Subject: Re: Reading in a non-rectangular array
Posted by Jean H. on Wed, 25 Oct 2006 17:52:11 GMT
View Forum Message <> Reply to Message

Hi,
Try to read each line as a string, then split this string using strSplit
 with the keywork Preserve_Null set. Then you can build your 6*n array,
filling the location having a null string value with a flag value...

Jean

willettk@gmail.com wrote:
> G'day,
>
> I'm trying to read in a data table to IDL that is (gasp)
> non-rectangular (thanks very much to the astronomers involved). The
> data is in an ASCII file, but certain cells will simply be blank if
> there was no data for that particular value (so out of 6 columns, any
> given row might have 0-6 actual values). Reading this into IDL, though,
> doesn't seem to work according to Hoyle. This is what I have tried (and
> received):
>
> ____________
>
> fname = 'table1.dat'
> openr, lun1, fname, /get_lun
> nlines = file_lines(fname)
> data = fltarr(6, nlines)
> readf, lun1, data
>
> % READF: Input conversion error. Unit: 100
>   File: /Users/willettk/table1
> % Execution halted at: $MAIN$
>
> ____________
>
> Any ideas that might help me read the data? Many thanks, if so.
>

Subject: Re: Reading in a non-rectangular array
Posted by willettk on Wed, 25 Oct 2006 19:42:18 GMT
View Forum Message <> Reply to Message

Thanks for the tip. I haven't used strmid and strsplit before, but they
seem suited to this sort of problem.

On Oct 25, 11:52 am, "Jean H." <jghas...@DELTHIS.ucalgary.ANDTHIS.ca>
wrote:
> Hi,
> Try to read each line as a string, then split this string using strSplit
>   with the keywork Preserve_Null set. Then you can build your 6*n array,
> filling the location having a null string value with a flag value...
>
> Jean
>
> wille...@gmail.com wrote:
>> G'day,
>
>> I'm trying to read in a data table to IDL that is (gasp)
>> non-rectangular (thanks very much to the astronomers involved). The
>> data is in an ASCII file, but certain cells will simply be blank if
>> there was no data for that particular value (so out of 6 columns, any
>> given row might have 0-6 actual values). Reading this into IDL, though,
>> doesn't seem to work according to Hoyle. This is what I have tried (and
>> received):
>
>> ___________
>
>> fname = 'table1.dat'
>> openr, lun1, fname, /get_lun
>> nlines = file_lines(fname)
>> data = fltarr(6, nlines)
>> readf, lun1, data
>
>> % READF: Input conversion error. Unit: 100
>>   File: /Users/willettk/table1
>> % Execution halted at: $MAIN$
>
>> ___________
>
>> Any ideas that might help me read the data? Many thanks, if so.

Subject: Re: Reading in a non-rectangular array
Posted by Braedley on Thu, 26 Oct 2006 14:50:46 GMT

View Forum Message <> Reply to Message

willettk@gmail.com wrote:
> Thanks for the tip. I haven't used strmid and strsplit before, but they
> seem suited to this sort of problem.
>
> On Oct 25, 11:52 am, "Jean H." <jghas...@DELTHIS.ucalgary.ANDTHIS.ca>
> wrote:
>> Hi,

>> Try to read each line as a string, then split this string using strSplit
>>   with the keywork Preserve_Null set. Then you can build your 6*n array,
>> filling the location having a null string value with a flag value...
>>
>> Jean
>>
>> wille...@gmail.com wrote:
>>> G'day,
>>
>>> I'm trying to read in a data table to IDL that is (gasp)
>>> non-rectangular (thanks very much to the astronomers involved). The
>>> data is in an ASCII file, but certain cells will simply be blank if
>>> there was no data for that particular value (so out of 6 columns, any
>>> given row might have 0-6 actual values). Reading this into IDL, though,
>>> doesn't seem to work according to Hoyle. This is what I have tried (and
>>> received):
>>
>>> ___________
>>
>>> fname = 'table1.dat'
>>> openr, lun1, fname, /get_lun
>>> nlines = file_lines(fname)
>>> data = fltarr(6, nlines)
>>> readf, lun1, data
>>
>>> % READF: Input conversion error. Unit: 100
>>>   File: /Users/willettk/table1
>>> % Execution halted at: $MAIN$
>>
>>> ___________
>>
>>> Any ideas that might help me read the data? Many thanks, if so.

There are other methods involving read_ASCII and ASCII_template, but
if the afforementioned method works for you, use that first.