Subject: Re: Case statement question
Posted by Paul Van Delst[1] on Tue, 22 Feb 2005 23:35:41 GMT
View Forum Message <> Reply to Message

Michael Wallace wrote:
> case x of
> 0: do_this()
> 1: do_that(x)
> 2: do_that(x)
> 3: do_that(x)
> 4: do_that(x)
> 5: do_that(x)
> 6: do_that(x)
> 7: do_that(x)
> 8: do_that(x)
> 9: do_that(x)
> 10: do_something_else()
> 11: do_something_more()
> endswitch
>
> Is there a way to group 1 - 9 into a single expression rather than
> having to list each distinct number in the range?

```
case 1 of
  (x eq 0):            do_this()
  (x ge 1 and x le 9 ): do_that(x)
  (x eq 10):           do_something_else()
  (x eq 11):           do_something_more()
  else: message, "Error!"
endcase
```

?? I think the syntax is correct, but I'm not sure.

paulv

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

Subject: Re: Case statement question
Posted by David Fanning on Tue, 22 Feb 2005 23:37:26 GMT
View Forum Message <> Reply to Message

Michael Wallace writes:

> case x of
> 0: do_this()

>   1: do_that(x)
>   2: do_that(x)
>   3: do_that(x)
>   4: do_that(x)
>   5: do_that(x)
>   6: do_that(x)
>   7: do_that(x)
>   8: do_that(x)
>   9: do_that(x)
>   10: do_something_else()
>   11: do_something_more()
> endswitch
>
> Is there a way to group 1 - 9 into a single expression rather than
> having to list each distinct number in the range?

I can't tell if this is too obvious or I'm being dense:

```
 case x of
     0: do_this()
    10: do_something_else()
    11: do_something_more()
  else: do_that(x)
 endcase
```

Cheers,

David

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

Subject: Re: Case statement question
Posted by Michael Wallace on Tue, 22 Feb 2005 23:44:12 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Michael Wallace writes:
>
>
>> case x of
>>   0: do_this()
>>   1: do_that(x)
>>   2: do_that(x)
>>   3: do_that(x)

>>   4: do_that(x)
>>   5: do_that(x)
>>   6: do_that(x)
>>   7: do_that(x)
>>   8: do_that(x)
>>   9: do_that(x)
>>   10: do_something_else()
>>   11: do_something_more()
>> endswitch
>>
>> Is there a way to group 1 - 9 into a single expression rather than
>> having to list each distinct number in the range?
>
>
> I can't tell if this is too obvious or I'm being dense:
>
>   case x of
>   0: do_this()
>   10: do_something_else()
>   11: do_something_more()
>   else: do_that(x)
>   endcase
>

It is too obvious! :-) I already have an else block statement in my
case which I didn't include in my example. My bad for leaving it out.
And if you notice, I made a second mistake by ending my case with an
endswitch. I'm getting loopy from all this IDL.

-Mike

Subject: Re: Case statement question
Posted by Michael Wallace on Tue, 22 Feb 2005 23:50:39 GMT
View Forum Message <> Reply to Message

>   case 1 of
>   (x eq 0):   do_this()
>   (x ge 1 and x le 9): do_that(x)
>   (x eq 10):   do_something_else()
>   (x eq 11):   do_something_more()
>   else: message, "Error!"
>   endcase

Sweet! That works. I would have never thought to switch the logic
around such that you're sending "true" into the case and looking for the
case that evaluates to "true." I guess every example I've seen has been
one where you send in the value you want to find, not the truth value

you want to find. Wow. Can you tell that I'm easily amused by
programming logic? Or maybe it's the caffeine talking. ;-)

-Mike

Subject: Re: Case statement question
Posted by Benjamin Hornberger on Tue, 22 Feb 2005 23:51:45 GMT
View Forum Message <> Reply to Message

Michael Wallace wrote:
> case x of
> 0: do_this()
> 1: do_that(x)
> 2: do_that(x)
> 3: do_that(x)
> 4: do_that(x)
> 5: do_that(x)
> 6: do_that(x)
> 7: do_that(x)
> 8: do_that(x)
> 9: do_that(x)
> 10: do_something_else()
> 11: do_something_more()
> endswitch
>
> Is there a way to group 1 - 9 into a single expression rather than
> having to list each distinct number in the range?
>
> -Mike

How about

switch x of
 0: do_this() & break
 1:
 2:
 ...
 9: do_that() & break
 10: do_something_else() & break
 11: do_something_more() & break
 else: whatever()
endswitch

Benjamin

Subject: Re: Case statement question
Posted by Mark Hadfield on Tue, 22 Feb 2005 23:57:35 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Michael Wallace writes:
>
>> case x of
>> 0: do_this()
>> 1: do_that(x)
>> 2: do_that(x)
>> 3: do_that(x)
>> 4: do_that(x)
>> 5: do_that(x)
>> 6: do_that(x)
>> 7: do_that(x)
>> 8: do_that(x)
>> 9: do_that(x)
>> 10: do_something_else()
>> 11: do_something_more()
>> endswitch
>>
>> Is there a way to group 1 - 9 into a single expression rather than
>> having to list each distinct number in the range?
>
> I can't tell if this is too obvious or I'm being dense:
>
> case x of
> 0: do_this()
> 10: do_something_else()
> 11: do_something_more()
> else: do_that(x)
> endcase

Another you might not have thought of:

```
case 1B of
   (x ge 0) and (x le 9): do_that(x)
   x eq 10: do_something_else()
   x eq 11: do_something_more()
   else: crash_and_burn
endcase
```

It's a bugger to indent, though.

