Subject: Re: str_sep bug
Posted by thompson on Tue, 26 Nov 1996 08:00:00 GMT
View Forum Message <> Reply to Message

Phil Williams <williams@irc.chmcc.org> writes:

> I am using v1.9 of str_sep dated December 1995.

Evidently there was a bug introduced somewhere between version 1.3 and 1.9.
Maybe somebody from RSI could comment. In the meantime, here's version 1.3.

Bill Thompson

```
 ============================================================
==================
; $Id: str_sep.pro,v 1.3 1995/01/06 21:59:22 dave Exp $
; Copyright (c) 1992-1995, CreaSo Creative Software Systems GmbH,
; and Research Systems, Inc.  All rights reserved.
; Unauthorized reproduction prohibited.
;+
; NAME:
;    STR_SEP
;
; PURPOSE:
;    This routine cuts a string into pieces which are separated by the
;    separator string.
; CATEGORY:
;    String processing.
; CALLING SEQUENCE:
;    arr = STR_SEP(string, separator)
;
; INPUTS:
;    str - The string to be separated.
;    sep - The separator.
;
; KEYWORDS:
;    ESC = escape character.  Only valid if separator is a single character.
;  Characters following the escape character are treated
;  literally and not interpreted as separators.
;  For example, if the separator is a comma,
;  and the escape character is a backslash, the character
;  sequence 'a\,b' is a single field containing the characters
;  'a,b'.
;    REMOVE = if set, remove all blanks from fields.
;    TRIM = if set, remove only leading and trailing blanks from fields.
;
; OUTPUT:
;    An array of strings as function value.
```

```
;
; COMMON BLOCKS:
;    None
;
; SIDE EFFECTS:
;    No known side effects.
;
; RESTRICTIONS:
;    None.
;
; EXAMPLE:
;    array = STR_SEP ("ulib.usca.test", ".")
;
; MODIFICATION HISTORY:
; July 1992, AH, CreaSo  Created.
; December, 1994, DMS, RSI Added TRIM and REMOVE.
;-
function STR_SEP, s, sep, REMOVE = remove, TRIM = trim, ESC=esc


spos = 0L
if n_elements(esc) gt 0 then begin  ;Check for escape character?
  if strpos(s, esc) lt 0 then goto, no_esc ;None in string, use fast case
  besc = (byte(esc))(0)
  bsep = (byte(sep))(0)
  new = bytarr(strlen(s)+1)
  new(0) = byte(s)
  j = 0
  for i=0, n_elements(new)-2 do begin
    if new(i) eq besc then begin
 new(j) = new(i+1)
 i = i + 1
    endif else if new(i) eq bsep then new(j) = 1b $   ;Change seps to 1b char
    else new(j) = new(i)
    j = j + 1
    endfor
  new = string(new(0:j-1))
  w = where(byte(new) eq 1b, count)  ;where seps are...
  arr = strarr(count+1)
  for i=0, count-1 do begin
 arr(i) = strmid(new, spos, w(i)-spos)
 spos = w(i) + 1
 endfor
  arr(count) = strmid(new, spos, strlen(s))  ;Last element
  goto, done
  endif   ;esc

no_esc:
```

```
if strlen(sep) eq 1 then begin ;Single character separator?
    w = where(byte(s) eq (byte(sep))(0), count)  ;where seps are...
    arr = strarr(count+1)
    for i=0, count-1 do begin
 arr(i) = strmid(s, spos, w(i)-spos)
 spos = w(i) + 1
 endfor
    arr(count) = strmid(s, spos, strlen(s))  ;Last element
endif else begin  ;Multi character separator....
    n = 0     ; Determine number of seperators in string.
    repeat begin
 pos = strpos (s, sep, spos)
 spos = pos + strlen(sep)
 n = n+1
    endrep until pos eq -1

    arr = strarr(n)    ; Create result array
    spos = 0
    for i=0, n-1 do begin   ; Separate substrings
      pos = strpos (s, sep, spos)
      if pos ge 0 then arr(i) = strmid (s, spos, pos-spos) $
      else arr(i) = strmid(s, spos, strlen(s))
      spos = pos+strlen(sep)
   endfor
endelse

done:
if keyword_set(trim) then arr = strtrim(arr,2) $
else if keyword_set(remove) then arr = strcompress(arr, /REMOVE_ALL)
return, arr
end
```