Subject: Re: function to convert string to array???
Posted by zawodny on Thu, 30 Mar 1995 08:00:00 GMT
View Forum Message <> Reply to Message

In article <1995Mar29.192708.17563@alw.nih.gov> cox <cox@colt.cpi.com> writes:

> Here is my routine which converts a string of space or comma-delimited
> items into a string array. It doesn't assume what the output should
> be, so it doesn't convert to FLOAT of INTEGER. It avoids the use of
> loops and makes as much use as possible of built-in IDL functions.
> It should work with ASCII and non-ASCII character sets.
>
> ;+
> ; NAME:
> ; parse_string
> ;
> ; PURPOSE:
> ; This procedure was written to parse a string for substrings
> ; separated by either spaces or commas.

Some stuff deleted

> SpacePos = where(ByteStrng eq Separators(1), count)
> count = count + 1
>
> SubString = strarr(count)
> SpacePos = [-1,SpacePos,strlen(strng_)+1]
> for i = 0, count - 1 do begin
> SubString(i) = strmid(strng_, SpacePos(i)+1, SpacePos(i+1)-SpacePos(i)-1)
> endfor
> return, SubString
> end

This is all great, but why stop here and return a vector of strings
when you can use READS to turn the string into array of numbers or a
structure. Actually, you can do this directly without doing any of
the searching for commas, tabs, or other delimiter and replacing them
with space characters.

Just do this:

 READS,input_string,numbers

Where numbers is an array (intarr(10), or fltarr(12), or whatever)
or a structure a={f1: 0, f2: 0.0, f3: bytarr(100), ... }

This seems to me to be much easier to do.

BTW, this has zero loops as well.

Just my $0.02.

--

Joseph M. Zawodny (KO4LW) NASA Langley Research Center
Internet: j.m.zawodny@larc.nasa.gov MS-475, Hampton VA, 23681-0001
TCP/IP: ko4lw@ko4lw.ampr.org Packet: ko4lw@n4hog.va.usa.na