Subject: Re: Dealing with list created by STRSPLIT?
Posted by Haje Korth on Fri, 20 May 2011 12:02:45 GMT
View Forum Message <> Reply to Message

Maybe i am missing something. STRSPLIT returns an array of string when using the /extract keyword. You simply access the elements by using [], splitnames[0]='diskname', etc. Haje

On May 19, 10:44 pm, Barry Lesht <ble...@gmail.com> wrote:
> I have an array of full-path filenames from which I need to extract
> some substrings. The array consists of a large number of strings of
> the form
>
> diskname/maindir/subdir1/subdir2/subdir3/filename
> diskname/maindir/subidr1/subdir2/subdir3/filename etc.
>
> The lengths of the subdirXs can be different. What I wanted to do was
> create some arrays the elements of which would be the subdirXs. I
> thought this would be simple now that STRSPLIT allows arrays as
> input. I had the array of n filenames in a string array and I used
>
> splitnames = STRSPLIT(filenames, '/', /EXTRACT).
>
> The result 'splitnames' is a list with n elements and each element is
> of the form
>
> diskname maindir subdir1 subdir2 subdir3 filename as it should be.
> Each element is a string array.
>
> What I can't seem to figure out is how to now extract one of these
> terms from the list element. That is, I'd like to end up with arrays
> that have the n subdir2s (for example). I've tried using the
> LIST::TOARRAY () feature but this fails for some mysterious reason
> with the message
>
> IDL> namearray=splitnames.ToArray()
> % LIST::TOARRAY: Unable to convert to type STRING: Element 798
> % Error occurred at: LIST::TOARRAY
> % $MAIN$
> % Execution halted at: $MAIN$
>
> Element 798 is no different from any of the elements in the list, by
> the way, but this is not the real problem (at least I don't think it
> is)
>
> There must be an easy way to do this; I'm not seeing it. Any help
> would be much appreciated.

Subject: Re: Dealing with list created by STRSPLIT?
Posted by BLesht on Fri, 20 May 2011 14:20:18 GMT

View Forum Message <> Reply to Message

```
On May 20, 7:02 am, Haje Korth <hajeko...@gmail.com> wrote:
>  Maybe i am missing something. STRSPLIT returns an array of string when
>  using the /extract keyword. You simply access the elements by using
>  [], splitnames[0]='diskname', etc. Haje
>
>  On May 19, 10:44 pm, Barry Lesht <ble...@gmail.com> wrote:
>
>>  I have an array of full-path filenames from which I need to extract
>>  some substrings.  The array consists of a large number of strings of
>>  the form
>
>>  diskname/maindir/subdir1/subdir2/subdir3/filename
>>  diskname/maindir/subidr1/subdir2/subdir3/filename  etc.
>
>>  The lengths of the subdirXs can be different.  What I wanted to do was
>>  create some arrays the elements of which would be the subdirXs.  I
>>  thought this would be simple now that STRSPLIT allows arrays as
>>  input.  I had the array of n filenames in a string array and I used
>
>>  splitnames = STRSPLIT(filenames, '/', /EXTRACT).
>
>>  The result 'splitnames' is a list with n elements and each element is
>>  of the form
>
>>  diskname maindir subdir1 subdir2 subdir3 filename as it should be.
>>  Each element is a string array.
>
>>  What I can't seem to figure out is how to now extract one of these
>>  terms from the list element.  That is, I'd like to end up with arrays
>>  that have the n subdir2s (for example).  I've tried using the
>>  LIST::TOARRAY () feature but this fails for some mysterious reason
>>  with the message
>
>>  IDL> namearray=splitnames.ToArray()
>>  % LIST::TOARRAY: Unable to convert to type STRING: Element 798
>>  % Error occurred at: LIST::TOARRAY
>>  %                    $MAIN$
>>  % Execution halted at: $MAIN$
>
>>  Element 798 is no different from any of the elements in the list, by
>>  the way, but this is not the real problem (at least I don't think it
>>  is)
>
>>  There must be an easy way to do this; I'm not seeing it.  Any help
>>  would be much appreciated.
```

>
>

Thanks, Haje. Maybe I'm being especially dense this morning, but I don't understand your suggestion. Indeed, the elements of list splitnames are string arrays, and, for each individual list element (e.g. splitnames[0]) I could use WHERE to determine the array element that matches my selection criterion (e.g. disk2s = WHERE(splitnames[0] EQ 'disk2'), but that doesn't solve my problem. Maybe my initial post wasn't clear. For example, if I start with an array of filenames

a1/b1/c1/d1/e1
a1/b1/c2/d1/e2
a1/b1/c2/d2/e3
a1/b1/c2/d3/e4 etc.

