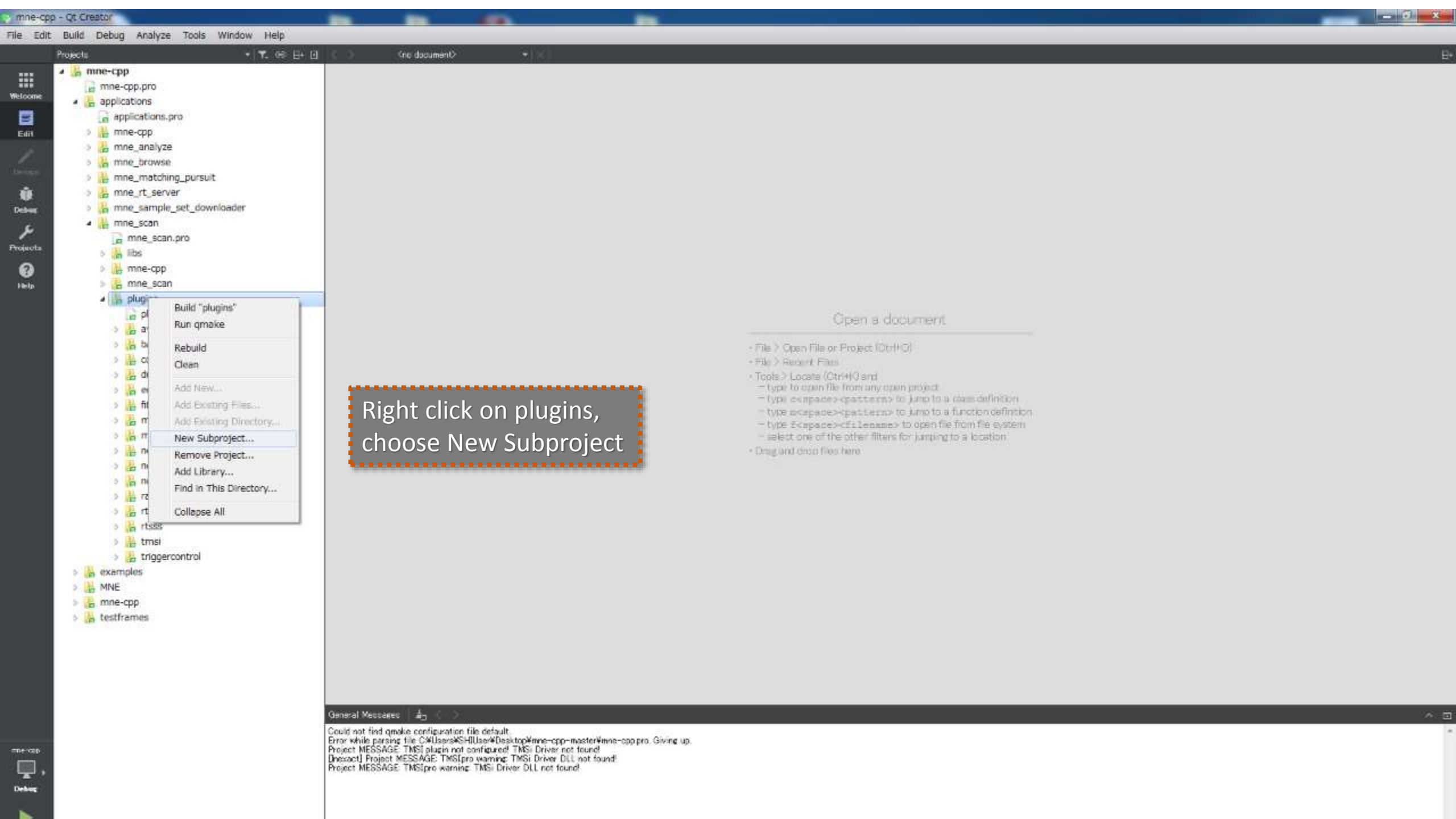
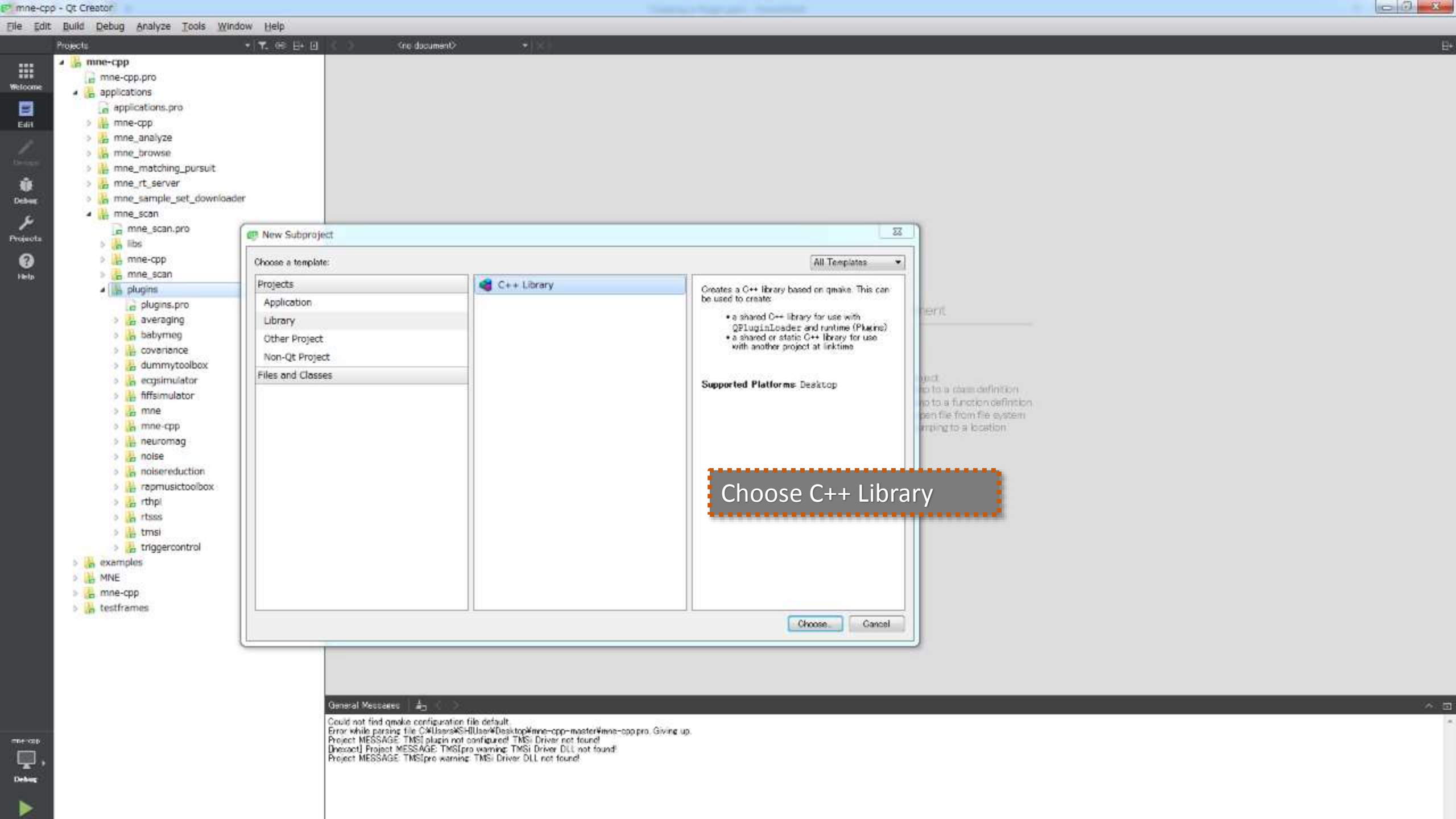
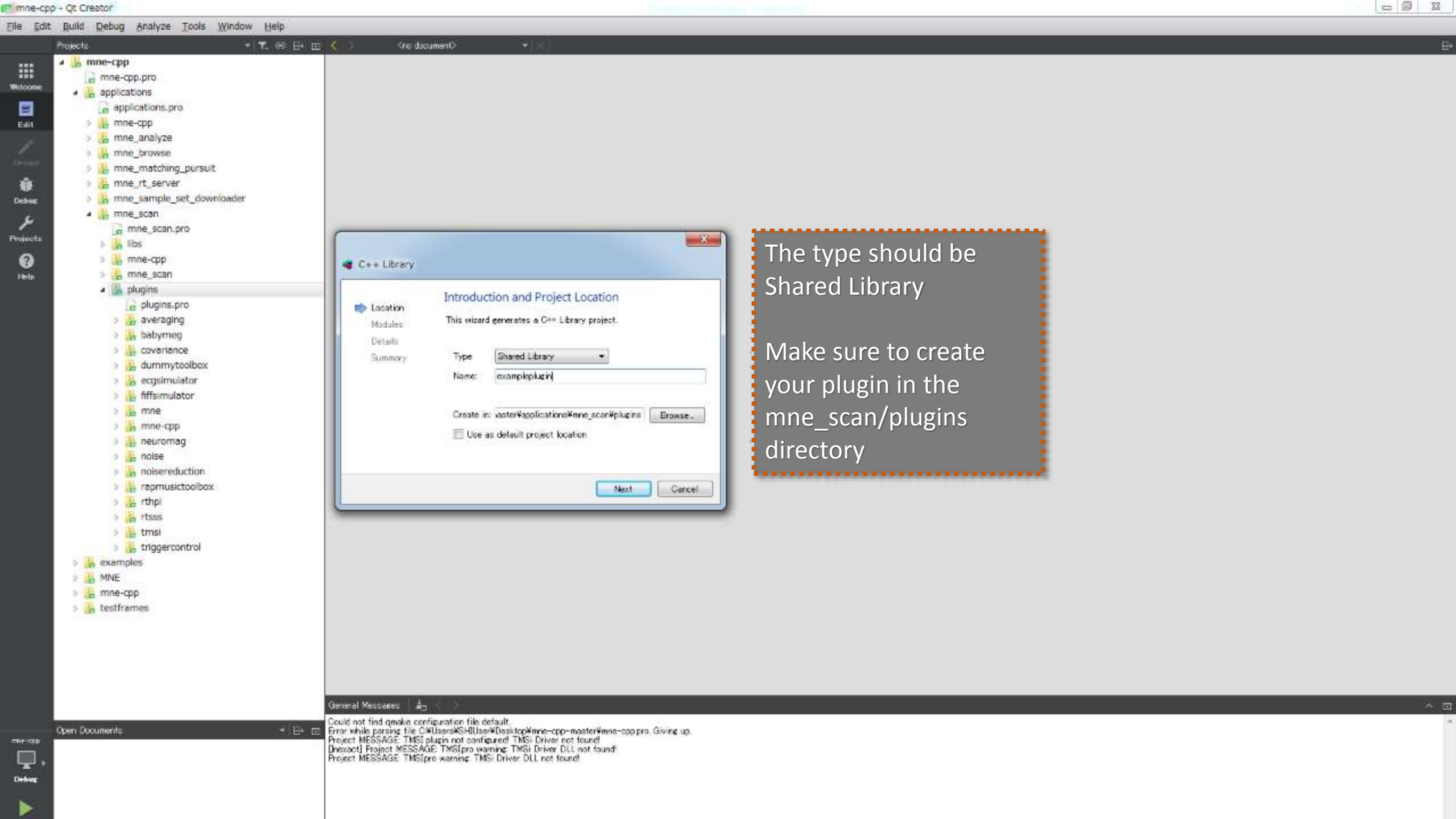
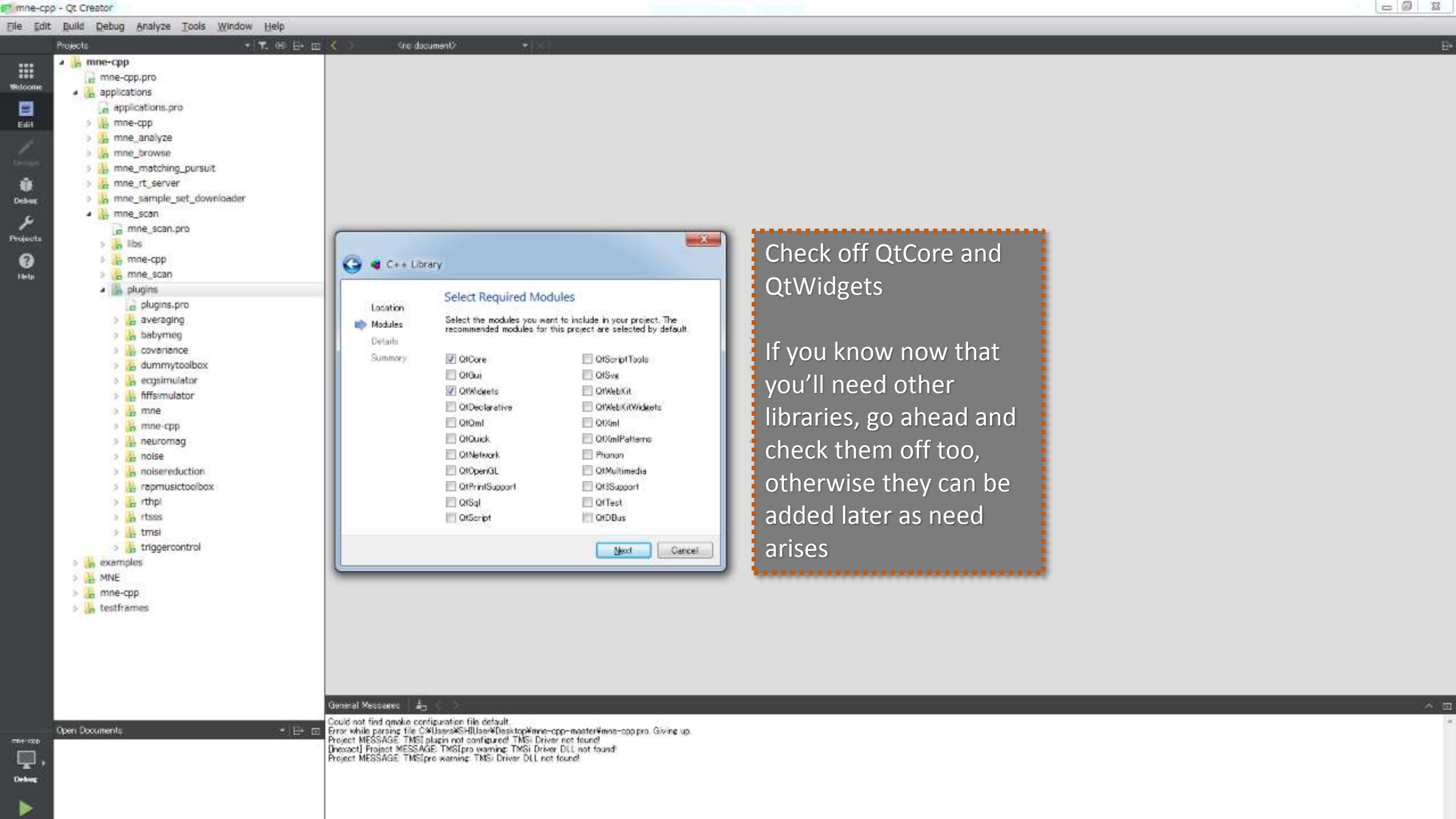
Creating a Subproject



Right click on plugins, choose New Subproject

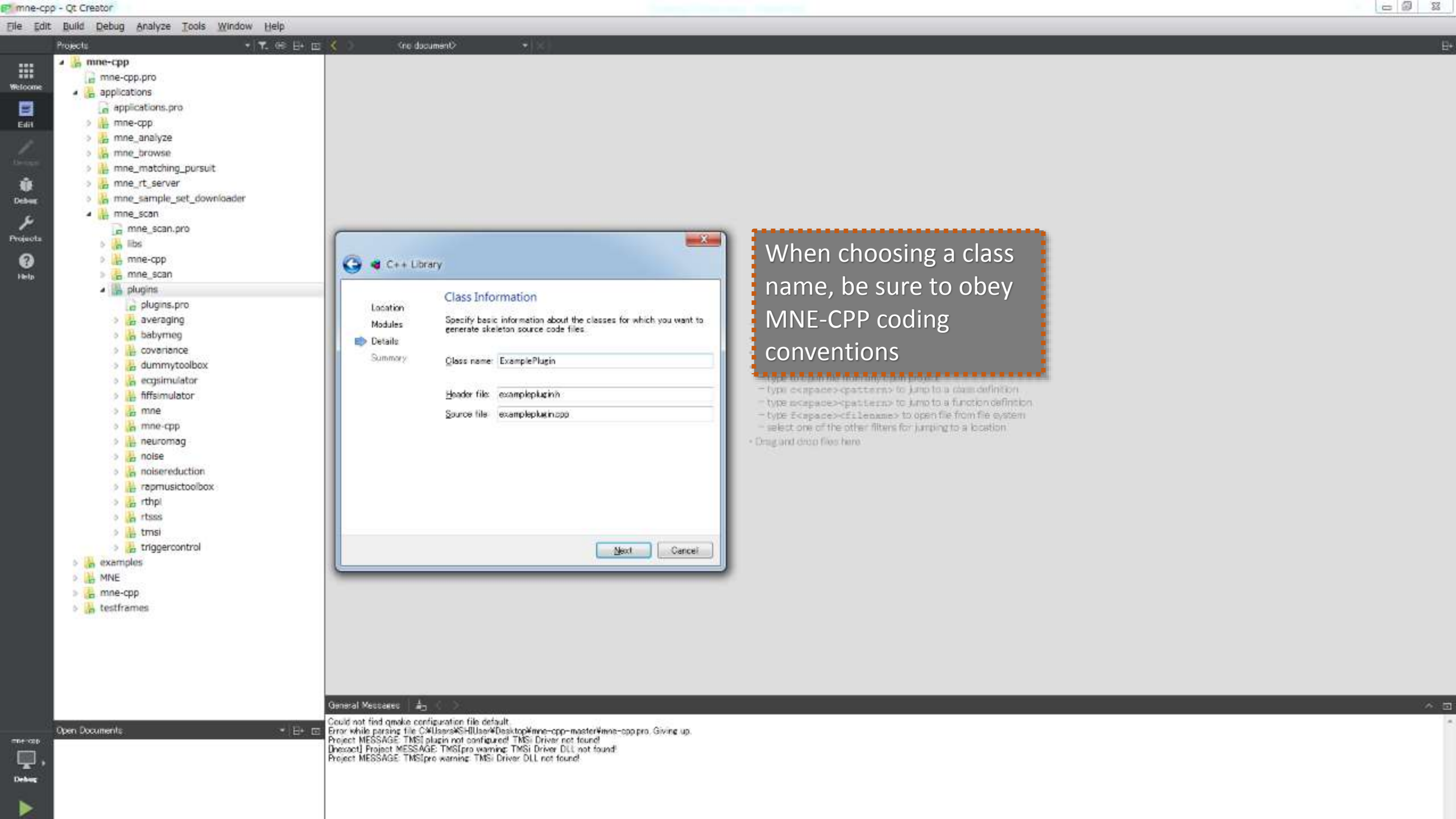

mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
rthpi
rtsss
tmsi
triggercontrol
examples
MNE
mne-cpp
testframes
New Subproject
Choose a template:
All Templates
Projects
Application
Library
Other Project
Non-Qt Project
Files and Classes
C++ Library
Creates a C++ library based on qmake. This can be used to create:
a shared C++ library for use with QPluginLoader and runtime (Plugins)
a shared or static C++ library for use with another project at linktime
Supported Platforms: Desktop
Choose...
Cancel
Choose C++ Library
General Messages
Could not find qmake configuration file default
Error while parsing file C:\Users\SHI\User\Desktop\mne-cpp-master\mne-cpp.pro. Giving up.
Project MESSAGE: TMSI plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!


mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
<no document>
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
rthpi
rtsss
tmsi
triggercontrol
examples
MNE
mne-cpp
testframes
Welcome
Edit
Debug
Projects
Help
Open Documents
General Messages
Could not find qmake configuration file default.
Error while parsing file C:¥Users¥SHIUser¥Desktop¥mne-cpp-master¥mne-cpp.pro. Giving up.
Project MESSAGE: TMSI plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
C++ Library
Introduction and Project Location
Location
Modules
Details
Summary
This wizard generates a C++ Library project.
Type
Shared Library
Name:
exampleplugin
Create in:
aster¥applications¥mne_scan¥plugins
Browse...
Use as default project location
Next
Cancel
The type should be Shared Library
Make sure to create your plugin in the mne_scan/plugins directory



Check off QtCore and QtWidgets

If you know now that you'll need other libraries, go ahead and check them off too, otherwise they can be added later as need arises


mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
rthpi
rtsss
tmsi
triggercontrol
examples
MNE
mne-cpp
testframes
Welcome
Edit
Debug
Projects
Help
Open Documents
General Messages
Could not find qmake configuration file default.
Error while parsing file C:\Users\SHI\User\Desktop\mne-cpp-master\mne-cpp.pro. Giving up.
Project MESSAGE: TMSI plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
C++ Library
Location
Modules
Details
Summary
Class Information
Specify basic information about the classes for which you want to generate skeleton source code files.
Class name: ExamplePlugin
Header file: exampleplugin.h
Source file: exampleplugin.cpp
Next
Cancel
When choosing a class name, be sure to obey MNE-CPP coding conventions

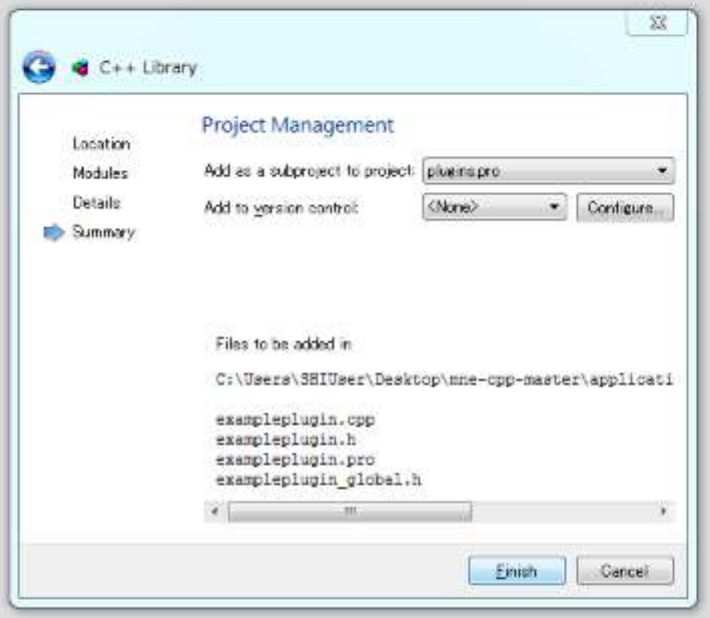
mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

- mne-cpp
  - mne-cpp.pro
  - applications
    - applications.pro
    - mne-cpp
    - mne_analyze
    - mne_browse
    - mne_matching_pursuit
    - mne_rt_server
    - mne_sample_set_downloader
    - mne_scan
      - mne_scan.pro
      - libs
      - mne-cpp
      - mne_scan
      - plugins
        - plugins.pro
        - averaging
        - babymeg
        - covariance
        - dummytoolbox
        - ecgsimulator
        - fiffsimulator
        - mne
        - mne-cpp
        - neuromag
        - noise
        - noisereduction
        - rapmusictoolbox
        - rthpi
        - rtsss
        - tmsi
        - triggercontrol
  - examples
  - MNE
  - mne-cpp
  - testframes

Open Documents

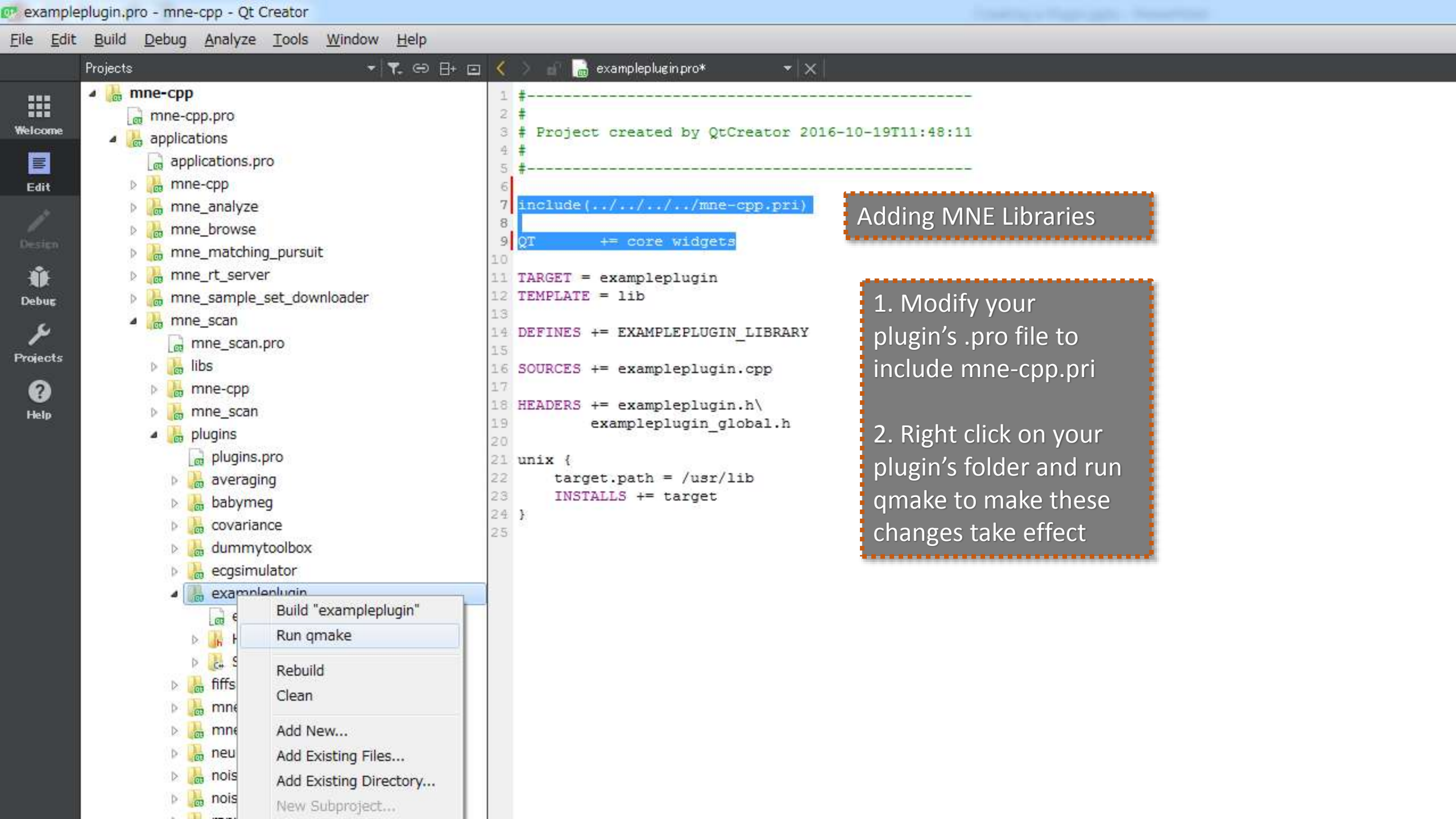


Nothing needs to be done here. Click Finish.

General Messages

Could not find qmake configuration file default
Error while parsing file C:¥Users¥SBIUser¥Desktop¥mne-cpp-master¥mne-cpp.pro. Giving up.
Project MESSAGE: TMSI plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!

Linking MNE Libraries


exampleplugin.pro - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome
Edit
Design
Debug
Projects
Help
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
Build "exampleplugin"
Run qmake
Rebuild
Clean
Add New...
Add Existing Files...
Add Existing Directory...
New Subproject...
exampleplugin.pro*
#---
#
Project created by QtCreator 2016-10-19T11:48:11
#
#---
include(../../../../mne-cpp.pri)
QT += core widgets
TARGET = exampleplugin
TEMPLATE = lib
DEFINES += EXAMPLEPLUGIN_LIBRARY
SOURCES += exampleplugin.cpp
HEADERS += exampleplugin.h\
exampleplugin_global.h
unix {
target.path = /usr/lib
INSTALLS += target
}
Adding MNE Libraries
1. Modify your plugin's .pro file to include mne-cpp.pri
2. Right click on your plugin's folder and run qmake to make these changes take effect

exampleplugin.pro - mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

- mne-cpp
  - mne-cpp.pro
  - applications
    - applications.pro
    - mne-cpp
    - mne_analyze
    - mne_browse
    - mne_matching_pursuit
    - mne_rt_server
    - mne_sample_set_downloader
    - mne_scan
      - mne_scan.pro
      - libs
      - mne-cpp
      - mne_scan
      - plugins
        - plugins.pro
        - averaging
        - babymeg
        - covariance
        - dummytoolbox
          - dummytoolbox.pro
          - mne-cpp
          - Headers
          - Sources
          - Forms
          - Other files
        - ecgsimulator
        - exampleplugin
          - exampleplugin.pro
          - Headers
          - Sources
        - fiffsimulator
        - mne
        - mne-cpp
        - neuromag
        - noise
        - noisereduction
        - rapmusictoolbox
        - rthpi
        - rtsss
        - tmsi
        - triggercontrol
  - examples

Open Documents

- averaging.pro
- dummytoolbox.pro
- exampleplugin.cpp
- exampleplugin.pro*

exampleplugin.pro*

```
#-------------------------------------------------
#
# Project created by QtCreator 2016-10-19T11:48:11
#
#-------------------------------------------------

include(../../../../mne-cpp.pri)

QT       += core widgets

TARGET = exampleplugin
TEMPLATE = lib

DEFINES += EXAMPLEPLUGIN_LIBRARY

SOURCES += exampleplugin.cpp

HEADERS += exampleplugin.h\
        exampleplugin_global.h

unix {
    target.path = /usr/lib
    INSTALLS += target
}

LIBS += -L$$(MNE_LIBRARY_DIR)
CONFIG(debug, debug|release) {
    LIBS += -lMNE$$(MNE_LIB_VERSION)Genericsd \
            -lscMeasd \
            -lscDispd \
            -lscSharedd
}
else {
    LIBS += -lMNE$$(MNE_LIB_VERSION)Generics \
            -lscMeas \
            -lscDisp \
            -lscShared
}

DESTDIR = $$(MNE_BINARY_DIR)/mne_scan_plugins

INCLUDEPATH += $$(EIGEN_INCLUDE_DIR)
INCLUDEPATH += $$(MNE_INCLUDE_DIR)
INCLUDEPATH += $$(MNE_SCAN_INCLUDE_DIR)

# Put generated form headers into the origin --> cause other src is pointing at them
UI_DIR = $$PWD

unix: QMAKE_CXXFLAGS += -isystem $$EIGEN_INCLUDE_DIR

# suppress visibility warnings
unix: QMAKE_CXXFLAGS += -Wno-attributes
```

3. Add this chunk of code to the .pro file. Most of it can be copied from the .pro file of another mne_scan plugin

Executed on debug builds

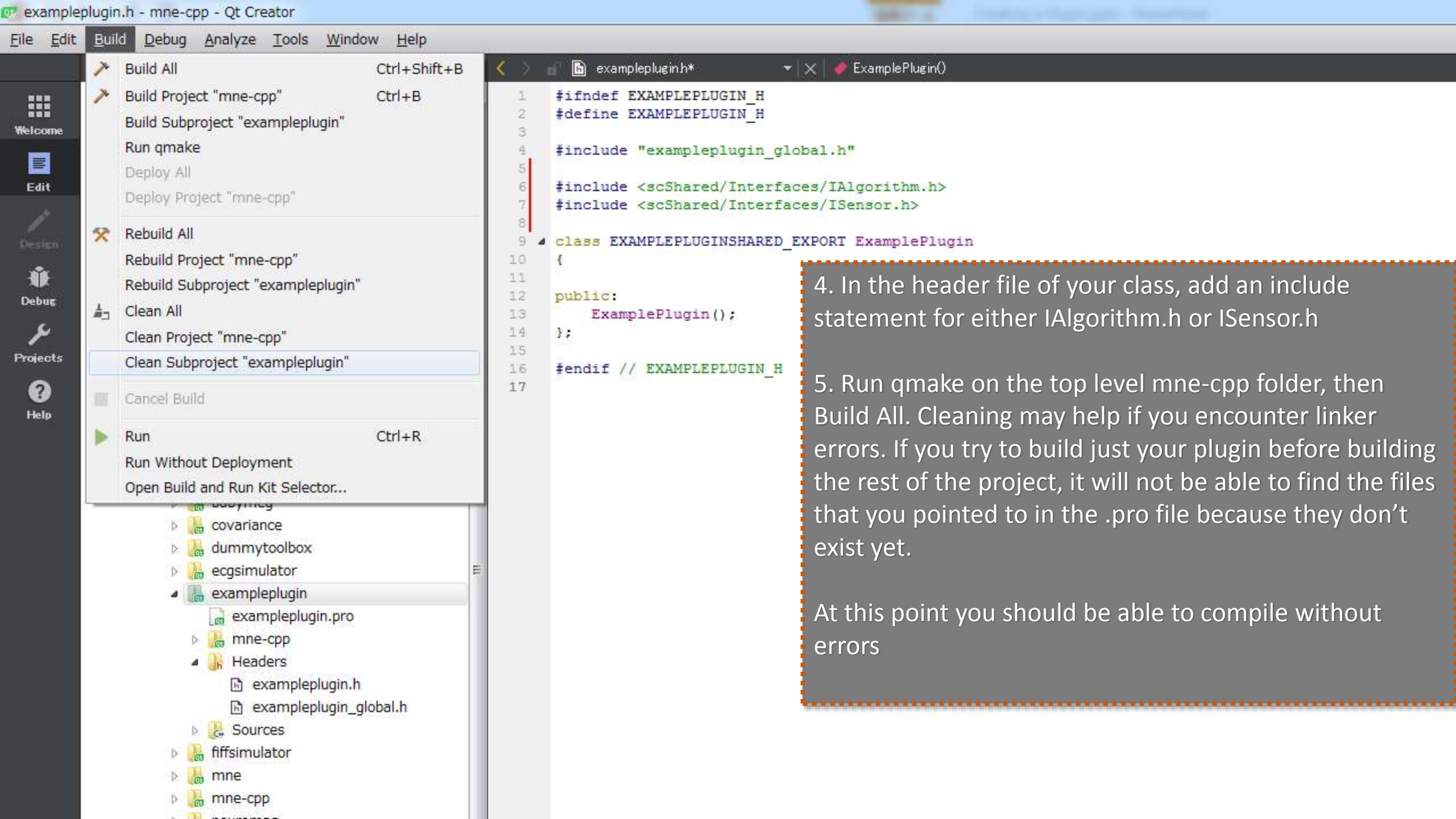
Files meant for debug builds have a “d” on the end

Executed on release builds

MNE_LIBRARY_DIR, MNE_INCLUDE_DIR, etc. are compiler macros defined in mne-cpp.pri

General Messages

```
Could not find qmake configuration file default.
Error while parsing file C:\Users\SHIUser\Desktop\mne-cpp-master\mne-cpp.pro. Giving up.
Project MESSAGE: TMSI plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSI plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSI plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
```

