is < 0, this error will occur.

In the message text:

num
number

Example: HPE20713 Value of node is -1 in Offset macro

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the parameter passed to the macro.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20714 Value of node is *num* in Length macro

Explanation: If the value of node is < 0, this error will occur.

In the message text:

num
number

Example: HPE20714 value of node is -1 in Length macro

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the parameter passed to the macro.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20716 ndbm store returned *num*, errno *num*, key "*string*"

Explanation: This warning is emitted when you try to store a key or a value that is too long. It means that the change was not recorded in the database.

In the message text:

string
string

num
number

Example: HPE20716 ndbm store returned -1, errno 22, key "..."

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: There are a number of limits on the size of the data that you can store in the SDBM file. The most important is that the length of a key, plus the length of its associated value, may not exceed 1008 bytes.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20717 No write permission to ndbm file

Explanation: You can read the file but cannot write to it.

Example: HPE20717 No write permission to ndbm file

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the permissions and change it to write mode.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20747 Unrecognized escape *char* in character class passed through

Explanation: You used a backslash-character combination which is not recognized by Perl inside character classes. The character was understood literally.

In the message text:

char
A character class.

Example: HPE20747 Unrecognized escape \d in character class passed through

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use a backslash-character combination in a character class.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20751 strxfrm() gets absurd

Explanation: This error occurs while setting up a new collation locale.

Example: HPE20751 strxfrm() gets absurd

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure your using the right parameters.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20752 PERL_SIGNALS illegal: "*SignalVal*"

Explanation: When attempting to get the environmental variable, a legal value must be provided, or this error will occur.

In the message text:

SignalVal
Signal value.

Example: HPE20752 PERL_SIGNALS illegal: "Language"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: For legal values, refer to perlrun/PERL_SIGNALS and then select one for use.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20753 Cannot use '*code*' after -mname

Explanation: While using the -m option, if it is followed by any other character, this error occurs.

In the message text:

code
codes.

Example: HPE20753 Cannot use 'n' after -mname

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure -m is not followed by any other characters.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20755 Wrong syntax (suid) fd script name "*filename*"

Explanation: In the fd script, the file will always be present in the /dev/fd/ . Therefore, if you indicate that the file is in the current directory, this error will occur.

In the message text:

filename
file name

Example: HPE20755 Wrong syntax (suid) fd script name \"%s\"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you are using fd script, make sure your file is in the /dev/fd/ directory.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20756 Missing (suid) fd script name

Explanation: No file name was provided while attempting to run the script.

Example: HPE20756 Missing (suid) fd script name

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure you provide the current file name.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20757 Cannot run with cpp -P with CPPSTDIN undefined

Explanation: Since CPPSTDIN is undefined, you cannot use -P .

Example: HPE20757 Cannot run with cpp -P with CPPSTDIN undefined

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure you do not use -P if CPPSTDIN is undefined.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20758 Cannot fix broken locale name "*LocaleCat*"

Explanation: Occasionaly, setlocale function presents this error.

In the message text:

LocaleCat
category and locale argument passed for setlocale

Example: HPE20758 Cannot fix broken locale name "es_ES@euro LANG=es_ES.ISO-8859-15 ".

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check if you have given the category and locale argument properly.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20759 Need (suid) fdscript in suidperl

Explanation: If you are using suidperl, then you should provide fdscript which should be present in /dev/fd/.

Example: HPE20759 Need (suid) fdscript in suidperl

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you are using suidperl to run the script, make sure you are using fdscript.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20760 Cannot access() script

Explanation: Access to the script was denied. This is because the path could not be found, or any of the desired access modes were not granted.

Example: HPE20760 Cannot access() script

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the permissions for the specified path.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20761 Real UID cannot exec script

Explanation: An attempt was made to execute a script for which permissions were not given.

Example: HPE20761 Real UID cannot exec script

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check if your real uid has the permission to run the script. Otherwise, log in as superuser and run the script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20762 Very long #! line

Explanation: The first line #! of the script may be too long.

Example: HPE20762 Very long #! line

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure the length of the script's first line (which is provided to specify the path of the perl interpretor) is within the normal range.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20763 Cannot change argv to have fd script

Explanation: Arguments cannot be changed to fd script.

Example: HPE20763 Can't change argv to have fd script

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: In suidperl, do not attempt to change the original argument list.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20764 Effective UID cannot exec script

Explanation: This error occured because an attempt was made to execute a script for which permissions were not given.

