Subject: Re: converting bytarr into strarr
Posted by Benjamin Hornberger on Tue, 21 Jun 2005 21:59:44 GMT
View Forum Message <> Reply to Message

It's weird that the string function doesn't offer a keyword for this.
But how about first converting to integer?

b=string(fix(a))

Benjamin

Paul Van Delst wrote:
> Hello there,
>
> This is probably a simple question, but I haven't found out how to do it.
>
> How does one convert a byte array into a equivalent sized string array?
> E.g. consider the following:
>
> IDL> a=bindgen(20,40)
> IDL> help, a
> A BYTE = Array[20, 40]
> IDL> b=string(a)
> IDL> help, b
> B STRING = Array[40]
> IDL>
>
> or
>
> IDL> a=bindgen(2,4,5,6)
> IDL> help, a
> A BYTE = Array[2, 4, 5, 6]
> IDL> b=string(a)
> IDL> help, b
> B STRING = Array[4, 5, 6]
>
>
> What I want to do is convert "a" into a string array, "b", that has the
> same dimensions as the original byte array, [20, 40], or [2, 4, 5, 6]
> i.e. each element of "b" is a single character corresponding to the byte
> value in "a".
>
> Everything I've read in the IDL help tells me that a byte array will
> always "lose" the first dimension as the bytes are converted into
> concatenated strings.
>
> Is there a simple way around this? (I could loop over the dimensions of

> "a" I guess).
>
> thanks,
>
> paulv
>

Subject: Re: converting bytarr into strarr
Posted by Benjamin Hornberger on Tue, 21 Jun 2005 22:01:54 GMT
View Forum Message <> Reply to Message

Or adding another dimension:

b=string(reform(a,1,20,40))

Benjamin Hornberger wrote:
> It's weird that the string function doesn't offer a keyword for this.
> But how about first converting to integer?
>
> b=string(fix(a))
>
> Benjamin
>
>
> Paul Van Delst wrote:
>
>> Hello there,
>>
>> This is probably a simple question, but I haven't found out how to do it.
>>
>> How does one convert a byte array into a equivalent sized string
>> array? E.g. consider the following:
>>
>> IDL> a=bindgen(20,40)
>> IDL> help, a
>> A BYTE = Array[20, 40]
>> IDL> b=string(a)
>> IDL> help, b
>> B STRING = Array[40]
>> IDL>
>>
>> or
>>
>> IDL> a=bindgen(2,4,5,6)
>> IDL> help, a
>> A BYTE = Array[2, 4, 5, 6]

>> IDL> b=string(a)
>> IDL> help, b
>> B STRING = Array[4, 5, 6]
>>
>>
>> What I want to do is convert "a" into a string array, "b", that has
>> the same dimensions as the original byte array, [20, 40], or [2, 4, 5,
>> 6] i.e. each element of "b" is a single character corresponding to the
>> byte value in "a".
>>
>> Everything I've read in the IDL help tells me that a byte array will
>> always "lose" the first dimension as the bytes are converted into
>> concatenated strings.
>>
>> Is there a simple way around this? (I could loop over the dimensions
>> of "a" I guess).
>>
>> thanks,
>>
>> paulv
>>

Subject: Re: converting bytarr into strarr
Posted by Michael Wallace on Tue, 21 Jun 2005 22:03:55 GMT
View Forum Message <> Reply to Message

How about doing a two step conversion: byte -> integer -> string.

IDL> a = bindgen(20, 40)
IDL> b = string(a)
IDL> c = string(fix(a))
IDL> help, a, b, c
A BYTE = Array[20, 40]
B STRING = Array[40]
C STRING = Array[20, 40]

-Mike

Subject: Re: converting bytarr into strarr
Posted by Paul Van Delst[1] on Tue, 21 Jun 2005 23:11:40 GMT
View Forum Message <> Reply to Message

Michael Wallace wrote:
> How about doing a two step conversion: byte -> integer -> string.

>
> IDL> a = bindgen(20, 40)
> IDL> b = string(a)
> IDL> c = string(fix(a))
> IDL> help, a, b, c
> A   BYTE   = Array[20, 40]
> B   STRING   = Array[40]
> C   STRING   = Array[20, 40]
>
>
> -Mike

Hi,

Yeah, that was the first thing that occurred to me too, but when you use STRING on a non-byte input, all you get is the number as a string, rather than the ASCII byte code converted to a string:

```
IDL> a = bindgen(2, 4) + 65B
IDL> b = string(a)
IDL> c = string(fix(a))
IDL> help, a, b, c
A               BYTE      = Array[2, 4]
B               STRING    = Array[4]
C               STRING    = Array[2, 4]
IDL> print, b
AB CD EF GH
IDL> print, c
      65      66
      67      68
      69      70
      71      72
```

paulv

--
Paul van Delst
CIMSS @ NOAA/NCEP/EMC