Subject: Re: Bug in SWITCH - ELSE: statement?
Posted by larkn10 on Thu, 11 Oct 2007 15:39:15 GMT

View Forum Message <> Reply to Message

Hi Chris,

My understanding of SWITCH is that as soon as one option is true every subsequent option is executed. The ELSE is the default if there is no option that works, but it is always executed in a SWITCH statement.

You brought up something interesting (to me anyway.) If you place the ELSE earlier in the SWITCH statement, then everything below it is executed. If you do the same for a CASE statement then only then ELSE is executed. It never occurred to me until I saw your post that it was possible to place the ELSE anywhere but at the end.

Anyway, I don't know if this is helpful.

```
EXAMPLE:
--------

pro switchtest

  a = 5

  switch a of
  else:print,'switchelse'
  1:print,'no1'
  3:print,'no3'
  5:print,'yes5'
  6:print,'6'
  else:print,'switchelse'
  endswitch

  case a of
  else:print,'caseelse'
  1:print,'case1'
  5:print,'case5'
  endcase

end;switchtest
```

OUTPUT:

```
-------
switchelse
no1
no3
yes5
6
switchelse
caseelse
```

-Larry

Subject: Re: Bug in SWITCH - ELSE: statement?
Posted by Paul Van Delst[1] on Thu, 11 Oct 2007 16:27:44 GMT

View Forum Message <> Reply to Message

Jean H wrote:

>> You brought up something interesting (to me anyway.) If you place
>> the
>> ELSE earlier in the SWITCH statement, then everything below it is
>> executed. If you do the same for a CASE statement then only then
>> ELSE is executed. It never occurred to me until I saw your post
>> that it was possible to place the ELSE anywhere but at the end.
>
> This is useless and can bring serious headaches...
>
> Both the Case and Switch run from the 1st condition to the last one...
> so anything you write below the "else" will never (Case) or always
> (Switch) be executed... then there is no need for a Case or Switch
> anymore! ... just delete (Case) your code or write it outside of the
> Switch block!

Goodness.... I never even thought of that. Why on Earth should the order of the case selectors matter in a CASE construct? I just wrote:

```
pro test_case, number
  case number of
    else: print, 'I do not like this number!'
    1: print, '1 is a good number'
    2: print, '2 is even better'
    3: print, '3 is weird looking'
  endcase
end
```

```
IDL> .run test_case
% Compiled module: TEST_CASE.
```

```
IDL> test_case, 1
I do not like this number!
IDL> test_case, 3
I do not like this number!
IDL> test_case, 2
I do not like this number!
IDL> test_case,1000
I do not like this number!
```

That is just ridiculous behaviour.

The Fortran95 equivalent:

```
program test_case
   integer :: number
   write(*,'(/,"Enter a number:")',advance='no')
   read(*,'(i10)') number

   select case (number)
     case default
       write(*,*) 'I do not like this number!'
     case (1:3)
       write(*,*) number,' is a good number!'
   end select

end program test_case
```

doesn not depend on the order of the "CASE"s:

```
lnx:scratch : lf95 --f95 test_case.f90
Encountered 0 errors, 0 warnings in file test_case.f90.
lnx:scratch : a.out

Enter a number:3
3  is a good number!
lnx:scratch : a.out

Enter a number:1000
I do not like this number!
lnx:scratch : a.out

Enter a number:2
2  is a good number!
```

that is what I would expect to happen. Seems like the IDL case construct cribs code from the switch construct. Tsk tsk.

cheers,

paulv

Subject: Re: Bug in SWITCH - ELSE: statement?
Posted by Paul Van Delst[1] on Thu, 11 Oct 2007 16:42:18 GMT
View Forum Message <> Reply to Message

Paul van Delst wrote:

```
>  Jean H wrote:
>>>  You brought up something interesting (to me anyway.)  If you place
>>>  the
>>>  ELSE earlier in the SWITCH statement, then everything below it is
>>>  executed.  If you do the same for a CASE statement then only then
>>>  ELSE is executed.  It never occurred to me until I saw your post
>>>  that it was possible to place the ELSE anywhere but at the end.
>>
>>  This is useless and can bring serious headaches...
>>
>>  Both the Case and Switch run from the 1st condition to the last one...
>>  so anything you write below the "else" will never (Case) or always
>>  (Switch) be executed... then there is no need for a Case or Switch
>>  anymore! ... just delete (Case) your code or write it outside of the
>>  Switch block!
>
>  Goodness.... I never even thought of that. Why on Earth should the order
>  of the case selectors matter in a CASE construct? I just wrote:
>
>  pro test_case, number
>    case number of
>      else: print, 'I do not like this number!'
>      1: print, '1 is a good number'
>      2: print, '2 is even better'
>      3: print, '3 is weird looking'
>    endcase
>  end
>
>  IDL> .run test_case
>  % Compiled module: TEST_CASE.
>  IDL> test_case, 1
>  I do not like this number!
>  IDL> test_case, 3
>  I do not like this number!
>  IDL> test_case, 2
>  I do not like this number!
>  IDL> test_case,1000
>  I do not like this number!
>
>  That is just ridiculous behaviour.
```

Ruby forces you to put it last:

```
#!/usr/bin/env ruby
number=ARGV[0].to_i
case number
   else
     puts("I don't like #{number}")
   when 1..3
     puts("#{number} is a good number")
end
```

```
lnx:scratch : ruby test_case.rb 5
test_case.rb:4: syntax error, unexpected kELSE, expecting kWHEN
test_case.rb:6: syntax error, unexpected kWHEN, expecting $end
   when 1..3
       ^
```

