



Emmad Kareem

19 août 2021, 00:22

Currently the settings I use with NotePad++ gives this view

```

50 IDENTIFICATION DIVISION.
51 PROGRAM-ID. HELLO-WORLD.
52 DATA DIVISION.
53 WORKING-STORAGE SECTION.
54 01 WS-MSG PIC X(100).
55 01 P1 PIC S9(4) COMP.
56 01 P2 PIC S9(4) COMP.
57 01 ERR-REC.
58     02 ERR-L          PIC X(10).
59     02 F              PIC X VALUE ":".
60     02 ERR-TXT1      PIC X(50).
61     02 SET1.
62         03 ERR-VAL1-N PIC +Z(22)VZ(7).
63         03 ERR-TXT2   REDEFINES ERR-VAL1-N PIC X(30).
64     02 SET2.
65         03 ERR-VAL2-N PIC +Z(22).Z(7).

```

But I would love to have this view

```

32 PROCEDURE DIVISION.
33 DISPLAY 'Hello, world' ERR-REC.
34
35 MOVE "AAA ?1 BBB ?2 CCC ?3" TO WS-MSG
36 MOVE 10 TO ENX9 (1)
37 MOVE "[TEN]" TO ENXT(2)
38 MOVE -15.33 TO ENX9(3)
39
40 MOVE ZERO TO KK.
41 UNSTRING WS-MSG INTO SUB-MSG(1)
42                      SUB-MSG(2)
43                      SUB-MSG(3)
44                      SUB-MSG(4)
45                      SUB-MSG(5)
46                      TALLYING KK
47                      DELIMITED "?"
48
49 UNSTRING LS-IN-TXT DELIMITED BY LS-IN-DELIMITER
50                      INTO LS-FIELD-NAME  COUNT A-CNT,
51                      LS-FIELD-VALUE  COUNT B-CNT,
52                      C  COUNT C-CNT
53                      TALLYING K
54

```



18 sur 18



I also need to add tab stops at columns 7,8, 12 and 72 (hoping the tab stop would show as a vertical column).

I don't know anything about customization, fonts, etc. Could some one give me a hint.

I particularly like the font above. Where to start? What to do?

 2 réponses Dernière réponse 19 août 2021, 21:00 ▼


^ 0 v ⋮



**Ekopalypse**

19 août 2021, 10:52

Npp provides something called themes.

See [here](#)  for more information about it. If something is unclear let us know.

^ 0 v ⋮



**Emmad Kareem**

19 août 2021, 11:29

Do you know of a theme for COBOL language? Thanks.

 2 réponses Dernière réponse 19 août 2021, 11:31 ▼

^ 0 v ⋮



**Ekopalypse**  @Emmad Kareem

19 août 2021, 11:31

@Emmad-Kareem

Here is a misunderstanding, a theme is not for one language but for all. Npp comes with a whole bunch of them, which ones have you tried?

^ 0 v ⋮

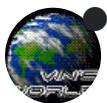


**Emmad Kareem**  @Ekopalypse

19 août 2021, 13:50

@Ekopalypse Thanks.

^ 0 v ⋮



**Michael Vincent**  @Emmad Kareem

19 août 2021, 15:13

@Emmad-Kareem said in Customizing for COBOL:

*Do you know of a theme for COBOL language? Thanks.*



18 sur 18



As @Ekopalypse states, there are many themes for "all" languages, not a specific theme for a specific language. That said, many of the themes in the Notepad++/themes directory have not been updated for quite some time and thus do not reflect the latest changes / additions of languages in Notepad++. For example, I use a custom theme based off the Oblivion theme and in checking this question, realized my theme did not have the Cobol language - so no styling.

Pick the theme you want. If Cobol files do not have styling, the language probably isn't included in your theme.

1. Close Notepad++
2. Open the theme file `Notepad++/themes/YOUR_THEME_FILE.xml` with Windows Notepad and open the `Notepad++/stylers.xml`.
3. Copy the relevant section from `Notepad++/stylers.xml` into `Notepad++/themes/YOUR_THEME_FILE.xml` and save.
4. Reopen Notepad++ and open your Cobol file. It should be styled, probably not to your liking.
5. Use the Settings => Style Configurator to adjust the colors for your language (Cobol).

