Subject: Re: quoted null character
Posted by Martin Schultz on Fri, 25 Feb 2000 08:00:00 GMT
View Forum Message <> Reply to Message

George McCabe wrote:
>
> hello,
>
> i had one hell of a time today figuring out some unexpected results in
> IDL string processing, and i wonder of any one has encountered similar
> problems.
>
> take the following code segment -
>
> IDL> ; the variable 'F' is string type result from another procedure
> IDL> cmd = 'perl -e '+""+'$_ = '+""+F+""+' ; print if (
> m/AC[0-9]{13}.L0/)'+""
> IDL> print, cmd
> IDL> spawn,cmd,res
> IDL> print, res
>
> what is executed by the unix shell is -
> perl -e '$_ = "<string>" ; print if (m/AC[0-9]{13}.L0/)'
>
> except, when...
>
> in case when the string in variable 'F' ends in a null character.
> then one gets an incomplete concatenation beyond the null. since i'ld
> already spent an hour or so rearranging quote marks and parting up the
> process to find the problem, i didn't spend time after discovering the
> cause to figure out if IDL was interpreting the null as a quote or
> other weird possibilities.
>
> anyone?
>
> Thank you, George
>

hmmm. All I can say is that print,byte('') yields 0. This may be the
problem and the
solution ?

To get rid of all null characters in a string (i.e. replace them with
blanks e.g.),
do the following:
```
  bf = byte(F)   ; byte representation of string F
  rc = (byte(' '))[0]  ; byte repr. of replacement character (blank)
  w = where(bf eq 0,count)
```

```
   if count gt 0 then bf[w] = rc
   newF = string(bf)
...
```

This functionality is implemented in my (heavily revised) strrepl function which I
attach below (and which has become obsolete by IDL 5.3 ?).

Cheers,
Martin

--
[[[[[[[[[[
[[Dr. Martin Schultz   Max-Planck-Institut fuer Meteorologie   [[
[[Bundesstr. 55, 20146 Hamburg   [[
[[phone: +49 40 41173-308   [[
[[fax: +49 40 41173-298   [[
[[martin.schultz@dkrz.de   [[
[[[[[[[[[[

```
; $Id: strrepl.pro,v 1.10 1999/01/22 20:12:17 mgs Stab $
;-------------------------------------------------------------- --
;+
; NAME:
;       STRREPL (function)
;
; PURPOSE:
;       Replace all occurences of one character in a string
;       with another character. The character to be replaced
;       can either be given as string of length 1 or as an
;       index array containing the character positions
;       (see strwhere). This function also works for string arrays.
;
; CATEGORY:
;       String Routines
;
; CALLING SEQUENCE:
;       Result = STRREPL(str, fromchar, tochar [,/IGNORECASE])
;
; INPUTS:
;       STR      -> the string to be changed
;
;       FROMCHAR -> either: a string of length 1 (the character to
;                   be replaced)
;                 or: an index array with the character positions
;
;       TOCHAR   -> replacement character
```

```
;
; KEYWORD PARAMETERS:
;       IGNORECASE -> if set, fromchar will be treated case-insensitive
;             (only if fromchar is a character)
;
; OUTPUTS:
;       A string of same length as the input string
;
; SUBROUTINES:
;
; REQUIREMENTS:
;
; NOTES:
;       Uses SIZE(/TYPE) available since IDL 5.2
;
; EXAMPLES:
;       ; Convert a Unix filename to Windows
;       ufile = '/usr/local/idl/lib/test.pro'
;       wfile = 'c:' + strrepl(ufile,'/','\')
;       print,wfile
;       ; prints "c:\usr\local\idl\lib\test.pro"
;
;       ; Use with index (uses strwhere function)
;       a = 'abcdabcdabcd'
;       index = [ strwhere(a,'a'), strwhere(a,'b') ] > 0
;       print,strrepl(a,index,'#')
;       ; prints  "##cd##cd##cd"
;
; MODIFICATION HISTORY:
;       mgs, 02 Jun 1998: VERSION 1.00
;       mgs, 24 Feb 2000: - rewritten
;                         - now accepts character argument
;                         - added IGNORECASE keyword
;
;-
; Copyright (C) 1998-2000, Martin Schultz
; This software is provided as is without any warranty
; whatsoever. It may be freely used, copied or distributed
; for non-commercial purposes. This copyright notice must be
; kept with any copy of this software. If this software shall
; be used commercially or sold as part of a larger package,
; please contact the author to arrange payment.
; Bugs and comments should be directed to mgs@io.harvard.edu
; with subject "IDL routine strrepl"
 ;-------------------------------------------------------------- --


function strrepl,str,fromchar,tochar,IGNORECASE=ignorecase
```

```
ON_ERROR,2    ; return to caller

    ; argument testing
    if n_params() lt 3 then begin
      message,'Usage: strrepl,str,fromchar,tochar[,/IGNORECASE])'
    endif

    ; make working copy of string and convert to a byte array
    bstr = byte(string(str))

    ; fromchar is given as character
    if size(fromchar,/TYPE) eq 7 then begin
      ; ignore case?
      if keyword_set(ignorecase) then begin
        ; call strrepl recursively w/o the IGNORE_CASE keyword
        res1 = strrepl(str,strupcase(fromchar),tochar)
        res2 = strrepl(res1,strlowcase(fromchar),tochar)
        return,res2
      endif else begin
        ; find all character occurences
        ; must be a single character - use the first
        bfc = (byte(fromchar))[0]
        ; go and search
        w = where(bstr eq bfc,count)
        ; if not found, return original string
        if count eq 0 then return,str
      endelse
    endif else begin
    ; fromchar is already an index array
      w = long(fromchar)
    endelse

    ; make sure index is in range
    test = where(w lt 0 OR w ge n_elements(bstr),tcount)
    if tcount gt 0 then begin
      message,'WARNING: Index out of range!',/Continue
      ; restrict to valid index values
      test = where(w ge 0 AND w lt n_elements(bstr),tcount)
      if tcount gt 0 then begin
        w = w[test]
      endif else begin
      ; no valid indices: return original string
        return,str
      endelse
    endif

    ; convert tochar to a byte value
```

```
    btc  = (byte(tochar))[0]

    ; replace
    bstr[w] = btc

    ; return result as string
    return,string(bstr)

end
```

File Attachments

1) strrepl.pro, downloaded 87 times