[How to execute a cmd command using QProcess?](https://stackoverflow.com/questions/21596104/how-to-execute-a-cmd-command-using-qprocess)

QProcess::startDetached will take the first parameter as the command to execute and the following parameters, delimited by a space, will be interpreted as separate arguments to the command.

Therefore, in this case: -

QProcess::startDetached("cmd /c net stop \"MyService\"");

The function sees **cmd** as the command and passes /c, net, stop and "MyService" as arguments to cmd. However, other than /c, the others are parsed separately and are not valid arguments.

What you need to do is use quotes around the "net stop \"MyService\" to pass it as a single argument, so that would give you: -

QProcess::startDetached("cmd /c \"net stop \"MyService\"\"");

Alternatively, using the string list you could use: -

QProcess::startDetached("cmd", QStringList() << "/c" << "net stop \"MyService\"");

# [Qt - reading from a text file](https://stackoverflow.com/questions/2612103/qt-reading-from-a-text-file)

You have to replace string line

QString line = in.readLine();

into while:

QFile file("/home/hamad/lesson11.txt");

if(!file.open(QIODevice::ReadOnly)) {

QMessageBox::information(0, "error", file.errorString());

}

QTextStream in(&file);

while(!in.atEnd()) {

QString line = in.readLine();

QStringList fields = line.split(",");

model->appendRow(fields);

}

file.close();

# [QT C++ remove file with \* (name contains)](https://stackoverflow.com/questions/52133207/qt-c-remove-file-with-name-contains)

The main task is to filter the files so we can use QDir with the nameFilter as shown below:

QDir dir("/path/of/directory", {"test\*.txt"});

for(const QString & filename: dir.entryList()){

dir.remove(filename);

}

Or use QDirIterator:

QDirIterator it("/path/of/directory", {"test\*.txt"});

while (it.hasNext())

QFile(it.next()).remove();

//QDir().remove(it.next());

QMainWindow – widget window title

QIcon icon("src/Me.png");

setWindowIcon(icon);

setWindowTitle(QStringLiteral("主窗口"));

QWidget

QIcon icon("src/Me.png");

setWindowIcon(icon);

setWindowTitle(QStringLiteral("XX窗口"));

QApplication

QApplication a(argc, argv);

QIcon icon("src/Me.png");

# [Get the Highlighted/Selected text](https://stackoverflow.com/questions/5379120/get-the-highlighted-selected-text)

Getting the text the user has selected is relatively simple. There's no benefit to be gained by involving jQuery since you need nothing other than the window and document objects.

function getSelectionText() {

var text = "";

if (window.getSelection) {

text = window.getSelection().toString();

} else if (document.selection && document.selection.type != "Control") {

text = document.selection.createRange().text;

}

return text;

}

If you're interested in an implementation that will also deal with selections in <textarea> and texty <input> elements, you could use the following. Since it's now 2016 I'm omitting the code required for IE <= 8 support but I've posted stuff for that in many places on SO.

function getSelectionText() {

var text = "";

var activeEl = document.activeElement;

var activeElTagName = activeEl ? activeEl.tagName.toLowerCase() : null;

if (

(activeElTagName == "textarea") || (activeElTagName == "input" &&

/^(?:text|search|password|tel|url)$/i.test(activeEl.type)) &&

(typeof activeEl.selectionStart == "number")

) {

text = activeEl.value.slice(activeEl.selectionStart, activeEl.selectionEnd);

} else if (window.getSelection) {

text = window.getSelection().toString();

}

return text;

}

document.onmouseup = document.onkeyup = document.onselectionchange = function() {

document.getElementById("sel").value = getSelectionText();

};

Selection:

<br>

<textarea id="sel" rows="3" cols="50"></textarea>

<p>Please select some text.</p>

<input value="Some text in a text input">

<br>

<input type="search" value="Some text in a search input">

<br>

<input type="tel" value="4872349749823">

<br>

<textarea>Some text in a textarea</textarea>

# [QtWebEngine - synchronously execute JavaScript to read function result](https://stackoverflow.com/questions/45330481/qtwebengine-synchronously-execute-javascript-to-read-function-result)

The callback is asynchronous because the JavaScript execution occurs not only in another thread but in another process. So there is no way to make it fully synchronous.

void ranJavaScript()

{

emit notifyRanJavaScript();

}

QString get()

{

QString result;

QEventLoop loop;

QObject::connect(this, SIGNAL(notifyRanJavaScript()), &loop, SLOT(quit()));

view->page()->runJavaScript("test();", [this](const QVariant &v)

{

result = v.toString();

this.ranJavaScript();

});

loop.exec();

return result;

}