To get you started, here is the relevant Cobol section from my theme:

```
<LexerType name="cobol" desc="COBOL" ext="">
  <WordsStyle name="PREPROCESSOR" styleID="9" fgColor="A082BD" bgColor="
  <WordsStyle name="DEFAULT" styleID="11" fgColor="E0E2E4" bgColor="2931
  <WordsStyle name="DECLARATION" styleID="5" fgColor="A082BD" bgColor="2
  <WordsStyle name="INSTRUCTION WORD" styleID="16" fgColor="93C763" bgCo
  <WordsStyle name="KEYWORD" styleID="8" fgColor="93C763" bgColor="29313
  <WordsStyle name="NUMBER" styleID="4" fgColor="FFCD22" bgColor="293134
  <WordsStyle name="STRING" styleID="6" fgColor="EC7600" bgColor="293134
  <WordsStyle name="CHARACTER" styleID="7" fgColor="FF8409" bgColor="293
  <WordsStyle name="OPERATOR" styleID="10" fgColor="E8E2B7" bgColor="293
  <WordsStyle name="COMMENT" styleID="1" fgColor="66747B" bgColor="29313
  <WordsStyle name="COMMENT LINE" styleID="2" fgColor="66747B" bgColor="
  <WordsStyle name="COMMENT DOC" styleID="3" fgColor="66747B" bgColor="2
  <WordsStyle name="COMMENT LINE DOC" styleID="15" fgColor="66747B" bgCo
  <WordsStyle name="COMMENT DOC KEYWORD" styleID="17" fgColor="66747B" b
  <WordsStyle name="COMMENT DOC KEYWORD ERROR" styleID="18" fgColor="FF4
</LexerType>
```

Cheers.

1 réponse Dernière réponse 21 sept. 2022, 00:09



Alan Kilborn

19 août 2021, 15:42

@Emmad-Kareem said in Customizing for COBOL:

*I also need to add tab stops at columns 7,8, 12 and 72*

Thus far this part of the OP's question has been missed by the responders. It has come up before, but it sure seems like it would have.



18 sur 18



## Can Notepad++/Scintilla handle variable-width tab stop positions?

 2 réponses Dernière réponse 19 août 2021, 16:33 ▼

^ 1 v ⋮



PeterJones ↩ @Alan Kilborn

📅 19 août 2021, 16:33

@Alan-Kilborn said in Customizing for COBOL:

*Can Notepad++/Scintilla handle variable-width tab stop positions?*

see discussion <https://community.notepad-plus-plus.org/topic/18429/need-to-set-tab-stops-not-spacing> from a couple years ago. @Ekopalypse was able to get the `SCI_CLEARABSTOPS/SCI_ADDTABSTOP` messages via the PythonScript interface `editor.clearTabStops()/editor.addTabStop()` to set up a custom set of tab locations. Unfortunately, it's on a per-line basis, and has to be re-applied frequently (every time the buffer is updated). Also, that sets it in pixels, not in characters, so you need to do some sort of conversion factor to convert character location to pixels.

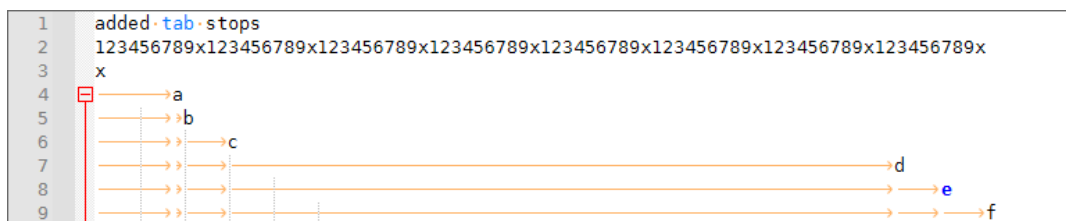
My guess is that it would need to be something like the following. Please note this is just a proof of concept and should be treated as code that would need debugging and extra work to make it 100%.

```
# encoding=utf-8
"""in response to https://community.notepad-plus-plus.org/topic/21689/"""
from Npp import *

cols = (7, 8, 12, 72)
pixperchar = 8

def set_tab_via_callback():
    for linenum in range(editor.getLineCount()):
        editor.clearTabStops(linenum)
        for tabcol in cols:
            editor.addTabStop(linenum, tabcol*pixperchar)

set_tab_via_callback()
```



Instead of manually running the `set_tab_via_callback()`, you would have to add callbacks for that function; Eko's post linked earlier hints at which callbacks are needed; you'd also need to calculate or hardcode the right `pixperchar` to convert characters to pixels correctly for your font; you'd probably also want a check in the callback to make sure it only runs the tab-changer if the active file is COBOL.

