Subject: Re: splitting difficult(?) strings
Posted by Craig Markwardt on Tue, 08 Jan 2002 19:42:57 GMT
View Forum Message <> Reply to Message

hradilv.nospam@yahoo.com writes:

- > Any ideas on what would be the easiest (best?) way to split strings
- > that have the following characteristics? Perhaps best understood by
- > looking at examples?
- > > 1)
- > buf= '4 5 6'
- > strsplit(buf,' ',/extract) yields ['4','5','6'] {good}

•••

- > Maybe I can summarize: I would like something that will split the
- > string by spaces, PROTECTED by "s. ie there may be spaces within the
- > "s which should not cause a split.

Using STRSPLIT is like sending a boy to do a man's job. I think there is no other choice but to hand-code what you want. [Counter-examples welcomed.] It \*may\* be possible to code it into a regular expression, but I'm not sure how to handle the "quotes-optional" aspect.

Craig

\_\_\_\_\_

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@cow.physics.wisc.edu Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response

.....

Subject: Re: splitting difficult(?) strings
Posted by Martin Downing on Tue, 08 Jan 2002 22:49:24 GMT
View Forum Message <> Reply to Message

"Craig Markwardt" <craigmnet@cow.physics.wisc.edu> wrote in message news:onr8p0k4se.fsf@cow.physics.wisc.edu...

- > hradilv.nospam@yahoo.com writes:
- >> Any ideas on what would be the easiest (best?) way to split strings
- >> that have the following characteristics? Perhaps best understood by
- >> looking at examples?
- >>
- >> 1)
- >> buf= '4 5 6'

```
>> strsplit(buf,' ',/extract) yields ['4','5','6'] {good}
>> Maybe I can summarize: I would like something that will split the
>> string by spaces, PROTECTED by "s. ie there may be spaces within the
>> "s which should not cause a split.
> Using STRSPLIT is like sending a boy to do a man's job. I think there
> is no other choice but to hand-code what you want. [ Counter-examples
> welcomed. ] It *may* be possible to code it into a regular
> expression, but I'm not sure how to handle the "quotes-optional"
> aspect.
Hi Vince, this works for your examples:
FUNCTION vince_sep, s
IF strpos(s, "") NE -1 THEN BEGIN
 sa = strsplit(s, '^"|" "|"$', /EXTRACT,/REGEX, /PRESERVE_NULL)
 n = n elements(sa)
IF n GT 2 THEN sa = sa[1:n-2]
 ENDIF ELSE BEGIN
 sa = strsplit(s,' ', /EXTRACT)
ENDELSE
return, sa
END
cheers
Martin
Martin Downing,
Clinical Research Physicist,
Grampian Orthopaedic RSA Research Centre,
Woodend Hospital, Aberdeen, AB15 6LS.
```

Subject: Re: splitting difficult(?) strings
Posted by hradilv.nospam on Wed, 09 Jan 2002 15:14:42 GMT
View Forum Message <> Reply to Message

/PRESERVE\_NULL ??? WOW!!! Thanks. Maybe I should read the online help

a little closer next time 8^) On Tue, 8 Jan 2002 22:49:24 -0000, "Martin Downing" <martin.downing@ntlworld.com> wrote: > "Craig Markwardt" <craigmnet@cow.physics.wisc.edu> wrote in message > news:onr8p0k4se.fsf@cow.physics.wisc.edu... >> >> hradilv.nospam@yahoo.com writes: >> >>> Any ideas on what would be the easiest (best?) way to split strings >>> that have the following characteristics? Perhaps best understood by >>> looking at examples? >>> >>> 1) >>> buf= '4 5 6' >>> strsplit(buf,' ',/extract) yields ['4','5','6'] {good} >>> Maybe I can summarize: I would like something that will split the >>> string by spaces, PROTECTED by "s. ie there may be spaces within the >>> "s which should not cause a split. >> >> Using STRSPLIT is like sending a boy to do a man's job. I think there >> is no other choice but to hand-code what you want. [Counter-examples >> welcomed. ] It \*may\* be possible to code it into a regular >> expression, but I'm not sure how to handle the "quotes-optional" >> aspect. >> > > Hi Vince, this works for your examples: > -----> FUNCTION vince\_sep, s > IF strpos(s, "") NE -1 THEN BEGIN sa = strsplit(s, '^"|" "|"\$', /EXTRACT,/REGEX, /PRESERVE\_NULL)  $n = n_elements(sa)$ > IF n GT 2 THEN sa = sa[1:n-2] ENDIF ELSE BEGIN sa = strsplit(s,' ', /EXTRACT) > ENDELSE

> return, sa

> > END

```
cheers
Martin
--
Martin Downing,
Clinical Research Physicist,
Grampian Orthopaedic RSA Research Centre,
Woodend Hospital, Aberdeen, AB15 6LS.
```

Subject: Re: splitting difficult(?) strings Posted by hradilv on Wed, 09 Jan 2002 17:46:03 GMT View Forum Message <> Reply to Message

I got something easier that makes some assumptions. Namely that all string delimiters are paired. It's based on the getstr() function that I found on the web (JHU?)

First I check to see if there is a " in the string. If not then I just use strsplit(), else I use getdstring() [get delimited string(s)]:

function getdstring, txt, delimiter=delimiter

```
if n_elements(delimiter) eq 0 then d='"' else d=delimiter
idx = where( byte(txt) eq (byte(d))[0], cnt )
if cnt eq 0 then return, "

lo = idx[2*lindgen(cnt/2)]
hi = idx[2*lindgen(cnt/2)+1]

return, strmid(txt,lo+1,hi-lo-1)
end
```

PS: I especially like the 'byte trick' and the fact that strmid() accepts and returns arrays. These are the simple things that really make my day 8^)

"Martin Downing" <martin.downing@ntlworld.com> wrote in message news:<nNK\_7.14521\$6q2.3132910@news2-win.server.ntlworld.com>... > "Craig Markwardt" <craigmnet@cow.physics.wisc.edu> wrote in message > news:onr8p0k4se.fsf@cow.physics.wisc.edu...

>>

```
>> hradilv.nospam@yahoo.com writes:
>>
>>> Any ideas on what would be the easiest (best?) way to split strings
>>> that have the following characteristics? Perhaps best understood by
>>> looking at examples?
>>>
>>> 1)
>>> buf= '4 5 6'
>>> strsplit(buf,' ',/extract) yields ['4','5','6'] {good}
>
>>> Maybe I can summarize: I would like something that will split the
>>> string by spaces, PROTECTED by "s. ie there may be spaces within the
>>> "s which should not cause a split.
>>
>> Using STRSPLIT is like sending a boy to do a man's job. I think there
>> is no other choice but to hand-code what you want. [Counter-examples
>> welcomed. 1 It *may* be possible to code it into a regular
>> expression, but I'm not sure how to handle the "quotes-optional"
>> aspect.
>>
> Hi Vince, this works for your examples:
>
 FUNCTION vince_sep, s
>
  IF strpos(s, '"') NE -1 THEN BEGIN
    sa = strsplit(s, '^"|" "|"$', /EXTRACT,/REGEX, /PRESERVE_NULL)
    n = n elements(sa)
>
  IF n GT 2 THEN sa = sa[1:n-2]
   ENDIF ELSE BEGIN
    sa = strsplit(s,' ', /EXTRACT)
  ENDELSE
  return, sa
>
> END
  -----
>
> cheers
> Martin
```