Example: HPE20764 Effective UID cannot exec script

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check the permissions of the file for the effective UID.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20765 (suid) fdscript needed in suidperl

Explanation: You should have superuser permission and use suidperl to run the fdscript, otherwise this error will occur.

Example: HPE20765 (suid) fdscript needed in suidperl

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Use suidperl to run fdscript.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20766 Syntax error

Explanation: If the syntax is not correct in the script, an error will be shown.

Example: HPE20766 Syntax error

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: First compile the code using -c option so that you will get to know the systax correctness. Then execute the script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20767 No *opt* allowed while running setgid

Explanation: Certain operations are deemed to be too insecure for a setuid or setgid script to even be allowed to attempt. There will be other ways to do what you want that are, if not secure, at least securable. See perlsec for more information.

In the message text:

opt
option like -P

Example: HPE20767 No -P allowed while running setuid

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: When you are using setgid, you cannot use certain options like -P.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20768 No *opt* allowed with (suid) fdscript

Explanation: Certain operations are deemed to be too insecure for a setuid or setgid script to even be allowed to attempt. There are other ways to do what you want that are, if not secure, at least securable.

In the message text:

opt
option Like -P

Example: HPE20768 No -e allowed with (suid) fdscript

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You can not use -e option with (suid) fdscript.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20769 No *opt* allowed in suidperl

Explanation: -e option used in suidperl.

In the message text:

opt
The option.

Example: HPE20769 No -e allowed in suidperl

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure -e is not used in suidperl .

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20770 Unrecognized switch: -*opt* (-h will show valid options)

Explanation: You specified an illegal option to Perl.

In the message text:

opt
The option.

Example: HPE20770 Unrecognized switch: -p (-h will show valid options)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure all provided options are acceptable (use -h to show valid options). If all options provided are valid, check the #! line to see if it's supplying the bad switch on your behalf.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20771 suidperl needs (suid) fd script

Explanation: suidperl requires fdscript to include the scripting language.

Example: HPE20771 suidperl needs (suid) fd script

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you are using suidperl, make sure it has an fdscript frame.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20772 Unbalanced scopes: *num* **more ENTERs than LEAVEs**

Explanation: The exit code detected an internal inconsistency in how many blocks were entered and left. This is an internal error that cannot be resolved by the user.

In the message text:

num
Number.

Example: HPE20773 Unbalanced saves: 1000 more ENTERs than LEAVEs

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20773 Unbalanced saves: *num* **more saves than restores**

Explanation: The exit code detected an internal inconsistency in how many values were temporarily localized. This is an internal error that cannot be resolved by the user.

In the message text:

num
Number.

Example: HPE20773 Unbalanced saves: 1000 more saves than restores

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20774 Unbalanced tmps: *num* **more allocs than frees**

Explanation: The exit code detected an internal inconsistency in how many mortal scalars were allocated and freed. This is an internal error that cannot be resolved by the user.

In the message text:

num
Number.

Example: HPE20774 Unbalanced tmps: 1000 more allocs than frees

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20775 Unbalanced context: *num* **more PUSHes than POPs**

Explanation: The exit code detected an internal inconsistency in how many execution contexts were entered and left. This is an internal error that cannot be resolved by the user.

In the message text:

num
Number.

Example: HPE20775 Unbalanced context: 1000 more PUSHes than POPs

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20776 Unbalanced string table refcount: (*num*) for "*function*"

Explanation: The unbalanced string table refcount is because the refcount on the struct refcounted_he isn't being dropped properly. This is an internal error occurred that cannot be resolved by the user.

In the message text:

function
function name

num
number

Example: HPE20776 Unbalanced string table refcount: (1) for "signal_connect"

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20777 Scalars leaked: *var*

Explanation: Something went wrong in Perl's internal bookkeeping of scalars: not all scalar variables were deallocated by the time Perl exited. This is an internal error that cannot be resolved by the user.

In the message text:

var
The variable may be long double.

Example: HPE20777 Scalars leaked: 0

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: This usually indicates is a memory leak, which is detrimental, especially if the Perl program is intended to be long-running. Check to make sure all scalars are deallocated before the program exits.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20778 Invalid option -D*opt*, use -D'' to see choices

Explanation: You used an invalid switch along with -D.

