

NAME

README.vos - Perl for Stratus VOS

SYNOPSIS

This file contains notes for building perl on the Stratus VOS operating system. Perl is a scripting or macro language that is popular on many systems. See *perlbook* for a number of good books on Perl.

These are instructions for building Perl from source. Most people can simply download a pre-compiled distribution from the VOS anonymous FTP site. If you are running VOS Release 14.2.0 or earlier, download Perl from <ftp://ftp.stratus.com/pub/vos/posix/alpha/alpha.html> If you are running VOS Release 14.3.0 or later, download Perl from <ftp://ftp.stratus.com/pub/vos/posix/ga/ga.html> Instructions for unbundling the Perl distribution file are at <ftp://ftp.stratus.com/pub/vos/utility/utility.html>

If you are running VOS Release 14.4.1 or later, you can obtain a pre-compiled, supported copy of perl by purchasing Release 2.0.1 (or later) of the VOS GNU C++ and GNU Tools product from Stratus Technologies.

Multiple methods to build perl for VOS

If you elect to build perl from its source code, you have several different ways that you can build perl. The method that you use depends on the version of VOS that you are using and on the architecture of your Stratus hardware platform.

- 1 If you have a Stratus XA2000 (Motorola 68k-based) platform, you must build perl using the alpha version of VOS POSIX support and using the VOS Standard C Cross-compiler. You must build perl on VOS Release 14.1.0 (or later) on an XA/R or Continuum platform.
This version of perl is properly called "miniperl" because it does not contain the complete perl functionality.
You must build perl with the `compile_perl.cm` command macro found in the `vos` subdirectory.
- 2 If you have a Stratus XA/R (Intel i860-based) platform, you must build perl using the alpha version of VOS POSIX support and using the VOS Standard C compiler or cross-compiler. You must build perl on VOS Release 14.1.0 (or later) on an XA/R or Continuum platform.
This version of perl is properly called "miniperl" because it does not contain the complete perl functionality.
You must build perl with the `compile_perl.cm` command macro found in the `vos` subdirectory.
- 3 If you have a Stratus Continuum (PA-RISC-based) platform that is running a version of VOS earlier than VOS 14.3.0, you must build perl using the alpha version of VOS POSIX support and using the VOS Standard C compiler or cross-compiler. You must build perl on VOS Release 14.1.0 (or later) on an XA/R or Continuum platform.
This version of perl is properly called "miniperl" because it does not contain the complete perl functionality.
You must build perl with the `compile_perl.cm` command macro found in the `vos` subdirectory.
- 4 If you have a Stratus Continuum (PA-RISC-based) platform that is running VOS Release 14.3.0 through VOS Release 14.4.1, you must build perl using the generally-available version of VOS POSIX support, and using either the VOS Standard C compiler or the VOS GNU C compiler. You must build perl on VOS Release 14.3.0 (or later) on a Continuum platform.
This version of perl is properly called "miniperl" because it does not contain the complete perl functionality.
You must build perl with the `compile_perl.cm` command macro found in the `vos` subdirectory.
- 5 If you have a Stratus Continuum (PA-RISC-based) platform that is running VOS Release 14.5.0 or later, you can either use the previous method to build "miniperl" or you can build

"full perl", which contains the complete functionality of perl. I strongly recommend that you build full perl. To build full perl, you must use the generally-available version of VOS POSIX support. You must use the VOS GNU C compiler and the VOS GNU C/C++ and GNU Tools Release 2.0.1 (or later) product. You must build full perl on VOS Release 14.5.0 (or later) on a Continuum platform.

You must build full perl with the `compile_full_perl.cm` command macro found in the `vos` subdirectory.

Stratus POSIX Support

Note that there are two different implementations of POSIX.1 support on VOS. There is an alpha version of POSIX that is available from the Stratus anonymous ftp site (<ftp://ftp.stratus.com/pub/vos/posix/alpha/alpha.html>). There is a generally-available version of POSIX that comes with VOS Release 14.3.0 or higher. This port of POSIX will compile and bind with either version of POSIX.

