

convert structured strings to possibly deep hash of hashes

Source: <http://coding.derkeiler.com/Archive/Perl/comp.lang.perl.misc/2006-08/msg00571.html>

- *From:* "seven.reeds" <seven.reeds@xxxxxxxx>
 - *Date:* 8 Aug 2006 22:52:15 -0700
-

Hi,

I have a list of well structured strings, actually they are file paths. This just means there are strings of '/' separated sub-strings. I can easily split these into an array. My question is really one of building a HoH based on all of the string records. My goal is to take strings like:

```
/file.txt
/a/file.txt
/a/b/c
/a/b/c/file.txt
/z/m/w/file.txt
```

and produce something like:

```
%dir_hash(
'file.txt' => "",
'a' => {
'file.txt' => "",
'b' => {
'c' => {
'file.txt'
}
}
},
'z' => {
'm' => {
'w' => {
'file.txt' => ""
}
}
}
)
```

My current feeble attempt was the following. I am getting lost in the "\\%"s and the "%{}"s. To pre-answer some obvious suggestions: yes, I have read perlref and perlreftut and my perl book(s) but I am not

convert structured strings to possibly deep hash of hashes

seeing how to recursively build/populate a HOH in this situation. I have looked through CPAN and did find the IO::Dir module which might help if the data was not static and i was pulling it from the originating directory instead of a flat file.

suggestions?

```
#!/usr/local/bin/perl -w

use strict;
select(STDOUT);
$|++;

use Data::Dumper;

my @path = ();
my %path_hash = ();

@path = split('/', "/a/b/c/d/e/f/g.ics");
shift(@path) if ($path[0] eq "");

&make_path_hash(\%path_hash, @path);

print STDOUT Dumper(%path_hash);

sub make_path_hash
{
my ($path_hash, $this, @list) = @_;

return if (! defined($this));

$this =~ s/^\s+//s;
$this =~ s/\s+$//s;

return if ($this eq "");

$path_hash->{$this} = 1 if (! defined($path_hash->{$this}));

&make_path_hash(\% { $path_hash->{$this} }, @list);
}

.
```