Subject: Re: IDLWAVE 4.7/Tutorial
Posted by John-David T. Smith on Mon, 11 Dec 2000 15:56:55 GMT
View Forum Message <> Reply to Message

Jack Saba wrote:
>
> Thanks, JD.
>
> The section with the <M>, <B>, ..., clarifies the four variables used to control
> positioning very nicely. And I can get almost everything I want with them.
>
> However, it doesn't look like it is possible to use the continuation indentation
> scheme I prefer. I usually want the first nonblank character in the second line
> to line up with the beginning of the second word in the preceding line. Using
> your example, the code layout would be
>
```
>>  pro foo
>>     ThisPro, a
>>     for i=1,10 do begin
>>        print, 'yay' + $
>>               AnotherVar
>>     endfor
>>  end
```
>
> That's what would happen in text mode (for me, using NTEmacs in W98) if I hit 2
> tabs at the start of the "AnotherVar" line. If I can set up
> continuation-extra-indent to do this, it would probably be sufficient, but I
> would still prefer to be able to tell idlwave mode to use the same tab control
> that text mode does. Is there a way to do either of these?

I guess maybe I don't know what you mean by same tab control as Text
Mode. There, <TAB> runs the command indent-relative (which you can find
out with C-h k <TAB>). To get this exact command to run in
IDLWAVE-mode, you'd need to undo the Tab functionality it sets up, using
(in your idlwave-mode-hook):

```
(local-set-key "\t" 'indent-relative)
```

The problem is, if "smart" indenting is on ever, or if you run
indent-region (M-C-\), this line will be reindented from your chosen
position.

So it's not really a good solution, not to mention that it requires 4 or
more key presses to achieve when starting on the first line (<SPACE> $
<RET> <TAB> <TAB> ...), when it should really require only one (M-<RET>)
-- you have tried M-<RET>, yes?

However, there is a probable solution.

Note that IDLWAVE does use some intelligence when it comes to certain continuations. I suppose I should have noted that <C> only operates when some more compelling continuation indentation is not identified. Witness:

```
print, convol(a, $
              b, $
              c)
```

For your scheme, you'd get:

```
print, convol(a, $
       b, $
       c)
```

also, what about something like

```
a=myfunc(thisvar,thatvar,theothervar,THISKEYWORD=1, $
                                     THATKEYWORD=2)
```

Yuck. Here I'd bet you'd opt for lining it up by hand with 10 spaces. The flaw in your algorithm is demanding the second *word* of the first line line up. For this case, IDLWAVE already does what you want (I think):

```
a=myfunc(thisvar,thatvar,theothervar,THISKEYWORD=1, $
         THATKEYWORD=2)
```

It seems what you *really* want is for smart indenting to line up things with the first non-blank character after the comma in a procedure call, in the same way the "(" is treated in a function call, e.g.:

```
print, a, b, c, d, 1, 2, 3, $
       x,y,z
```

This looks like a reasonable option, in light of IDLWAVE's behavior with continued functions. The "normal" continuation offset is still needed in other cases, e.g.:

```
a=b+c+d+e+ $
   f+g+h
```

Perhaps even this could be a special case, always yielding:

```
a=b+c+d+e+ $
```

  f+g+h

Since functions and procedures are already being identified for other
reasons, I should think this is quite doable. This is exactly the kind
of thing that Carsten is eager to hear about, and also the sort of issue
that usually prompts a new alpha version addressing it the very same day
(unless it's really harebrained)!

It is difficult to relinquish the notion of lining up all your bits of
code yourself, but once you give in and let the program do what it was
designed for, your energies are freed for far more productive things.
As I said before, thinking about indentation is not quality use of time.

JD

--
 J.D. Smith | WORK: (607) 255-6263
 Cornell Dept. of Astronomy | (607) 255-5842
 304 Space Sciences Bldg. | FAX: (607) 255-5875
 Ithaca, NY 14853 |