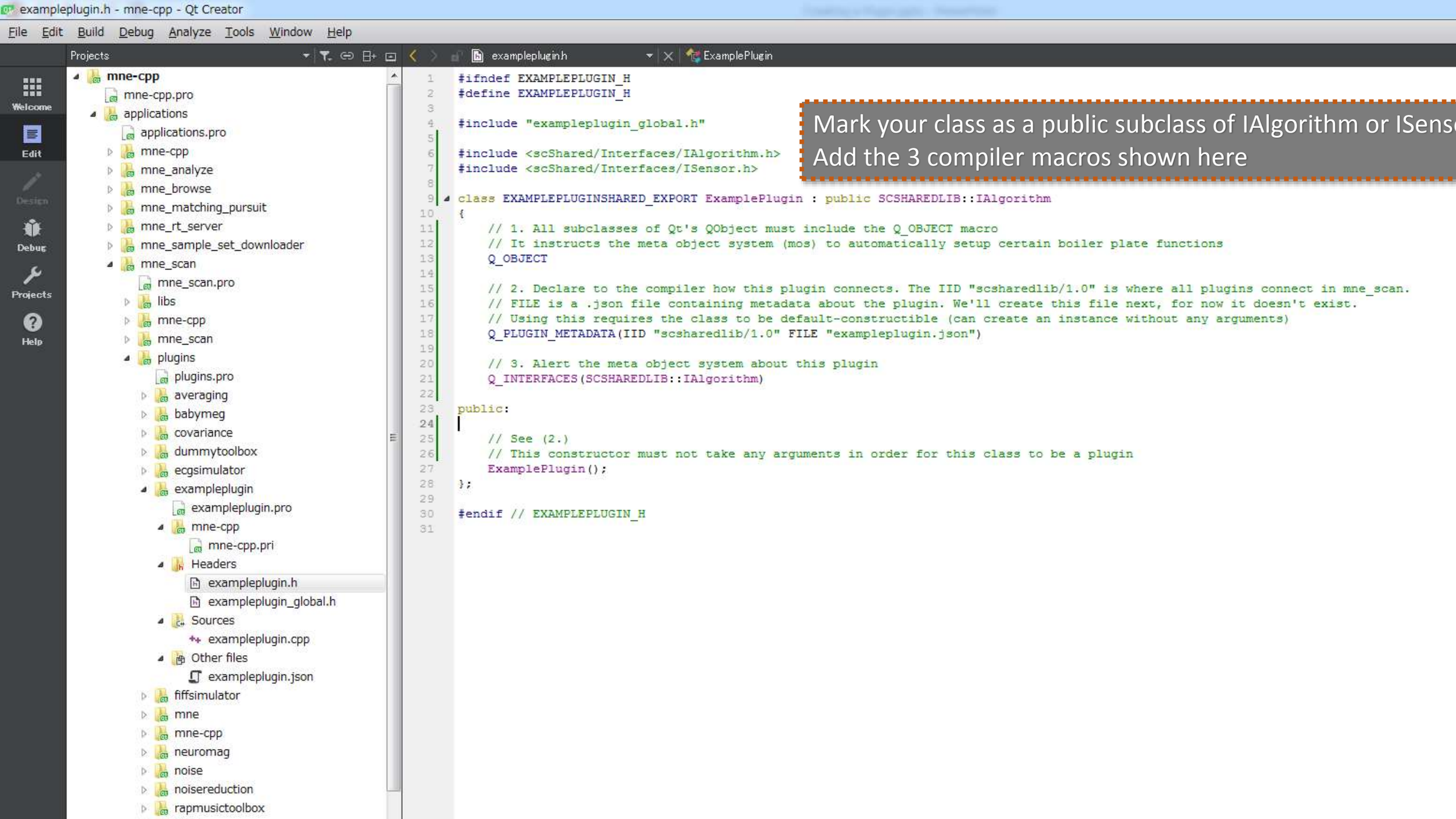


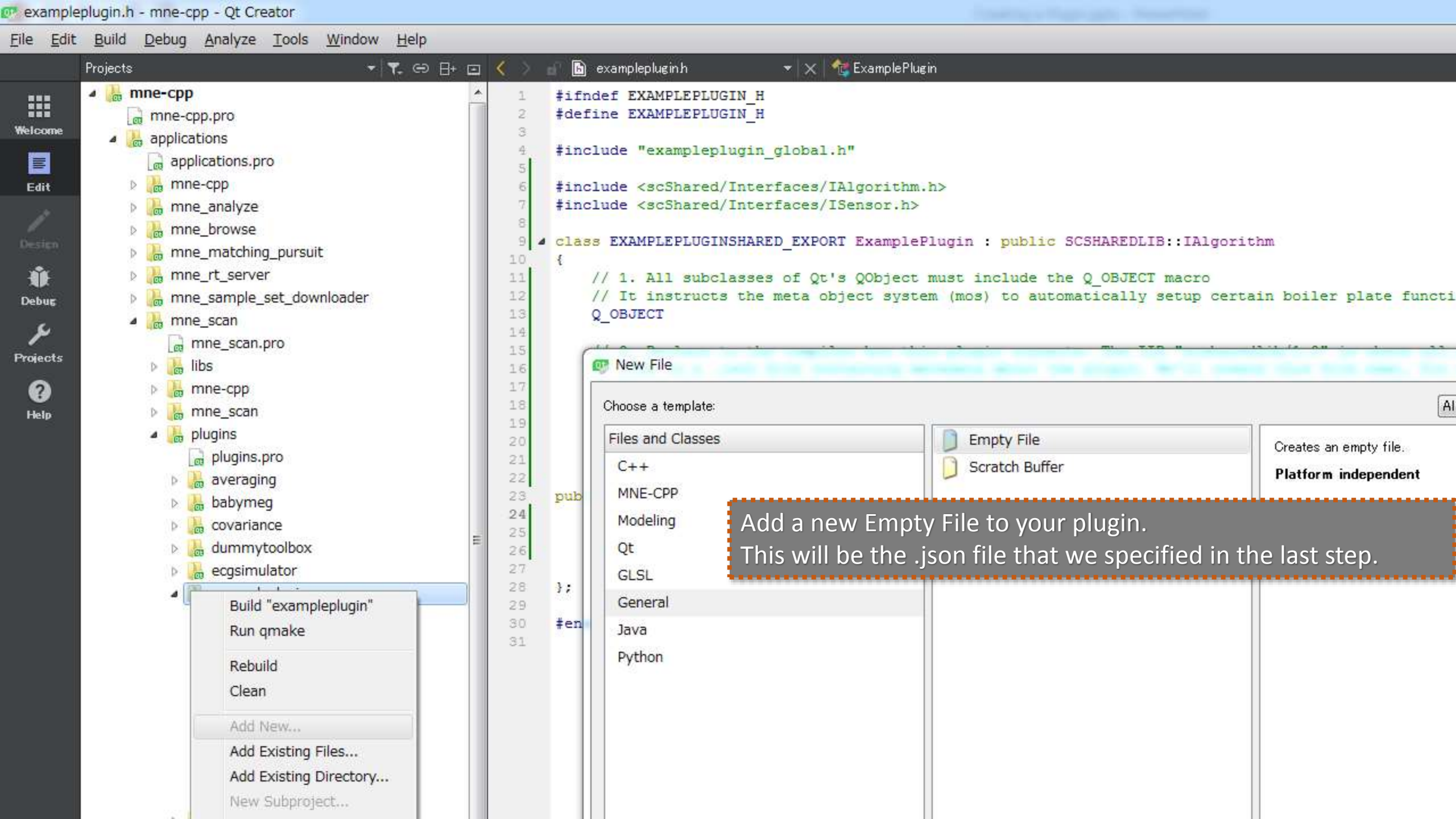
4. In the header file of your class, add an include statement for either IAlgorithm.h or ISensor.h

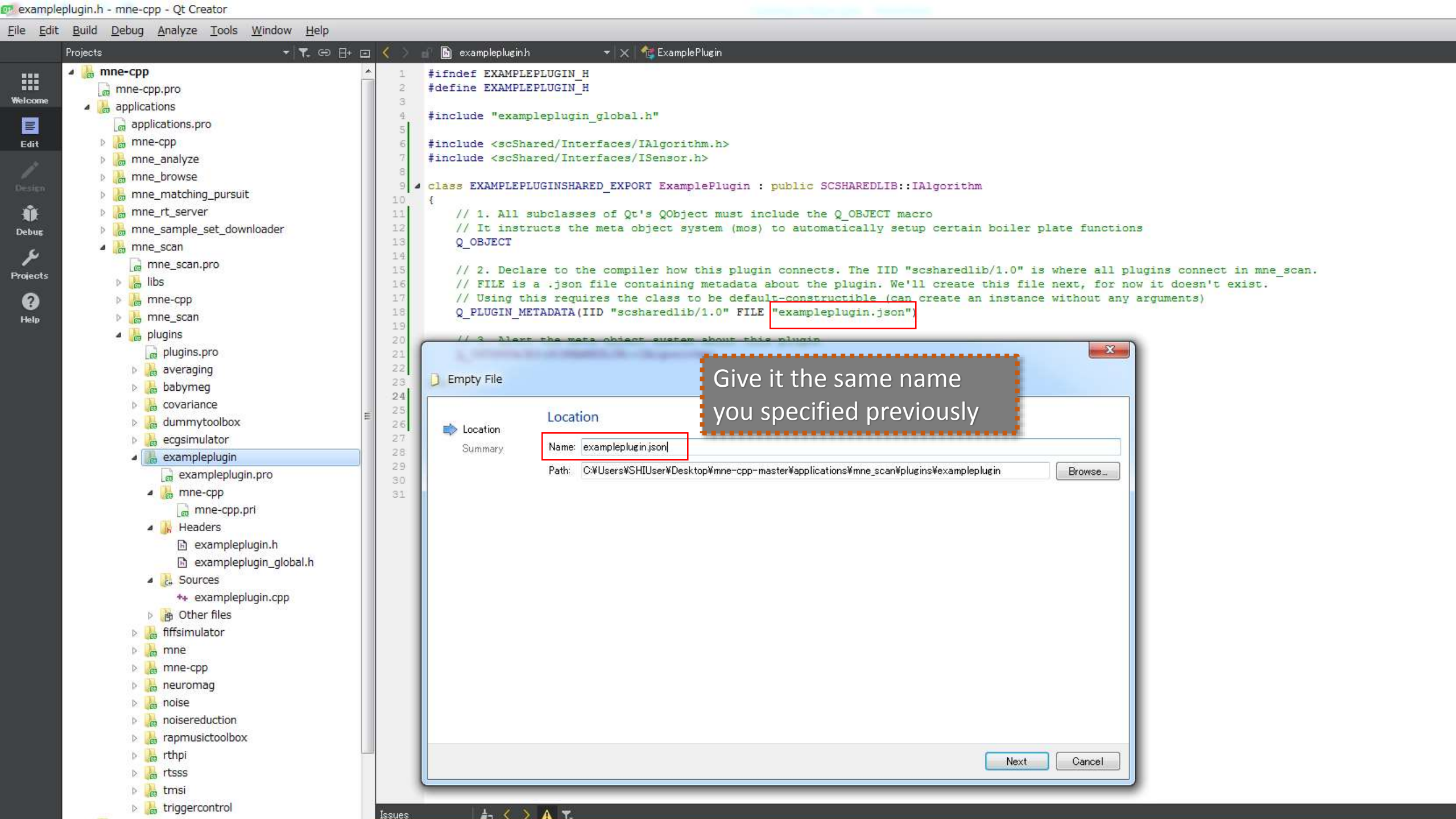
5. Run qmake on the top level mne-cpp folder, then Build All. Cleaning may help if you encounter linker errors. If you try to build just your plugin before building the rest of the project, it will not be able to find the files that you pointed to in the .pro file because they don't exist yet.

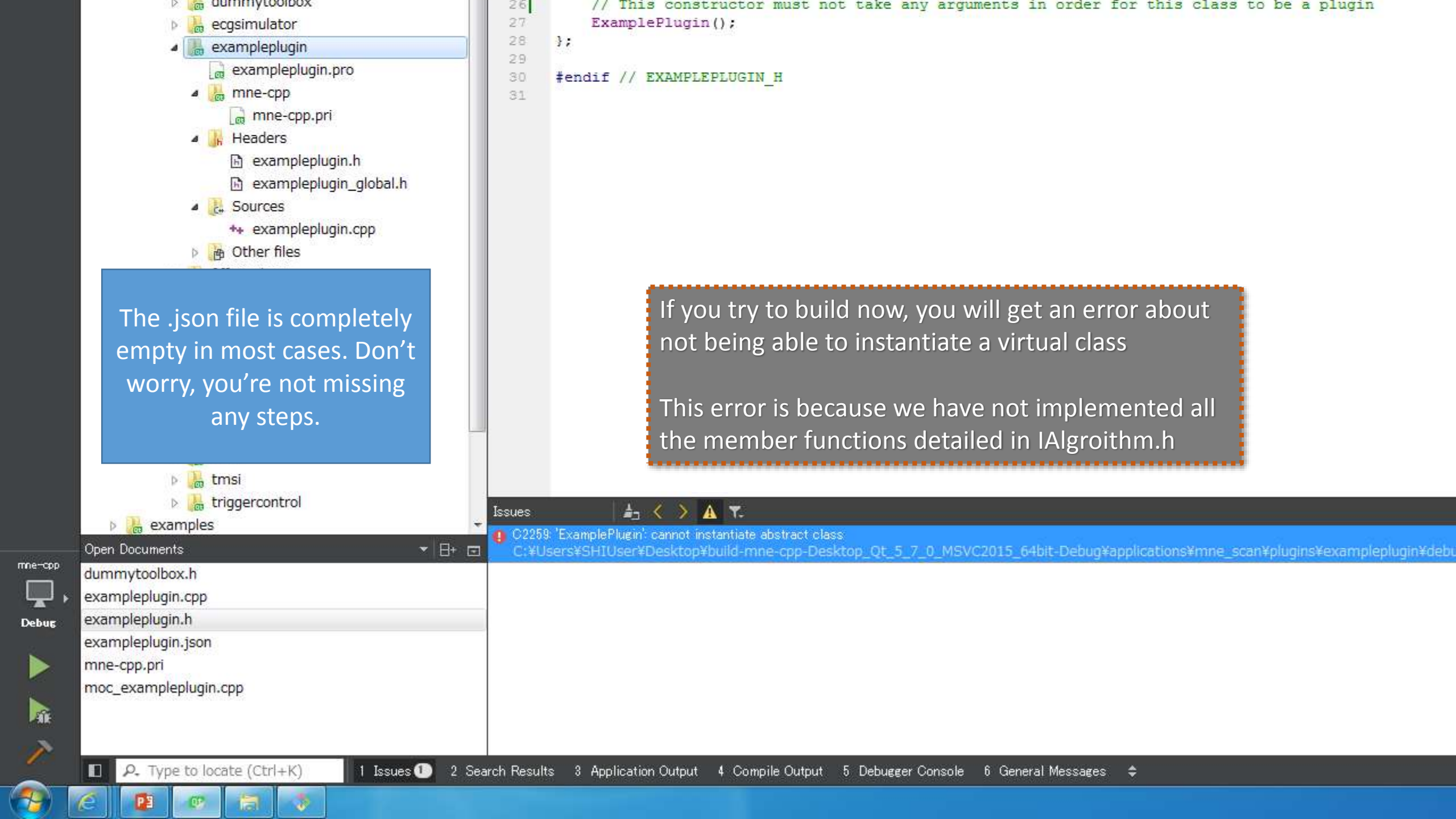
At this point you should be able to compile without errors

Subclassing IPlugin


exampleplugin.h - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
mne-cpp.pri
Headers
exampleplugin.h
exampleplugin_global.h
Sources
exampleplugin.cpp
Other files
exampleplugin.json
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
Welcome
Edit
Design
Debug
Projects
Help
exampleplugin.h
ExamplePlugin
#ifndef EXAMPLEPLUGIN_H
#define EXAMPLEPLUGIN_H
#include "exampleplugin_global.h"
#include <scShared/Interfaces/IAlgorithm.h>
#include <scShared/Interfaces/ISensor.h>
class EXAMPLEPLUGINSHARED_EXPORT ExamplePlugin : public SCSHAREDLIB::IAlgorithm
{
// 1. All subclasses of Qt's QObject must include the Q_OBJECT macro
// It instructs the meta object system (mos) to automatically setup certain boiler plate functions
Q_OBJECT
// 2. Declare to the compiler how this plugin connects. The IID "scsharedlib/1.0" is where all plugins connect in mne_scan.
// FILE is a .json file containing metadata about the plugin. We'll create this file next, for now it doesn't exist.
// Using this requires the class to be default-constructible (can create an instance without any arguments)
Q_PLUGIN_METADATA(IID "scsharedlib/1.0" FILE "exampleplugin.json")
// 3. Alert the meta object system about this plugin
Q_INTERFACES(SCSHAREDLIB::IAlgorithm)
public:
// See (2.)
// This constructor must not take any arguments in order for this class to be a plugin
ExamplePlugin();
};
#endif // EXAMPLEPLUGIN_H
Mark your class as a public subclass of IAlgorithm or ISens
Add the 3 compiler macros shown here


exampleplugin.h - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
exampleplugin.h
ExamplePlugin
Welcome
Edit
Design
Debug
Projects
Help
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
Build "exampleplugin"
Run qmake
Rebuild
Clean
Add New...
Add Existing Files...
Add Existing Directory...
New Subproject...
#ifndef EXAMPLEPLUGIN_H
#define EXAMPLEPLUGIN_H
#include "exampleplugin_global.h"
#include <scShared/Interfaces/IAlgorithm.h>
#include <scShared/Interfaces/ISensor.h>
class EXAMPLEPLUGINSHARED_EXPORT ExamplePlugin : public SCSHAREDLIB::IAlgorithm
{
// 1. All subclasses of Qt's QObject must include the Q_OBJECT macro
// It instructs the meta object system (mos) to automatically setup certain boiler plate functi
Q_OBJECT
};
#en
New File
Choose a template:
Files and Classes
C++
MNE-CPP
Modeling
Qt
GLSL
General
Java
Python
Empty File
Scratch Buffer
Creates an empty file.
Platform independent
Add a new Empty File to your plugin.
This will be the .json file that we specified in the last step.


exampleplugin.h - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
exampleplugin.h
ExamplePlugin
Welcome
Edit
Design
Debug
Projects
Help
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
mne-cpp.pri
Headers
exampleplugin.h
exampleplugin_global.h
Sources
exampleplugin.cpp
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
rthpi
rtsss
tmsi
triggercontrol
#ifndef EXAMPLEPLUGIN_H
#define EXAMPLEPLUGIN_H
#include "exampleplugin_global.h"
#include <scShared/Interfaces/IAlgorithm.h>
#include <scShared/Interfaces/ISensor.h>
class EXAMPLEPLUGINSHARED_EXPORT ExamplePlugin : public SCSHAREDLIB::IAlgorithm
{
// 1. All subclasses of Qt's QObject must include the Q_OBJECT macro
// It instructs the meta object system (mos) to automatically setup certain boiler plate functions
Q_OBJECT
// 2. Declare to the compiler how this plugin connects. The IID "scsharedlib/1.0" is where all plugins connect in mne_scan.
// FILE is a .json file containing metadata about the plugin. We'll create this file next, for now it doesn't exist.
// Using this requires the class to be default-constructible (can create an instance without any arguments)
Q_PLUGIN_METADATA(IID "scsharedlib/1.0" FILE "exampleplugin.json")
Empty File
Location
Summary
Location
Name: exampleplugin.json
Path: C:¥Users¥SHIUser¥Desktop¥mne-cpp-master¥applications¥mne_scan¥plugins¥exampleplugin
Browse...
Next
Cancel
Give it the same name
you specified previously
Issues


dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
mne-cpp.pri
Headers
exampleplugin.h
exampleplugin_global.h
Sources
exampleplugin.cpp
Other files
tmsi
triggercontrol
examples
// This constructor must not take any arguments in order for this class to be a plugin
ExamplePlugin();
};
#endif // EXAMPLEPLUGIN_H
The .json file is completely empty in most cases. Don't worry, you're not missing any steps.
If you try to build now, you will get an error about not being able to instantiate a virtual class
This error is because we have not implemented all the member functions detailed in IAlgroithm.h
Issues
C2259: 'ExamplePlugin': cannot instantiate abstract class
C:¥Users¥SHIUser¥Desktop¥build-mne-cpp-Desktop_Qt_5_7_0_MSVC2015_64bit-Debug¥applications¥mne_scan¥plugins¥exampleplugin¥debu
Open Documents
dummytoolbox.h
exampleplugin.cpp
exampleplugin.h
exampleplugin.json
mne-cpp.pri
moc_exampleplugin.cpp
mne-cpp
Debug
Type to locate (Ctrl+K)
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 Debugger Console
6 General Messages

IAlgorithm.h - mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

- mne-cpp
  - mne-cpp.pro
  - applications
    - applications.pro
    - mne-cpp
    - mne_analyze
    - mne_browse
    - mne_matching_pursuit
    - mne_rt_server
    - mne_sample_set_downloader
    - mne_scan
      - mne_scan.pro
      - libs
      - mne-cpp
      - mne_scan
      - plugins
        - plugins.pro
        - averaging
        - babymeg
        - covariance
        - dummytoolbox
        - ecgsimulator
        - exampleplugin
          - exampleplugin.pro
          - mne-cpp
            - mne-cpp.pri
          - Headers
            - exampleplugin.h
            - exampleplugin_global.h
          - Sources
            - exampleplugin.cpp
          - Other files
        - fiffsimulator
        - mne
        - mne-cpp
        - neuromag
        - noise
        - noisereduction
        - rapmusictoolbox
        - rthpi
        - rtsss
        - tmsi
        - triggercontrol
  - examples