In the message text:

opt
An invalid option

Example: HPE20778 invalid option -Dz, use -D'' to see choices

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check for valid options using perl -D.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20779 -Dp not implemented on this platform

Explanation: This switch is not available on the EBCDIC platform.

Example: HPE20779 -Dp not implemented on this platform

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use this switch.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20790 Cleaning loops: SV at 0x"*address*"

Explanation: Called by sv_clean_all() for each live SV. However, some barnacles may remain, clinging to typeglobs.

In the message text:

address
address

Example: HPE20790 Cleaning loops: SV at 0x%"UVxf"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: This is not a user level error.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20791 oops: oopsAV

Explanation: An internal warning that the grammar is confused. This is an internal error that cannot be resolved by the user.

Example: HPE20791 oops: oopsAV

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20792 oops: oopsHV

Explanation: An internal warning that the grammar is confused. This is an internal error that cannot be resolved by the user.

Example: HPE20792 oops: oopsHV

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20793 Page overflow

Explanation: A single call to write() produced more lines than can fit on a page. See perlform for more information.

Example: HPE20793 Page overflow

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Reduce the number of lines to be written.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20794 perl: warning: Setting locale failed.

Explanation: This means that your locale settings had LC_ALL set to ``En_US'', and LANG exists but has no value. Perl tried to believe you but could not. Instead, Perl gave up and resorted to the ``C'' locale, the default locale that should always work. See perllocale section **LOCALE PROBLEMS**.

Example: HPE20794 perl: warning: Setting locale failed.

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check your locale setting, or the locale installation on your system, for any problems (for example, broken or missing system files).

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20795 perl: warning: Setting locale failed for the categories:

Explanation: This locale setting warning is displayed if you have not given a proper value for a particular locale.

Example: HPE20795 perl: warning: Setting locale failed for the categories:

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: This error will occur if you are trying to set the following: LC_CTYPE LC_COLLATE

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20796 perl: warning: Please check that your locale settings:

Explanation: This locale setting warning is displayed if you have not given a proper value.

Example: HPE20796 perl: warning: Please check that your locale settings:

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The following are some appropriate locale settings that you can use:

LC_CTYPE LC_ALL LANGUAGE

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20797 perl: warning: Falling back to the standard locale ("C").

Explanation: The whole warning message will look something like: perl: warning: Setting locale failed. perl: warning: Please check that your locale settings: LC_ALL = "En_US", LANG = (unset) are supported and installed on your system. perl: warning: Falling back to the standard locale ("C").

Example: HPE20797 perl: warning: Falling back to the standard locale ("C").

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: If your system cannot take the locale setting that you have provided, try providing a different locale setting, or accept the default.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20798 perl: warning: Failed to fall back to the standard locale ("C").

Explanation: Sometimes the system cannot set the default locale.

Example: HPE20798 perl: warning: Failed to fall back to the standard locale ("C").

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check your setting.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20799 perl: warning: Cannot fall back to the standard locale ("C").

Explanation: You tried to set locale but failed. As a result, the system tried to set the default value, but failed as well.

Example: HPE20799 perl: warning: Cannot fall back to the standard locale ("C").

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check your setting.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20830 Hexadecimal number > 0xffffffff non-portable

Explanation: The hexadecimal number you specified is larger than 2**32-1 (4294967295) and therefore non-portable between systems. See perlport for more on portability concerns.

Example: HPE20830 Hexadecimal number > 0xffffffff non-portable

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Use a hexadecimal number less than or equal to 2**32-1 (4294967295).

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20831 Illegal switch in PERL5OPT: -*opt*

Explanation: The PERL5OPT environment variable may only be used to set the following switches: **-[DIMUdmtw]**.

In the message text:

opt
The switch option.

Example: HPE20831 Illegal switch in PERL5OPT: -D

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You should only use the following switches **-[DIMUdmtw]** .

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: X- A very fatal error (non-trappable).

Automation: Not applicable.

HPE20834 Integer overflow in binary number

Explanation: The binary number you have specified, either as a literal or as an argument, is too big for your architecture, and has been converted to a floating point number.

Example: HPE20834 Integer overflow in binary number

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: On a 32-bit architecture, the largest binary number representable without overflow is 0b11111111111111111111111111111111. Note that Perl transparently promotes all numbers to a floating point representation internally--subject to loss of precision errors in subsequent operations.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20835 Integer overflow in hexadecimal number

Explanation: The hexadecimal number you have specified either as a literal or as an argument to hex() is too big for your architecture, and has been converted to a floating point number.

