Subject: Re: Testing validity of numerical content of a string
Posted by Sven Geier on Thu, 07 Dec 2006 06:21:26 GMT
View Forum Message <> Reply to Message

Brian Larsen wrote:

> I just read you post a little closer and the numbers you have are more
> complicated but not too bad.
>
> IDL> print, stregex('hello', '[a-df-zA-DF-Z]', /boolean)
> 1
> IDL> print, stregex('3.14159', '[a-df-zA-DF-Z]', /boolean)
> 0
> IDL> print, stregex('3.14159e3', '[a-df-zA-DF-Z]', /boolean)
> 0
>
> To make it work perfectly you would need to know a little more info
> about what could be there but if's based on these regular expressions
> can tell the difference between 'hello' and '-.3e-4' which can then
> build valid_float.pro.

I kinda mostly agree with the regexp direction, but I'd recommend looking for a lot more than just "what characters are in the string". Order counts, for example: -1.e4 -1e.4 1e.-4 1-e4. all contain the same characters, but not all are well-formed doubles.

(And if the OP manages to get bit-errors that turn "3.141" into "hello", then I'd like to meet those bit-errors... ;)

I'm somewhat rusty on this, but the full regexp for a double would probably be something something like

-?[0-9]\.?[0-9]*([de]-?[0-9]{1,2})?

i.e. zero or more spaces, then zero or one minus sign, then zero or more digits, then zero or one literal period followed by zero or more additional digits followed by zero or one instance of (the letter 'e' or 'd' followed by zero or one minus sign followed by one or two digits).

And this probably doesn't cover all of it...

I always found IDLs implementation of the regexp library to be rather slow; so this might not be practical for a large amount of data anyways.

In a pinch one might just force the conversion and catch any resulting error-messages with ON_ERROR -- that's ugly, but might be a lot easier than trying to parse strings...

- S

--
http://www.sgeier.net
My real email address does not contain any "Z"s.