Open Documents

IAlgorithm.h | IAlgorithm

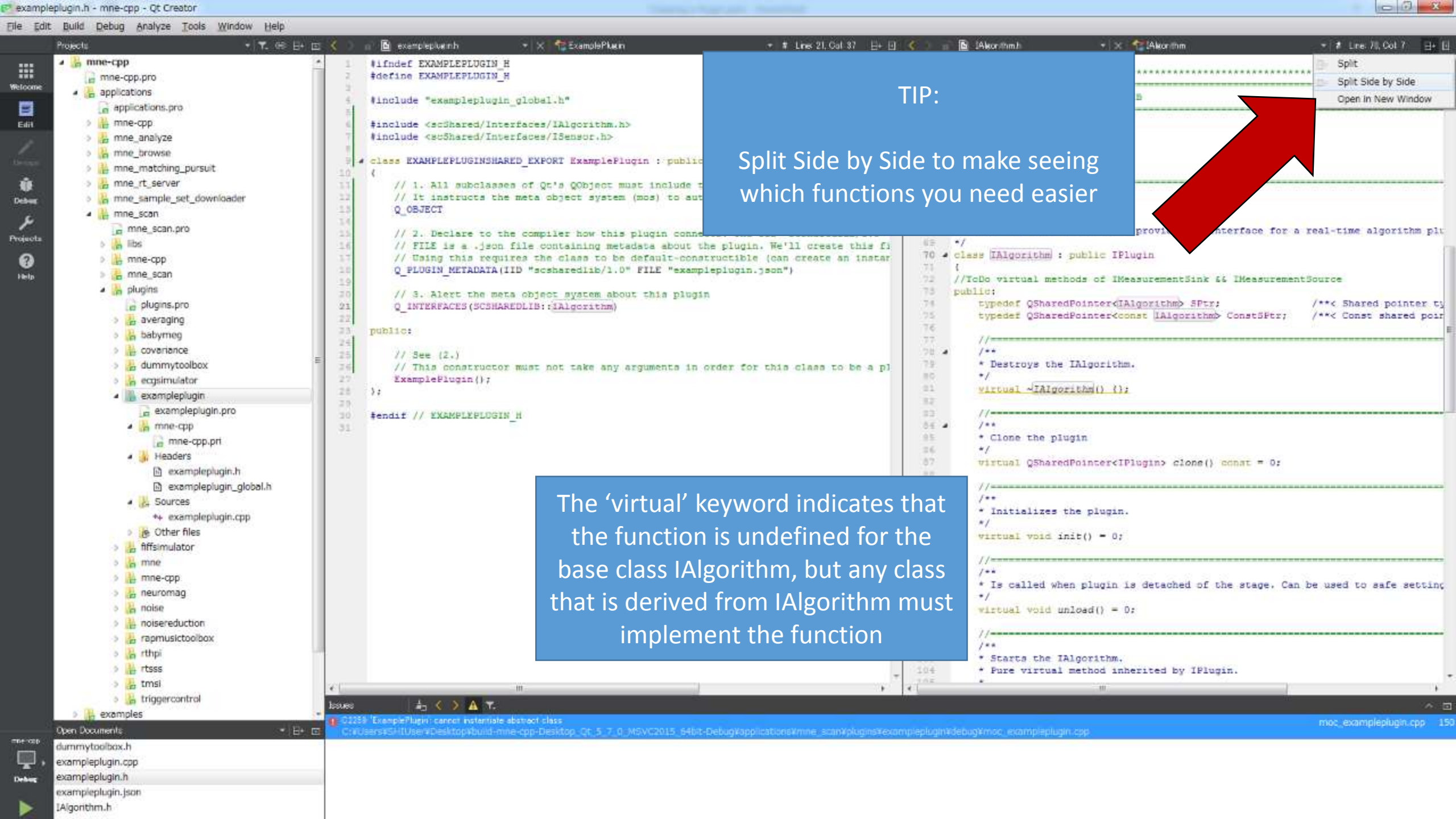
```
54
55  //*************************************************************************************************************
56  //=============================================================================================================
57  // DEFINE NAMESPACE SCSHAREDLIB
58  //=============================================================================================================
59
60  namespace SCSHAREDLIB
61  {
62
63
64  //=============================================================================================================
65  /**
66  * DECLARE CLASS IAlgorithm
67  *
68  * @brief The IAlgorithm class provides an interface for a real-time algorithm plugin.
69  */
70  class IAlgorithm : public IPlugin
71  {
72  //ToDo virtual methods of IMeasurementSink && IMeasurementSource
73  public:
74      typedef QSharedPointer<IAlgorithm> SPtr;              /**< Shared pointer type for IAlgorithm. */
75      typedef QSharedPointer<const IAlgorithm> ConstSPtr;   /**< Const shared pointer type for IAlgorithm. */
76
77      //=========================================================================================================
78      /**
79      * Destroys the IAlgorithm.
80      */
81      virtual ~IAlgorithm() {};
82
83      //=========================================================
84      /**
85      * Clone the plugin
86      */
87      virtual QSharedPointer<IPlugin> clone() const = 0;
88
89      //=========================================================
90      /**
91      * Initializes the plugin.
92      */
93      virtual void init() = 0;
94
95      //=========================================================================================================
96      /**
97      * Is called when plugin is detached of the stage. Can be used to safe settings.
98      */
99      virtual void unload() = 0;
100
101     //=========================================================================================================
102     /**
103     * Starts the IAlgorithm.
104     * Pure virtual method inherited by IPlugin.
105     *
106     * @return true if success, false otherwise
```

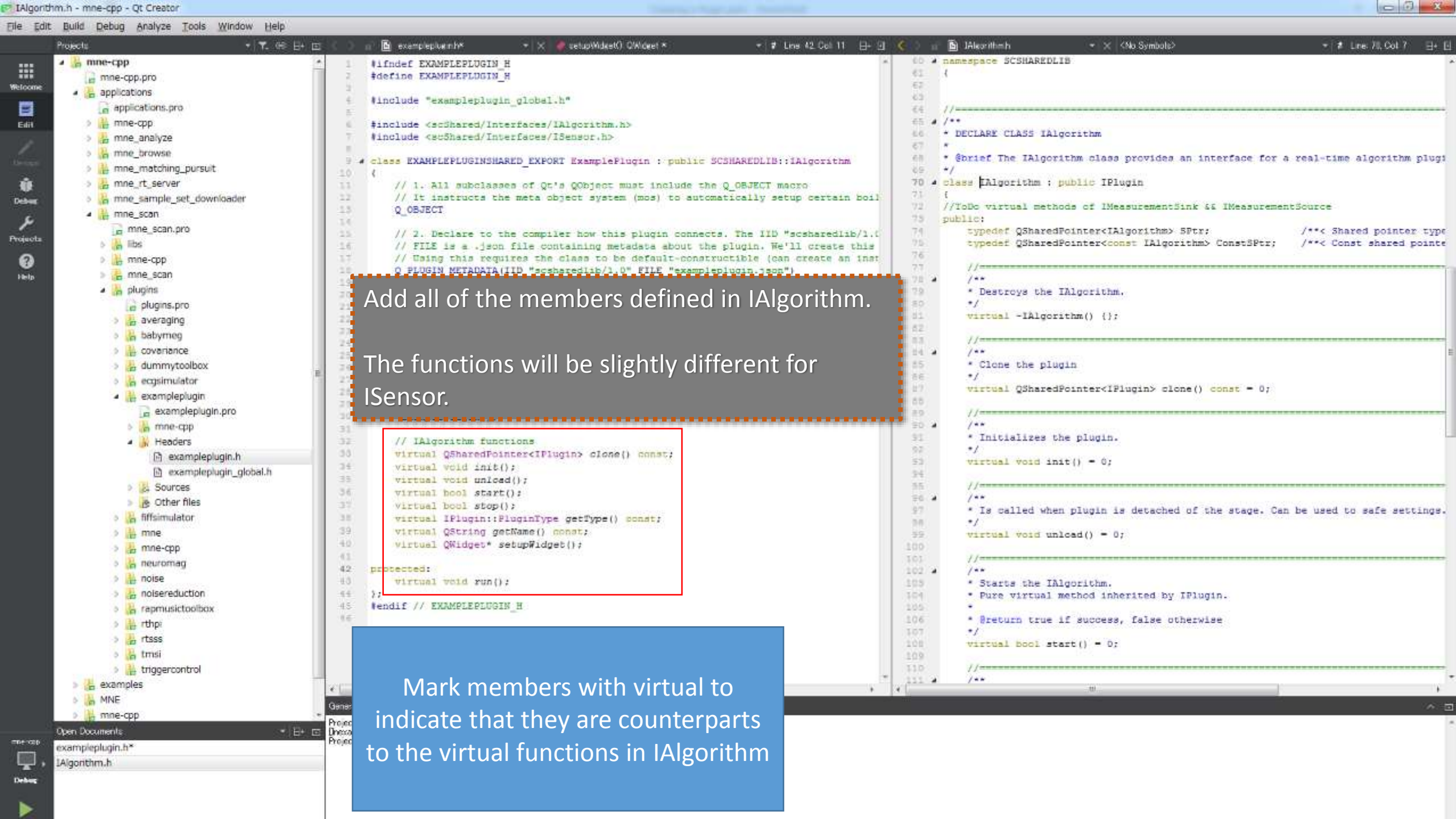
Hold Ctrl (or Cmnd) and click on IAlgorithm to jump to its header file and see its member functions.

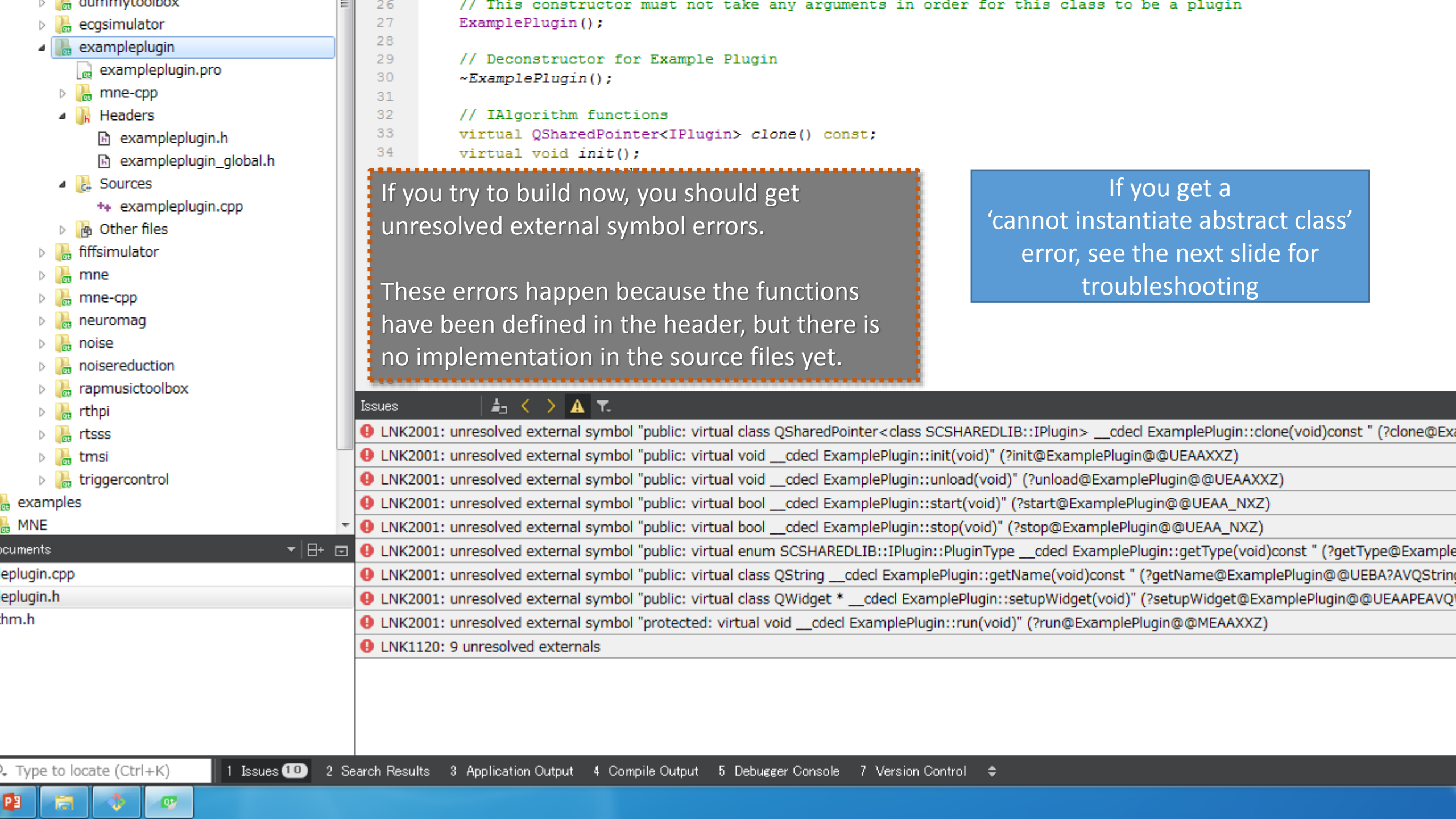
All of the public functions marked virtual must be implemented by our plugin before it will build.

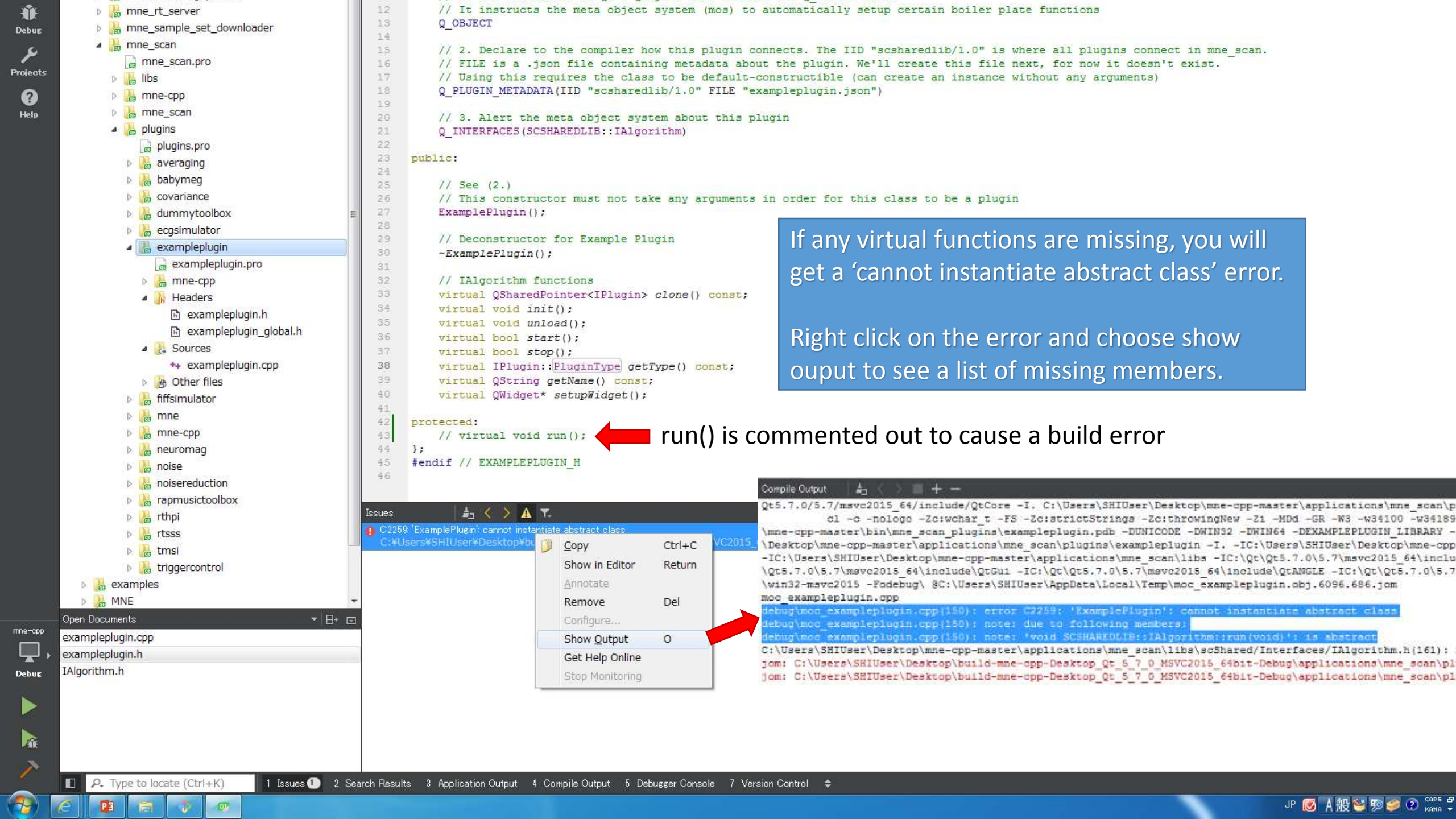
Issues

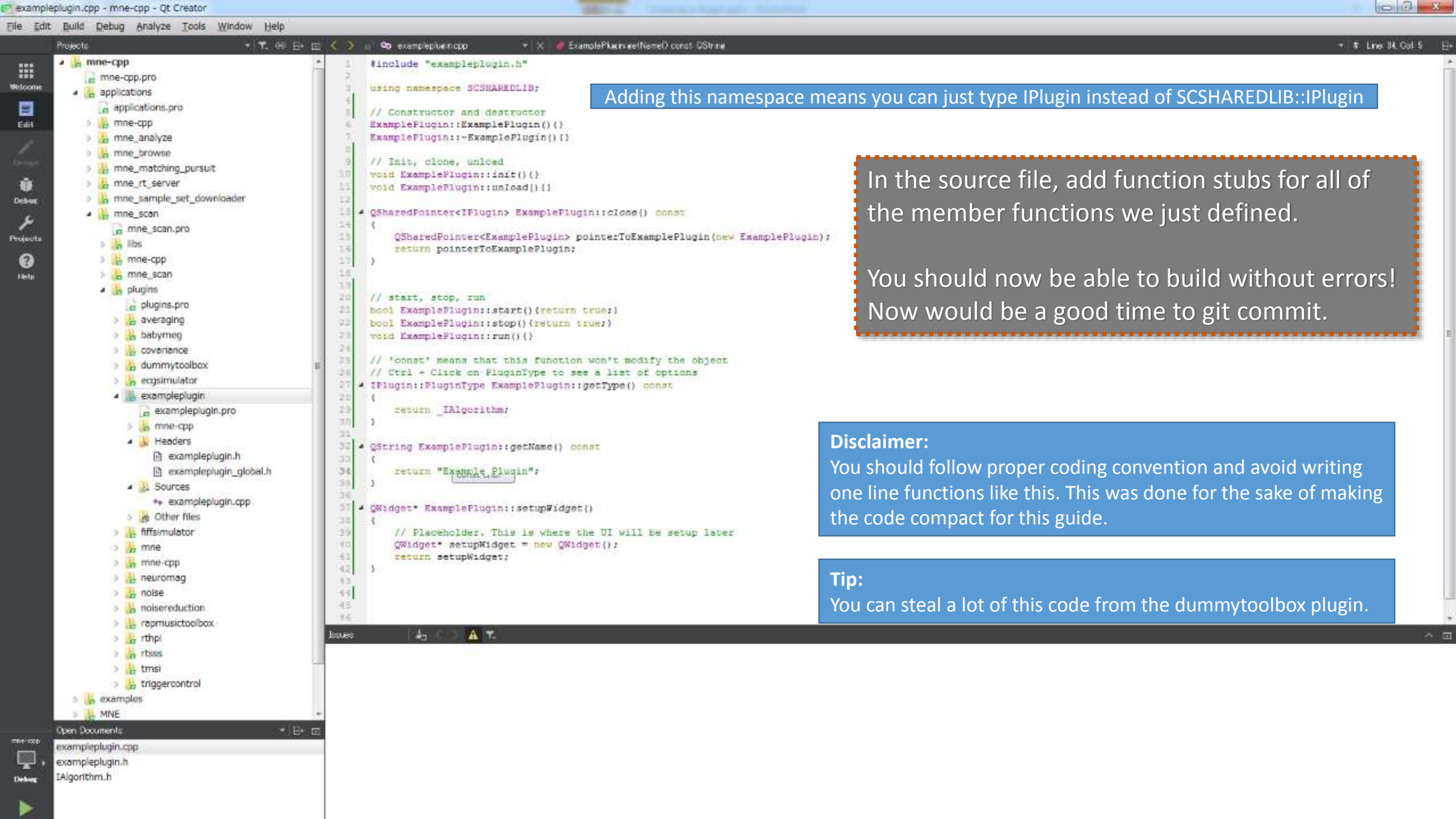
C2259: 'ExamplePlugin': cannot instantiate abstract class
C:¥Users¥SHIUser¥Desktop¥build-mne-cpp-Desktop_Qt_5_7_0_MSVC2015_64bit-Debug¥applications¥mne_scan¥plugins¥exampleplugin¥debug¥moc_exampleplugin.cpp

