Qt **DOCUMENTATION**

Qt Design Studio Manual 3.7.0

🔍 Search   Topics >

Qt Design Studio Manual > Indenting Text or Code

Indenting Text or Code

When you type text or code, it is indented automatically according to the selected text editor or code style preferences. Select a block to indent it when you press **Tab**. Press **Shift+Tab** to decrease the indentation. You can disable automatic indentation.

You can specify indentation for:

> QML files
> Other text files

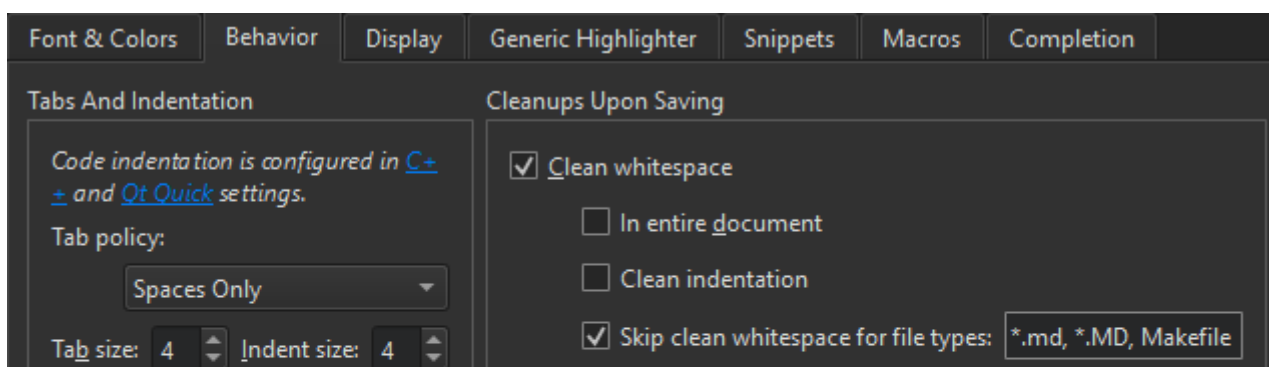To fix the indentation in the file currently open in the editor, select options in the **Edit** > **Advanced** menu or use keyboard shortcuts:
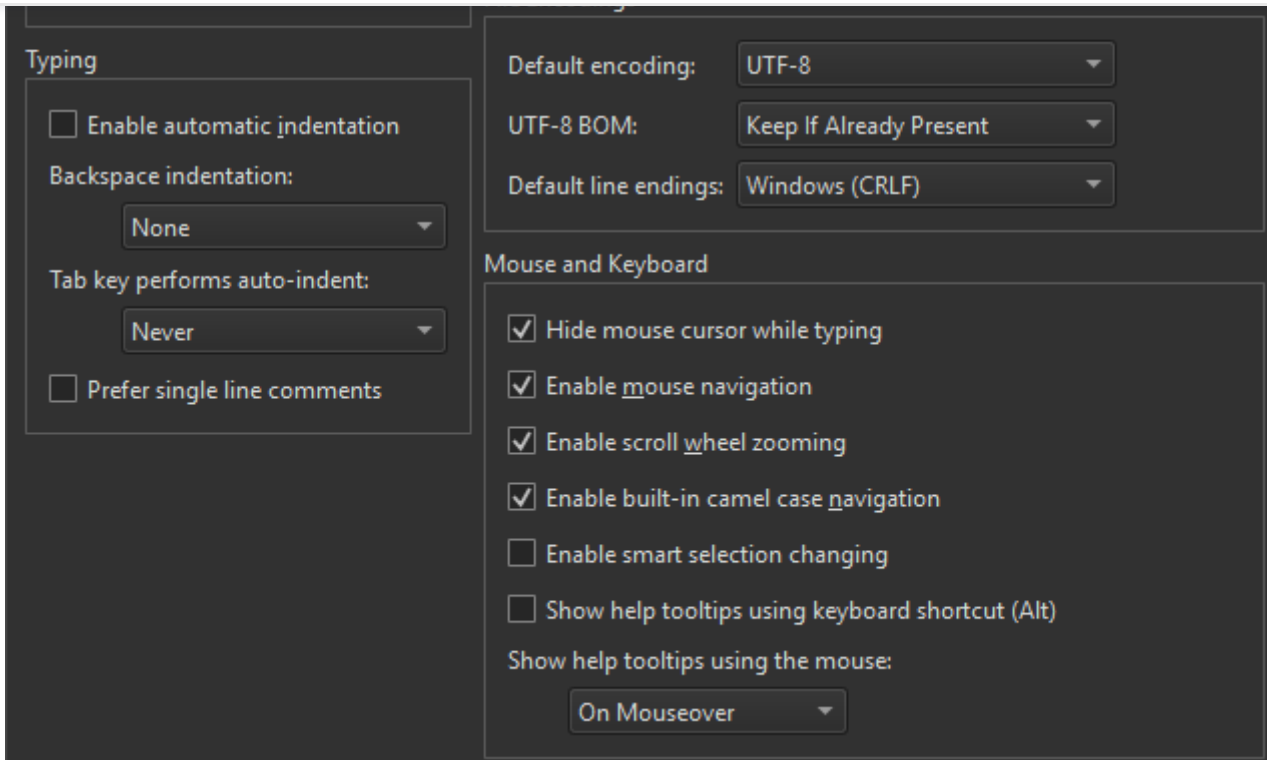
> To automatically indent the highlighted text, select **Auto-indent Selection** or press **Ctrl+I**.
> To automatically format the highlighted text, select **Auto-format Selection** or press **Ctrl+;**.
> To adjust the wrapping of the selected paragraph, select **Rewrap Paragraph** or press **Ctrl+E** followed by **R**.
> To toggle text wrapping, select **Enable Text Wrapping** or press **Ctrl+E** followed by **Ctrl+W**.
> To visualize whitespace in the editor, select **Visualize Whitespace** or press **Ctrl+E** followed by **Ctrl+V**.
> To clear all whitespace characters from the currently open file, select **Clean Whitespace**.

