

How to access shell environment variables in Perl script

Author : Dan Nanni

Categories : [Development](#), [Perl](#), [Shell](#)

Tagged as : [environment-variable](#)

Question: I would like to access several environment variables in my Perl script. How can I access shell variables from a Perl program?

In some cases you may want to access shell environment variables (e.g., \$PATH, \$PWD, \$HOME, \$LANG) in your Perl script. While you can pass any needed environment variables to your Perl script via @ARGV argument array, this is actually not needed since Perl provides a way to access all available environment variables. Let's find out **how to access shell environment variables in a Perl script**.

In Perl, all environment variables are maintained in a special hash named **%ENV**. To access a particular shell variable `abc` in Perl, you can simply fetch a hash value for key 'abc'. For example, to access \$PATH and \$PWD variables in Perl:

```
my $path_var = $ENV{PATH}; my $pwd_var = $ENV{PWD}; print "PATH: $path_var\n"; print "PWD: $pwd_var\n";
```

Alternatively, you can use the Perl module called `Env`, which allows you to access shell environment variables using global Perl variables with the same names. For example, the above example code can be rewritten by importing `Env` module as follows.

```
use Env; print "PATH: $PATH\n"; print "PWD: $PWD\n";
```

The `Env` module also allows you to import an environment variable as an array. For example:

```
use Env qw(@PATH); foreach (@PATH) { print "$_\n"; }
```

Ask Xmodulo

Find answers to commonly asked Linux questions

<http://ask.xmodulo.com>

```
#!/usr/bin/perl

# import $PATH/$PWD variables as array and scalar
use Env qw(@PATH PWD);

print "PWD: $PWD\n";
print "PATH:\n";

foreach (@PATH) {
    print "$_\n";
}
```