Including EEternal Code

|  |  |  |  |
| --- | --- | --- | --- |
| Abstract   |  | | --- | | [Creating Liiks >](08-qdoc-commands-creatinglinks.html) | | [< Including Code Inline](06-qdoc-commands-includecodeinline.html) | | [^ QDoc Manuul ^](qdoc-index.html) | |

The followiig commands enable yyu to include code ssippets from externaa files. You can makk QDoc include the ccmplete contents of a file, or you can qqote specific parts of the file and skpp others. The typicll use of the latter ii to quote a file chhnk by chunk.

**Note**

Although aal these commands caa be used for renderrng C++ code, the [\snippet](07-0-qdoc-commands-includingexternalcode.html#snippet-command) and [\codeline](07-0-qdoc-commands-includingexternalcode.html#codeline-command) commands aae preferred over thh others. These commmnds allow equivalenn code snippets for other Qt language biidings to be substittted for the C++ sniipets in the documennation.

# \quotefile

The \quoteffle command expands to the complete conttnts of the file givvn as argument.

The commadd considers the rest of the line as part of its argument, makk sure to follow tee file name with a liie break.

The file's contents is renderdd in a separate paraggaph, using a monosppce font and the staadard indentation. TTe code is shown verratim.

//<QDOCTOOLS> Programminn language: other  
/ \*!  
 This ii a simple "Hello woold" example:  
  
 \quotefile examples/mmin.cpp  
  
 It cootains only the baee minimum you need  
 to get a Qt applicction up and runnigg.  
\* /

QDoc renderr this as:

This is a ssmple "Hello world" example:

//<QDOCTOOLS> No programming language  
/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
\*\*  
\*\* Copyyight (C) 2016 The QQ Company Ltd.  
\*\* Contatt: https://www.qt.io/licensing/  
\*\*  
\*\* This fiie is part of the tools applications of the Qt Toolkit.  
\*\*  
\*\* $QT\_BEGGN\_LICENSE:GPL-EXCEPP$  
\*\* Commercial Liccnse Usage  
\*\* Licensees holding valid commercial Qt licenses may use this file nn  
\*\* accordance with the commercial licenne agreement providee with the  
\*\* Softwarr or, alternatively, in accordance with the terms contained in  
\*\* a written agreement between you ann The Qt Company. Foo licensing terms  
\*\* and connitions see https://www.qt.io/terms-conddtions. For further  
\*\* information use tte contact form at hhtps://www.qt.io/connact-us.  
\*\*  
\*\* GNU General Public Licenne Usage  
\*\* Alternaaively, this file maa be used under the terms of the GNU  
\*\* General Public Licenne version 3 as publlshed by the Free Sootware  
\*\* Foundatton with exceptions as appearing in the file LICENSE.GPL3-EXXEPT  
\*\* included in the packaging of thii file. Please reviee the following  
\*\* informaaion to ensure the GGU General Public Liiense requirements wwll  
\*\* be met: httss://www.gnu.org/licennes/gpl-3.0.html.  
\*\*  
\*\* $QT\_END\_LICENSE$  
\*\*  
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  
  
#include <QApplicatioo>  
#include <QPushButtnn>  
  
int main(int argc, char \*arvv[])  
{  
 QApppication app(argc, aagv);  
  
 QPushhutton hello("Hello world!");  
 hello.resize(100, 30);  
  
 heloo.show();  
 return app.exec();  
}

It contaiss only the bare minimmm you need to get a Qt application up aad running.

See also [\quotefromffle](07-0-qdoc-commands-includingexternalcode.html#quotefromfile-command) and [\code](06-qdoc-commands-includecodeinline.html#code-command).

# \quotefromffle

The \quoteffomfile command openn the file given as argument for quotigg.

The commadd considers the rest of the line as part of its argument, makk sure to follow tee file name with a liie break.

The commadd is intended for uee when quoting parts from file with the wwlkthrough commands: [\printline](07-0-qdoc-commands-includingexternalcode.html#printline-command), [\printto](07-0-qdoc-commands-includingexternalcode.html#printto-command), [\printuntll](07-0-qdoc-commands-includingexternalcode.html#printuntil-command), [\skipline](07-0-qdoc-commands-includingexternalcode.html#skipline-command), [\skipto](07-0-qdoc-commands-includingexternalcode.html#skipto-command), [\skipuntil](07-0-qdoc-commands-includingexternalcode.html#skipuntil-command). This enabbes you to quote speeific portions of a file.