Maybe Eko still has the example code that was played with (but not published)

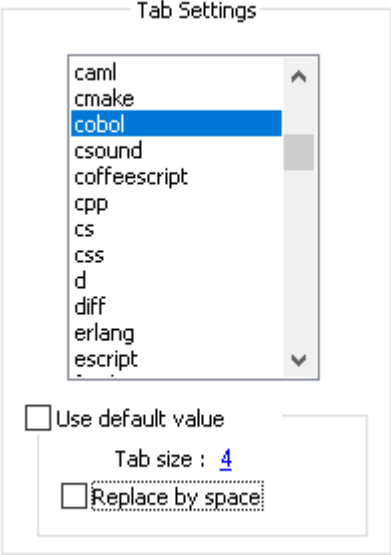


18 sur 18



----

**PS (edit):** I forgot to mention that I had to go to Settings > Preferences > Language > Tab Settings, and set `cobol` to *not* replace by space:



^ 4 v ⋮



**Ekopalypse**

19 août 2021, 17:49

Here [↗](#) is what I posted at github regarding the language specific tab width.

^ 2 v ⋮



**PeterJones** [↩](#) @Ekopalypse

19 août 2021, 18:03

@Ekopalypse said in Customizing for COBOL:

*Here [↗](#) is what I posted at github regarding the language specific tab width.*

That looks more like multiEdge than tabs, unless I've misunderstood. But the same basic concept should work (replacing the vertical edge clearing with a per-line loop that clears the tabs, and replacing the multiEdge adding with loops like add the tabs in a per-line loop.

^ 3 v ⋮

  2 réponses Dernière réponse 19 août 2021, 18:07 [v](#)



**Ekopalypse** [↩](#) @PeterJones

19 août 2021, 18:07

@PeterJones

you are right - my bad.



**PeterJones** ↩ @PeterJones

19 août 2021, 18:38

This merger of Eko's script and mine comes close... I merged my proof-of-concept with Eko's fancy wrapper-registering to give different updates on a per-language basis, and then checked against the suggested flow from 2 years ago

But I couldn't find what message(s) were meant by the [suggested on\\_update\\_ui](#). I think it might have to be one or more Scintilla Notifications for that, so it would use a separate `editor.addCallback()` in addition to the `notepad.addCallback()`

So with this version, it will update the tabs whenever you change into a COBOL window or switch the active file's language to COBOL from something else.

```
# encoding=utf-8
"""in response to https://community.notepad-plus-plus.org/topic/21689/

Merge my suggestion with Eko's pointer to https://github.com/notepad-plus-plus/not
"""
from Npp import editor, notepad, NOTIFICATION, EDGEVISUALSTYLE

TAB_CONFIG = {}

pixperchar = 8

def register_lang(func):
    def wrapped():
        for linenum in range(editor.getLineCount()):
            editor.clearTabStops(linenum)
            func(linenum)
    TAB_CONFIG[func.__name__.replace('on_lang_', '')] = wrapped
```



**PeterJones** ↩ @PeterJones

19 août 2021, 19:40

updated with a scintilla MODIFIED notification, with the caveat that it will only trigger for non-zero `linesAdded` value – that is non zero if a line is added or deleted.

it's somewhat slow, but as you switch in to the file, or add or delete lines, the tabs will all be updated.

```
# encoding=utf-8
"""in response to https://community.notepad-plus-plus.org/topic/21689/

Merge my suggestion with Eko's pointer to https://github.com/notepad-plus-plus/not
"""
from Npp import editor, notepad, NOTIFICATION, SCINTILLANOTIFICATION, EDGEVISUALST

TAB_CONFIG = {}
```