Most of the Perl features should work on VOS regardless of which version of POSIX that you are using. However, the alpha version of POSIX is missing a number of key functions, and therefore any attempt by `perl.pm` to call the following unimplemented POSIX functions will result in an error message and an immediate and fatal call to the VOS debugger. They are "dup", "fork", and "waitpid". The lack of these functions prevents you from starting VOS commands and grabbing their output in perl. The workaround is to run the commands outside of perl, then have perl process the output file. These functions are all available in the generally-available version of POSIX.

INSTALLING PERL IN VOS

Compiling Perl 5 on VOS

Before you can build Perl 5 on VOS, you need to have or acquire the following additional items.

- 1 The VOS Standard C Compiler (or the VOS Standard C Cross-Compiler) and the VOS C Runtime. If you are using the generally-available version of POSIX support, you may instead use the VOS GNU C/C++ Compiler. These are standard Stratus products.
- 2 Either the VOS OS TCP/IP or STCP product set. If you are building with the alpha version of POSIX you need the OS TCP/IP product set. If you are building with the generally-available version of POSIX you need the STCP product set. These are standard Stratus products.
- 3 Either the alpha or generally-available version of the VOS POSIX.1 environment.
The alpha version of POSIX.1 support is available on the Stratus FTP site. Login anonymously to [ftp.stratus.com](ftp://ftp.stratus.com) and get the file `/pub/vos/posix/alpha/posix.save.evf.gz` in binary file-transfer mode. Or use the Uniform Resource Locator (URL) <ftp://ftp.stratus.com/pub/vos/posix/alpha/posix.save.evf.gz> from your web browser. Instructions for unbundling this file are at <ftp://ftp.stratus.com/pub/vos/utility/utility.html> This is NOT a standard Stratus product.
In VOS Release 14.3.0, the generally-available version of POSIX.1 support is bundled with the VOS Standard C compiler (or Standard C Cross-Compiler). In VOS Release 14.4.0 or higher, it is also bundled with the VOS C Runtime. These are standard Stratus products.
- 4 You must compile this version of Perl 5 on VOS Release 14.1.0 or higher because some of the perl source files contain more than 32,767 source lines. Due to VOS release-compatibility rules, this port of perl may not execute on VOS Release 12 or earlier.
- 5 If you are using the generally-available version of VOS POSIX support, then you should also acquire the VOS GNU C/C++ Compiler and GNU Tools product. When perl is built with this version of POSIX support, it assumes that it can find "bash", "sed" and other POSIX-compatible commands in the directory `/system/gnu_library/bin`.

To build perl using the supplied VOS command macros, change to the "vos" subdirectory and type the command `"compile_perl -processor X"`, where X is the processor type (`mc68020`, `i80860`, `pa7100`,

pa8000) that you wish to use. Note that the generally-available version of POSIX.1 support is not available for the mc68020 or i80860 processors.

Use the "-version alpha" control argument to build perl with the alpha version of POSIX support, and use the "-version ga" control argument to build it with the generally-available version of POSIX. The default is "ga".

Use the "-compiler cc" control argument to build perl with the VOS Standard C compiler. Use the "-compiler gcc" control argument to build it with the GNU GCC compiler. The default is "cc".

You must have purchased the VOS Standard C Cross Compiler in order to compile perl for a processor type that is different from the processor type of the module.

Note that code compiled for the pa7100 processor type can execute on the PA7100, PA8000, PA8500 and PA8600 processors, and that code compiled for the pa8000 processor type can execute on the PA8000, PA8500 and PA8600 processors.

To build full perl using the supplied Configure script and makefiles, change to the "vos" subdirectory and type the command "compile_full_perl" or "start_process compile_full_perl". This will configure, build, and test perl.