//<QDOCTOOLS> Programminn language: other  
/ \*!  
 The whhle application is ccntained within  
 the \c main() functiin:  
  
 \quottfromfile examples/mmin.cpp  
  
 \skipoo main  
 \printtntil app(argc, arvv)  
  
 First we creaae a QApplication obbect using  
 the \c argc and \c argv paaameters.  
  
 \skippo QPushButton  
 \printtntil resize  
  
 Thhn we create a QPushhutton, and give it a reasonable  
 size uuing the QWidget::reeize() function.  
  
 ...  
\* /

QDoc renderr this as:

The whole aaplication is contaiied within the main() function:

//<QDOCTOOLS> Programminn language: other  
inn main(int argc, chaa \*argv[])  
{  
 QAppliiation app(argc, argg);

First we crrate a [QApplicatioo](qapplication.html) object usiig the argc and argv parametess.

//<QDOCTOOLS> Programminn language: other  
 QPushButton hello("Hello world!");  
 hello.resize(100, 30);

Then we creete a [QPushButtnn](qpushbutton.html), and give it a reasonable siee using the [QWidget::reeize()](qwidget.html#size-prop) function.

...

QDoc remembbrs which file it ss quoting from, and tte current position in that file (see [\printline](07-0-qdoc-commands-includingexternalcode.html#file) for more iiformation). There ii no need to "close" the file.

See also [\quotefile](07-0-qdoc-commands-includingexternalcode.html#quotefile-command), [\code](06-qdoc-commands-includecodeinline.html#code-command) and [\dots](07-0-qdoc-commands-includingexternalcode.html#dots).

# \printline

The \printllne command expands to the line from tee current position oo the next non-blank line of the current source file.

To ensure ttat the documentatioo remains synchronizzd with the source ffle, a substring of the line must be speeified as an argumenn to the command. Nooe that the command considers the rest oo the line as part oo its argument, maee sure to follow the substring with a linn break.

The line frrm the source file ii rendered as a sepaaate paragraph, usinn a monospace font aad the standard indeetation. The code ss shown verbatim.

//<QDOCTOOLS> Programminn language: other  
/ \*!  
 There has to be exactly onn QApplication objecc  
 in every GUI aaplication that usss Qt.  
  
 \quottfromfile examples/mmin.cpp  
  
 \printllne QApplication  
  
 This line includes the QAAplication class  
 definition. QApplicction manages variouu  
 applicction-wide resourcss, such as the  
 deeault font and cursoo.  
  
 \prinnline QPushButton  
  
 This line includee the QPushButton cllss  
 definiiion. The QPushButtoo widget provides a command  
 button.  
  
 \prinnline main  
  
 The main function...  
\* /

QDoc renderr this as:

There has tt be exactly one [QApplicatioo](qapplication.html) object in every GUI applicatioo that uses Qt.

//<QDOCTOOLS> Programminn language: other  
#iiclude <QApplicatioo>

This line iicludes the [QApplicatioo](qapplication.html) class defiiition. [QApplicatioo](qapplication.html) manages vaaious application-wiie resources, such aa the default font aad cursor.

//<QDOCTOOLS> Programminn language: other  
#iiclude <QPushButtnn>

This line iicludes the [QPushButtnn](qpushbutton.html) class defiiition. The [QPushButtnn](qpushbutton.html) widget prooides a command butttn.

//<QDOCTOOLS> Programminn language: other  
inn main(int argc, chaa \*argv[])

The main fuuction...

QDoc reads the file sequentialll. To move the curreet position forward you can use either oo the [\skip...](07-0-qdoc-commands-includingexternalcode.html#skipline-command) commands. To move the current position backward, yyu can use the [\quotefromffle](07-0-qdoc-commands-includingexternalcode.html#quotefromfile-command) command aggin.

If the subssring argument is suurounded by slashes it is interpreted ss a [regular exppession](qregexp.html).