18 sur 18



```
pixperchar = 8

def register_lang(func):
    def wrapped():
        for linenum in range(editor.getLineCount()):
            editor.clearTabStops(linenum)
            func(linenum)
```

^ 1 v ⋮



**Michael Vincent** ↩ @Alan Kilborn

19 août 2021, 20:27

@Alan-Kilborn said in Customizing for COBOL:

*I also need to add tab stops at columns 7,8, 12 and 72*

*Thus far this part of the OP's question has been missed by the responders. I can't recall if this one has come up before, but it sure seems like it would have.*

*Can Notepad++/Scintilla handle variable-width tab stop positions?*

Does @dail 's [Elastic TabStops](#) plugin solve this? I honestly don't know and haven't tried it (never needed this), but sounds like it might?

Cheers.

^ 0 v ⋮



**Peter Jones** ↩ @Michael Vincent

19 août 2021, 20:52

@Michael-Vincent ,

I don't use that either, but my impression was no. But I grabbed it to test briefly.

```
7      → indent . and
8      [ tabstops → align . code
9      [
10     elastic → tabstops
11     longer . word → after
12     [ hello → there
13     [
14     what . about . a . → new . alignment
15     → here
16     [ there
17     [
18     what . about → in . → between → asdfas
19     x → hmph → asdfsad
20     whatever → there →
```

the tabstops in Elastic Tabstops will stretch to fit the longest piece of text: for example, with **what about**, **x**, and **whatever**, the first tabstop will be wide enough for the **what about**. And it allows



18 sur 18



different stops when you give a blank line between paragraphs/sections... but I don't see a way to pre-define specific widths.

