

## Re: Splitting and comparing file names

---

*Source:* <http://coding.derkeiler.com/Archive/Perl/perl.beginners/2006-12/msg00511.html>

---

- *From:* "Jake" <Jake.Newly@xxxxxxxx>
  - *Date:* 14 Dec 2006 10:13:12 -0800
- 

On Dec 14, 9:42 am, "Paul Lalli" <mri...@xxxxxxxx> wrote:

```
hostname#val1#val2#val3.load
```

Are those # symbols actually part of the file name, or are you using them to demark the different values? Giving actual sample data along with the general format would really be much preferred...

That is a pretty accurate representation of the data. The filename is a contatation of the hostname and uptime values for that host with the # as the value delimiter.

The idea now is to read in each of the file names, generate a mean  
(meanval = (val1+val2+val3)/3)

Are there *\*always\** three values, or is this just the formula for when there are three values? Again, sample data would be nice.

Yes, there are always three values and there will always be the three files unless the write script never ran.

```
my ($hostname, $val1, $val2, $val3, $filename, @filesfound);
```

You have a case of premature declaration. Don't do that. Declare your variables as you need them. Reduces the possibility of name collisions, makes the code more readable, reduces memory consumption.

Great point! I wasn't sure why some scripts did it one way and other did it another and your explanation certainly helps clarify that.

## Re: Splitting and comparing file names

```
chdir ("/store/admin/tools/load");
```

Why? Changing the current local directory is not necessary for opening that directory for reading.

Thanks for pointing that out. I actually had copied that logic from my previous script that required it because of a write to the directory. For some reason opendir doesn't set the path for when you use open...unless I'm missing something there as well.

```
opendir (DIR, "/store/admin/tools/load")
```

You should use lexical file and directory handles these days, not global barewords.

I'm afraid I'm not savvy enough on the lingo to really understand what that means. My apologies...

```
|| die "Directory not found or unable to be opened. $!\n";
```

Don't guess as to the problem in your hard-coded error message. That's what \$! is there for.

```
opendir my $DIR, '/store/admin/tools/load' or die "Could not open directory: $!";
```

I think this is a general case where I dislike most error messages I see and wanted to add more detail. I can understand the need to cut down on unnecessary text or guesses.

```
foreach $filename(@filesfound) {
```

```
use List::Utils qw/sum/; #should really be near the top, not here.
```

```
my @vals = $filename =~ /#(\d+)/g;
```

```
my $avg = sum(@vals) / @vals;
```

Thank you for all the information. I'm sure it will help make me a bit better at writing perl in the future.

.