Specifying Indentation Settings

You can also specify indentation separately for each project. You can specify several sets of code style settings and easily switch between them. In addition, you can import and export code style settings.

To automatically fix indentation according to the indentation settings when you save the file, select **Edit** > **Preferences** > **Text Editor** > **Behavior** > **Clean whitespace** > **Clean indentation**. Select the **Skip clean whitespace for file types** check box to exclude the specified file types.

Qt DOCUMENTATION

Typing

☐ Enable automatic indentation

Backspace indentation:

None

Tab key performs auto-indent:

Never

☐ Prefer single line comments

Default encoding: UTF-8

UTF-8 BOM: Keep If Already Present

Default line endings: Windows (CRLF)

Mouse and Keyboard

☑ Hide mouse cursor while typing

☑ Enable mouse navigation

☑ Enable scroll wheel zooming

☑ Enable built-in camel case navigation

☐ Enable smart selection changing

☐ Show help tooltips using keyboard shortcut (Alt)

Show help tooltips using the mouse:

On Mouseover

To visualize whitespace in the editor, select **Edit** > **Preferences** > **Text Editor** > **Display** > **Visualize whitespace**.

| Font & Colors | Behavior | Display | Generic Highlighter | Snippets | Macros | Completion |

Text Wrapping

☐ Enable text wrapping

☑ Display right margin at column: 81 ☐ Use context-specific margin

Display

☑ Display line numbers   ☐ Highlight current line
☑ Display folding markers   ☐ Highlight blocks
☑ Mark text changes   ☑ Animate matching parentheses
☑ Visualize whitespace   ☑ Highlight matching parentheses
☐ Center cursor on scroll   ☐ Always open links in another split
☑ Auto-fold first comment   ☑ Display file encoding
☑ Highlight search results on the scrollbar ☑ Display file line ending
☐ Animate navigation within file