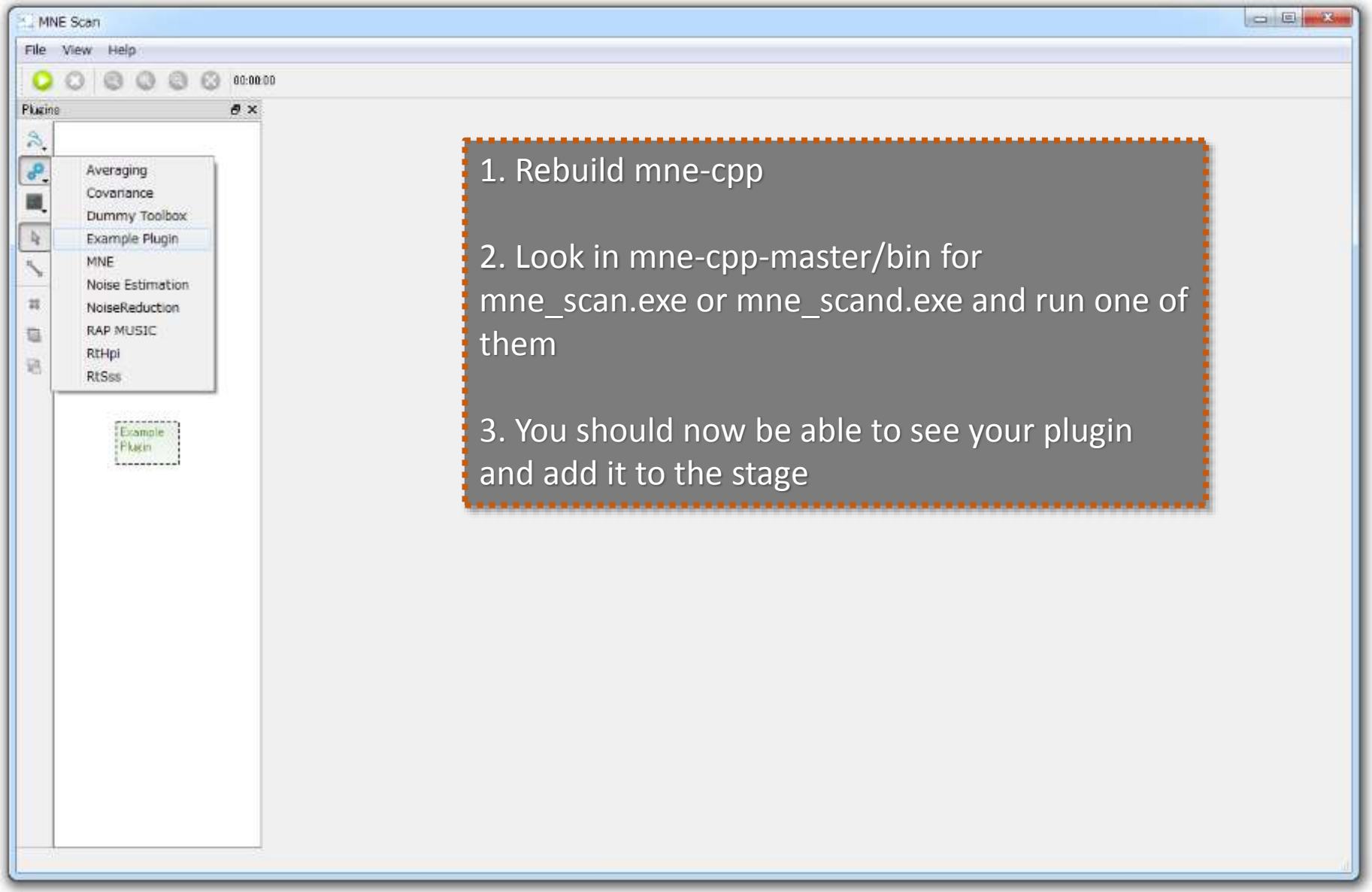

exampleplugin.h - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
mne-cpp.pri
Headers
exampleplugin.h
exampleplugin_global.h
Sources
exampleplugin.cpp
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
rtfpi
rtsss
tmsi
triggercontrol
examples
Open Documents
dummytoolbox.h
exampleplugin.cpp
exampleplugin.h
exampleplugin.json
IAlgorithm.h
#ifndef EXAMPLEPLUGIN_H
#define EXAMPLEPLUGIN_H
#include "exampleplugin_global.h"
#include <scShared/Interfaces/IAlgorithm.h>
#include <scShared/Interfaces/ISensor.h>
class EXAMPLEPLUGINSHARED_EXPORT ExamplePlugin : public
{
Q_OBJECT
Q_PLUGIN_METADATA(IID "scsharedlib/1.0" FILE "exampleplugin.json")
Q_INTERFACES(SCSHAREDLIB::IAlgorithm)
public:
ExamplePlugin();
};
#endif // EXAMPLEPLUGIN_H
class IAlgorithm : public IPlugin
typedef QSharedPointer<IAlgorithm> SPtr;
typedef QSharedPointer<const IAlgorithm> ConstSPtr;
virtual ~IAlgorithm() {};
virtual QSharedPointer<IPlugin> clone() const = 0;
virtual void init() = 0;
virtual void unload() = 0;
Issues
C2259: 'ExamplePlugin': cannot instantiate abstract class
moc_exampleplugin.cpp
Split
Split Side by Side
Open in New Window
TIP:
Split Side by Side to make seeing which functions you need easier
The ‘virtual’ keyword indicates that the function is undefined for the base class IAlgorithm, but any class that is derived from IAlgorithm must implement the function


IAlgorithm.h - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
exampleplugin.h
exampleplugin_global.h
Sources
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
rthpi
rtsss
tmsi
triggercontrol
examples
MNE
mne-cpp
Open Documents
exampleplugin.h*
IAlgorithm.h
#ifndef EXAMPLEPLUGIN_H
#define EXAMPLEPLUGIN_H
#include "exampleplugin_global.h"
#include <scShared/Interfaces/IAlgorithm.h>
#include <scShared/Interfaces/ISensor.h>
class EXAMPLEPLUGINSHARED_EXPORT ExamplePlugin : public SCSHAREDLIB::IAlgorithm
{
// 1. All subclasses of Qt's QObject must include the Q_OBJECT macro
// It instructs the meta object system (mos) to automatically setup certain boil
Q_OBJECT
// 2. Declare to the compiler how this plugin connects. The IID "scsharedlib/1.0
// FILE is a .json file containing metadata about the plugin. We'll create this
// Using this requires the class to be default-constructible (can create an inst
Q_PLUGIN_METADATA(IID "scsharedlib/1.0" FILE "exampleplugin.json")
// IAlgorithm functions
virtual QSharedPointer<IPlugin> clone() const;
virtual void init();
virtual void unload();
virtual bool start();
virtual bool stop();
virtual IPlugin::PluginType getType() const;
virtual QString getName() const;
virtual QWidget* setupWidget();
protected:
virtual void run();
};
#endif // EXAMPLEPLUGIN_H
Add all of the members defined in IAlgorithm.
The functions will be slightly different for ISensor.
Mark members with virtual to indicate that they are counterparts to the virtual functions in IAlgorithm
namespace SCSHAREDLIB
{
class IAlgorithm : public IPlugin
{
public:
typedef QSharedPointer<IAlgorithm> SPtr;
typedef QSharedPointer<const IAlgorithm> ConstSPtr;
virtual ~IAlgorithm() {};
virtual QSharedPointer<IPlugin> clone() const = 0;
virtual void init() = 0;
virtual void unload() = 0;
virtual bool start() = 0;


dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
exampleplugin.h
exampleplugin_global.h
Sources
exampleplugin.cpp
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
rthpi
rtsss
tmsi
triggercontrol
examples
MNE
// This constructor must not take any arguments in order for this class to be a plugin
ExamplePlugin();
// Deconstructor for Example Plugin
~ExamplePlugin();
// IAlgorithm functions
virtual QSharedPointer<IPlugin> clone() const;
virtual void init();
If you try to build now, you should get unresolved external symbol errors.
These errors happen because the functions have been defined in the header, but there is no implementation in the source files yet.
If you get a 'cannot instantiate abstract class' error, see the next slide for troubleshooting
Issues
LNK2001: unresolved external symbol "public: virtual class QSharedPointer<class SCSHAREDLIB::IPlugin> __cdecl ExamplePlugin::clone(void)const " (?clone@Ex
LNK2001: unresolved external symbol "public: virtual void __cdecl ExamplePlugin::init(void)" (?init@ExamplePlugin@@UEAAXXZ)
LNK2001: unresolved external symbol "public: virtual void __cdecl ExamplePlugin::unload(void)" (?unload@ExamplePlugin@@UEAAXXZ)
LNK2001: unresolved external symbol "public: virtual bool __cdecl ExamplePlugin::start(void)" (?start@ExamplePlugin@@UEAA_NXZ)
LNK2001: unresolved external symbol "public: virtual bool __cdecl ExamplePlugin::stop(void)" (?stop@ExamplePlugin@@UEAA_NXZ)
LNK2001: unresolved external symbol "public: virtual enum SCSHAREDLIB::IPlugin::PluginType __cdecl ExamplePlugin::getType(void)const " (?getType@Example
LNK2001: unresolved external symbol "public: virtual class QString __cdecl ExamplePlugin::getName(void)const " (?getName@ExamplePlugin@@UEBA?AVQString
LNK2001: unresolved external symbol "public: virtual class QWidget * __cdecl ExamplePlugin::setupWidget(void)" (?setupWidget@ExamplePlugin@@UEAAPEAVQ
LNK2001: unresolved external symbol "protected: virtual void __cdecl ExamplePlugin::run(void)" (?run@ExamplePlugin@@MEAAXXZ)
LNK1120: 9 unresolved externals
eplugin.cpp
eplugin.h
hm.h
Type to locate (Ctrl+K)
1 Issues 10
2 Search Results
3 Application Output
4 Compile Output
5 Debugger Console
7 Version Control


mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
exampleplugin.h
exampleplugin_global.h
Sources
exampleplugin.cpp
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
rthpi
rtsss
tmsi
triggercontrol
examples
MNE
Open Documents
exampleplugin.cpp
exampleplugin.h
IAlgorithm.h
// It instructs the meta object system (mos) to automatically setup certain boiler plate functions
Q_OBJECT
// 2. Declare to the compiler how this plugin connects. The IID "scsharedlib/1.0" is where all plugins connect in mne_scan.
// FILE is a .json file containing metadata about the plugin. We'll create this file next, for now it doesn't exist.
// Using this requires the class to be default-constructible (can create an instance without any arguments)
Q_PLUGIN_METADATA(IID "scsharedlib/1.0" FILE "exampleplugin.json")
// 3. Alert the meta object system about this plugin
Q_INTERFACES(SCSHAREDLIB::IAlgorithm)
public:
// See (2.)
// This constructor must not take any arguments in order for this class to be a plugin
ExamplePlugin();
// Deconstructor for Example Plugin
~ExamplePlugin();
// IAlgorithm functions
virtual QSharedPointer<IPlugin> clone() const;
virtual void init();
virtual void unload();
virtual bool start();
virtual bool stop();
virtual IPlugin::PluginType getType() const;
virtual QString getName() const;
virtual QWidget* setupWidget();
protected:
// virtual void run();
};
#endif // EXAMPLEPLUGIN_H
If any virtual functions are missing, you will get a ‘cannot instantiate abstract class’ error.
Right click on the error and choose show ouput to see a list of missing members.
run() is commented out to cause a build error
Issues
C2259: 'ExamplePlugin': cannot instantiate abstract class
Copy Ctrl+C
Show in Editor Return
Annotate
Remove Del
Configure...
Show Output O
Get Help Online
Stop Monitoring
Compile Output
Qt5.7.0/5.7/msvc2015_64/include/QtCore -I. C:\Users\SHIUser\Desktop\mne-cpp-master\applications\mne_scan\p
cl -c -nologo -Zc:wchar_t -FS -Zc:strictStrings -Zc:throwingNew -Zi -MDd -GR -W3 -w34100 -w34189
\mne-cpp-master\bin\mne_scan_plugins\exampleplugin.pdb -DUNICODE -DWIN32 -DWIN64 -DEXAMPLEPLUGIN_LIBRARY -
\Desktop\mne-cpp-master\applications\mne_scan\plugins\exampleplugin -I. -IC:\Users\SHIUser\Desktop\mne-cpp
-IC:\Users\SHIUser\Desktop\mne-cpp-master\applications\mne_scan\libs -IC:\Qt\Qt5.7.0\5.7\msvc2015_64\inclu
\Qt5.7.0\5.7\msvc2015_64\include\QtGui -IC:\Qt\Qt5.7.0\5.7\msvc2015_64\include\QtANGLE -IC:\Qt\Qt5.7.0\5.7
\win32-msvc2015 -Fodebug\ @C:\Users\SHIUser\AppData\Local\Temp\moc_exampleplugin.obj.6096.686.jom
moc_exampleplugin.cpp
debug\moc_exampleplugin.cpp(150): error C2259: 'ExamplePlugin': cannot instantiate abstract class
debug\moc_exampleplugin.cpp(150): note: due to following members:
debug\moc_exampleplugin.cpp(150): note: 'void SCSHAREDLIB::IAlgorithm::run(void)': is abstract
C:\Users\SHIUser\Desktop\mne-cpp-master\applications\mne_scan\libs\scShared/Interfaces/IAlgorithm.h(161):
jom: C:\Users\SHIUser\Desktop\build-mne-cpp-Desktop_Qt_5_7_0_MSVC2015_64bit-Debug\applications\mne_scan\pl
jom: C:\Users\SHIUser\Desktop\build-mne-cpp-Desktop_Qt_5_7_0_MSVC2015_64bit-Debug\applications\mne_scan\pl
Type to locate (Ctrl+K)
1 Issues
2 Search Results
3 Application Output
4 Compile Output
5 Debugger Console
7 Version Control


exampleplugin.cpp - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
exampleplugin.h
exampleplugin_global.h
Sources
exampleplugin.cpp
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
rtfpl
rtsss
tmsi
triggercontrol
examples
MNE
Open Documents
exampleplugin.cpp
exampleplugin.h
IAlgorithm.h
Welcome
Edit
Debug
Projects
Help
Debug
Issues
#include "exampleplugin.h"
using namespace SCSHAREDLIB;
// Constructor and destructor
ExamplePlugin::ExamplePlugin(){}
ExamplePlugin::~ExamplePlugin(){}
// Init, clone, unload
void ExamplePlugin::init(){}
void ExamplePlugin::unload(){}
QSharedPointer<IPlugin> ExamplePlugin::clone() const
{
QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
return pointerToExamplePlugin;
}
// start, stop, run
bool ExamplePlugin::start(){return true;}
bool ExamplePlugin::stop(){return true;}
void ExamplePlugin::run(){}
// 'const' means that this function won't modify the object
// Ctrl + Click on PluginType to see a list of options
IPlugin::PluginType ExamplePlugin::getType() const
{
return _IAlgorithm;
}
QString ExamplePlugin::getName() const
{
return "Example Plugin";
}
QWidget* ExamplePlugin::setupWidget()
{
// Placeholder. This is where the UI will be setup later
QWidget* setupWidget = new QWidget();
return setupWidget;
}
Adding this namespace means you can just type IPlugin instead of SCSHAREDLIB::IPlugin
In the source file, add function stubs for all of the member functions we just defined.
You should now be able to build without errors!
Now would be a good time to git commit.
Disclaimer:
You should follow proper coding convention and avoid writing one line functions like this. This was done for the sake of making the code compact for this guide.
Tip:
You can steal a lot of this code from the dummytoolbox plugin.

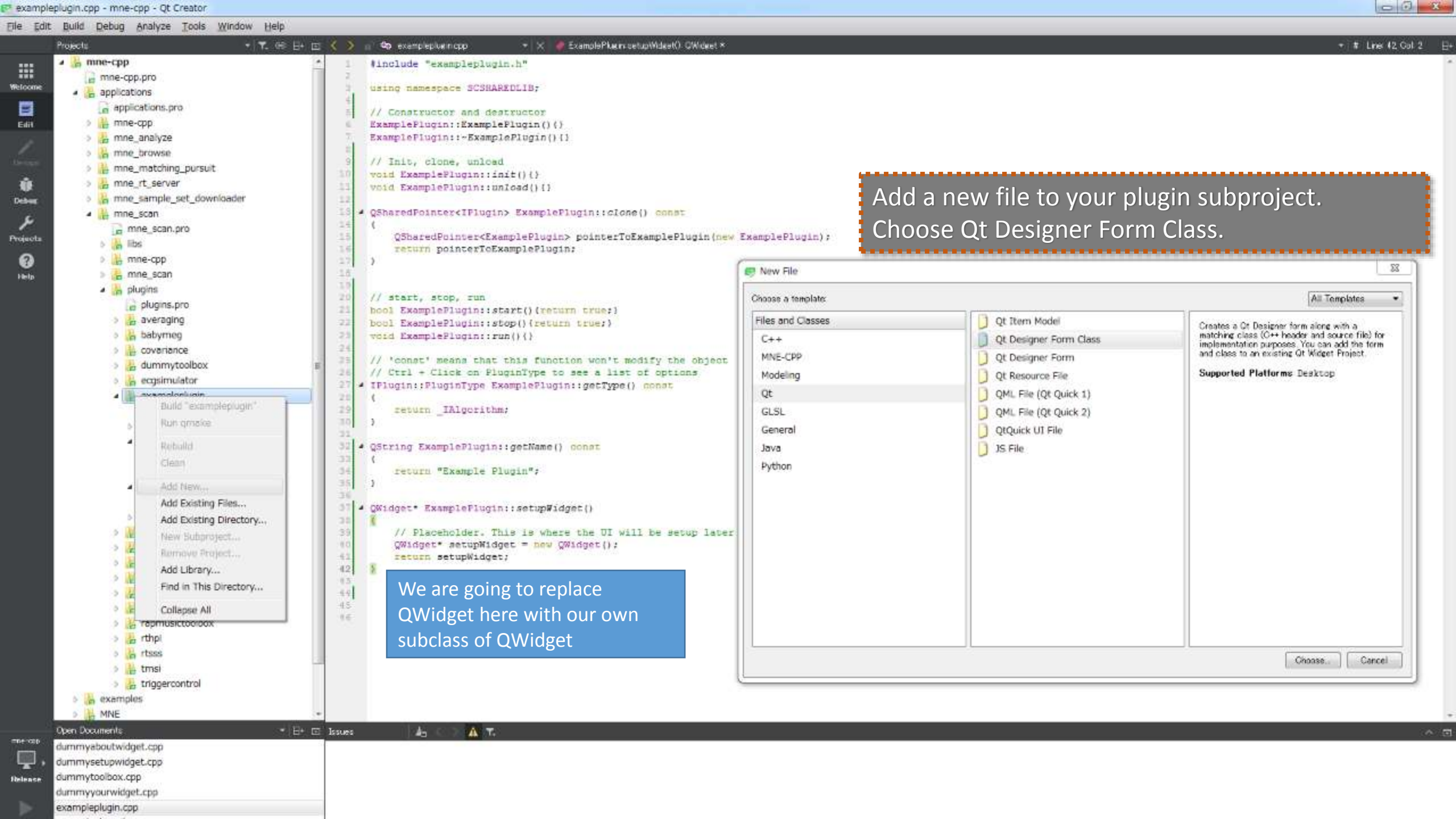


1. Rebuild mne-cpp

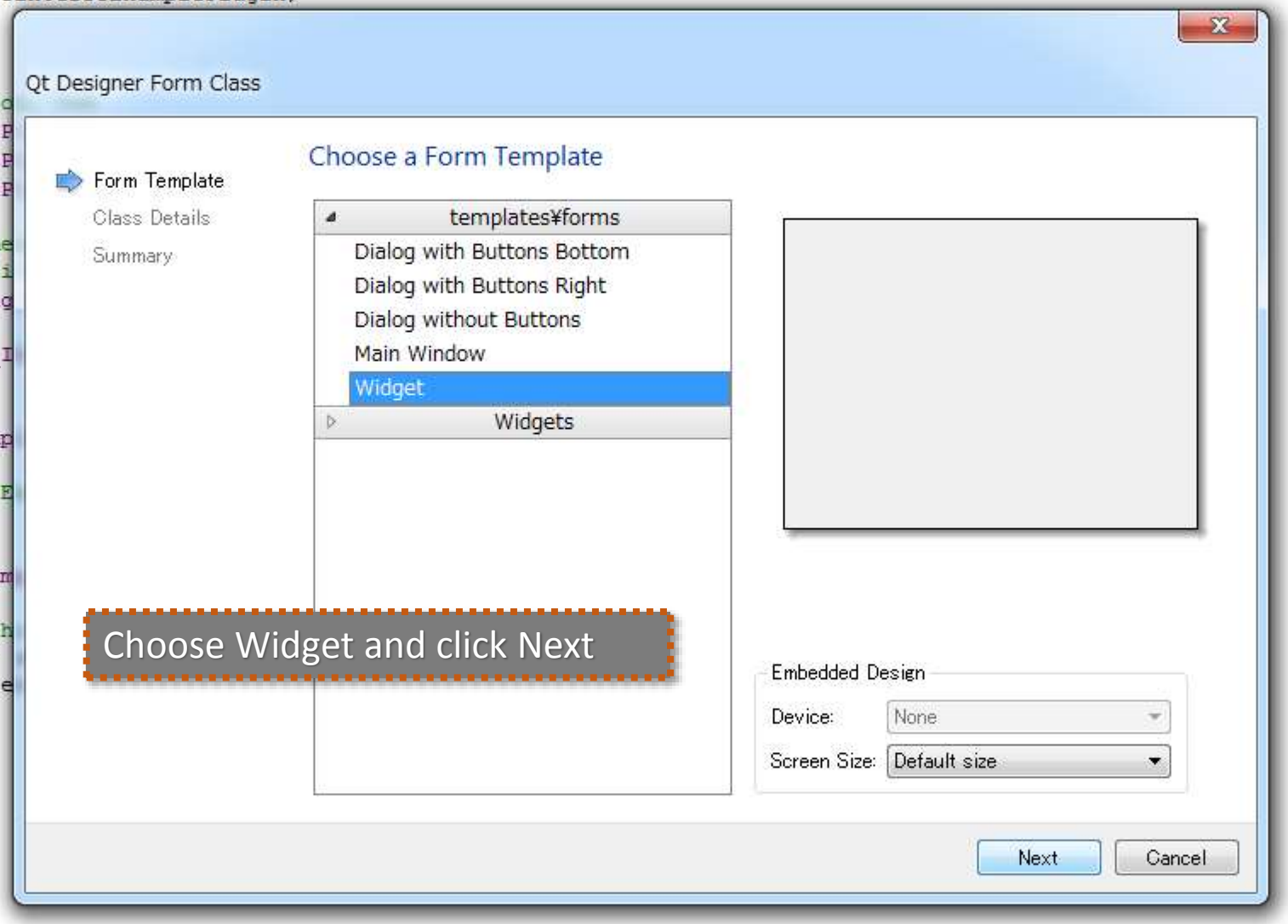
2. Look in mne-cpp-master/bin for mne_scan.exe or mne_scand.exe and run one of them

