Subject: parse subdirectories
Posted by Helder Marchetto on Mon, 13 Mar 2017 16:32:11 GMT
View Forum Message <> Reply to Message

Hi,
I have widget application that takes a directory as input and generates a widget_tree of the (sub)directory structure.
I've have done this using basically something like (there's more to it that this...):

```
pro dirParser::addsubtree, dir
subs = file_search(dir+'*', /test_directory, count=cnt)
subParent = self.widgets.treeSub[-1]
if cnt gt 0 then foreach sub,subs do self->addSubsTree, sub, subParent
end
```

This works fine, meaning it returns the directory structure and that's great. However, I would like to switch the windows powershell to get the tree structure. Why? Because the above is really slow.

So, I can call the powershell like this:
```
spawn, 'powershell -WindowStyle Hidden "Get-ChildItem -Recurse | ?{ $_.PSIsContainer } | Select-Object FullName"', result, /noshell
```

which gives me a very quick response with something like:
```
IDL> print, transpose(result)

FullName
--------
K:\data\sub-1\2002
K:\data\sub-1\2004
K:\data\sub-1\2005
K:\data\sub-1\2017
K:\data\sub-1\2002\02_01_26
K:\data\sub-1\2002\02_01_28
K:\data\sub-1\2004\04_12_02
K:\data\sub-1\2004\04_12_03
```

and so on (there are many more subdirectories).
Does anybody have a good suggestion how to parse the text contained in the above result array?

I know how to handle strings, but I don't have a good way to sort the subdirectories (for instance "K:\data\sub-1\2002\02_01_26" is a subdirectory of "K:\data\sub-1\2002", but comes only after all the other same level directories are listed).

I would appreciate any suggestion on how to solve the directory listing chaos.

Regards,
Helder

PS: Just for the time comparison: running the above powershell command over a structure of >2000 directories/subdirectories took "just" 8 seconds. The older method took minutes...