Example: HPE20835 Integer overflow in hexadecimal number

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: On a 32-bit architecture, the largest hexadecimal number representable without overflow is 0xFFFFFFFF. Perl transparently promotes all numbers to a floating point representation internally--subject to loss of precision errors in subsequent operations.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20836 Integer overflow in octal number

Explanation: The octal number you have specified either as a literal or as an argument to oct() is too big for your architecture, and has been converted to a floating point number.

Example: HPE20836 Integer overflow in octal number

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: On a 32-bit architecture, the largest octal number representable without overflow is 037777777777. Perl transparently promotes all numbers to a floating point representation internally--subject to loss of precision errors in subsequent operations.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20837 Illegal hexadecimal digit '*num*' ignored

Explanation: You may have tried to use a character other than 0 - 9 or A - F, a - f in a hexadecimal number.

In the message text:

num
The number.

Example: HPE20837 Illegal hexadecimal digit 'm' ignored

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Interpretation of the hexadecimal number stopped before the illegal character. Use a valid digit.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20839 Invalid conversion in *str*printf:

Explanation: The conversion you specified for printf or sprintf is invalid. For more information, see perlfunc/sprintf

In the message text:

str
The string will be "s" or "" which will make it either sprintf or printf .

Example: HPE20839 Invalid conversion in printf:

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Refer to perlfunc/sprintf for valid conversions and use any that you require.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20840 Invalid separator character *char1char2char1* in PerlIO layer specification *layer*

Explanation: When pushing layers onto the Perl I/O system, something other than a colon or whitespace was seen between the elements of a layer list. If the previous attribute had a parenthesised parameter list, perhaps that list was terminated too soon.

In the message text:

char1
The character can be " or \.

char2
The invalid separator character can be \ .

layer
The layername.

Example: HPE20840 Invalid separator character $/$ in PerlIO layer specification PerlIO

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: You should use only a colon or whitespace for seperating PerlIO layers.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20841 *function* not implemented on this architecture

Explanation: This function is not available on your machine.

In the message text:

function
The function name.

Example: HPE20841 foo not implemented on this architecture

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use this function.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20842 SHMLBA is not defined on this architecture

Explanation: SHMLBA is not defined on your machine.

Example: HPE20842 SHMLBA is not defined on this architecture

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use SHMLBA.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20843 nl_langinfo() not implemented on this architecture

Explanation: The nl_langinfo() function is not implemented on your machine.

Example: HPE20843 nl_langinfo() not implemented on this architecture

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: This function cannot be used on your architecture and should not be called from your script.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20844 POSIX::*function* not implemented on this architecture

Explanation: The POSIX function indicated is not implemented on this architecture.

In the message text:

function
The function name.

Example: HPE20844 POSIX::setsid not implemented

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not use this function on your machine.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20845 lstat() on filehandle *filehandle*

Explanation: You tried to do an lstat on a filehandle.

In the message text:

filehandle
The filehandle.

Example: HPE20845 lstat() on filehandle STD

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: lstat() is used only on filenames, so

do not use lstat on filehandles.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20846 (Might be a runaway multi-line *charchar*string starting on line *n*)

Explanation: An advisory indicating that the previous error may have been caused by a missing delimiter on a string or pattern, because it eventually ended earlier on the current line.

In the message text:

char
Any character.

n Line no.

Example: HPE20846 (Might be a runaway multi-line "" string starting on line 1)

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check for a missing end quote.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20847 "*sym*" never introduced

Explanation: The symbol in question was declared but somehow went out of scope before it could possibly have been used.

In the message text:

sym
The Symbol.

Example: HPE20847 "s" never introduced

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Either declare the symbol if you want to use it now, or declare it globally.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: S- A severe warning (default).

Automation: Not applicable.

HPE20848 No dbm on this machine

Explanation: This is counted as an internal error, as every machine should supply dbm nowadays, and because Perl comes with SDBM. This is internal error that cannot be resolved by the user. For more information, see SDBM_File.

Example: HPE20848 No dbm on this machine

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20850 NULL OP IN RUN

Explanation: An internal routine called run() with a null opcode pointer. This is an internal error that cannot be resolved by the user.