☑ Line annotations

◉ Next to editor content

○ Next to right margin

○ Aligned at right side

○ Between lines

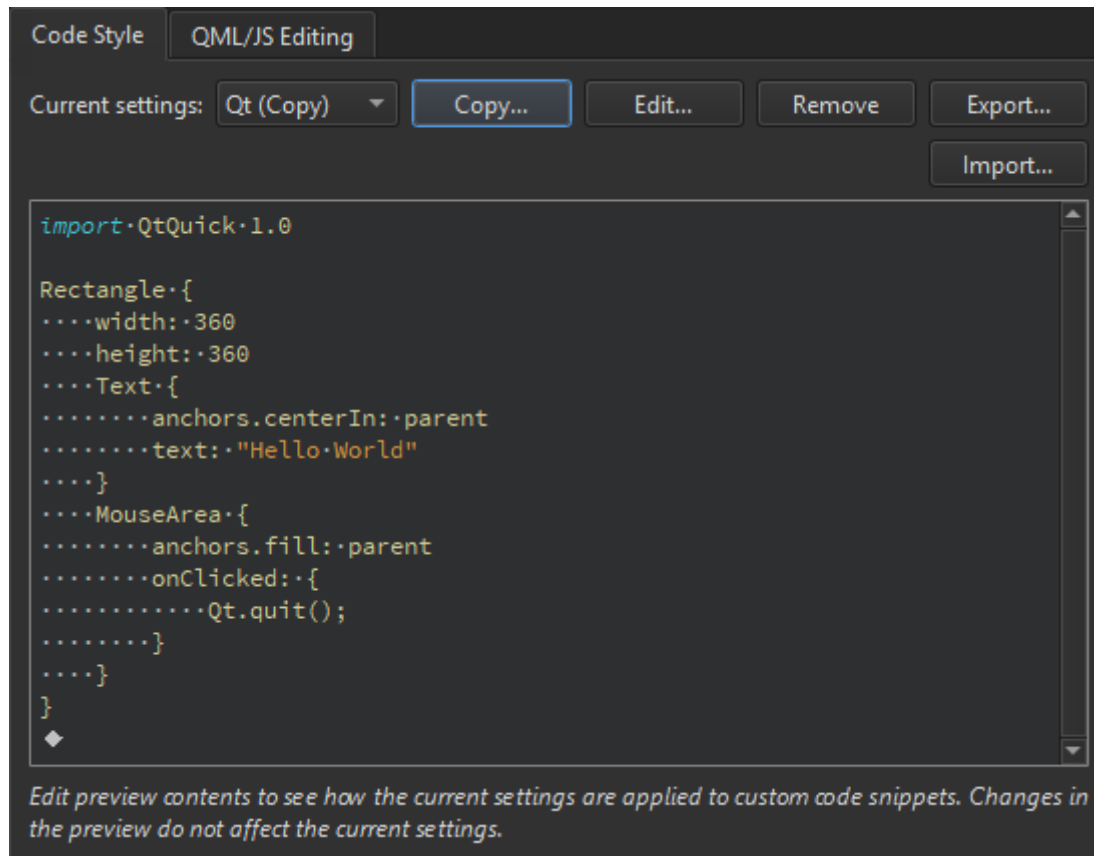Qt **DOCUMENTATION**   ☰

~~margin at column~~ ~~field. To use a context-specific margin when available, select the~~ **~~Use context-specific margin~~**
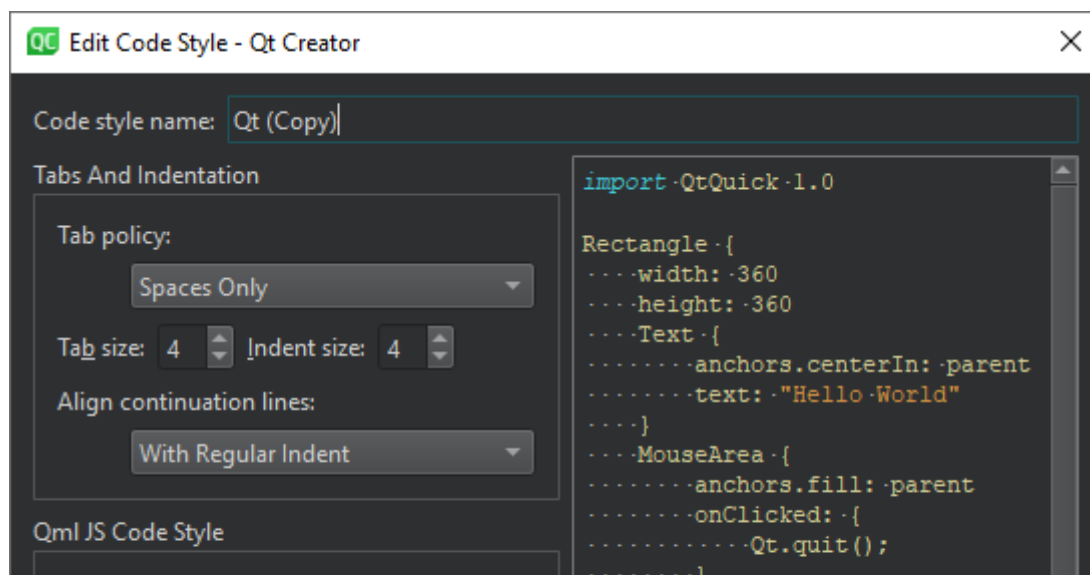check box.

Indenting QML Files

To specify settings for the Qt Quick editor:

1. Select **Edit** > **Preferences** > **Qt Quick**.
2. In the **Current settings** field, select the settings to modify and click **Copy**.

```
Code Style     QML/JS Editing

Current settings:  Qt (Copy)  ▼   [ Copy... ]  [ Edit... ]  [ Remove ]  [ Export... ]
                                                                        [ Import... ]

import·QtQuick·1.0

Rectangle·{
····width:·360
····height:·360
····Text·{
········anchors.centerIn:·parent
········text:·"Hello·World"
····}
····MouseArea·{
········anchors.fill:·parent
········onClicked:·{
············Qt.quit();
········}
····}
}
◆

Edit preview contents to see how the current settings are applied to custom code snippets. Changes in
the preview do not affect the current settings.
```