3. You should now be able to see your plugin and add it to the stage

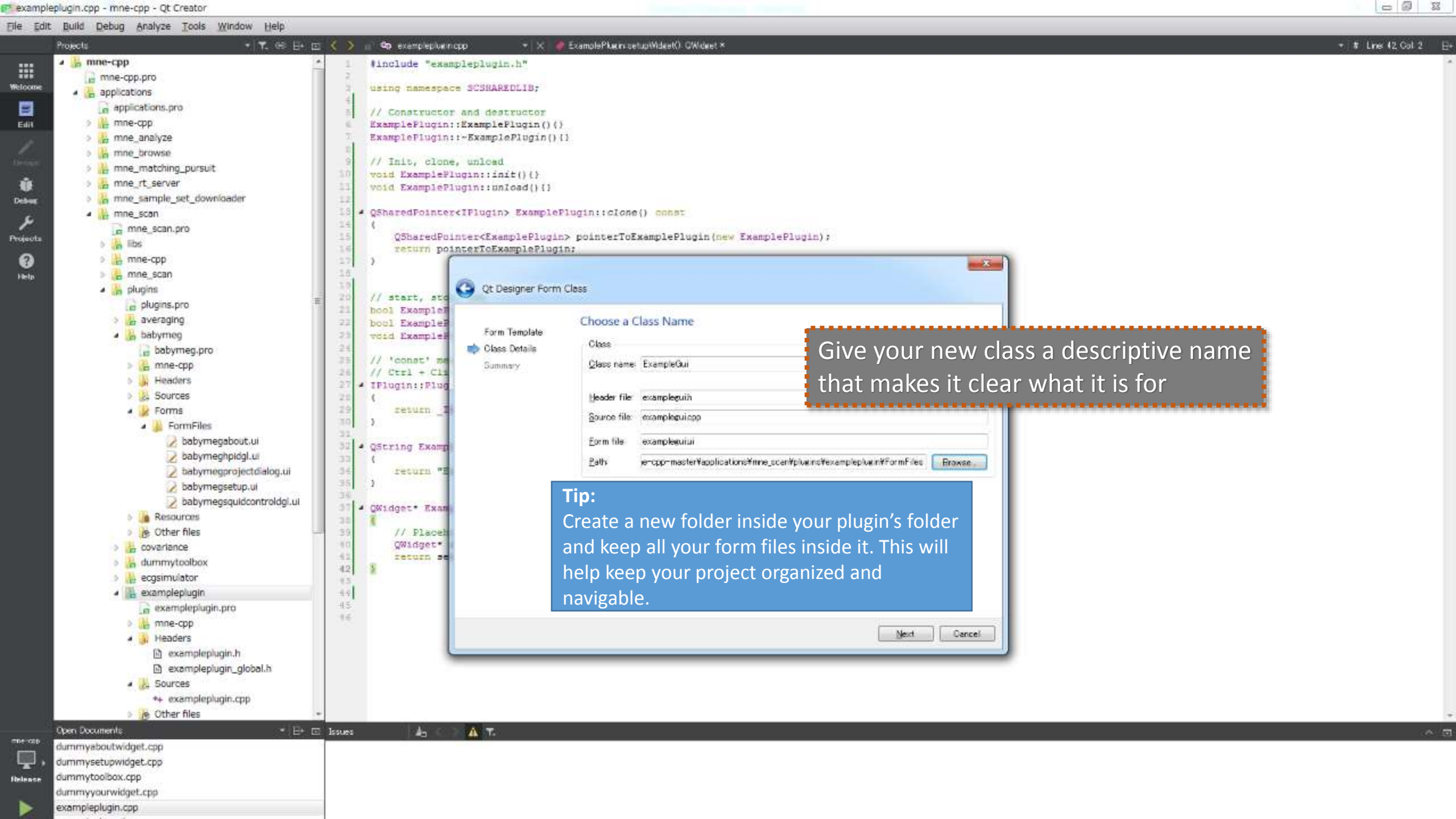
Adding a GUI

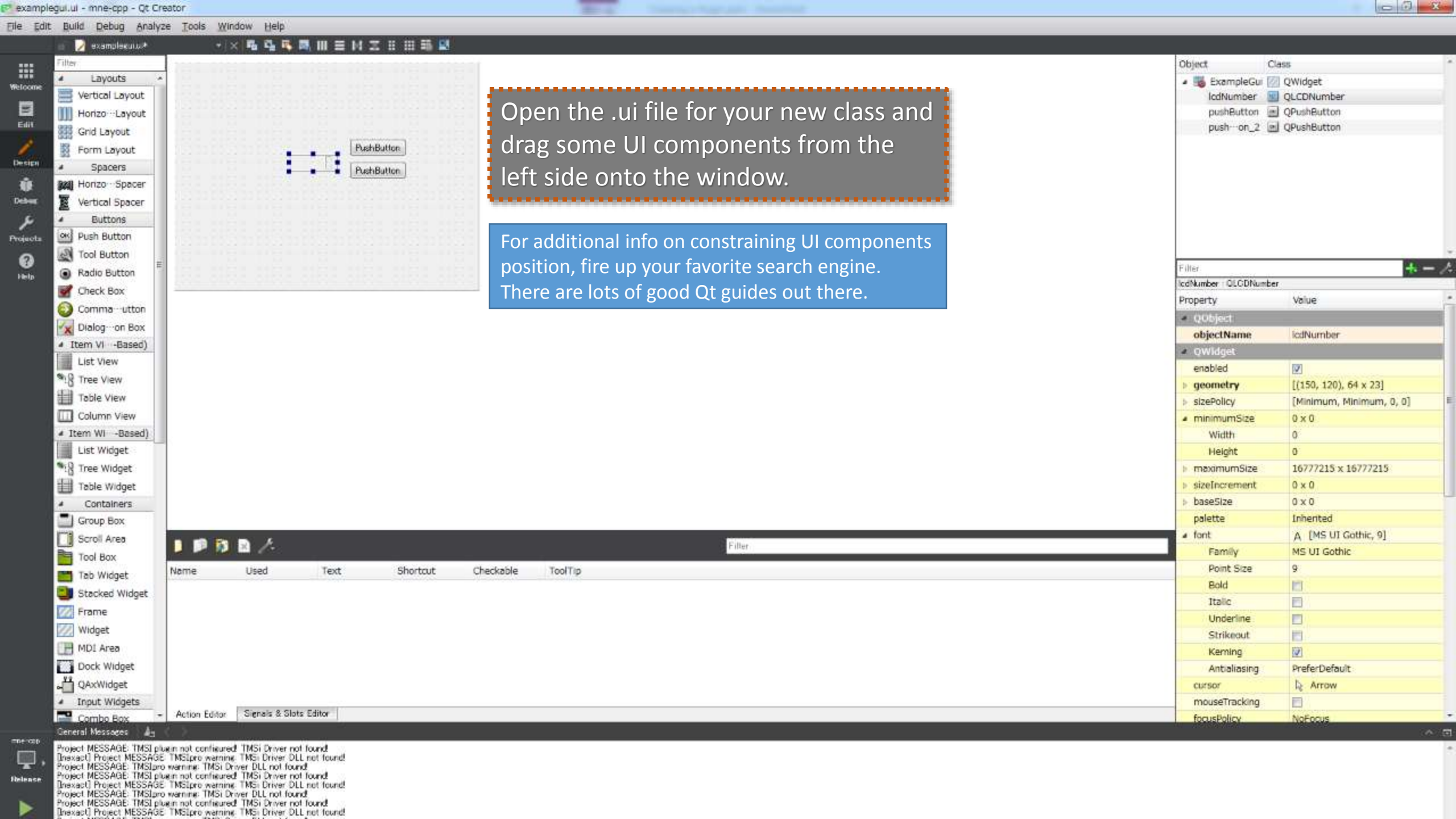

exampleplugin.cpp - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
Build "exampleplugin"
Run qmake
Rebuild
Clean
Add New...
Add Existing Files...
Add Existing Directory...
New Subproject...
Remove Project...
Add Library...
Find in This Directory...
Collapse All
rtmsi
rtsss
tmsi
triggercontrol
examples
MNE
Open Documents
dummyaboutwidget.cpp
dummysetupwidget.cpp
dummytoolbox.cpp
dummyyourwidget.cpp
exampleplugin.cpp
Issues
#include "exampleplugin.h"
using namespace SCSHAREDLIB;
// Constructor and destructor
ExamplePlugin::ExamplePlugin(){}
ExamplePlugin::~ExamplePlugin(){}
// Init, clone, unload
void ExamplePlugin::init(){}
void ExamplePlugin::unload(){}
QSharedPointer<IPlugin> ExamplePlugin::clone() const
{
QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
return pointerToExamplePlugin;
}
// start, stop, run
bool ExamplePlugin::start(){return true;}
bool ExamplePlugin::stop(){return true;}
void ExamplePlugin::run(){}
// 'const' means that this function won't modify the object
// Ctrl + Click on PluginType to see a list of options
IPlugin::PluginType ExamplePlugin::getType() const
{
return _IAlgorithm;
}
QString ExamplePlugin::getName() const
{
return "Example Plugin";
}
QWidget* ExamplePlugin::setupWidget()
{
// Placeholder. This is where the UI will be setup later
QWidget* setupWidget = new QWidget();
return setupWidget;
}
We are going to replace QWidget here with our own subclass of QWidget
Add a new file to your plugin subproject.
Choose Qt Designer Form Class.
New File
Choose a template:
All Templates
Files and Classes
C++
MNE-CPP
Modeling
Qt
GLSL
General
Java
Python
Qt Item Model
Qt Designer Form Class
Qt Designer Form
Qt Resource File
QML File (Qt Quick 1)
QML File (Qt Quick 2)
QtQuick UI File
JS File
Creates a Qt Designer form along with a matching class (C++ header and source file) for implementation purposes. You can add the form and class to an existing Qt Widget Project.
Supported Platforms: Desktop
Choose...
Cancel

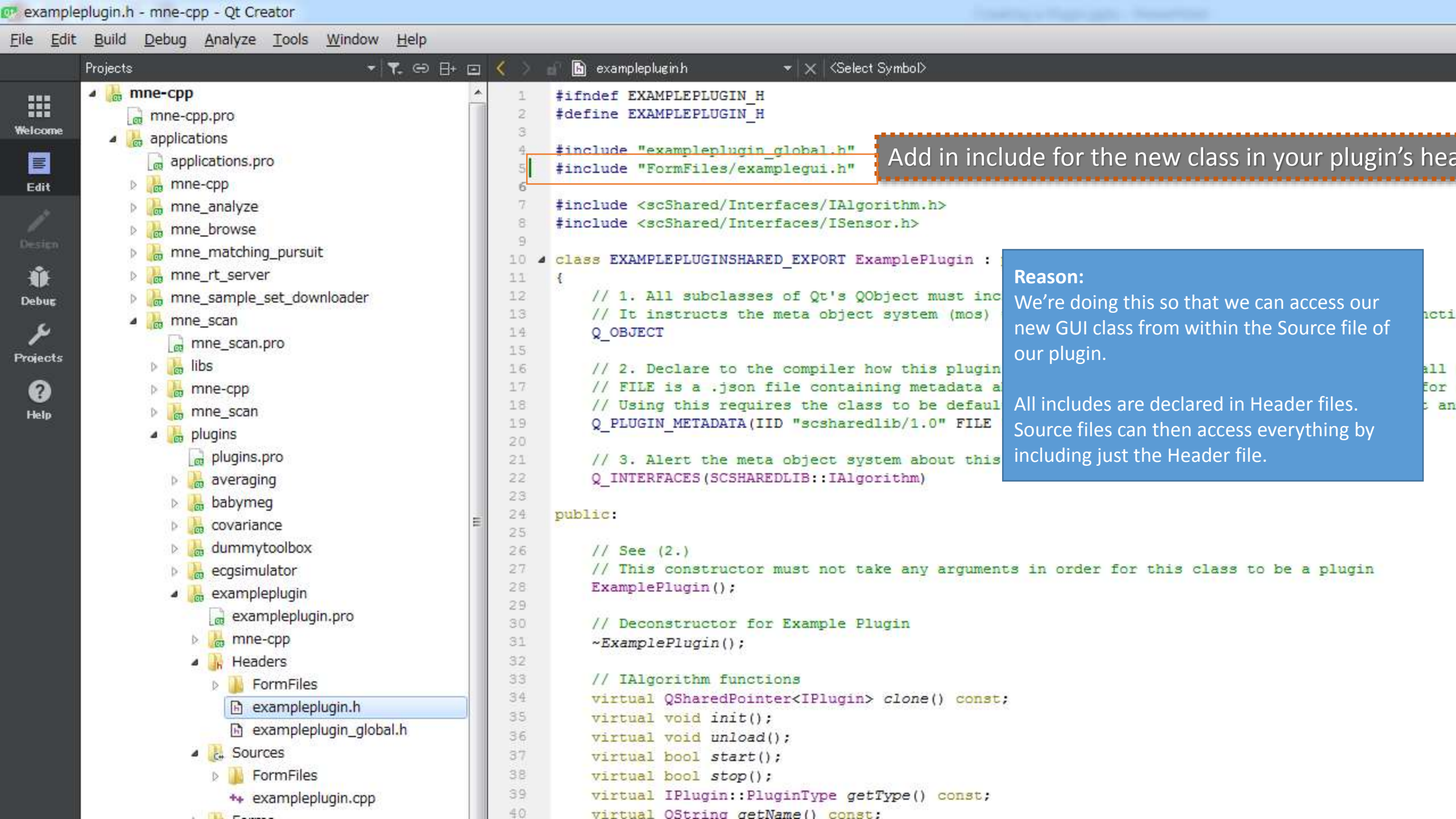
```
12
13  QSharedPointer<IPlugin> ExamplePlugin::clone() const
14  {
15      QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
16      return pointerToExamplePlugin;
17  }
18
19
20  // start, sto
21  bool ExampleP
22  bool ExampleP
23  void ExampleP
24
25  // 'const' me
26  // Ctrl + Cli
27  IPlugin::Plug
28  {
29      return _I
30  }
31
32  QString Examp
33  {
34      return "E
35  }
36
37  QWidget* Exam
38  {
39      // Placeh
40      QWidget*
41      return se
42  }
43
44
45
46
```



Choose Widget and click Next


exampleplugin.cpp - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
babymeg.pro
mne-cpp
Headers
Sources
Forms
FormFiles
babymegabout.ui
babymeghpidgl.ui
babymegprojectdialog.ui
babymegsetup.ui
babymegsquidcontroldgl.ui
Resources
Other files
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
exampleplugin.h
exampleplugin_global.h
Sources
exampleplugin.cpp
Other files
#include "exampleplugin.h"
using namespace SCSHAREDLIB;
// Constructor and destructor
ExamplePlugin::ExamplePlugin()()
ExamplePlugin::~ExamplePlugin()()
// Init, clone, unload
void ExamplePlugin::init()()
void ExamplePlugin::unload()()
QSharedPointer<IPlugin> ExamplePlugin::clone() const
{
QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
return pointerToExamplePlugin;
}
Qt Designer Form Class
Choose a Class Name
Form Template
Class Details
Summary
Class
Class name: ExampleGui
Header file: exampleguih
Source file: examplegui.cpp
Form file: exampleguiui
Path:
Browse...
Next
Cancel
Give your new class a descriptive name that makes it clear what it is for
Tip:
Create a new folder inside your plugin's folder and keep all your form files inside it. This will help keep your project organized and navigable.
Open Documents
dummyaboutwidget.cpp
dummysetupwidget.cpp
dummytoolbox.cpp
dummyyourwidget.cpp
exampleplugin.cpp
Issues


examplegui.ui - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Welcome
Edit
Design
Debug
Projects
Help
Filter
Layouts
Vertical Layout
Horizo···Layout
Grid Layout
Form Layout
Spacers
Horizo···Spacer
Vertical Spacer
Buttons
Push Button
Tool Button
Radio Button
Check Box
Comma···utton
Dialog···on Box
Item Vi···-Based)
List View
Tree View
Table View
Column View
Item Wi···-Based)
List Widget
Tree Widget
Table Widget
Containers
Group Box
Scroll Area
Tool Box
Tab Widget
Stacked Widget
Frame
Widget
MDI Area
Dock Widget
QAxWidget
Input Widgets
Combo Box
PushButton
PushButton
Open the .ui file for your new class and drag some UI components from the left side onto the window.
For additional info on constraining UI components position, fire up your favorite search engine. There are lots of good Qt guides out there.
Object
Class
ExampleGui
QWidget
lcdNumber
QLCDNumber
pushButton
QPushButton
push···on_2
QPushButton
Filter
lcdNumber : QLCDNumber
Property
Value
QObject
objectName
lcdNumber
QWidget
enabled
geometry
[(150, 120), 64 x 23]
sizePolicy
[Minimum, Minimum, 0, 0]
minimumSize
0 x 0
Width
0
Height
0
maximumSize
16777215 x 16777215
sizeIncrement
0 x 0
baseSize
0 x 0
palette
Inherited
font
[MS UI Gothic, 9]
Family
MS UI Gothic
Point Size
9
Bold
Italic
Underline
Strikeout
Kerning
Antialiasing
PreferDefault
cursor
Arrow
mouseTracking
focusPolicy
NoFocus
Filter
Name
Used
Text
Shortcut
Checkable
ToolTip
Action Editor
Signals & Slots Editor
General Messages
Project MESSAGE: TMSi plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSi plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSi plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
mne-cpp
Release


exampleplugin.h - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
exampleplugin.h
<Select Symbol>
Welcome
Edit
Design
Debug
Projects
Help
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
FormFiles
exampleplugin.h
exampleplugin_global.h
Sources
FormFiles
exampleplugin.cpp
#ifndef EXAMPLEPLUGIN_H
#define EXAMPLEPLUGIN_H
#include "exampleplugin_global.h"
#include "FormFiles/examplegui.h"
#include <scShared/Interfaces/IAlgorithm.h>
#include <scShared/Interfaces/ISensor.h>
class EXAMPLEPLUGINSHARED_EXPORT ExamplePlugin :
{
// 1. All subclasses of Qt's QObject must inc
// It instructs the meta object system (mos)
Q_OBJECT
// 2. Declare to the compiler how this plugin
// FILE is a .json file containing metadata a
// Using this requires the class to be defaul
Q_PLUGIN_METADATA(IID "scsharedlib/1.0" FILE
// 3. Alert the meta object system about this
Q_INTERFACES(SCSHAREDLIB::IAlgorithm)
public:
// See (2.)
// This constructor must not take any arguments in order for this class to be a plugin
ExamplePlugin();
// Deconstructor for Example Plugin
~ExamplePlugin();
// IAlgorithm functions
virtual QSharedPointer<IPlugin> clone() const;
virtual void init();
virtual void unload();
virtual bool start();
virtual bool stop();
virtual IPlugin::PluginType getType() const;
virtual QString getName() const;
Add in include for the new class in your plugin's hea
Reason:
We're doing this so that we can access our new GUI class from within the Source file of our plugin.
All includes are declared in Header files. Source files can then access everything by including just the Header file.

```
28  {
29      return _IAlgorithm;
30  }
31
32  QString ExamplePlugin::getName() const
33  {
34      return "Example Plugin";
35  }
36
37  QWidget* ExamplePlugin::setupWidget()
38  {
39      // Setup the UI using our custom form class here
40      ExampleGUI* setupWidget = new ExampleGUI();
41      return setupWidget;
42  }
43
44
45
46
```

