Subject: Custom formatting under version 7.0?
Posted by Pierre V. on Tue, 04 Dec 2007 21:45:14 GMT
View Forum Message <> Reply to Message

I have been using the new eclipse-based DE for the past few days and overall I like what I see. One thing I can't figure out yet is how to modify the auto-formatting rules to force my code to line up with parentheses, like this:

```
res = my_func(arg0, arg1, $
key1 = key1, $
key2 = key2)
```

I would like the lines that follow an open parenthesis to automatically indent to that same column. I don't know if the above example will show it since it probably depends on viewing this post with a fixed width font.

Anyhow, does anyone know if this is even possible with the new DE? Thanks!

Pierre

Subject: Re: Custom formatting under version 7.0? Posted by Pierre V. on Wed, 05 Dec 2007 18:06:27 GMT View Forum Message <> Reply to Message

```
On Dec 5, 9:09 am, Doug Edmundson <do...@ittvis.com> wrote:
> Pierre wrote:
>> On Dec 4, 2:40 pm, sboli...@gmail.com wrote:
>>> On Dec 4, 2:45 pm, Pierre <pierre.villene...@gmail.com> wrote:
>
>>>> I have been using the new eclipse-based DE for the past few days and
>>> overall I like what I see. One thing I can't figure out yet is how to
>>> modify the auto-formatting rules to force my code to line up with
>>> parentheses, like this:
>>> res = my_func(arg0, arg1, $
                   key1 = key1.$
>>>>
                   key2 = key2
>>>>
>>>> I would like the lines that follow an open parenthesis to
>>> automatically indent to that same column. I don't know if the above
>>> example will show it since it probably depends on viewing this post
>>>> with a fixed width font.
>>> Anyhow, does anyone know if this is even possible with the new DE?
>>>> Thanks!
>>>> Pierre
>>> Pierre.
```

```
>
      Unfortunately, the auto-formatter is not that sophisticated at this
>>>
>>> point in time.
      It will just identify an IDL line as a continuation of a previous
>>> line, and
>>> then indent by the amount dictated by the preference IDL->Editor-
>
>>> Displayed tab width
>>> -Scott
>
>> Thanks for the reply Scott. Is it at all possible to modify the
>> formatting rules from the user's end, or are they currently "hard
>> coded" into the system?
>
>> -Pierre
>
> Hi Pierre.
> Now might be a good time to bring up code templates. They don't resolve
  formatting issues, but you might find them useful.
>
> If you hit ctrl+space while focused in an editor, a context menu will
> pop up. Normally this is used for word completion, but if you haven't
> typed part of a word yet, the full list will start with several
> templates. The IDL Workbench has the following templates built-in:
>
    CASE, CATCH, FOR, FUNCTION, IF, PROCEDURE, REPEAT, SWITCH, and WHILE
>
> If you select one of those, code will be inserted for you. It's great
> if you can't remember the syntax for if-then-else. Note that some of
> them allow you to select text and then apply the template. For example,
if you have "a = 1", select it and apply "IF", that text will be put
> into an IF statement.
> Note that if you type some letters and then press ctrl+space, the pop-up
 list will trim the list of matches down.
> The on-line documentation goes into more of the details. One thing you
> can do is create your own templates. A good use of this might be to
> insert standard code, comments, copyrights, etc. For more templates, go
> to Preferences > IDL > Templates. The little "?" in the lower left will
> bring up the documentation.
>
> Doug
Hi Doug,
```

I had noticed the templates menu and I have started using them. I am

always forgetting exactly how to set up case statements, and this has made it easier for me.