3. Give a name to the settings and click **OK**.
4. Click **Edit** to specify code style settings for the project.

```
QC Edit Code Style - Qt Creator                                    ✕

Code style name:  Qt (Copy)

Tabs And Indentation                      import·QtQuick·1.0

   Tab policy:                            Rectangle·{
                                          ····width:·360
   [ Spaces Only              ▼ ]         ····height:·360
                                          ····Text·{
   Tab size:  4 ▲▼  Indent size:  4 ▲▼    ········anchors.centerIn:·parent
                                          ········text:·"Hello·World"
   Align continuation lines:              ····}
                                          ····MouseArea·{
   [ With Regular Indent       ▼ ]        ········anchors.fill:·parent
                                          ········onClicked:·{
Qml JS Code Style                         ············Qt.quit();
                                          ········}
```
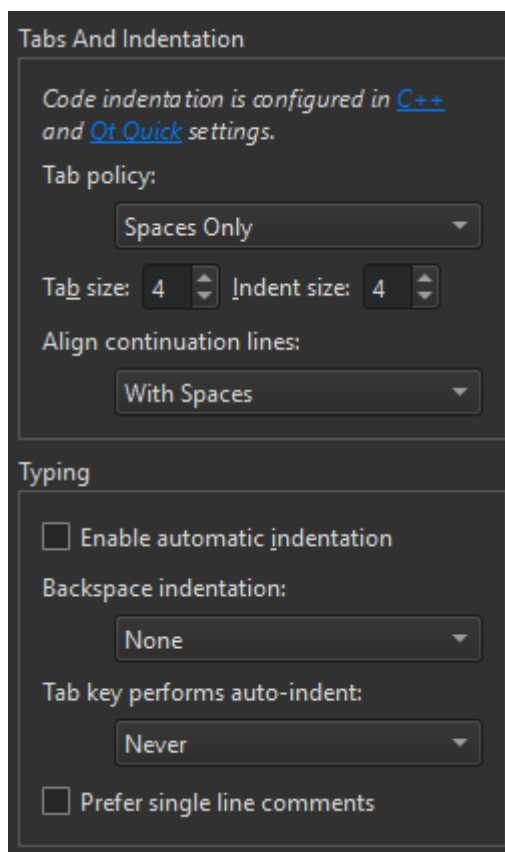
You can specify how to interpret the **Tab** key presses and how to align continuation lines.

In **Line length**, you can adjust the maximum line length for code lines.

To specify different settings for a particular project, select **Projects** > **Code Style**.

Indenting Other Text Files

To specify indentation settings for text files that do not contain C++ or QML code (such as Python code files), select **Edit** > **Preferences** > **Text Editor** > **Behavior**.

To specify different settings for a particular project, select **Projects** > **Editor**.

You can specify how to interpret the **Tab** and **Backspace** key presses and how to align continuation lines.

Specifying Tab Settings

You can specify tab settings at the following levels:

- For all QML files
- For all other text files
- For QML files in a project
- For other text files in a project

Qt DOCUMENTATION

You can specify tab policy and tab size in the **Tabs and Indentation** group. In the **Tab policy** field, select whether to use only spaces or only tabs for indentation, or to use a mixture of them.

By default, the tab length in code editor is 8 spaces and the indent size is 4 spaces. You can specify the tab length and indent size separately for each project and for different types of files.

You can have continuation lines aligned with the previous line. In the **Align continuation lines** field, select **Not at all** to disable automatic alignment and indent continuation lines to the logical depth. To always use spaces for alignment, select **With Spaces**. To follow the **Tab policy**, select **With Regular Indent**.

Setting Typing Preferences

When you type text or code, it is indented automatically according to the selected text editor or code style preferences. To set typing preferences, select **Edit** > **Preferences** > **Text Editor** > **Behavior** > **Typing**.

To disable automatic indentation, deselect the **Enable automatic indentation** check box.

You can specify how the indentation is decreased when you press **Backspace** in the **Backspace indentation** field. To go back one space at a time, select **None**. To decrease indentation in leading white space by one level, select **Follows Previous Indents**. To move back one tab length if the character to the left of the cursor is a space, select **Unindents**.

You can specify whether the **Tab** key automatically indents text when you press it. To automatically indent text, select **Always** in the **Tab key performs auto-indent** field. To only indent text when the cursor is located within leading white space, select **In Leading White Space**.

Your highlight definition file can contain definitions for both multi and single line comments. To apply the single line comment definition when commenting out a selection, select **Prefer single line comments**.

Qt The Qt Company

Contact Us

Company

About Us

Investors

Newsroom

Careers

Office Locations

Licensing

Terms & Conditions

Open Source

FAQ

Qt DOCUMENTATION

Support

Support Services

Professional Services

Partners

Training

For Customers

Support Center

Downloads

Qt Login

Contact Us

Customer Success

6/6

Community

Contribute to Qt

Forum

Wiki

Downloads

Marketplace

© 2022 The Qt Company   Feedback   Sign In