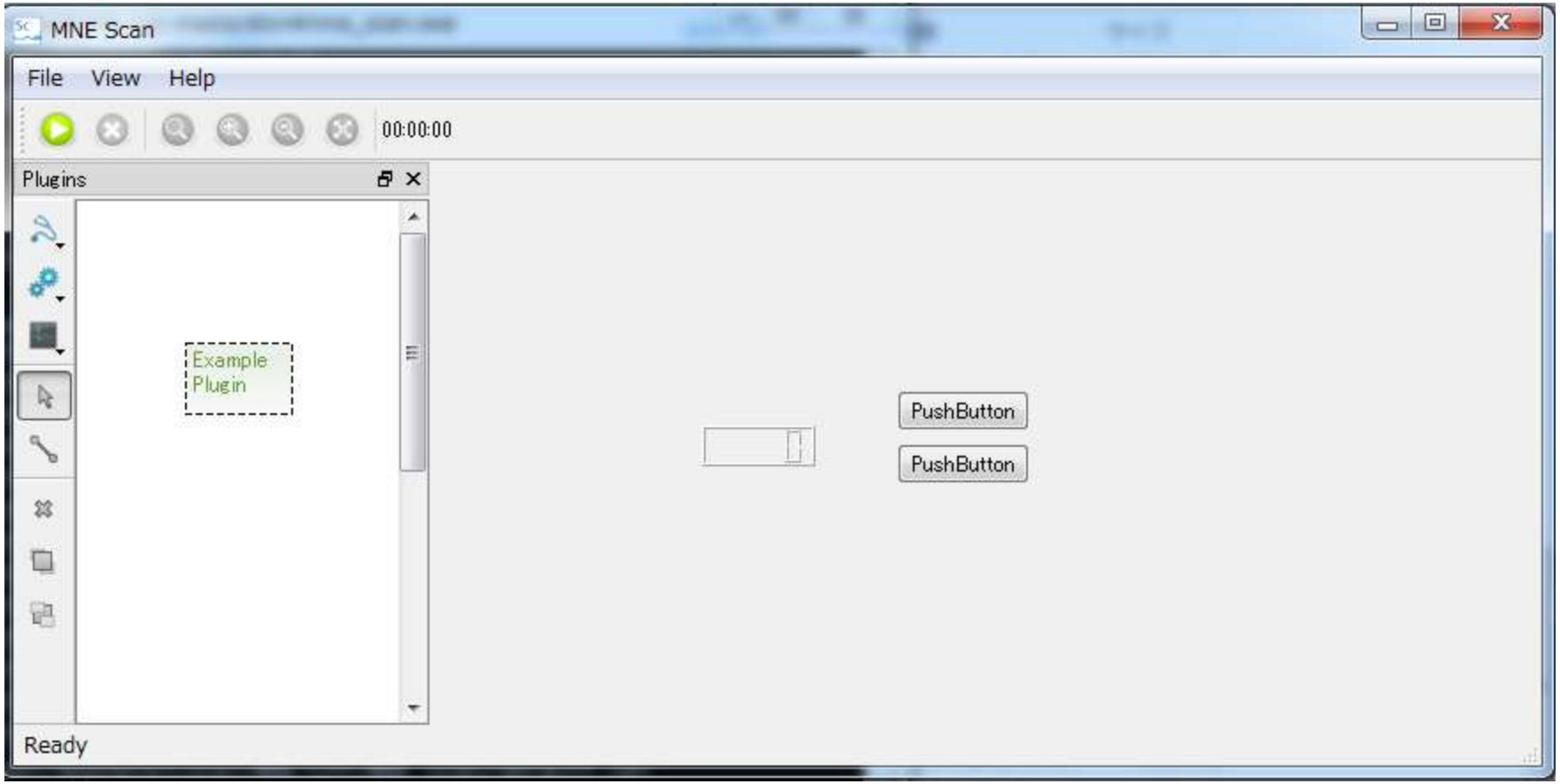
Update the setupWidget() method to return an instance of the class we just created, and try building again

General Messages

```
Project MESSAGE: TMSI plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSI plugin not configured! TMSi Driver not found!
[Inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
```

2 Search Results 3 Application Output 4 Compile Output 5 Debugger Console 7 Version Control

JP



Run mne_scan.exe, drop your plugin on the stage, and now you should see your first GUI.

examplegui.ui - mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

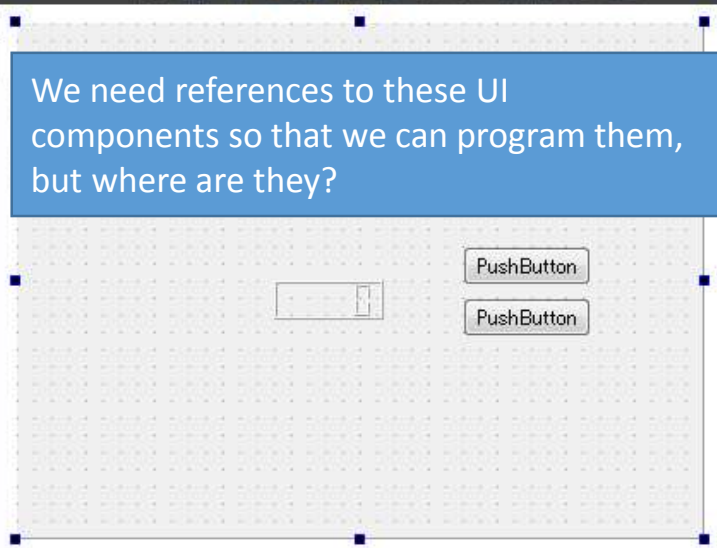
examplegui.ui

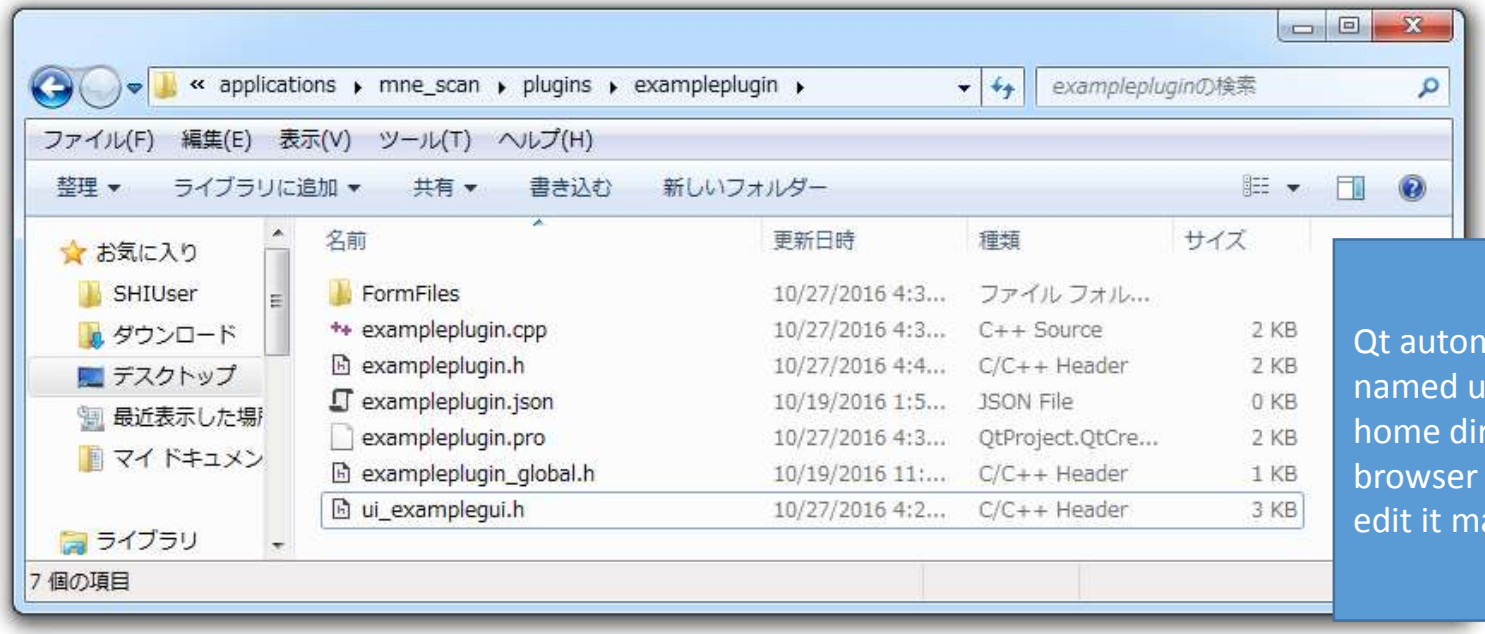
Welcome
Edit
Design
Debug
Projects
Help

Filter

- Layouts
  - Vertical Layout
  - Horizo···Layout
  - Grid Layout
  - Form Layout
- Spacers
  - Horizo···Spacer
  - Vertical Spacer
- Buttons
  - Push Button
  - Tool Button
  - Radio Button
  - Check Box
  - Comma···utton
  - Dialog···on Box
- Item Vi···-Based)
  - List View
  - Tree View
  - Table View
  - Column View
- Item Wi···-Based)
  - List Widget
  - Tree Widget
  - Table Widget
- Containers
  - Group Box
  - Scroll Area
  - Tool Box
  - Tab Widget
  - Stacked Widget
  - Frame
  - Widget

We need references to these UI components so that we can program them, but where are they?





Qt automatically generates a header file named ui_yourlcassname.h in your project's home directory. It doesn't appear in Qt's file browser because you're not intended to edit it manually

examplegui.h - mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

- plugins
  - plugins.pro
  - averaging
  - babymeg
  - covariance
  - dummytoolbox
    - dummytoolbox.pro
    - mne-cpp
    - Headers
      - FormFiles
        - dummyaboutwidget.h
        - dummysetupwidget.h
        - dummyyourwidget.h
      - dummytoolbox.h
      - dummytoolbox_global.h
    - Sources
      - FormFiles
        - dummyaboutwidget.cpp
        - dummysetupwidget.cpp
        - dummyyourwidget.cpp
      - dummytoolbox.cpp
    - Forms
      - FormFiles
        - dummyabout.ui
        - dummysetup.ui
        - dummyyourtoolbarwidget.ui
      - Other files
  - ecgsimulator
  - exampleplugin
    - exampleplugin.pro
    - mne-cpp
    - Headers

Welcome
Edit
Design
Debug
Projects
Help

exampleguih* ~ExampleGUI()

```
1  #ifndef EXAMPLEGUI_H
2  #define EXAMPLEGUI_H
3
4  #include <QWidget>
5
6  namespace Ui {
7  class ExampleGUI;
8  }
9
10 class ExampleGUI : public QWidget
11 {
12     Q_OBJECT
13
14 public:
15     explicit ExampleGUI(QWidget *parent = 0);
16     ~ExampleGUI();
17
18 private:
19
20     // Control + Clock on ExampleGUI to see
21     // the code generated from the designer file we made
22     Ui::ExampleGUI *ui;
23 };
24
25 #endif // EXAMPLEGUI_H
26
```

Return to your UI's Header file and ctrl + click on ExampleGUI

```
26      QPushButton *pushButton;
27      QPushButton *pushButton_2;
28      QLCDNumber *lcdNumber;
29
30      void setupUi(QWidget *ExampleGUI)
31      {
32          if (ExampleGUI->objectName().isEmpty())
33              ExampleGUI->setObjectName(QStringLiteral("ExampleGUI"));
34          ExampleGUI->resize(400, 300);
35          pushButton = new QPushButton(ExampleGUI);
36          pushButton->setObjectName(QStringLiteral("pushButton"));
37          pushButton->setGeometry(QRect(260, 130, 75, 23));
38          pushButton_2 = new QPushButton(ExampleGUI);
39          pushButton_2->setObjectName(QStringLiteral("pushButton_2"));
40          pushButton_2->setGeometry(QRect(260, 160, 75, 23));
41          lcdNumber = new QLCDNumber(ExampleGUI);
42          lcdNumber->setObjectName(QStringLiteral("lcdNumber"));
43          lcdNumber->setGeometry(QRect(150, 150, 64, 23));
44
45          retranslateUi(ExampleGUI);
46
47          QMetaObject::connectSlotsByName(ExampleGUI);
48      } // setupUi
49
50      void retranslateUi(QWidget *ExampleGUI)
51      {
52          ExampleGUI->setWindowTitle(QApplication::translate("ExampleGUI", "Form", 0));
53          pushButton->setText(QApplication::translate("ExampleGUI", "PushButton", 0));
54          pushButton_2->setText(QApplication::translate("ExampleGUI", "PushButton", 0));
55      } // retranslateUi
56
57  };
58
59  namespace Ui {
60      class ExampleGUI: public Ui_ExampleGUI {};
61  } // namespace Ui
62
63  QT_END_NAMESPACE
64
65  #endif // UI_EXAMPLEGUI_H
66
```

We're especially interested in these buttons and the number LCD

setupUi is run when the widget is created

Code in retranslateUi is run when the window is resized

General Messages

examplegui.ui - mne-cpp - Qt Creator

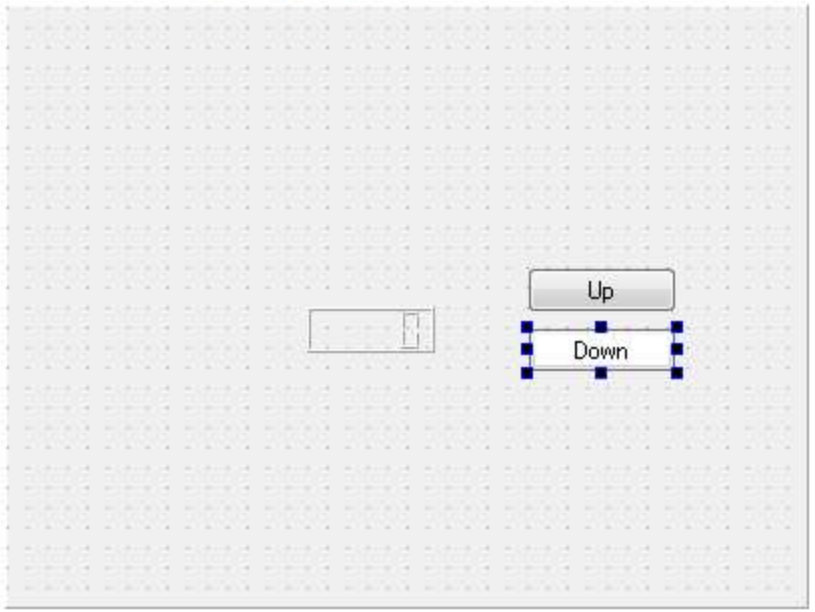
File Edit Build Debug Analyze Tools Window Help

examplegui.ui*

Welcome
Edit
Design
Debug
Projects
Help

Filter

- Layouts
  - Vertical Layout
  - Horizo···Layout
  - Grid Layout
  - Form Layout
- Spacers
  - Horizo···Spacer
  - Vertical Spacer
- Buttons
  - Push Button
  - Tool Button
  - Radio Button
  - Check Box
  - Comma···utton
  - Dialog···on Box
- Item Vi···-Based)
  - List View
  - Tree View
  - Table View
  - Column View
- Item Wi···-Based)
  - List Widget
  - Tree Widget
  - Table Widget
- Containers
  - Group Box
  - Scroll Area



Go back to designer and change the names of the buttons.
Rebuild.