Example: HPE20850 NULL OP IN RUN

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20851 Cannot declare *descriptor* in *var*

Explanation: Only scalar, array, and hash variables may be declared as "my" or "our" variables. They must have ordinary identifiers as names.

In the message text:

descriptor
The opcode descriptor.

var
The var will be "my" or "our".

Example: HPE20851 Cannot declare scalar dereference in my

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Do not declare opcode descriptors such as "null operation", "stub", "scalar", "pushmark", "wantarray", as "my" or "our".

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20852 lock can only be used on shared values

Explanation: Attempting to lock unshared values.

Example: HPE20852 lock can only be used on shared values

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you are trying something like the following,

```
$hashref->{key};
lock $hashref->{key};
```

then this will create an error. To lock an element, you first need to share it before locking it.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20853 cond_wait can only be used on shared values

Explanation: cond_wait only accepts locked variables as arguments, so the variables need to be shared.

Example: HPE20853 cond_wait can only be used on shared values

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check if you are trying to pass an unshared variable as the parameter for cond_wait.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20854 cond_wait lock must be a shared value

Explanation: The argument passed to cond_wait must be a shared variable or shared value.

Example: HPE20854 cond_wait lock must be a shared value

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: When you are using cond_wait lock, make sure you are doing it on a shared variable.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20855 You need a lock before you can cond_wait

Explanation: cond_wait takes only locked variables as arguments, so the variables need to be shared.

Example: HPE20855 You need a lock before you can cond_wait

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You need a lock before you can cond_wait a variable.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20856 cond_timedwait can only be used on shared values

Explanation: cond_timedwait takes only locked variables as arguments, so the variable must be shared.

Example: HPE20856 cond_timedwait can only be used on shared values

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that cond_timedwait calls only shared values.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20857 cond_timedwait lock must be a shared value

Explanation: cond_timedwait lock was applied to an unshared value.

Example: HPE20857 cond_timedwait lock must be a shared value

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: cond_timedwait must only be applied to a shared value.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20858 cond_broadcast can only be used on shared values

Explanation: The cond_broadcast function works similarly to cond_wait. Trying to call cond_broadcast on non-shared values.

Example: HPE20858 cond_broadcast can only be used on shared values

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure the variable, for which you are calling cond_broadcast on, is shared.

System programmer response: No System

Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20859 Internal error, could not set TLS

Explanation: The error occured while setting the thread. This is an internal error that cannot be resolved by the user.

Example: HPE20859 Internal error, could not set TLS

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check to make sure the parameters are correct while threading.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20860 Internal error, could not get TLS

Explanation: TLS (Transport Layer Security) error occured due to some internal problem. This is an internal error that cannot be resolved by the user.

Example: HPE20860 Internal error, couldn't get TLS

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Contact the system administrator.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20861 DProf: don't know what subroutine to profile

Explanation: -d:DProf failed to figure out which subroutine is taking more time.

Example: HPE20861 DProf: don't know what subroutine to profile

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Check to see if there is a bug in Devel::DProf mail perl5-porters@perl.org, and correct any that are found.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20863 DProf: run perl with -d to use DProf.

Explanation: If -d is missed out while using DProf, this error will occur.

Example: HPE20863 DProf: run perl with -d to use DProf.

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Syntax for using DProf -> perl -d:DProf scriptname.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20864 Wide character in Socket::inet_ntoa

Explanation: Perl encountered a wide character (greater than 255 bytes) when it was not expecting one.

Example: HPE20864 Wide character in Socket::inet_ntoa

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Add the `:utf8` layer to the output, `binmode STDOUT, ':utf8'`

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20866 Wide character in Socket::pack_sockaddr_in

Explanation: Perl encountered a wide character (greater than 255) when it was not expecting one.

Example: HPE20866 Wide character in Socket::pack_sockaddr_in

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Add the `:utf8` layer to the output, `binmode STDOUT, ':utf8' .`

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20867 Bad getcc subscript

Explanation: The argument passed to getcc was more than 10.

Example: HPE20867 Bad getcc subscript

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check that the argument passed to getcc is less than or equal to 10.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20868 Bad setcc subscript

Explanation: Subscipt for setcc is not correct.

Example: HPE20868 Bad setcc subscript

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Set a value in the c_cc field of a termios object. The c_cc field is an array so an index must be specified. $termios->setcc(&POSIX::VEOF, 1);

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20869 Action is not of type POSIX::SigAction

Explanation: While performing a POSIX sigaction, if the sv_isa function returns false, this error will occur.