Installing Perl 5 on VOS

- 1 If you have built perl using the Configure script, ensure that you have modify permission to >system>ported and type

```
gmake install
```

- 2 If you have built perl using any of the other methods, type

```
install_perl -processor PROCESSOR -name NAME
```

where PROCESSOR is mc68020, i80860, pa7100, or pa8000, as appropriate, and NAME is perl or perl5, according to which name you wish to use.

This command macro will install perl and all of its related files in the proper directories.

- 3 While there are currently no architecture-specific extensions or modules distributed with perl, the following directories can be used to hold such files:

```
>system>ported>lib>perl5>5.8.0>68k
>system>ported>lib>perl5>5.8.0>860
>system>ported>lib>perl5>5.8.0>7100
>system>ported>lib>perl5>5.8.0>8000
```

- 4 Site-specific perl extensions and modules can be installed in one of two places. Put architecture-independent files into:

```
>system>ported>lib>perl5>site_perl>5.8.0
```

Put site-specific architecture-dependent files into one of the following directories:

```
>system>ported>lib>perl5>site_perl>5.8.0>68k
>system>ported>lib>perl5>site_perl>5.8.0>860
>system>ported>lib>perl5>site_perl>5.8.0>7100
>system>ported>lib>perl5>site_perl>5.8.0>8000
```

- 5 You can examine the @INC variable from within a perl program to see the order in which Perl searches these directories.

USING PERL IN VOS

Unimplemented Features of Perl on VOS

If perl is built with the alpha version of VOS POSIX.1 support and if it attempts to call an unimplemented VOS POSIX.1 function, it will print a fatal error message and enter the VOS debugger. This error is not recoverable. See `vos_dummies.c` for a list of the unimplemented POSIX.1 functions. To see what functions are unimplemented and what the error message looks like, compile and execute `"test_vos_dummies.c"`.

Restrictions of Perl on VOS

This port of Perl version 5 to VOS prefers Unix-style, slash-separated pathnames over VOS-style greater-than-separated pathnames. VOS-style pathnames should work in most contexts, but if you have trouble, replace all greater-than characters by slash characters. Because the slash character is used as a pathname delimiter, Perl cannot process VOS pathnames containing a slash character in a directory or file name; these must be renamed.

This port of Perl also uses Unix-epoch date values internally. As long as you are dealing with ASCII character string representations of dates, this should not be an issue. The supported epoch is January 1, 1980 to January 17, 2038.

See the file `pod/perlport.pod` for more information about the VOS port of Perl.

Handling of underflow and overflow

Prior to VOS Release 14.7.0, VOS does not support automatically mapping overflowed floating-point values to +infinity, nor automatically mapping underflowed floating-point values to zero, unlike many other platforms. The Perl `pack` function has been modified to perform such mapping in software on VOS. Performing other floating-point computations that underflow or overflow will probably result in SIGFPE. Don't push your luck.

As of VOS Release 14.7.0, the VOS POSIX runtime sets up the PA-RISC hardware floating-point status register so that the overflow and underflow exceptions do not trap, but instead automatically convert the result to infinity or zero, as appropriate. As of this writing, there are still floating-point operations that can trap, for example, subtracting two infinite values. This is recorded as suggestion `posix-1022`, which is not yet fixed.

TEST STATUS

When Perl 5.8.3 is built using the native build process on VOS Release 14.7.0 and GNU C++/GNU Tools 2.0.2a, all but three attempted tests either pass or result in TODO (ignored) failures. The tests that fail are:

`t/io/tell.t`, test 28 `t/op/pack.t`, test 39 `lib/Net/ing/t/450_service.t`, test 8

SUPPORT STATUS

I'm offering this port "as is". You can ask me questions, but I can't guarantee I'll be able to answer them. There are some excellent books available on the Perl language; consult a book seller.

If you want a supported version of perl for VOS, purchase the VOS GNU C++ and GNU Tools Release 2.0.1 (or later) product from Stratus Technologies, along with a support contract (or from anyone else who will sell you support).

AUTHOR

Paul Green (Paul.Green@stratus.com)

LAST UPDATE

January 15, 2004