

---

Subject: Re: IDLWAVE 4.7/Tutorial

Posted by [John-David T. Smith](#) on Fri, 08 Dec 2000 19:16:09 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Jack Saba wrote:

- > OK, Carsten, here's some feedback from a would-be user.
- >
- > I've tried IDLWave in the past, and always ended up disabling it because I was
- > unable to customize it to my satisfaction. It looks like a very useful tool.
- > This tutorial certainly helps, but I'm still unable to figure out how to
- > control some of the settings.
- >
- > Basically, I want complete control of text layout in the buffer.
- >
- > 1. I enter
- >
- > FOR i=0,n
- >   stuff
- > ENDFOR
- >
- > When I hit the <CR>, IDLWave reformats this by indenting the ENDFOR 3 spaces:
- >
- > FOR i=0,n
- >   stuff
- >    ENDFOR
- >
- > How do I prevent this reformatting?
- >
- > 2. How do I make tabs in IDLwave mode work the way they do in text mode?
- > Currently, the first tab moves the cursor under the first character on the last
- > line, but subsequent tabs fall into a black hole.

Think of <TAB> as more of a code cleaner-upper than a tab insertion mechanism. Almost all programming modes do it this way. Once a line of code is cleaned, cleaning it again will make no change. You *really* shouldn't be lining code up by hand, it's far too tedious.

If you would just like code to line up as in your first example, try:

```
idlwave-block-indent 3      ; Indentation settings
idlwave-end-offset -3
```

This will line the "ENDFOR" up with the "FOR".

Here's a little pictogram to help with the indents:

```
pro foo
<M>ThisPro, a
```

```
<M>for i=1,10 do begin
<M><B>print, 'yay' + $
<M><B><C>AnotherVar
<M><B><E>endfor
end
```

<M> = Main block indent  
<B> = Block indent  
<C> = Continuation extra indent  
<E> = End offset

in particular: <M>=3, <B>=3, <C>=1, <E>=-3 would produce:

```
pro foo
  ThisPro, a
  for i=1,10 do begin
    print, 'yay' + $
    AnotherVar
  endfor
end
```

see how <E> cancels <B> in the endfor line? Pretty simple. Your end offset is leaving things hanging.

<M>=0, <B>=5, <C>=3, <E>=-3 would make it:

```
pro foo
ThisPro, a
for i=1,10 do begin
  print, 'yay' + $
  AnotherVar
endfor
end
```

Oooh, ugly. But any degree of ugliness is tolerated.

Currently, <RET> indents the line after inserting the newline. If you would prefer "<RET>" not to indent the line for you, you can simply set:

```
(local-set-key "\r" 'newline)
```

in your idlwave-mode-hook. But I can't see why anyone would really want this... you'll just end up hitting <TAB> after <RET> everytime. But give it a try if you like.

If you don't want <TAB> to behave specially at all (by the way, you can always use C-<TAB> to get a "real" tab in any case), you can simply:

(local-set-key "\t" 'idlwave-hard-tab)

but this is getting really silly. The point of this indentation is to pick a style you like, and let IDLWAVE use it everywhere. It makes your code so much more maintainable. It would be nice if we could all agree on a code indentation scheme and commenting style, but *\*at least\** a personally enforced standard is necessary.

Also, if you ever find that you don't like what your customizations have done, disable them. Carsten has worked hard to make the default settings very useable.

JD

---