Using STRSPLIT results in a list

a1 b1 c1 d1 e1
a1 b1 c2 d1 e2
a1 b1 c2 d2 e3
a1 b1 c2 d3 e4 etc.

and what I want is something like like c2s = WHERE({third element of} splitnames[*] EQ 'c2') that will return an array of the list element numbers that include 'c2'. I think I need to play with WHERE and some of the other string processing routines.

Subject: Re: Dealing with list created by STRSPLIT?
Posted by David Fanning on Fri, 20 May 2011 14:45:14 GMT

View Forum Message <> Reply to Message

Barry Lesht writes:

> What I can't seem to figure out is how to now extract one of these
> terms from the list element. That is, I'd like to end up with arrays
> that have the n subdir2s (for example). I've tried using the
> LIST::TOARRAY () feature but this fails for some mysterious reason
> with the message
>
> IDL> namearray=splitnames.ToArray()
> % LIST::TOARRAY: Unable to convert to type STRING: Element 798
> % Error occurred at: LIST::TOARRAY
> % $MAIN$
> % Execution halted at: $MAIN$
>
> Element 798 is no different from any of the elements in the list, by

> the way, but this is not the real problem (at least I don't think it
> is)
>
> There must be an easy way to do this; I'm not seeing it. Any help
> would be much appreciated.

I think you are going to have to figure out what is up
with element 798. You might want to use the MISSING
keyword in the ToArray method.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Dealing with list created by STRSPLIT?
Posted by chris_torrence@NOSPAM on Fri, 20 May 2011 16:54:54 GMT

View Forum Message <> Reply to Message

On May 19, 8:44 pm, Barry Lesht <ble...@gmail.com> wrote:

```
>  IDL> namearray=splitnames.ToArray()
>  % LIST::TOARRAY: Unable to convert to type STRING: Element 798
>  % Error occurred at: LIST::TOARRAY
>  %                    $MAIN$
>  % Execution halted at: $MAIN$
```

>
> Element 798 is no different from any of the elements in the list, by
> the way, but this is not the real problem (at least I don't think it
> is)
>
> There must be an easy way to do this; I'm not seeing it. Any help
> would be much appreciated.

Hi Barry,

If you have IDL 8.1, you can now use multiple indices with List and
Hash. The first index gets the actual list (or hash) element, while
the remaining indices are used to index into the returned array. So in
your case, you could simply do:
print, splitnames[27,3]

This would take the 28th element of the list and extract the 4th
string. You can also use the usual array ranges, *, etc.

Regarding your particular problem, I am curious about what is wrong
with element 798. I'm worried there is a bug either in the strsplit
list code, or in the List::ToArray code, and I'd like to fix it. Can
you try the ::ToArray again, and when it halts with the error, print
out some info about element 798? For example:
HELP, splitnames[798]
PRINT, splitnames[798]

Thanks!
-Chris
ITTVIS

Subject: Re: Dealing with list created by STRSPLIT?
Posted by David Fanning on Fri, 20 May 2011 17:28:33 GMT
View Forum Message <> Reply to Message

Chris Torrence writes:

> If you have IDL 8.1, you can now use multiple indices with List and
> Hash. The first index gets the actual list (or hash) element, while
> the remaining indices are used to index into the returned array. So in
> your case, you could simply do:
> print, splitnames[27,3]
> This would take the 28th element of the list and extract the 4th
> string. You can also use the usual array ranges, *, etc.

Well, I have IDL 8.1 and this certainly doesn't work:

```
  IDL> print, splitnames[*,2]
  % Multiple list elements cannot be indexed.
```

This works:

```
  IDL> print, splitnames[3,2]
  c2
```

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Dealing with list created by STRSPLIT?
Posted by BLesht on Fri, 20 May 2011 20:45:40 GMT

View Forum Message <> Reply to Message

On May 20, 11:54 am, Chris Torrence <gorth...@gmail.com> wrote:
> On May 19, 8:44 pm, Barry Lesht <ble...@gmail.com> wrote:
>
>> IDL> namearray=splitnames.ToArray()
>> % LIST::TOARRAY: Unable to convert to type STRING: Element 798
>> % Error occurred at: LIST::TOARRAY
>> % $MAIN$
>> % Execution halted at: $MAIN$
>
>> Element 798 is no different from any of the elements in the list, by
>> the way, but this is not the real problem (at least I don't think it
>> is)
>
>> There must be an easy way to do this; I'm not seeing it. Any help
>> would be much appreciated.
>
> Hi Barry,
>
> If you have IDL 8.1, you can now use multiple indices with List and
> Hash. The first index gets the actual list (or hash) element, while
> the remaining indices are used to index into the returned array. So in
> your case, you could simply do:
> print, splitnames[27,3]
> This would take the 28th element of the list and extract the 4th
> string. You can also use the usual array ranges, *, etc.
>
> Regarding your particular problem, I am curious about what is wrong
> with element 798. I'm worried there is a bug either in the strsplit
> list code, or in the List::ToArray code, and I'd like to fix it. Can
> you try the ::ToArray again, and when it halts with the error, print
> out some info about element 798? For example:
> HELP, splitnames[798]
> PRINT, splitnames[798]
>
> Thanks!
> -Chris
> ITTVIS

