

---

Subject: Re: Deleting files from IDL

Posted by [David S. Foster/Admin](#) on Mon, 22 May 1995 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

cabr69@ccsun.strath.ac.uk ( "D.H.Brooks") wrote:

>  
> Hello all,  
>  
> I have a file selection widget which gives me an option to pick an old file, pick  
> no file or create a new one. The name for the new file is entered in an editable  
> text widget. The problem is that if i make a mistake with the name the file  
> gets created immediately, and when i enter a new name i end up with two files. Ok i  
> could create the file after i'm finished rather than straight away but for my  
> application this is not practicable. I'm running under Unix and if i use  
>  
> spawn,'rm filename'  
>  
> it doesn't work since filename in the IDL code is a variable (e.g.state.file).What i  
> would really like is an IDL command which removes a file but i don't know of any.  
> I'm thinking of something like  
>  
> IDL>remove\_file,state.file  
>  
> Does anyone know if such a thing exists?  
>  
> Thanks in advance,  
>  
> David

In UNIX you can use the SPAWN routine to do this. I've written a short routine called DELETE\_FILES:

```
=====
; DELETE_FILES.PRO  11-22-94
;
; Function to delete specified files. Returns 0 if successful, -1 if not.
; First searches for files and if not found returns 0 (otherwise an
; "error" occurs when you try to delete them).
;
```

```
FUNCTION delete_files, to_delete
```

```
ret = file_find(to_delete, temp) ; First search for files
case (ret) of
  -1: return, 0 ; No files found (ok)
  -2: return, -1 ; Argument To_Delete undefined/null
  else:
endcase
```

```

command = 'rm ' + to_delete + ' 2>&1'
SPAWN, [ "/bin/sh", "-c", command ], results, /NOSHELL
if (results(0) eq "") then begin
    return, 0
endif else begin
    message, ' (rm) ' + results(0), /continue
    return, -1
endelse

END

```

=====

Here is the FILE\_FIND function used above. I like it better than the FINDFILE IDL function:

=====

```

;-----
; function File_Find
;
; Takes file specification, expands it and returns a list of matching
; filenames. Returns -1 if no files are found (and argument FNAMES will
; be undefined). Returns -2 if arg FILESPEC is undefined or null.
; Returns the number of files found if no errors. Mostly borrowed from
; the PICKFILE widget.
;-----

```

```

FUNCTION file_find, filespec, fnames

```

```

if (n_elements(filespec) eq 0) then return, -2
if (strlen(filespec) eq 0) then return, -2

```

```

ON_IOERROR, io_error

```

```

SPAWN, ["/bin/sh", "-c", "ls -l " + filespec + $,
    " 2> /dev/null"], results, /NOSHELL

```

```

IF KEYWORD_SET(results) THEN BEGIN
    firsts = STRUPCASE(STRMID(results, 0, 1))
    fileinds = (WHERE(((firsts EQ "F") OR (firsts EQ "-") OR $
        (firsts EQ "I")), found))
    IF (found GT 0) THEN BEGIN
        results = results(fileinds)
        FOR i = 0, N_ELEMENTS(results) - 1 DO BEGIN
            spaceinds = WHERE(BYTE(results(i)) EQ 32)
            spaceindex = spaceinds(N_ELEMENTS(spaceinds) - 1)

```

```
        results(i) = STRMID(results(i), spaceindex + 1, 100)
    ENDFOR
    fnames = results    ; Store list of file names
ENDIF ELSE RETURN, -1
ENDIF ELSE RETURN, -1

return, n_elements(fnames)

io_error: return, -1

END
```

=====

Hope this helps.

Dave

---