Example: HPE20869 Action is not of type POSIX::SigAction

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure that the parameter passed to POSIX::sigaction is proper.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20870 Cannot supply an old action without a HANDLER

Explanation: Perl failed to remember the old action. This is an internal error that cannot be resolved by the user.

Example: HPE20870 Cannot supply an old action without a HANDLER

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20871 Cannot supply an action without a HANDLER

Explanation: Vector new Perl handler through %SIG. (The core signal handlers read %SIG to dispatch.). Action without the HANDLER causes this error.

Example: HPE20871 Cannot supply an action without a HANDLER

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Make sure you are passing the correct handler.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20872 sigset is not of type POSIX::SigSet

Explanation: While performing POSIX SigSet addset, if it returns false, this error will occur.

Example: HPE20872 sigset is not of type POSIX::SigSet

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check if the parameter passed is correct.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20873 oldsigset is not of type POSIX::SigSet

Explanation: While performing POSIX OldSigSet addset, if it returns false, this error will occur.

Example: HPE20873 oldsigset is not of type POSIX::SigSet

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Check if the parameter passed is correct.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20874 Close with partial character

Explanation: This error occurs when you try to close a file which is opened in write mode, and it contains a partial character in the buffer to be written.

Example: HPE20874 Close with partial character

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Flush the buffers before closing the file.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20875 Cannot tell at partial character

Explanation: Perl cannot tell the current file position when the UTF-8 character is partial character.

Example: HPE20875 Cannot tell at partial character

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: The way to get a postion is to re-translate the UTF-8 character in the buffer, and then ask layer below.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20876 Opcode tag "*OpTag*" already defined

Explanation: Store a new tag definition. Always a mask. The tag must not already be defined. SV *mask is copied not referenced.

In the message text:

OpTag
Operator tag.

Example: HPE20876 Opcode tag "define_tag" already defined

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Each operator tag should be unique.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20877 Unknown operator tag "*OpName*"

Explanation: Operator name cannot be a tag. If operator name has : in it, then this error will be displayed.

In the message text:

OpName
Example operator names include gv and mg .

Example: HPE20877 Unknown operator tag ":gv"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Operator names should not contain numbers.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20878 Unknown operator name "*OpName*"

Explanation: If the operator name has any numeric value in it, this error will occur.

In the message text:

OpName
Example operator names include gv and mg.

Example: HPE20878 Unknown operator name "3gv"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Operator names should not contain numbers.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20879 Unknown operator prefix "*code*"

Explanation: Only a specific operator can be used as prefix operator.

In the message text:

code
codes.

Example: HPE20879 Unknown operator prefix "^"

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: For example, prefix dereferencing operators are typed: $, @, %, and &. So make sure in the error you are using the proper prefix operator.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20880 Invalid opset: *str*

Explanation: The operator set `opset` supplied to verify_opset, which verifies the set is invalid. This is an internal error that cannot be resolved by the user.

In the message text:

str
The str will be "undefined", "wrong type" or "wrong size".

Example: HPE20880 Invalid opset: undefined

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20881 Bad symbol for form (GV is unique)

Explanation: An internal request asked to add a form entry to something that was not a symbol table entry. This is an internal error that cannot be resolved by the user.

Example: HPE20881 Bad symbol for form (GV is unique)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20882 Pseudo-hashes are deprecated

Explanation: Pseudo-hashes were deprecated in Perl 5.8.0 and they will be removed in Perl 5.10.0. You can continue to use the `fields` pragma. This error message is displayed if the script is run using -w option or declaring `use warnings` at the start of script.

Example: HPE20882 Pseudo-hashes are deprecated

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Do not use pseudo-hashes.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: D- A deprecation (optional).

Automation: Not applicable.

HPE20883 panic: null array

Explanation: One of the internal array routines was passed a null AV pointer. This is an internal error that cannot be resolved by the user.

Example: HPE20883 panic: null array

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: AV pointer cannot be null. Follow your local procedures for reporting a problem to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20884 Attempt to clear deleted array

Explanation: Operation attempted on an array that has been deleted.

Example: HPE20884 Attempt to clear deleted array

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: You might have previously deleted the array that you are attempting to clear again.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20885 av_reify called on tied array

Explanation: When trying to tie the array to some class, av_verify is being called. This is an internal error that cannot be resolved by the user.