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz

Subject: Re: Case statement question
Posted by Paul Van Delst[1] on Wed, 23 Feb 2005 16:31:35 GMT
View Forum Message <> Reply to Message

Benjamin Hornberger wrote:
> Michael Wallace wrote:
>
>> case x of
>> 0: do_this()
>> 1: do_that(x)
>> 2: do_that(x)
>> 3: do_that(x)
>> 4: do_that(x)
>> 5: do_that(x)
>> 6: do_that(x)
>> 7: do_that(x)
>> 8: do_that(x)
>> 9: do_that(x)
>> 10: do_something_else()
>> 11: do_something_more()
>> endswitch
>>
>> Is there a way to group 1 - 9 into a single expression rather than
>> having to list each distinct number in the range?
>>
>> -Mike
>
>
> How about
>
> switch x of
> 0: do_this() & break
> 1:
> 2:
> ...
> 9: do_that() & break
> 10: do_something_else() & break
> 11: do_something_more() & break
> else: whatever()
> endswitch

Oooo - I prefer this to the solution I posted. Looks cleaner. Of course, I prefer the Fortran solution above all:

SELECT CASE (x)

```
    CASE (0);     do_this()
    CASE (1:9);   do_that(x)
    CASE (10);    do_something_else()
    CASE (11);    do_something_more()
    CASE DEFAULT; whatever()
  END SELECT
```

And you can do stuff like
    CASE (:-1)
to specify any negative numbers or
    CASE (1:)
to specify any positive numbers. And multiple single/ranges too:
    CASE (1, 3, 7:10, 13, 15:20)

Very bloody handy.

Personally I think IDL should've used different syntax for the CASE/SWITCH constructs. I would prefer something more like the Fortran example above:

```
case x of
    0)   do_this()
    1:9) do_that(x)
   10)   do_something_else()
   11)   do_something_more()
   else) whatever()
endcase
```

so you could use ":" to specify ranges.

paulv

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

Subject: Re: Case statement question
Posted by Paul Van Delst[1] on Wed, 23 Feb 2005 16:32:43 GMT
View Forum Message <> Reply to Message

Michael Wallace wrote:

```
>>   case 1 of
>>     (x eq 0):            do_this()
>>     (x ge 1 and x le 9 ): do_that(x)
>>     (x eq 10):           do_something_else()
>>     (x eq 11):           do_something_more()
>>     else: message, "Error!"
>>   endcase
```

>
>
> Sweet! That works. I would have never thought to switch the logic
> around such that you're sending "true" into the case and looking for the
> case that evaluates to "true." I guess every example I've seen has been
> one where you send in the value you want to find, not the truth value
> you want to find. Wow. Can you tell that I'm easily amused by
> programming logic? Or maybe it's the caffeine talking. ;-)

I think it's the caffiene. :o)

Don't attribute any programming smarts to me - that example is straight out of the docs.

paulv

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

Subject: Re: Case statement question
Posted by R.G.Stockwell on Wed, 23 Feb 2005 17:19:23 GMT
View Forum Message <> Reply to Message

"Paul Van Delst" <paul.vandelst@noaa.gov> wrote in message
news:cvib57$7n3$1@news.nems.noaa.gov...
>> How about
>>
>> switch x of
>> 0: do_this() & break
>> 1:
>> 2:
>> ...
>> 9: do_that() & break
>> 10: do_something_else() & break
>> 11: do_something_more() & break else: whatever()
>> endswitch
>
> Oooo - I prefer this to the solution I posted. Looks cleaner.

I disagree with you Paul, I like your solution best.
In fact I was about to post a response, and then saw yours, and
sent my inelegant solution to the intergalactic bitbucket in the sky.
I don't think the above is really any different than the original question
(it is easy to paste do_that(x) 10 times.)

> Of course, I prefer the Fortran solution above all:

> SELECT CASE (x)
> CASE (0); do_this()
> CASE (1:9); do_that(x)

COOL!
How about something along the lines of:
commandstrings = strarray(12)
commandstrings[0] = 'do_this()'
commandstrings[1:9] = 'do_that(x)'
commandstrings[10] = 'do_something_else()'
commandstrings[11] = 'do_something_more()'
commandstrings[12] = 'do_elsedefault()'

r = execute(commandstring[x])

NOTE: the
commandstrings[12] = 'do_elsedefault()'
takes advantage of IDLs array overrun "feature" where any x
gt 11 will call that last element (thereby reproducing the effect of the
"else" statement in the case command).

Cheers,
bob

Subject: Re: Case statement question
Posted by Mark Hadfield on Wed, 23 Feb 2005 20:11:19 GMT
View Forum Message <> Reply to Message

R.G.Stockwell wrote:
>
> NOTE: the
> commandstrings[12] = 'do_elsedefault()'
> takes advantage of IDLs array overrun "feature" where any x
> gt 11 will call that last element (thereby reproducing the effect of the
> "else" statement in the case command).

You really like living on the edge, don't you Bob?

--
Mark Hadfield "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)
