
Subject: Read_ASCII and 'invalid' ascii files

Posted by [jaden](#) on Fri, 10 Dec 2004 19:01:36 GMT

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

Hi Everyone,

I have been using IDL's read_ascii function to read in data files using a template. The only problem is that all the source files have a +/- symbol in the header, and IDL reports that these files are not valid ascii files, when they clearly are. When I open with a text editor, and remove the suspect character, the function runs perfectly. Any suggestions?

Jaden

Subject: Re: Read_ASCII and 'invalid' ascii files

Posted by [Benjamin Hornberger](#) on Fri, 10 Dec 2004 19:15:10 GMT

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

Jaden wrote:

> Hi Everyone,

>

> I have been using IDL's read_ascii function to read in data files
> using a template. The only problem is that all the source files have a
> +/- symbol in the header, and IDL reports that these files are not
> valid ascii files, when they clearly are. When I open with a text
> editor, and remove the suspect character, the function runs perfectly.
> Any suggestions?

>

> Jaden

I think the ASCII code defines only 128 characters (7 bit). The +/- sign is part of an "extended" ASCII code, which makes use of the 8th bit but is not strictly defined.

Benjamin

Subject: Re: Read_ASCII and 'invalid' ascii files

Posted by [Michael Wallace](#) on Fri, 10 Dec 2004 20:14:32 GMT

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

>> I have been using IDL's read_ascii function to read in data files
>> using a template. The only problem is that all the source files have a
>> +/- symbol in the header, and IDL reports that these files are not
>> valid ascii files, when they clearly are. When I open with a text

```
>> editor, and remove the suspect character, the function runs perfectly.
>> Any suggestions?
>>
>> Jaden
>
>
> I think the ASCII code defines only 128 characters (7 bit). The +/- sign
> is part of an "extended" ASCII code, which makes use of the 8th bit but
> is not strictly defined.
```

There's a pretty standard technique for determining whether a file is ASCII or binary. Basically, you look at the bytes within the file and determine what percentage of the bytes fall into the printable standard ASCII range. If this percentage is really high, you can guess that the file is probably text. Otherwise the file is probably binary. It appears that IDL's check is a lot more rigorous. If any byte falls outside printable standard ASCII, it's throwing the error you see.

Now, what to do about it? I don't know what system you're on, but if you're using *nix, you can use a simple little sed command to remove or replace the plus/minus sign. For example, the following command finds all occurrences of the plus/minus, removes them and saves the result in a new file.

```
$ sed "s/±½/" file.txt > new_file.txt
```

-Mike