Example: HPE20885 av_reify called on tied array

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: av_reify should not be called while tieing the array to any class. Follow your local procedures for reporting a problem to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20886 Cannot use anonymous symbol table for method lookup

Explanation: The internal routine that does method lookup was handed a symbol table that does not have a name.

Example: HPE20886 Cannot use anonymous symbol table for method lookup

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: If you have undefined stashes, avoid them.

System programmer response: No System Programmer response is required.

Problem determination: Symbol tables can become anonymous, for example, by undefining stashes: `undef %Some::Package::`.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20887 Looking for method *method* in package *pkg*

Explanation: It is looking for method(bar) in package(foo), which is not available.

In the message text:

method
The method name.

pkg
The package name.

Example: HPE20887 Looking for method bar in package foo

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: The method is not defined in this package. Select the right one.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20889 Too late to run CHECK block

Explanation: A CHECK block is being defined during run time proper, when the opportunity to run them has already passed.

Example: HPE20889 Too late to run CHECK block

System action: Displays the warning and continues interpreting the script.

Operator response: No System Operator response is required.

User response: Instead of loading a file with `require` or `do`, try using `use`instead. If this does not fix the problem, try putting the `require` or `do` inside a BEGIN block.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: W- A warning (optional).

Automation: Not applicable.

HPE20890 tried to access per-thread data in non-threaded perl

Explanation: If the version of Perl on your machine is a non-threaded version of Perl, this error occurs. This is an internal error that cannot be resolved by the user.

Example: HGU20890 tried to access per-thread data in non-threaded perl

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20891 Split loop

Explanation: The split was looping infinitely. A split should not iterate more times than there are characters of input, which is what happened. This is an internal error that cannot be resolved by the user. For more information, see perlfunc/split

Example: HGU20891 Split loop

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20892 Substitution loop

Explanation: The substitution was looping infinitely. A substitution should not iterate more times than there are characters of input, which is what happened. This is an internal error that cannot be resolved by the user. For more information, see perlop/"Quote and Quote-like Operators"

Example: HGU20892 Substitution loop

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: Follow your local procedures for reporting problems to IBM.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: P- An internal error you should never see (trappable).

Automation: Not applicable.

HPE20893 Runaway format

Explanation: Your format contained the ~~ repeat-until-blank sequence, but it produced 200 lines at once, and the 200th line looked exactly like the 199th line. Apparently you didn't arrange for the arguments to exhaust themselves. For more information, see perlform

Example: HGU20893 Runaway format

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You should arrange for the arguments to exhaust themselves either by using ^ instead of @ (for scalar variables), or by shifting or popping (for array variables).

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20894 Repeated format line will never terminate (~~ and @#)

Explanation: Your format contains the ~~ repeat-until-blank sequence and a numeric field that will never go blank, so that the repetition never terminates. For more information, see perlform.

Example: HGU20894 Repeated format line will never terminate (~~ and @#)

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: You might want to use ^# instead.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20905 Usage:

Explanation: A function was called for which this error is displayed, because of incorrect arguments. See perlfunctions for more information.

Example: if the error is something like `Usage: POSIX::sysconf(name)`, it means POSIX::sysconf takes one argument.

Example: HPE20905 Usage:

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: See the error and provide the arguments as shown.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

HPE20906 Usage:

Explanation: A function was called for which this error is displayed, because of incorrect arguments. See perlfunctions for more information.

For example, if the error is something like `Usage: POSIX::sysconf(name)`, it means POSIX::sysconf takes one argument.

Example: HPE20906 Usage:

System action: Perl immediately stops interpreting the file (before any statements are run) and exits.

Operator response: No System Operator response is required.

User response: See the error and provide the arguments as shown.

System programmer response: No System Programmer response is required.

Problem determination: No additional information.

Source: Perl

Module: Not applicable.

Routing code: Not applicable.

Descriptor code: F- A fatal error (trappable).

Automation: Not applicable.

Chapter 7. Related Messages

The following message is not generated by Perl for z/OS, however, because it may be seen in some Perl usage scenarios, it has been documented here. The explanation provided is in the context of Perl only and those encountering this message in other environments should reference the primary message document pertaining to this message.

CEE3501S The module libperl.so was not found.

Explanation: This message indicates that the directory containing the DLL `libperl.so` was not found in the LIBPATH environment variable. Those encountering this message in other environments than Perl should refer to the complete message description in *z/OS Language Environment Run-Time Messages*.