Hi Chris - Thanks! I'm using 8.0.1 but will update shortly. Maybe
that will fix the problem. Here is the output corresponding to your
suggestion:

```
IDL> help, splitname[798]
<Expression>    STRING    = Array[7]
IDL> print, splitname[798]
Volumes 2TBDrive SeaWiFS L2standard62 Superior 2009
S2009238193838.L2.x.hdf
IDL>
```

Also, look at this:

```
IDL> print, splitname[796:798]
Volumes 2TBDrive SeaWiFS L2standard62 Superior 2008
S2008239191527.L2.x.hdf
Volumes 2TBDrive SeaWiFS L2standard62 Superior 2008
S2008239191527.L2.x.hdf
Volumes 2TBDrive SeaWiFS L2standard62 Superior 2009
S2009238193838.L2.x.hdf
IDL>
```

and this:

```
IDL> newnames = splitname[0:796]
IDL> tryarray = newnames.ToArray()
% LIST::TOARRAY: Unable to convert to type STRING: Element 791
% Error occurred at: LIST::TOARRAY
%                  $MAIN$
% Execution halted at: $MAIN$
```

Barry

Subject: Re: Dealing with list created by STRSPLIT?
Posted by BLesht on Fri, 20 May 2011 20:46:19 GMT

View Forum Message <> Reply to Message

On May 20, 12:28 pm, David Fanning <n...@idlcoyote.com> wrote:
> Chris Torrence writes:
>> If you have IDL 8.1, you can now use multiple indices with List and
>> Hash. The first index gets the actual list (or hash) element, while
>> the remaining indices are used to index into the returned array. So in
>> your case, you could simply do:
>> print, splitnames[27,3]
>> This would take the 28th element of the list and extract the 4th
>> string. You can also use the usual array ranges, *, etc.
>
> Well, I have IDL 8.1 and this certainly doesn't work:
>
>   IDL> print, splitnames[*,2]

```
>     % Multiple list elements cannot be indexed.
>
>  This works:
>
>     IDL> print, splitnames[3,2]
>     c2
>
>  Cheers,
>
>  David
>
>  --
>  David Fanning, Ph.D.
>  Fanning Software Consulting, Inc.
>  Coyote's Guide to IDL Programming:http://www.idlcoyote.com/
>  Sepore ma de ni thui. ("Perhaps thou speakest truth.")
```

Thanks, David. I'll see if Chris replies again and in the meantime
install 8.1.

Regards, Barry

Subject: Re: Dealing with list created by STRSPLIT?
Posted by chris_torrence@NOSPAM on Mon, 23 May 2011 20:43:09 GMT
View Forum Message <> Reply to Message

On May 20, 2:46 pm, Barry Lesht <ble...@gmail.com> wrote:

```
>  On May 20, 12:28 pm, David Fanning <n...@idlcoyote.com> wrote:
>
>
>
>
>
>
>
>
>
>>  Chris Torrence writes:
>>>  If you have IDL 8.1, you can now use multiple indices with List and
>>>  Hash. The first index gets the actual list (or hash) element, while
>>>  the remaining indices are used to index into the returned array. So in
>>>  your case, you could simply do:
>>>  print, splitnames[27,3]
>>>  This would take the 28th element of the list and extract the 4th
>>>  string. You can also use the usual array ranges, *, etc.
>
>>  Well, I have IDL 8.1 and this certainly doesn't work:
```

>
>> IDL> print, splitnames[*,2]
>> % Multiple list elements cannot be indexed.
>
>> This works:
>
>> IDL> print, splitnames[3,2]
>> c2
>
>> Cheers,
>
>> David
>
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming:http://www.idlcoyote.com/
>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>
> Thanks, David. I'll see if Chris replies again and in the meantime
> install 8.1.
>
> Regards, Barry

I should have said "you can also use the usual array ranges, *, etc."
for the second and subsequent indices. The first index MUST be a
scalar.
-Chris