In the right order it works as expected:

```
#!/usr/bin/env ruby
number=ARGV[0].to_i
case number
   when 1..3
     puts("#{number} is a good number")
   else
     puts("I don't like #{number}")
end
```

```
lnx:scratch : ruby test_case.rb 1
1 is a good number
lnx:scratch : ruby test_case.rb 5
I don't like 5
```

Ideally I would prefer the order to not matter, but if it does then I think IDL should
force the ELSE to be the last selector in CASE constructs (and probably SWITCH ones too)

cheers,

paulv

Subject: Re: Bug in SWITCH - ELSE: statement?
Posted by R.G.Stockwell on Thu, 11 Oct 2007 16:47:47 GMT
View Forum Message <> Reply to Message

"Paul van Delst" <Paul.vanDelst@noaa.gov> wrote in message
news:feliu2$p9t$1@news.nems.noaa.gov...
...

```
>  Goodness.... I never even thought of that. Why on Earth should the order
>  of the case selectors matter in a CASE construct? I just wrote:
>
>  pro test_case, number
>    case number of
>      else: print, 'I do not like this number!'
>      1: print, '1 is a good number'
>      2: print, '2 is even better'
>      3: print, '3 is weird looking'
>    endcase
>  end
```

That seems exactly like it should behave to me.
It executes the first match "else' and jumps out.

I figure the code above is something like

```
pro myjunk
 print,'hi'
 return
 print, 'never executed'
end
```

Cheers,
bob

Subject: Re: Bug in SWITCH - ELSE: statement?
Posted by JD Smith on Fri, 12 Oct 2007 19:43:52 GMT
View Forum Message <> Reply to Message

On Thu, 11 Oct 2007 09:44:27 +0000, Spon wrote:

> Hi everyone,
>
> can someone explain this to me please?
>
>> [quoted text muted]
> The ELSE clause of the SWITCH statement is optional. If included, it

> matches any selector expression, causing its code to be executed. For
> this reason, it is usually written as the last clause in the switch
> statement. **The ELSE statement is executed only if none of the
> preceding statement expressions match.** If an ELSE clause is not
> included and none of the values match the selector, program execution
> continues immediately below the SWITCH without executing any of the
> SWITCH statements.
>
> But, for example:
>
> ```
> PRO SWITCHTEST
>
> N = FIX (6 * RANDOMU (S, 1) ) + 1
>
> SWITCH N OF
>  1: PRINT, 'One'
>  2: PRINT, 'Two or less'
>  3: PRINT, 'Three or less'
>  4: PRINT, 'Four or less'
>  5: PRINT, 'Five or less'
>  6: PRINT, 'Six or less'
>  ELSE: PRINT, 'Are you using loaded dice?'
> ENDSWITCH
>
> PRINT, N
> END
> ```

You probably wanted CASE, which has an implicit BREAK after each statement. Or put the BREAKs in yourself:

```
1: begin
  print, 'One'
  break
end
```

SWITCH is really useful only when you want to "fall through" to multiple processing steps. If you do want to fall through, you must BREAK somewhere before ELSE:

```
SWITCH N OF
 1: PRINT, 'One'
 2: PRINT, 'Two or less'
 3: PRINT, 'Three or less'
 4: PRINT, 'Four or less'
 5: PRINT, 'Five or less'
 6: begin
    PRINT, 'Six or less'
    break
```

```
 end
 ELSE: PRINT, 'Are you using loaded dice?'
ENDSWITCH
```

Yes, it's ugly.

JD

Subject: Re: Bug in SWITCH - ELSE: statement?
Posted by Matt[2] on Fri, 12 Oct 2007 20:40:02 GMT
View Forum Message <> Reply to Message

JD,

I think it's documentation bug. Look closely at the original poster's emphasis.

>> **The ELSE statement is executed only if none of the preceding statement expressions match.**

That line of documentation is inconsistent with the way switch works in IDL.

Of course Jean H just pointed out.

"SWITCH executes the first matching statement and any following
statements in
the SWITCH block. Once a match is found in the SWITCH block, execution
falls
through to any remaining statements. For this reason, the BREAK
statement is
commonly used within SWITCH statements to force an immediate exit from
the
SWITCH block."

So the documentation is orthogonal as far as I can tell, and the first statement is the wrong one.

Matt

--
Matthew Savoie - Scientific Programmer
National Snow and Ice Data Center
   http://nsidc.org

Subject: Re: Bug in SWITCH - ELSE: statement?
Posted by R.G.Stockwell on Fri, 12 Oct 2007 20:56:49 GMT
View Forum Message <> Reply to Message

"Matt" <savoie@nsidc.org> wrote in message
news:1192221602.770405.43150@q5g2000prf.googlegroups.com...
>
> JD,
>
> I think it's documentation bug. Look closely at the original poster's
> emphasis.
>
>>> **The ELSE statement is executed only if none of the preceding statement
>>> expressions match.**
>
> That line of documentation is inconsistent with the way switch works
> in IDL.
>
> Of course Jean H just pointed out.
>
> "SWITCH executes the first matching statement and any following
> statements in
> the SWITCH block. Once a match is found in the SWITCH block, execution
> falls
> through to any remaining statements. For this reason, the BREAK
> statement is
> commonly used within SWITCH statements to force an immediate exit from
> the
> SWITCH block."
>
> So the documentation is orthogonal as far as I can tell, and the first
> statement is the wrong one.

Right, looks like someone copy and pasted that paragraph from the
CASE statement help file.
It is correctly described in the previous paragraph though.

For the SWITCH help, it really should state:

**The ELSE statement is executed only if none of the preceding statement
expressions match,
or it any of the preceding statement expressions match.**

:)

Cheers,
bob