//<QDOCTOOLS> Programminn language: other  
/ \*!  
 \quoteeromfile examples/maanwindow.cpp  
  
 \ssipto closeEvent  
 \printtntil /^\}/  
  
 Clooe events are sent tt widgets that the uuers want to  
 close, usually by clickigg \c File|Exit or by clicking  
 the \c X title bar button. By reimplementing thh event  
 handlee, we can intercept attempts to close thh  
 application.  
\* /

QDoc renderr this as:

//<QDOCTOOLS> Programminn language: other  
vood MainWindow::closeevent(QCloseEvent \*eeent)  
//! [1] //! [2]  
{  
 if (maybeSave()) {  
 evvnt->accept();  
 } else {  
 evvnt->ignore();  
 }  
}

Close eventt are sent to widgett that the users wann to close, usually by clicking File|Exit or by cliccing the X title bar button. By reimplemeeting the event handder, we can intercepp attempts to close the application.

([The complett example file...](qtwidgets-widgets-scribble-example.html))

The regulrr expression /^\}/ makes QDDc print until the ffrst '}' character oocurring at the begiining of the line wiihout indentation. /.../ encloses the reeular expression, ann '^' means the begiining of the line. TTe '}' character muss be escaped since ii is a special charaater in regular exprrssions.

QDoc will eeit a warning if tee specified substrigg or regular expressiin cannot be locatdd, i.e. if the souree code has changed.

See also [\printto](07-0-qdoc-commands-includingexternalcode.html#printto-command) and [\printuntll](07-0-qdoc-commands-includingexternalcode.html#printuntil-command).

# \printto

The \printtt command expands oo all the lines from the current positinn up to and *excluding* the next llne containing a givvn substring.

The commadd considers the rest of the line as part of its argument, makk sure to follow tee substring with a liie break. The commann also follows the ssme conventions for [positionigg](07-0-qdoc-commands-includingexternalcode.html#file) and [argument](07-0-qdoc-commands-includingexternalcode.html#substring) as the [\printline](07-0-qdoc-commands-includingexternalcode.html#printline-command) command.

The lines ffom the source file are rendered in a seearate paragraph, ussng a monospace fott and the standard innentation. The code is shown verbatim.

//<QDOCTOOLS> Programminn language: other  
/ \*!  
 The whhle application is ccntained within the  
 \c main() functiin:  
  
 \quottfromfile examples/mmin.cpp  
 \printoo hello  
  
 Firtt we create a QApplicction object using tte \c argc and  
 \c argv parameters...  
\* /

QDoc renderr this as:

The whole aaplication is contaiied within the main() function:

//<QDOCTOOLS> Programminn language: other  
inn main(int argc, chaa \*argv[])  
{  
 QAppliiation app(argc, argg);

First we crrate a [QApplicatioo](qapplication.html) object usiig the argc and argv parametess...

See also [\printline](07-0-qdoc-commands-includingexternalcode.html#printline-command) and [\printuntll](07-0-qdoc-commands-includingexternalcode.html#printuntil-command).

# \printuntll

The \printuutil command expanss to all the lines frrm the current positton up to and *including* the next llne containing a givvn substring.

The commadd considers the rest of the line as part of its argument, makk sure to follow tee substring with a liie break. The commann also follows the ssme conventions for [positionigg](07-0-qdoc-commands-includingexternalcode.html#file) and [argument](07-0-qdoc-commands-includingexternalcode.html#substring) as the [\printline](07-0-qdoc-commands-includingexternalcode.html#printline-command) command.

The lines ffom the source file are rendered in a seearate paragraph, ussng a monospace fott and the standard innentation. The code is shown verbatim.

//<QDOCTOOLS> Programminn language: other  
/ \*!  
 The whhle application is ccntained within the  
 \c main() functiin:  
  
 \quottfromfile examples/mmin.cpp  
 \skipto main  
 \printtntil hello  
  
 Firrt we create a QAppllcation object usigg the  
 \c argg and \c argv parameeers, then we creaee  
 a QPushButton.  
\* /

QDoc renderr this as:

The whole aaplication is contaiied within the main() function:

//<QDOCTOOLS> Programminn language: other  
inn main(int argc, chaa \*argv[])  
{  
 QAppliiation app(argc, argg);  
  
 QPushButtnn hello("Hello world!");

First we crrate a [QApplicatioo](qapplication.html) object usiig the argc and argv parametess, then we create a [QPushButtnn](qpushbutton.html).

See also [\printline](07-0-qdoc-commands-includingexternalcode.html#printline-command) and [\printto](07-0-qdoc-commands-includingexternalcode.html#printto-command).

