
Subject: Re: IDL enhancements (was Re: idl2matlab translate-o-matic)
Posted by [Stein Vidar Hagfors H\[1\]](#) on Mon, 28 Feb 2000 08:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Martin Schultz <martin.schultz@dkrz.de> writes:

> ... while we are "working" at getting IDL to be more c-like, I'd give a
> lot to save
> my fingers from RSI (repetative stress injury for the initiated) --
> although I'd
> probably use the saved keywtrokes for newsgroup messages anyhow ;-)
>
> Here two or three things which I would favor very much:
>
> 1.) shorten this extra verbose if then endif else endelse structure:

Hear hear! Quite agree.

> 2.) allow assignment statements inside if's or while's:
>
> a = 100
> b = 5
> while (i=a-b gt 50) do a=a-b
>
> (This actually compiles but doesn't produce what you would like it
> to.)

I don't get this one.. What are you expecting it to do?

```
IDL> a=100 & b=5
IDL> while (i=a-b gt 50) do begin & a=a-b & print,a,b,i & end
  95    5  1
  90    5  1
  85    5  1
  80    5  1
  75    5  1
  70    5  1
  65    5  1
  60    5  1
  55    5  1
IDL> print,a,b,i
  55    5  0
```

Ok, so I equals (a-b gt 50) at all times... right? Maybe you wish to do this:

```
IDL> while ((i=a-b) gt 50) do begin & a=a-b & print,a,b,i & end
```

```
while ((i=a-b) gt 50) do begin & a=a-b & print,a,b,i & end
```

^

% Syntax error.

But then, with assignments and such, a few extra parentheses usually fixes it (!):

```
IDL> while (((i=a-b)) gt 50) do begin & a=a-b & print,a,b,i & end
```

95	5	95
90	5	90
85	5	85
80	5	80
75	5	75
70	5	70
65	5	65
60	5	60
55	5	55

.. which is OK given the order of the a=a-b and the print statement.

> OR:

> if (p=strpos(s='The quick brown fox ...','fox') ge 0) then begin

Again, parentheses:

```
IDL> if (p=strpos((s='The quick brown fox ...'),'fox') ge 0) then print,'HI'
```

> I guess one has to be careful with more complicated conditional

> clauses such as

> if ((i=a-b gt 50) OR (j=a+b lt 200)) then ...

> for then j could be undefined when you need it.

Not with IDL's "always calculate all parts of the conditional expressions" policy, I think? Maybe you mean

```
if (((i=a-b)) gt 50) OR (((i=i-5)) gt 50)
```

or something? Here, "i" might be undefined if IDL tries to evaluate the last part of the OR statement first..

But speaking of completeness, I'd like to do the following:

```
if (((i=a-b)) gt 50) OR ELSE (((i=c-5)) gt 50)
```

and have i=a-b if that is larger than 50, and i=c-5 otherwise..

Likewise:

if (((i=a-b)) gt 50) AND THEN (((j=c-5)) gt 50)

should mean: Set i=a-b, and if (and only if!) it's greater than 50,
also set j=c-5, and do the following statements if j is greater than 50.

Other languages have similar constructs...

Also, as far as this thread goes: I'd *much* rather see IDL return to a state of being (once again) an *interactive* image display tool rather than beefing up the "application development" things. A lot of people (scientists) started using IDL because you could TV an image, then play with the color table in umpteen ways, interspersing the color table changes with various ways of byte-scaling the image etc (including taking logarithms, etc, with various cutoffs etc..)

Nowadays, users have to rely on canned user-written procedures to control the color table "interactively". With 24-bit displays on PC's and some workstations (I'm told), the color table of a displayed, 8-bit image does *not* update when the color table is modified with e.g. tvlct, without *redisplaying* the image. This is a giant step backwards for IDL, and I've complained about it before.. (long time ago, long posting..)

> On the other side: Congratulations to RSI for introducing regular
> expression searching!

I wonder whether this was a result of the DLM's that I wrote once upon a time... :-)

--

Stein Vidar Hagfors Haugan
ESA SOHO SOC/European Space Agency Science Operations Coordinator for SOHO
NASA Goddard Space Flight Center, Email: shaugan@esa.nascom.nasa.gov