-Pierre

Subject: Re: Custom formatting under version 7.0?
Posted by Jim Pendleton, ITT Vi on Thu, 06 Dec 2007 16:54:48 GMT
View Forum Message <> Reply to Message

```
"Pierre" <pierre.villeneuve@gmail.com> wrote in message
news:86914214-2396-4802-bbdb-3398335b3805@v4q2000hsf.googleg roups.com...
> On Dec 5, 9:09 am, Doug Edmundson <do...@ittvis.com> wrote:
>> Pierre wrote:
>>> On Dec 4, 2:40 pm, sboli...@gmail.com wrote:
>>> On Dec 4, 2:45 pm, Pierre <pierre.villene...@gmail.com> wrote:
>>
>>>> I have been using the new eclipse-based DE for the past few days and
>>>> overall I like what I see. One thing I can't figure out yet is how
>>>> to
>>>> modify the auto-formatting rules to force my code to line up with
>>>> parentheses, like this:
>>>> res = my func(arg0, arg1, $
                    key1 = key1, $
>>>> >
                    key2 = key2
>>>> >
>>>> I would like the lines that follow an open parenthesis to
>>>> automatically indent to that same column. I don't know if the above
>>>> example will show it since it probably depends on viewing this post
>>>> with a fixed width font.
>>>> Anyhow, does anyone know if this is even possible with the new DE?
>>>> > Thanks!
>>>> > Pierre
>>>> Pierre,
>>
       Unfortunately, the auto-formatter is not that sophisticated at this
>>>>
>>>> point in time.
>>>> It will just identify an IDL line as a continuation of a previous
>>>> line, and
       then indent by the amount dictated by the preference IDL->Editor-
>>>>
>>
>>>> Displayed tab width
>>>> -Scott
>>
>>> Thanks for the reply Scott. Is it at all possible to modify the
>>> formatting rules from the user's end, or are they currently "hard
>>> coded" into the system?
>>> -Pierre
```

```
>>
>> Hi Pierre,
>>
>> Now might be a good time to bring up code templates. They don't resolve
   formatting issues, but you might find them useful.
>>
>> If you hit ctrl+space while focused in an editor, a context menu will
>> pop up. Normally this is used for word completion, but if you haven't
>> typed part of a word yet, the full list will start with several
>> templates. The IDL Workbench has the following templates built-in:
>>
     CASE, CATCH, FOR, FUNCTION, IF, PROCEDURE, REPEAT, SWITCH, and WHILE
>>
>>
>> If you select one of those, code will be inserted for you. It's great
>> if you can't remember the syntax for if-then-else. Note that some of
>> them allow you to select text and then apply the template. For example,
>> if you have "a = 1", select it and apply "IF", that text will be put
>> into an IF statement.
>>
>> Note that if you type some letters and then press ctrl+space, the pop-up
   list will trim the list of matches down.
>>
>> The on-line documentation goes into more of the details. One thing you
>> can do is create your own templates. A good use of this might be to
>> insert standard code, comments, copyrights, etc. For more templates, go
>> to Preferences > IDL > Templates. The little "?" in the lower left will
>> bring up the documentation.
>> Doug
>
 Hi Doug,
>
>
> I had noticed the templates menu and I have started using them. I am
  always forgetting exactly how to set up case statements, and this has
> made it easier for me.
> -Pierre
To follow up on Doug's comment, creating templates for IDLdoc-style
comments also helps discourage laziness in documenting one's code.
Now if only I could attach keyboard accelerators to individual template
insertion commands...
```

Subject: Re: Custom formatting under version 7.0?

Jim P.

Posted by Michael Galloy on Thu, 06 Dec 2007 18:38:39 GMT

View Forum Message <> Reply to Message

On Dec 6, 9:54 am, "Jim Pendleton, ITT Visual Information Solutions" <jimp@no_spam.ittvis.com> wrote:

- > To follow up on Doug's comment, creating templates for IDLdoc-style
- > comments also helps discourage laziness in documenting one's code.
- > Now if only I could attach keyboard accelerators to individual template
- > insertion commands...

I've created some templates for writing classes/subclass (getProperty, setProperty, cleanup, init, and __define, plus comment headers). Very handy. My one issue is that it indents every line, so I have to select all and move left. I don't see a way to fix that (and ITT VIS's function and procedure templates do it too).

Mike

__

www.michaelgalloy.com Tech-X Corporation Software Developer II

Subject: Re: Custom formatting under version 7.0? Posted by sbolin42 on Thu, 06 Dec 2007 20:21:11 GMT View Forum Message <> Reply to Message

On Dec 6, 11:38 am, "mgal...@gmail.com" <mgal...@gmail.com> wrote:

- > On Dec 6, 9:54 am, "Jim Pendleton, ITT Visual Information Solutions"
- > <jimp@no_spam.ittvis.com> wrote:
- >> To follow up on Doug's comment, creating templates for IDLdoc-style
- >> comments also helps discourage laziness in documenting one's code.
- >> Now if only I could attach keyboard accelerators to individual template
- >> insertion commands...

>

>

- > I've created some templates for writing classes/subclass (getProperty,
- > setProperty, cleanup, init, and __define, plus comment headers). Very
- > handy. My one issue is that it indents every line, so I have to select
- > all and move left. I don't see a way to fix that (and ITT VIS's
- > function and procedure templates do it too).
- > > Mike
- > --www.michaelgallov.com
- > Tech-X Corporation
- > Software Developer II

Mike.

It should insert the template at the current cursor position. If you are in column 1 of the editor when you select CTRL+SPACE, and then pick your template, it should insert it beginning in column 1. This is not the behavior you are seeing?

Scott Bolin

Subject: Re: Custom formatting under version 7.0?
Posted by Michael Galloy on Thu, 06 Dec 2007 20:46:56 GMT
View Forum Message <> Reply to Message

```
On Dec 6, 1:21 pm, Scott Bolin <sboli...@gmail.com> wrote:
> On Dec 6, 11:38 am, "mgal...@gmail.com" <mgal...@gmail.com> wrote:
>
>> On Dec 6, 9:54 am, "Jim Pendleton, ITT Visual Information Solutions"
>> <jimp@no_spam.ittvis.com> wrote:
>>> To follow up on Doug's comment, creating templates for IDLdoc-style
>>> comments also helps discourage laziness in documenting one's code.
>>> Now if only I could attach keyboard accelerators to individual template
>>> insertion commands...
>> I've created some templates for writing classes/subclass (getProperty,
>> setProperty, cleanup, init, and ___define, plus comment headers). Very
>> handy. My one issue is that it indents every line, so I have to select
>> all and move left. I don't see a way to fix that (and ITT VIS's
>> function and procedure templates do it too).
>> Mike
>> --www.michaelgalloy.com
>> Tech-X Corporation
>> Software Developer II
>
> Mike,
>
   It should insert the template at the current cursor position. If
> you are in column 1 of the editor when you select CTRL+SPACE, and then
> pick your template, it should insert it beginning in column 1. This is
> not the behavior you are seeing?
OK. It's only one use case (which was the one I was using all the
```