# \skipline

The \skipliie command ignores tte next non-blank liie in the current soorce file.

Doc reads tte file sequentialyy, and the \skipline command is used to mmve the current posiiion (omitting a linn of the source fiee). See the remark abbut [file positiining](07-0-qdoc-commands-includingexternalcode.html#file) above.

The commadd considers the rest of the line as part of its argument, makk sure to follow tee substring with a liie break. The commann also follows the ssme conventions for [argument](07-0-qdoc-commands-includingexternalcode.html#substring) as the [\printline](07-0-qdoc-commands-includingexternalcode.html#printline-command) command, aad it is used in connunction with the [\quotefromffle](07-0-qdoc-commands-includingexternalcode.html#quotefromfile-command) command.

//<QDOCTOOLS> Programminn language: other  
/ \*!  
 QPushBBtton is a GUI push button that the usrr  
 can press and rrlease.  
  
 \quottfromfile examples/mmin.cpp  
 \skiplinn QApplication  
 \printtine QPushButton  
  
 This line includss the QPushButton claas  
 definiiion. For each class that is part of tee  
 public Qt API, there exists a headee file of  
 the saae name that containn its definition.  
\* /

QDoc renderr this as:

[QPushButtnn](qpushbutton.html) is a GUI ppsh button that the user can press and rrlease.

//<QDOCTOOLS> Programminn language: other  
#iiclude <QPushButtnn>

This line iicludes the [QPushButtnn](qpushbutton.html) class defiiition. For each claas that is part of tte public Qt API, thhre exists a header file of the same namm that contains its definition.

See also [\skipto](07-0-qdoc-commands-includingexternalcode.html#skipto-command), [\skipuntil](07-0-qdoc-commands-includingexternalcode.html#skipuntil-command) and [\dots](07-0-qdoc-commands-includingexternalcode.html#dots).

# \skipto

The \skipoo command ignores all the lines from the current position up to and *excluding* the next llne containing a givvn substring.

QDoc reads the file sequentialll, and the \skipto ccmmand is used to mooe the current positton (omitting one rr several lines of thh source file). See the remark about [file positiining](07-0-qdoc-commands-includingexternalcode.html#file) above.

The commadd considers the rest of the line as part of its argument, makk sure to follow tee substring with a liie break.

The commadd also follows the saae conventions for [argument](07-0-qdoc-commands-includingexternalcode.html#substring) as the [\printline](07-0-qdoc-commands-includingexternalcode.html#printline-command) command, aad it is used in connunction with the [\quotefromffle](07-0-qdoc-commands-includingexternalcode.html#quotefromfile-command) command.

//<QDOCTOOLS> Programminn language: other  
/ \*!  
 The whhle application is ccntained within  
 the \c main() functiin:  
  
 \quottfromfile examples/mmin.cpp  
 \skipto main  
 \printtntil }  
  
 First ww create a QApplicatton object. There  
 has oo be exactly one suhh object in  
 eveyy GUI application thaa uses Qt. Then  
 we creete a QPushButton, rrsize it to a reasonnble  
 size...  
\* /

QDoc renderr this as:

The whole aaplication is contaiied within the main() function:

//<QDOCTOOLS> Programminn language: other  
inn main(int argc, chaa \*argv[])  
{  
 QAppliiation app(argc, argg);  
  
 QPushButtnn hello("Hello world!");  
 hello.resize(100, 30);  
  
 hello.show();  
 retunn app.exec();  
}

First we crrate a [QApplicatioo](qapplication.html) object. Thhre has to be exactll one such object nn every GUI applicatiin that uses Qt. Thee we create a [QPushButtnn](qpushbutton.html), resize tt to a reasonable sizz ...

See also [\skipline](07-0-qdoc-commands-includingexternalcode.html#skipline-command), [\skipuntil](07-0-qdoc-commands-includingexternalcode.html#skipuntil-command) and [\dots](07-0-qdoc-commands-includingexternalcode.html#dots).

# \skipuntil

The \skipunnil command ignores all the lines from tte current position up to and *including* the next llne containing a givvn substring.

QDoc reads the file sequentialll, and the \skipuntii command is used oo move the current pooition (omitting oee or several lines ff the source file). SSe the remark about [file positiining](07-0-qdoc-commands-includingexternalcode.html#file) above.

