



Carp & Friends

Alberto Manuel Simões
 ambs@di.uminho.pt

Perl, by default, includes just two ways to report errors to warn about anything that went wrong or to die stating that the program cannot continue. For these two tasks Perl provides the `warn` and `die` commands.

`warn` just prints the message I supply to `STDERR` and the program continues running. In its turn, `die` does the same, but aborts the program execution, returning to the shell or command prompt.

There are two interesting things about these two commands. If `$@` is defined and I do not supply a string to the command, Perl uses the contents of `$@` instead.

Also, if I do not supply a new line at the end of my error message, Perl will add a string for me automatically:

```
at myProgramName line errorLine
```

This additional part of the message tells me the program name and the line where the error message comes from.

Carp

With the previous two commands I can write most of my error handling, but sometimes I need more control.

When I want more information in the error report I use the `Carp` module, which comes with Perl so every installation should already have it ready to use. `Carp` includes function equivalents to `warn` and `die` as the `carp` and `croak` functions. They act exactly in the same way as the built-in ones, but print more interesting information to `STDERR`, to help me with finding the error.

For instance, if I have the following program where I want to open a file for writing and return the filehandle for it:

```
my $fh = create_log("/var/log/
mylog");

sub create_log {
    my $file = shift;
    my $fh;
    open $fh, ">$file"
        or die "Can't create $file";
    return $fh;
}
```

Since I check the result of `open`, if I do not have permissions to open that file, my program will `die`, printing the following error message:

```
Can't create /var/log/mylog
at log.pl line 8.
```

Often I include the system error variable `!` to get an idea what went wrong.

```
open $fh, ">$file"
    or die "Can't create $file: $!";
```

Now Perl has a chance to tell me a little more about what went wrong:

```
Can't create /var/log/mylog:
Permission denied
at log.pl line 8.
```

Want to read about a particular topic in Alberto's Module Beginner Seminar? Write to him to suggest a topic.