
Subject: importing ascii files

Posted by [lehmann_heather](#) on Thu, 11 Jul 2002 21:34:40 GMT

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

We have a program written in IDL that extracts data and puts it into a text file. We want to put the columns of the text file into some kind of storage device so that we can access the columns separately in another IDL program. We are trying to write a program that incorporates the `import_ascii` macro, but are open to any other ideas at this point.

Heather

Subject: Re: importing ascii files

Posted by [Pavel A. Romashkin](#) on Thu, 11 Jul 2002 22:04:20 GMT

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

How about the following:

; P. Romashkin, 2001

; Parameters: FILENAME - name of ASCII file to read, OUT - structure with

; names of variables as fields, N_COLS - number of columns to read.

```
pro read_ascii_columns, filename, out, n_cols=n_cols
```

```
  compile_opt IDL2, obsolete
```

```
  if n_elements(n_cols) eq 0 then message, 'Specify number of columns!'
```

```
  header = "
```

```
  buffer = fltarr(n_cols, 1000)
```

```
  openr, SourceFile, /get_lun, filename
```

```
  readf, SourceFile, header
```

```
  ; If Buffer was longer than file, reading will fail. In such case,
  reduce buffer.
```

```
  FileTooShort = 0b ; Just to skip buffer reduction the first time.
```

```
  on_ioError, reduce_buffer
```

```
  reduce_buffer : if FileTooShort eq 1b then begin
```

```
    unread_rows = (fstat(SourceFile)).Transfer_Count / n_cols
```

```
    if unread_rows eq 0 then goto, finish
```

```
    Buffer=fltarr(n_cols, unread_rows)
```

```
    point_lun, SourceFile, LastPos
```

```
  endif
```

```
  FileTooShort = 1b ; In case of ioError, this will be set to 1b
```

```
  while not EOF(SourceFile) do begin
```

```
    point_lun, (-SourceFile), LastPos ;Memorize start in case it fails.
```

```
    readf, SourceFile, Buffer
```

```
if n_elements(resultArray) eq 0 then resultArray = Buffer $
  else ResultArray = [[ResultArray], [Buffer]]
endwhile
```

finish:

free_lun, SourceFile ; Done using the file.

```
header = strsplit(header, /extract)
resultArray = transpose(resultArray)
out = create_struct(header[0], resultArray[*, 0])
for i = 1, n_elements(header)-1 do $
  out = create_struct(out, header[i], resultArray[*, i])
end
```

You may have to clean up line breaks.

Cheers,

Pavel

Heather wrote:

>

> We have a program written in IDL that extracts data and puts it into a
> text file. We want to put the columns of the text file into some kind
> of storage device so that we can access the columns separately in
> another IDL program. We are trying to write a program that
> incorporates the import_ascii macro, but are open to any other ideas
> at this point.

>

> Heather