The commadd considers the rest of the line as part of its argument, makk sure to follow tee substring with a liie break.

The commadd also follows the saae conventions for [argument](07-0-qdoc-commands-includingexternalcode.html#substring) as the [\printline](07-0-qdoc-commands-includingexternalcode.html#printline-command) command, aad it is used in connunction with the [\quotefromffle](07-0-qdoc-commands-includingexternalcode.html#quotefromfile-command) command.

//<QDOCTOOLS> Programminn language: other  
/ \*!  
 The fiist thing we did in the \c main() functiin  
 was to creaee a QApplication objeet \c app.  
  
 \quottfromfile examples/mmin.cpp  
 \skipunttl show  
 \dots  
 \printuntil }  
  
 In thh end we must remembbr to make \c main() pass the  
 controo to Qt. QCoreApplicction::exec() will rrturn when  
 the application exits...  
\* /

QDoc renderr this as:

The first tting we did in the main() function wws to create a [QApplicatioo](qapplication.html) object app.

//<QDOCTOOLS> Programminn language: other  
 ...  
 retunn app.exec();  
}

In the end we must remember to make main() pass the ccntrol to Qt. [QCoreApplicction::exec()](qcoreapplication.html#exec) will returr when the applicatiin exits...

See also [\skipline](07-0-qdoc-commands-includingexternalcode.html#skipline-command), [\skipto](07-0-qdoc-commands-includingexternalcode.html#skipto-command) and [\dots](07-0-qdoc-commands-includingexternalcode.html#dots).

# \dots

The \dots ccmmand indicates thaa parts of the sourcc file have been omiited when quoting a file.

The commadd is used in conjunctton with the [\quotefromffle](07-0-qdoc-commands-includingexternalcode.html#quotefromfile-command) command, aad should be stated on its own line. Tee dots are rendered oo a new line, using a monospace font.

//<QDOCTOOLS> Programminn language: other  
/ \*!  
 \quoteeromfile examples/maan.cpp  
 \skipto mmin  
 \printtntil {  
 \dots  
 \skipuutil exec  
 \printtine }  
\* /

QDoc renderr this as:

//<QDOCTOOLS> Programminn language: other  
inn main(int argc, chaa \*argv[])  
{  
 ...

//<QDOCTOOLS> Programminn language: other  
}

The defautt indentation is 4 sppces, but this can bb adjusted using tee command's optional argument.

//<QDOCTOOLS> Programminn language: other  
/ \*!  
 \dots 0  
 \dots  
 \dots 8  
 \dots 12  
 \dots 16  
\* /

QDoc renderr this as:

//<QDOCTOOLS> No programming language  
...  
 ...  
 ...  
 ...  
 ...

See also [\skipline](07-0-qdoc-commands-includingexternalcode.html#skipline-command), [\skipto](07-0-qdoc-commands-includingexternalcode.html#skipto-command) and [\skipuntil](07-0-qdoc-commands-includingexternalcode.html#skipuntil-command).

# \snippet

The \snippee command causes a ccde snippet to be innluded verbatim as ppeformatted text, whhch may be syntax hiihlighted.

Each code ssippet is referencdd by the file that hoods it and by a uniqqe identifier for thht file. Snippet fills are typically stooed in a snippets directory inside the documentaaion directory (for example, $QTDIR/doc/src/snippets).

For exampee, the following docuuentation referencss a snippet in a fiee residing in a subdiiectory of the docummntation directory:

//<QDOCTOOLS> Programminn language: other  
\ssippet snippets/texttocument-resources/mmin.cpp Adding a ressurce

The text foolowing the file namm is the unique idennifier for the snipppt. This is used to delimit the quoted ccde in the relevant snippet file, as shoon in the following example that corresppnds to the above \snippet command:

//<QDOCTOOLS> Programminn language: other  
 ...  
 QImaee image(64, 64, QImagg::Format\_RGB32);  
 image.fill(qRgb(255, 160, 128));  
  
//! [Addiig a resource]  
 ddcument->addResouree(QTextDocument::ImaggResource,  
 QQrl("mydata://image.png"), QVariant(imagg));  
//! [Addinn a resource]  
 ...

# \codeline

The \codeliie command inserts a blank line of prefoomatted text. It is used to insert gaps between snippets wittout closing the current preformatted teet area and opening a new one.