Subject: Re: parse subdirectories
Posted by Helder Marchetto on Wed, 15 Mar 2017 20:05:22 GMT
View Forum Message <> Reply to Message

On Wednesday, March 15, 2017 at 7:23:46 PM UTC+1, alx wrote:
> Le mercredi 15 mars 2017 16:02:42 UTC+1, Helder a écrit :
>> On Tuesday, March 14, 2017 at 12:04:15 AM UTC+1, Helder wrote:
>>> Thanks for the insight Wayne.
>>> I think I will go for the sort/count slashes option. Otherwise programming is no fun.
>>> Cheers, Helder
>>
>> Ok,
>> so I managed this and it is working fine. The improvement is there: crawling through 2000 sub-directories takes ~9 seconds against the >14 sec with the old file_search method.
>> Now the time to execute is limited by the spawn command (8.5 sec), whereas before it was limited by the file_search (13 sec).
>> If anybody requests it and I have time I'll put the code together in a "nicer" way.
>> Cheers,
>> Helder
>
> Maybe you could simply reorder your string array first by using SORT ?
>
> print, result[sort(result)]
>
> K:\data\sub-1\2002
> K:\data\sub-1\2002\02_01_26
> K:\data\sub-1\2002\02_01_28
> K:\data\sub-1\2004
> K:\data\sub-1\2004\04_12_02
> K:\data\sub-1\2004\04_12_03
> K:\data\sub-1\2005
> K:\data\sub-1\2017

Hi Alx,
that's indeed *one* of the things I have to do. However, there's more to it than plain sorting. This because I want to maintain the physical tree structure: therefore I have to determine if any given directory is a node or a leaf and index it accordingly.
Since I like to use this a lot - and wonder why this isn't already available - I will try to order/clean things up and share it when it's done.

What it does not do and will no do, is follow symbolic links and I will not test this on a linux/mac machine (simply don't have one). If anybody is interested, we can share the load :-)

Cheers,
Helder