Subject: Logic problem
Posted by Ryan. on Wed, 16 Jan 2008 21:42:29 GMT
View Forum Message <> Reply to Message

Hi All,

I need assistance with the logic of a particular routine to retrieve a list of files by date. I have tried a few methods but they haven't been successful. I'm posting here to hopefully get some assistance on the logic that I'm using. I can't seem to get it correct. Here are the details:

I have a directory of folders (As you can see the folder names correspond to a time period that overlap):
2004-02-19_2004-02-20
2004-02-20_2004-02-21
2004-02-21_2004-02-25
2004-02-25_2004-02-28
2004-03-06_2004-03-10
2004-03-10_2004-03-13
...

Each of these folders contains various files, I am interested in obtaining one with a particular extension, say *.SAS (I should note, that sometimes this file does not exist within the folder).

I want to have a special routine that given a date (or a start and end date) to return the full path of the *.SAS file(s) or if it doesn't exist to print out a statement saying it doesn't exist. If the desired dates are spread over 2 or more folders I want it to return all the paths of the files. If the date desired lands on the overlapping part of the date (e.g. 2004-02-20 in sample folders above), I want it to return the path of the two files.

Here are some examples of what I would like returned (using the list of folders above):

IDL> print, findsasfiles(JULDAY(2,19,2004))
full_path/2004-02-19_2004-02-20/file.SAS

IDL> print, findsasfiles(JULDAY(2,20,2004))
full_path/2004-02-19_2004-02-20/file.SAS
full_path/2004-02-20_2004-02-21/file.SAS

IDL> print, findsasfiles(JULDAY(2,19,2004), JULDAY(2,20,2004))
full_path/2004-02-19_2004-02-20/file.SAS
full_path/2004-02-20_2004-02-21/file.SAS

```
IDL> print, findsasfiles( JULDAY(2,20,2004), JULDAY(2,25,2004) )
full_path/2004-02-19_2004-02-20/file.SAS
full_path/2004-02-20_2004-02-21/file.SAS
full_path/2004-02-21_2004-02-25/file.SAS
full_path/2004-02-25_2004-02-28/file.SAS
```

Here is the function so far:

```
FUNCTION findsasfiles, date, enddate, MISSING=missing, $
 NMISSING=nmissing

   ;Get directory names:
   ;directory of SAS files
   sasdir = FILEPATH('', ROOT_DIR=rch_getrootdir(),
SUBDIRECTORY=['plan'])

   ;Getting the list of folder names in 'sasdir' directory
   dirlist = FILE_SEARCH(sasdir+'*',  COUNT=nDirs, /FULLY_QUALIFY_PATH, $
     /TEST_DIRECTORY)

   ;Remove erroneous directory names:
   ;make sure no extra folder names are found except for the one
;corresponding to a time span
   p = STRPOS(dirlist, '200')
   diridx = WHERE(p+1, ndirs)
   IF ndirs GT 0 THEN dirlist = STRMID(dirlist[WHERE(p+1)], 1#p) $
     ELSE dirlist = STRMID(dirlist, 1#p)

   ;some needed constants
   dirs = STRARR(nDirs)
   dirdates = DBLARR(nDirs, 2)

   ;Extract dates from folder names:
   FOR i = 0L, ndirs-1 DO BEGIN
     splitdir = STRSPLIT(dirlist[i], PATH_SEP(), COUNT=slashcnt)

     ;retrieve directory name
     dirs[i] = STRMID(dirlist[i], splitdir[slashcnt-1])

     ;retrieve folder name and dates
     datesplit = STRSPLIT(dirs[i], '_', /EXTRACT)
     dirdates[i,0] = JULDAY(STRMID(datesplit[0],5,2),
STRMID(datesplit[0],8,2), $
       STRMID(datesplit[0],0,4))
     dirdates[i,1] = JULDAY(STRMID(datesplit[1],5,2),
STRMID(datesplit[1],8,2), $
```

```
      STRMID(datesplit[1],0,4))
  ENDFOR


;****************************** NEED HELP AFTER THIS POINT *****
  ;Find dates that are searched for:
  idx = WHERE(dirdates[*,0] GE date, startcnt)
  IF startcnt LT 1 THEN BEGIN
    PRINT, 'Start Date Not Found. Returning...'
    RETURN, -1S
  ENDIF

  IF endcorrect THEN BEGIN
    endidx = WHERE(dirdates[*,1] GT enddate, endcnt)
    IF (endcnt GT 0) THEN BEGIN
      idx = [idx[0], endidx[0]]
      folders = dirlist[idx[0]:idx[1]]
    ENDIF
  ENDIF ELSE folders = dirlist[idx[0]]

  ;Discover if SAS file exists and return it if it does:
  nFiles = FIX(N_ELEMENTS(folders))
  files = STRARR(nFiles)
  missing = STRARR(nFiles)
  sascounter = 0S
  nMissing = 0S

  FOR j=0L, nFiles-1 DO BEGIN
    sasfind = FILE_SEARCH(sasdir+folders[j], '*.SAS',
COUNT=sasfindcount, /FULLY_QUALIFY_PATH)
    IF (sasfindcount GT 0) THEN BEGIN
      files[sascounter] = sasfind
      sascounter += 1
    ENDIF ELSE BEGIN
      PRINT, 'No SAS file found in folder: '+folders[j]
      missing[nMissing] = folders[j]
      nMissing += 1
    ENDELSE
  ENDFOR

  CASE 1 OF
    (nMissing EQ 0) AND (sascounter EQ 0): BEGIN
      files = -1S
      missing = -1S
    END
    (nMissing GT 0) AND (sascounter EQ 0): BEGIN
      files = -1S
      print, 'No files found'
```

```
    END
    (nMissing EQ 0) AND (sascounter GT 0): BEGIN
      missing = -1S
      print, 'All files found'
    END
    ELSE: BEGIN
      files = files[0:(sascounter-1)]
      missing = missing[0:(nMissing-1)]
      PRINT, 'A bit of this, a bit of that'
    END
  ENDCASE

;******************

  RETURN, files

END
```

The routine as it stands doesn't quite work because the indices returned (the final 'idx' variable) can be backwards (eg. [175,174]) and thus cause an error when trying to execute the line "folders = dirlist[idx[0]:idx[1]]".

To me it doesn't seem like an overly difficult problem but I've spent the last 2 or 3 days trying to get it right with no success. I need some new minds to help me with this.

Thanks,
Ryan.