Subject: function to convert string to array???
Posted by davis on Wed, 15 Mar 1995 19:45:40 GMT
View Forum Message <> Reply to Message

Hi,

I was wondering if there is a routine that will read a string and convert it to an array of floats. For example,

```
str = " 10.8  8.4              9.3";
x = string_to_array (str);
```

would yield an array of 3 elements: x = [10.8 8.4 9.3]

whereas string_to_array (" 20.1 9.2") yields the two element array [20.1 9.2]. Here is my implementation of this:

```
FUNCTION string_to_array, str
  str = strtrim (strcompress (str), 2);
  maxpos = strlen (str)

  this_pos = 0
  n = 1

  while (1) do begin
    this_pos = strpos (str, " ", this_pos)
    if (this_pos eq -1) then goto, break1
    n = n + 1
    this_pos = this_pos + 1
  endwhile
  break1:

  a = fltarr (n);

  ; Now go back and fill the array

  maxlen = strlen (str)

  for i = 0, n - 2 do begin
    reads, str, ai
    a(i) = ai
    str = strmid (str, 1 + strpos (str, " ", 0), maxlen)
  endfor
  ;
  ; Now the last element
  ;
  reads, str, ai
  a(n - 1) = ai
```

```
  return, a

 END
```

Thanks,
--John

Subject: Re: function to convert string to array???
Posted by cox on Wed, 29 Mar 1995 08:00:00 GMT
View Forum Message <> Reply to Message

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.
;
; CATEGORY:
;
; CALLING SEQUENCE:
; substrings = parse_string(strng)
;
; POSITIONAL PARAMETERS:
;   <INPUT>
; strng  STRING  A string containing multiple sub-
;     strings separated by commas or by
;     spaces.
;   <INPUT/OUTPUT>
; None.
;
;   <OUTPUT>
; None.
;
; KEYWORD PARAMETERS:
; None.
;
; MODIFICATION HISTORY:
; Written by:  R.J. Cox, CPI, 1993
; Modified by:  R.J. Cox, CPI, 2 Feb 1995
```

```
;    Extensively recoded to make more use of
;    built in IDL commands and to remove loops.
;-
 ;********************************************************** **********
;
function parse_string, strng
;
 ;********************************************************** **********
Separators = BYTE(', ')
ByteStrng = BYTE(strng)
;----------------------------------------
; replace commas by single spaces
;----------------------------------------
CommaPos = where(ByteStrng eq Separators(0), count)
if (count gt 0) then ByteStrng(CommaPos) = Separators(1)

;----------------------------------------
; Replace all whitespace by a single space
;----------------------------------------
strng_    = STRCOMPRESS(STRING(ByteStrng))
ByteStrng = BYTE(Strng_)
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
```

Robin Cox,
Computational Physics, Inc.

Subject: Re: function to convert string to array???
Posted by chase on Thu, 30 Mar 1995 08:00:00 GMT

View Forum Message <> Reply to Message

>>>> > "Joseph" == Joseph M Zawodny <zawodny@arbd0.larc.nasa.gov> writes:
In article <3lek18$pm0@reznor.larc.nasa.gov> zawodny@arbd0.larc.nasa.gov (Joseph M Zawodny) writes:

Joseph> This is all great, but why stop here and return a vector of strings

Joseph> when you can use READS to turn the string into array of numbers or a
Joseph> structure.

You may want to stop because you want to parse strings, i.e., you may
be working with fields instead of numerical fields. Although this
does not apply to the original application that started this thread,
it is helpful for other applications.

Chris
--
==========
Bldg 24-E188
The Applied Physics Laboratory
The Johns Hopkins University
Laurel, MD 20723-6099
(301)953-6000 x8529
chris.chase@jhuapl.edu

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 <3lek18$pm0@reznor.larc.nasa.gov> zawodny@arbd0.larc.nasa.gov (Joseph M
Zawodny) writes:
> 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.

>
> 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), ... }
>

I was not sure about the use of a structure in the READS call,
but the following example illustrates my point more completely.

IDL> s='12, 24 36 3.456 2 3 4 5, 6,7'

;   ^ there is a tab in this string here

IDL> a={f1:0, f2:0, f3:0L, f4:0., f5:intarr(6)}

```
IDL> help,/str,a
** Structure <4009be08>, 5 tags, length=24, refs=1:
   F1              INT           0
   F2              INT           0
   F3              LONG              0
   F4              FLOAT         0.00000
   F5              INT       Array(6)

IDL> reads,s,a

IDL> help,/str,a
** Structure <4009be08>, 5 tags, length=24, refs=1:
   F1              INT           12
   F2              INT           24
   F3              LONG              36
   F4              FLOAT         3.45600
   F5              INT       Array(6)
IDL> print,a.f5
      2       3       4       5       6       7
```

Have fun,
--
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

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