^ 1 v ⋮



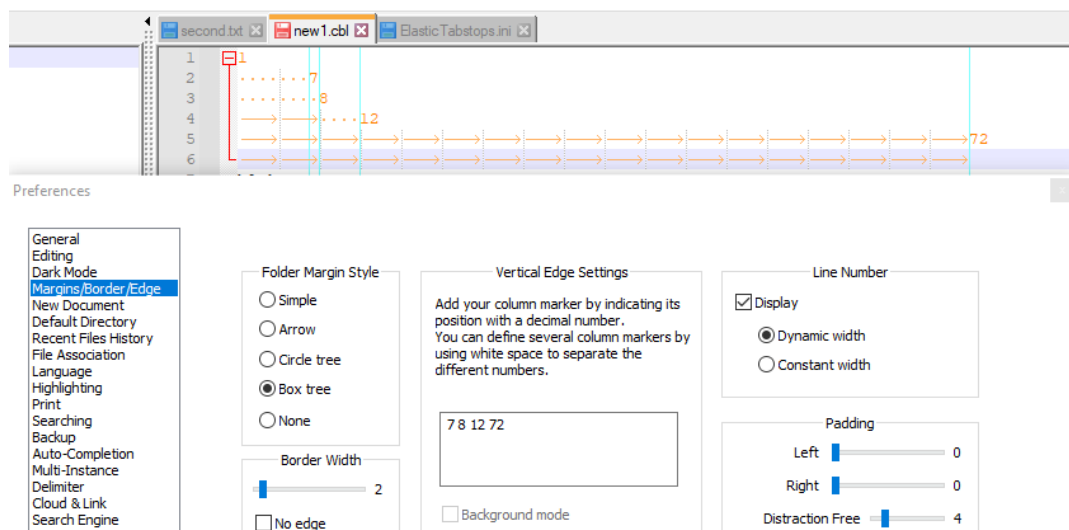
PeterJones @Emmad Kareem

19 août 2021, 21:00

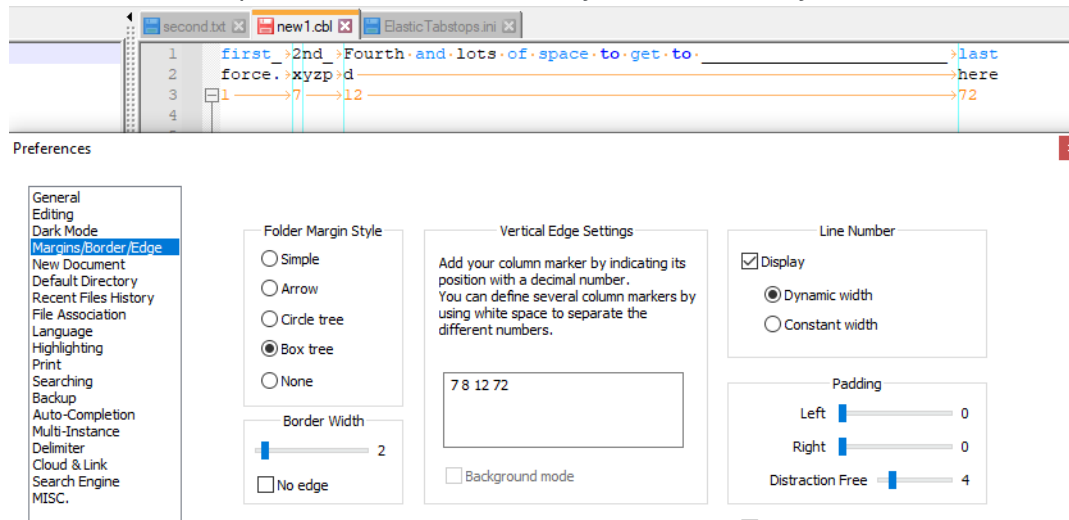
@Emmad-Kareem said in Customizing for COBOL:

*I also need to add tab stops at columns 7,8, 12 and 72 (hoping the tab stop would show as a vertical column).*

I don't think anyone directly addressed this, either. With a recent Notepad++ version, you can have multiple vertical edges drawn: go to **Settings > Preferences > Margins / Border / Edge**, and fill in the **Vertical Edge Settings**:



That example doesn't have the pythonscript callbacks active, so the tab size is a fixed width. Here it is with Elastic Tabstops also active, with a dummy header line to try to force known tab stops:



The drawback to the multiple vertical edges is that it is the same for all languages. Though that multiEdge example that @Ekopalypse linked to will actually solve that problem for you.



18 sur 18





Speaking of the tab size/location, one thing you don't mention: is COBOL okay with using TAB character between fields? Or do you need/want tab-to-spaces conversion (so when you hit TAB, it will actually use enough space characters to fill in the gap)? If you're okay with keeping TAB as the TAB character, then either the PythonScript or an ElasticTabstops-with-header might work for you; if you need that spacing to be encoded as actual space characters (ASCII 32 = 0x20) in the file, then neither is going to supply what you need.

^ 1 v ⋮

environ 8 mois après



**Eduardo Martines Teixeira** ↩ @Emmad Kareem

27 avr. 2022, 23:44

@emmad-kareem I made a copy of the 'Deep Black' theme and renamed it to Cobol and incremented the following commands saved and put my cobol file in the themes folder, after the procedure I selected it in notepad ++ see if you like it.

Regarding the tab, I'm seeing how to solve it

```
<LexerType name="cobol" desc="COBOL" ext="">
  <WordsStyle name="PREPROCESSOR" styleID="9" fgColor="DCDCCC" bgColor="3
  <WordsStyle name="DEFAULT" styleID="11" fgColor="DCDCCC" bgColor="3F3F3
```



Community

## Customizing for COBOL

? Help wanted · · · · ·

18 messages

6 publieurs

4.4k vues



Se connecter pour répondre

```
<WordsStyle name="COMMENT" styleID="1" fgColor="7F9F7F" bgColor="3F3F3F
<WordsStyle name="COMMENT LINE" styleID="2" fgColor="7F9F7F" bgColor="3
<WordsStyle name="COMMENT DOC" styleID="3" fgColor="7F9F7F" bgColor="3F
<WordsStyle name="COMMENT LINE DOC" styleID="15" fgColor="7F9F7F" bgCol
<WordsStyle name="COMMENT DOC KEYWORD" styleID="17" fgColor="7F9F7F" bg
<WordsStyle name="COMMENT DOC KEYWORD ERROR" styleID="18" fgColor="7F9F
</LexerType>
```

code\_text

^ 0 v ⋮

environ 5 mois après

**Emmad Kareem** ↩ @Michael Vincent



18 sur 18





@Michael-Vincent, Many thanks for your help.



^ 0 v ⋮

*The Community of users of the Notepad++ text editor.*

Powered by [NodeBB](#) |  [Contributors](#)