Example: CEE3501S The module libperl.so was not found. The traceback information could not be determined.

System action: Perl is not terminated abonormally and in some cases a CEEDUMP file is generated in the user's current working directory.

User response: At the user level, this problem can be remedied by adding the following directory to the LIBPATH variable using either the export command or immediately prior to executing perl or a perl script. Following are examples of each approach:

export libpath

```
export LIBPATH="/usr/lpp/perl/lib/5.8.7/os390-thread-multi/CORE:$LIBPATH"
perl myperlscript.pl
```

define prior to executing perl

```
LIBPATH="/usr/lpp/perl/lib/5.8.7/os390-thread-multi/CORE:$LIBPATH"  perl myperlscript.pl
```

System programmer response: The suggested solution to preventing this problem is for the system administrator to create a symbolic link to `libperl.so` in a directory which already exists in a directory in the default LIBPATH. Refer to “Add references to libperl.so” on page 6 for more information.

Appendix A. Code pages

There are two code pages that are commonly used when porting applications to z/OS UNIX: IBM-1047 and ISO-8859. For a graphic of the IBM-1047 page, see `http://www.ibm.com/servers/eserver/zseries/zos/unix/perl/graphics/hpezafg1.gif`. For a graphic of the IS0-8859 page, see `http://www.ibm.com/servers/eserver/zseries/zos/unix/perl/graphics/hpezafg2.gif`.

Appendix B. Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. The major accessibility features in z/OS enable users to:

- Use assistive technologies such as screen readers and screen magnifier software
- Operate specific or equivalent features using only the keyboard
- Customize display attributes such as color, contrast, and font size

Using assistive technologies

Assistive technology products, such as screen readers, function with the user interfaces found in z/OS. Consult the assistive technology documentation for specific information when using such products to access z/OS interfaces.

Keyboard navigation of the user interface

Users can access z/OS user interfaces using TSO/E or ISPF. Refer to *z/OS TSO/E Primer*, *z/OS TSO/E User's Guide*, and *z/OS ISPF User's Guide Vol I* for information about accessing TSO/E and ISPF interfaces. These guides describe how to use TSO/E and ISPF, including the use of keyboard shortcuts or function keys (PF keys). Each guide includes the default settings for the PF keys and explains how to modify their functions.

z/OS information

z/OS information is accessible using screen readers with the BookServer/Library Server versions of z/OS books in the Internet library at:

`http://www.ibm.com/systems/z/os/zos/bkserv/`

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
USA

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the products and/or the programs described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Mail Station P300
2455 South Road
Poughkeepsie, NY 12601-5400
USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

The following terms are trademarks of the IBM Corporation in the United States or other countries or both:

BookManager	MVS
C/MVS	OS/390
C/370	RACF
CICS	Resource Link
IBM	SP
IBMLink	VTAM
Language Environment	z/OS
Library Reader	zSeries
Library Server	z/VM

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Index

Readers' Comments — We'd Like to Hear from You

z/OS
IBM Ported Tools for z/OS: Perl for z/OS Feature User's Guide and Reference

Publication No. SA23-1347-00

We appreciate your comments about this publication. Please comment on specific errors or omissions, accuracy, organization, subject matter, or completeness of this book. The comments you send should pertain to only the information in this manual or product and the way in which the information is presented.

For technical questions and information about products and prices, please contact your IBM branch office, your IBM business partner, or your authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate without incurring any obligation to you. IBM or any other organizations will only use the personal information that you supply to contact you about the issues that you state on this form.

Comments:

Thank you for your support.

Submit your comments using one of these channels:

- Send your comments to the address on the reverse side of this form.
- Send your comments via email to: mhvrcfs@us.ibm.com

If you would like a response from IBM, please fill in the following information:

Name	Address
Company or Organization	
Phone No.	Email address

Readers' Comments — We'd Like to Hear from You
SA23-1347-00

IBM

Cut or Fold
Along Line

Fold and Tape | **Please do not staple** | Fold and Tape

NO POSTAGE
NECESSARY
IF MAILED IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

IBM Corporation
Department 55JA, Mail Station P384
2455 South Road
Poughkeepsie, NY
12601-5400

Fold and Tape | **Please do not staple** | Fold and Tape

Cut or Fold
Along Line

Product Number: 5655-M23

Printed in USA

SA23-1347-00