time). Open a new file. Start on the first column of the first row.

Hit content assist. The list of choices will come up. *Type*

"subclass" (for my SUBCLASS template). Hit enter to select. Everything

is indented two spaces. If instead of typing "subclass", I down arrow to "SUBCLASS" and hit enter, it is fine. Actually, the "PROCEDURE" and "FUNCTION" templates seem to behave just fine, my problem there was what you said -- I was already indented two spaces when I tried them.

My class template is shown below.

```
Mike
www.michaelgalloy.com
Tech-X Corporation
Software Developer
; docformat = 'rst'
Get properties.
pro ${classname}::getProperty, _ref_extra=e
 compile_opt strictarr
 if (n elements(e) gt 0) then begin
  self->${subclassname}::getProperty, _strict_extra=e
 endif
end
; Set properties.
pro ${classname}::setProperty, _ref_extra=e
 compile_opt strictarr
 if (n_elements(e) gt 0) then begin
  self->${subclassname}::setProperty, _strict_extra=e
 endif
end
; Free resources.
pro ${classname}::cleanup
 compile_opt strictarr
 self->${subclassname}::cleanup
end
```

```
;+
; Create ${classname} object.
;
;:Returns: 1 for success, 0 for failure
;-
function ${classname}::init
compile_opt strictarr

if (~self->${subclassname}::init()) then return, 0

return, 1
end

;+
; Define instance variables.
;
;:Fields:
;-
pro ${classname}___define
compile_opt strictarr

define = { ${classname}, inherits ${subclassname}, ${cursor}}
end
```

Subject: Re: Custom formatting under version 7.0?
Posted by sbolin42 on Thu, 06 Dec 2007 22:53:41 GMT
View Forum Message <> Reply to Message

```
>>> setProperty, cleanup, init, and __define, plus comment headers). Very
>>> handy. My one issue is that it indents every line, so I have to select
>>> all and move left. I don't see a way to fix that (and ITT VIS's
>>> function and procedure templates do it too).
>
>>> Mike
>>> --www.michaelgalloy.com
>>> Tech-X Corporation
>>> Software Developer II
>
>> Mike,
>
    It should insert the template at the current cursor position. If
>> you are in column 1 of the editor when you select CTRL+SPACE, and then
>> pick your template, it should insert it beginning in column 1. This is
>> not the behavior you are seeing?
>
> OK. It's only one use case (which was the one I was using all the
> time). Open a new file. Start on the first column of the first row.
> Hit content assist. The list of choices will come up. *Type*
> "subclass" (for my SUBCLASS template). Hit enter to select. Everything
> is indented two spaces. If instead of typing "subclass", I down arrow
> to "SUBCLASS" and hit enter, it is fine. Actually, the "PROCEDURE" and
> "FUNCTION" templates seem to behave just fine, my problem there was
> what you said -- I was already indented two spaces when I tried them.
>
  My class template is shown below.
> Mike
> --www.michaelgalloy.com
> Tech-X Corporation
> Software Developer
>
  ; docformat = 'rst'
>
 ; Get properties.
> ;-
> pro ${classname}::getProperty, _ref_extra=e
   compile_opt strictarr
>
   if (n_elements(e) gt 0) then begin
>
     self->${subclassname}::getProperty, _strict_extra=e
   endif
>
  end
>
>
> ; Set properties.
```

```
> ;-
> pro ${classname}::setProperty, _ref_extra=e
  compile_opt strictarr
   if (n_elements(e) gt 0) then begin
>
    self->${subclassname}::setProperty, _strict_extra=e
   endif
> end
> ;+
> ; Free resources.
> pro ${classname}::cleanup
   compile_opt strictarr
   self->${subclassname}::cleanup
 end
>
> ; Create ${classname} object.
> ; :Returns: 1 for success, 0 for failure
> ;-
> function ${classname}::init
   compile_opt strictarr
   if (~self->${subclassname}::init()) then return, 0
>
>
  return, 1
> end
> ; Define instance variables.
> ;:Fields:
>
> ;-
> pro ${classname}__define
   compile_opt strictarr
>
  define = { ${classname}, inherits ${subclassname}, ${cursor}}
> end
Mike,
 I can duplicate the problem you are reporting. I'll log this bug.
Scott Bolin
ITTVis
```