
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