ui_examplegui.h - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

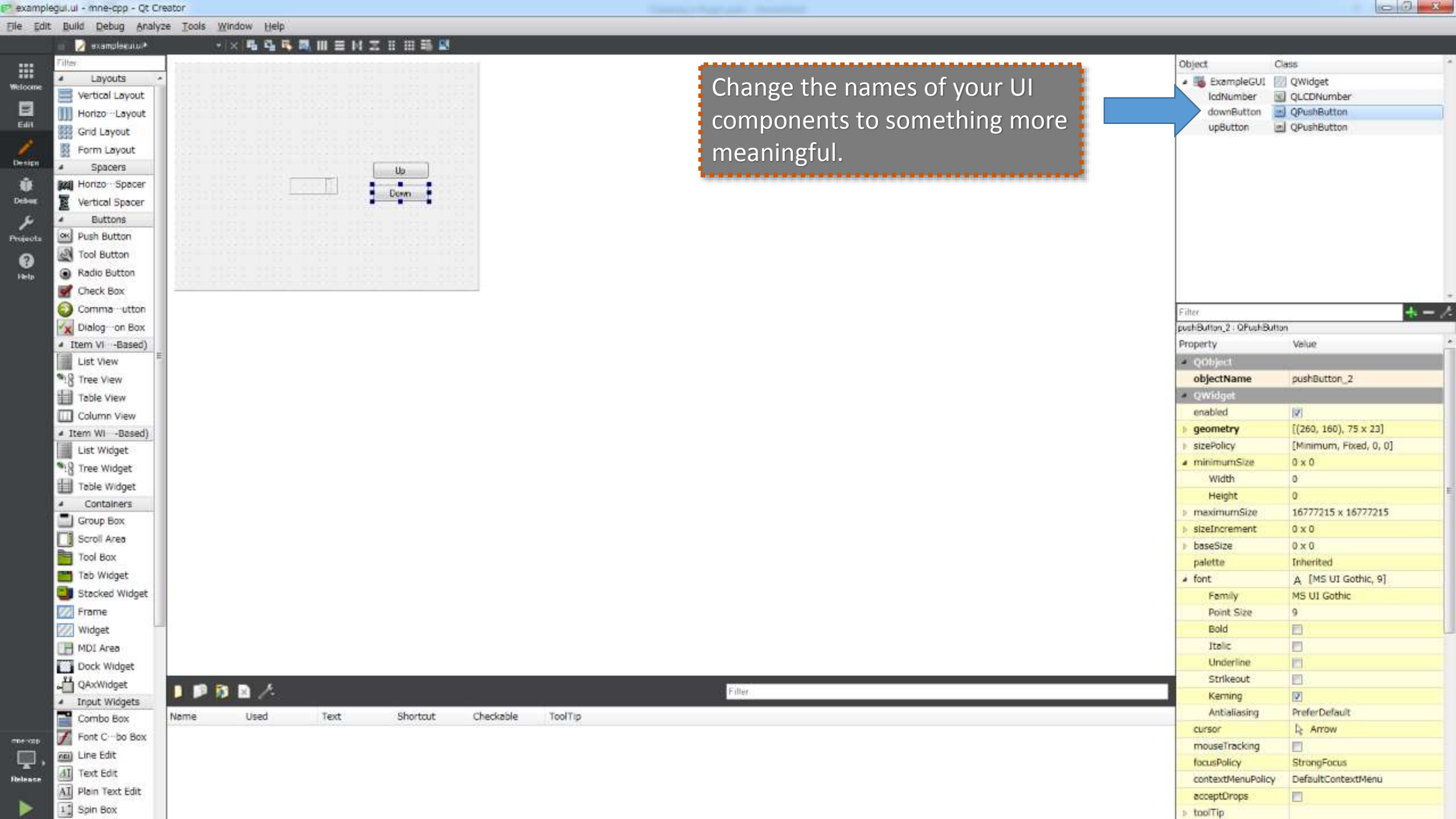
- plugins
  - plugins.pro
  - averaging
  - babymeg
  - covariance
  - dummytoolbox
    - dummytoolbox.pro
    - mne-cpp
    - Headers
      - FormFiles
        - dummyaboutwidget.h
        - dummysetupwidget.h
        - dummyyourwidget.h
      - dummytoolbox.h
      - dummytoolbox_global.h
    - Sources
      - FormFiles
        - dummyaboutwidget.cpp
        - dummysetupwidget.cpp
        - dummyyourwidget.cpp
      - dummytoolbox.cpp
    - Forms
      - FormFiles
        - dummyabout.ui
        - dummysetup.ui
        - dummyyourtoolbarwidget.ui
    - Other files
  - ecgsimulator
  - exampleplugin
    - exampleplugin.pro
    - mne-cpp
    - Headers
      - FormFiles
        - examplegui.h
      - exampleplugin.h
      - exampleplugin_global.h
    - Sources
      - FormFiles
        - examplegui.cpp
      - exampleplugin.cpp
    - Forms
      - FormFiles

ui_examplegui.h* | <No Symbols>

```
14  #include <QtWidgets/QApplication>
15  #include <QtWidgets/QButtonGroup>
16  #include <QtWidgets/QHeaderView>
17  #include <QtWidgets/QLCDNumber>
18  #include <QtWidgets/QPushButton>
19  #include <QtWidgets/QWidget>
20
21  QT_BEGIN_NAMESPACE
22
23  class Ui_ExampleGUI
24  {
25  public:
26      QPushButton *pushButton;
27      QPushButton *pushButton_2;
28      QLCDNumber *lcdNumber;
29
30      void setupUi(QWidget *ExampleGUI)
31      {
32          if (ExampleGUI->objectName().isEmpty())
33              ExampleGUI->setObjectName(QStringLiteral("ExampleGUI"));
34          ExampleGUI->resize(400, 300);
35          pushButton = new QPushButton(ExampleGUI);
36          pushButton->setObjectName(QStringLiteral("pushButton"));
37          pushButton->setGeometry(QRect(260, 130, 75, 23));
38          pushButton_2 = new QPushButton(ExampleGUI);
39          pushButton_2->setObjectName(QStringLiteral("pushButton_2"));
40          pushButton_2->setGeometry(QRect(260, 160, 75, 23));
41          lcdNumber = new QLCDNumber(ExampleGUI);
42          lcdNumber->setObjectName(QStringLiteral("lcdNumber"));
43          lcdNumber->setGeometry(QRect(150, 150, 64, 23));
44
45          retranslateUi(ExampleGUI);
46
47          QMetaObject::connectSlotsByName(ExampleGUI);
48      } // setupUi
49
50      void retranslateUi(QWidget *ExampleGUI)
51      {
52          ExampleGUI->setWindowTitle(QApplication::translate("ExampleGUI", "Form", 0));
53          pushButton->setText(QApplication::translate("ExampleGUI", "Up", 0));
54          pushButton_2->setText(QApplication::translate("ExampleGUI", "Down", 0));
55      } // retranslateUi
56  };
57
58  namespace Ui {
59      class ExampleGUI: public Ui_ExampleGUI {};
60  } // namespace Ui
61
62  QT_END_NAMESPACE
63
64  #endif // UI_EXAMPLEGUI_H
65
```

If you look back in ui_yourclass.h, you'll that the names of the buttons have ben updated

If you try to modify this file manually, all your changes will be overwritten the next time you build. Make all changes through Designer.


examplegui.ui - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Layouts
Vertical Layout
Horizo···Layout
Grid Layout
Form Layout
Spacers
Horizo···Spacer
Vertical Spacer
Buttons
Push Button
Tool Button
Radio Button
Check Box
Comma···utton
Dialog···on Box
Item Vi···-Based)
List View
Tree View
Table View
Column View
Item Wi···-Based)
List Widget
Tree Widget
Table Widget
Containers
Group Box
Scroll Area
Tool Box
Tab Widget
Stacked Widget
Frame
Widget
MDI Area
Dock Widget
QAxWidget
Input Widgets
Combo Box
Font C···bo Box
Line Edit
Text Edit
Plain Text Edit
Spin Box
Up
Down
Change the names of your UI components to something more meaningful.
Object
Class
ExampleGUI
QWidget
lcdNumber
QLCDNumber
downButton
QPushButton
upButton
QPushButton
pushButton_2 : QPushButton
Property
Value
QObject
objectName
pushButton_2
QWidget
enabled
geometry
[(260, 160), 75 x 23]
sizePolicy
[Minimum, Fixed, 0, 0]
minimumSize
0 x 0
Width
0
Height
0
maximumSize
16777215 x 16777215
sizeIncrement
0 x 0
baseSize
0 x 0
palette
Inherited
font
[MS UI Gothic, 9]
Family
MS UI Gothic
Point Size
9
Bold
Italic
Underline
Strikeout
Kerning
Antialiasing
PreferDefault
cursor
Arrow
mouseTracking
focusPolicy
StrongFocus
contextMenuPolicy
DefaultContextMenu
acceptDrops
toolTip
Name
Used
Text
Shortcut
Checkable
ToolTip

examplegui.cpp - mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

- babymeg
- covariance
- dummytoolbox
  - dummytoolbox.pro
  - mne-cpp
  - Headers
    - FormFiles
      - dummyaboutwidget.h
      - dummysetupwidget.h
      - dummyyourwidget.h
    - dummytoolbox.h
    - dummytoolbox_global.h
  - Sources
    - FormFiles
      - dummyaboutwidget.cpp
      - dummysetupwidget.cpp
      - dummyyourwidget.cpp
    - dummytoolbox.cpp
  - Forms
    - FormFiles
      - dummyabout.ui
      - dummysetup.ui
      - dummyyourtoolbarwidget.ui
  - Other files
- ecgsimulator
- exampleplugin
  - exampleplugin.pro
  - mne-cpp
  - Headers
    - FormFiles
      - examplegui.h
    - exampleplugin.h

examplegui.cpp | ExampleGUI::ExampleGUI(QWidget *)

```
1  #include "examplegui.h"
2  #include "ui_examplegui.h"
3
4  ExampleGUI::ExampleGUI(QWidget *parent) :
5      QWidget(parent),
6      ui(new Ui::ExampleGUI)
7  {
8      ui->setupUi(this);
9
10     // After setup has completed, connect the buttons (MUST BE DONE AFTER setupUi)
11     connect(ui->upButton, SIGNAL(released()), this, SLOT(incrementLcd()));
12     connect(ui->downButton, SIGNAL(released()), this, SLOT(decrementLcd()));
13
14 }
15
16 ExampleGUI::~ExampleGUI()
17 {
18     delete ui;
19 }
20
```

Head to your UI's source file and connect the buttons to some functions

This syntax links a signal emitted from a button to a function belonging to ExampleGUI (this) called incrementLcd.

The released() function is built-in, but we need to create

examplegui.cpp - mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

- babymeg
- covariance
- dummytoolbox
  - dummytoolbox.pro
  - mne-cpp
  - Headers
    - FormFiles
      - dummyaboutwidget.h
      - dummysetupwidget.h
      - dummyyourwidget.h
    - dummytoolbox.h
    - dummytoolbox_global.h
  - Sources
    - FormFiles
      - dummyaboutwidget.cpp
      - dummysetupwidget.cpp
      - dummyyourwidget.cpp
    - dummytoolbox.cpp
  - Forms
    - FormFiles
      - dummyabout.ui
      - dummysetup.ui
      - dummyyourtoolbarwidget.ui
  - Other files
- ecgsimulator
- exampleplugin
  - exampleplugin.pro
  - mne-cpp
  - Headers
    - FormFiles
      - examplegui.h
    - exampleplugin.h
    - exampleplugin_global.h
  - Sources
    - FormFiles
      - examplegui.cpp
    - exampleplugin.cpp
  - Forms
    - FormFiles
      - examplegui.ui
  - Other files
- fiffsimulator
- mne
- mne-cpp

Open Documents

mne-cpp

exampleguih | incrementLcd(): void

```
1  #ifndef EXAMPLEGUI_H
2  #define EXAMPLEGUI_H
3
4  #include <QWidget>
5
6  namespace Ui {
7  class ExampleGUI;
8  }
9
10 class ExampleGUI : public QWidget
11 {
12     Q_OBJECT
13
14 public:
15     explicit ExampleGUI(QWidget *parent = 0);
16     ~ExampleGUI();
17
18 public slots:
19
20     void incrementLcd();
21     void decrementLcd();
22
23 private:
24
25     // Control + Clock on ExampleGUI to see
26     // the code generated from the designer file we made
27     Ui::ExampleGUI *ui;
28 };
29
30 #endif // EXAMPLEGUI_H
31
```

Declare our two new functions in the Header

examplegui.cpp | ExampleGUI::decrementLcd(): void

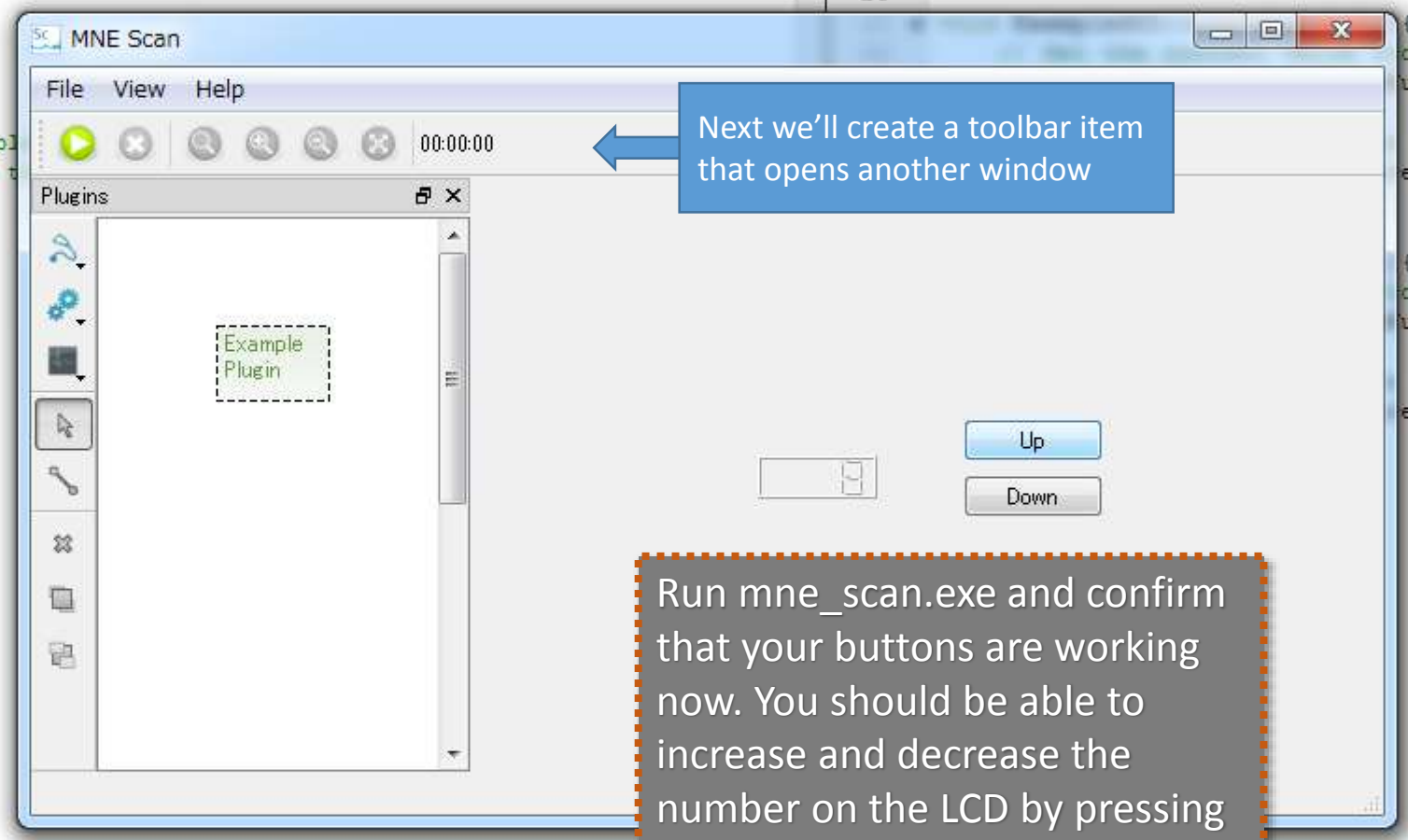
```
1  #include "examplegui.h"
2  #include "ui_examplegui.h"
3
4  ExampleGUI::ExampleGUI(QWidget *parent) :
5      QWidget(parent),
6      ui(new Ui::ExampleGUI)
7  {
8      ui->setupUi(this);
9
10     // After setup has completed, connect the buttons (MUST BE DONE AFTER setupUi)
11     connect(ui->upButton, SIGNAL(released()), this, SLOT(incrementLcd()));
12     connect(ui->downButton, SIGNAL(released()), this, SLOT(decrementLcd()));
13
14 }
15
16 ExampleGUI::~ExampleGUI()
17 {
18     delete ui;
19 }
20
21 void ExampleGUI::incrementLcd(){
22     // Get the current value from the lcd
23     int currentValue = ui->lcdNumber->intValue();
24
25     // Set the new value as one greater
26     ui->lcdNumber->display(currentValue + 1);
27 }
28
29 void ExampleGUI::decrementLcd(){
30     // Get the current value from the lcd
31     int currentValue = ui->lcdNumber->intValue();
32
33     // Set the new value as one less
34     ui->lcdNumber->display(currentValue - 1);
35 }
36
```

And implement the in the Source file

Issues

```
    explicit ExampleGUI(QWidget *parent = 0);
    ~ExampleGUI();

public slots:

    void incrementLcd();
    void decrementLcd();

private:

    // Control + Clock on Exampl
    // the code generated from t
    Ui::ExampleGUI *ui;
};

#endif // EXAMPLEGUI_H
```

```
ExampleGUI::~ExampleGUI()
{
    delete ui;
}
```

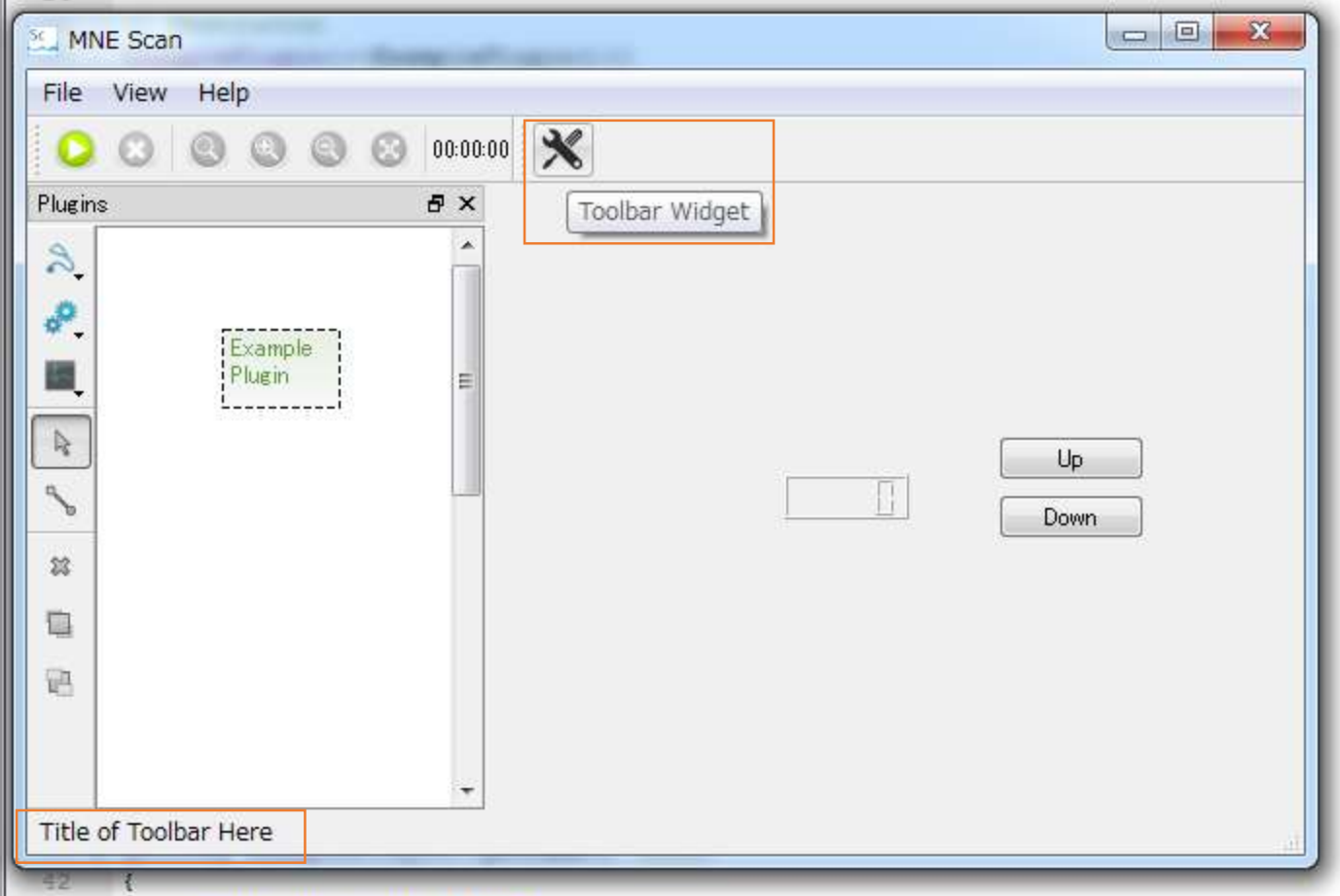


Next we'll create a toolbar item that opens another window

Run mne_scan.exe and confirm that your buttons are working now. You should be able to increase and decrease the number on the LCD by pressing the two buttons

```
eral Messages
ect MESSAGE: TMSI plugin not configured! TMSi Driver not found!
xact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
ect MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
```

```
    showSettingsWidgetAction->setStatusTip(tr("Title of Toolbar Here"));
    addPluginAction(showSettingsWidgetAction);
}
```




```
{
    return "Example Plugin";
}

QWidget* ExamplePlugin::setupWidget()
{
    // Setup the UI using our custom form class here
    ExampleGUI* setupWidget = new ExampleGUI();
    return setupWidget;
}
```

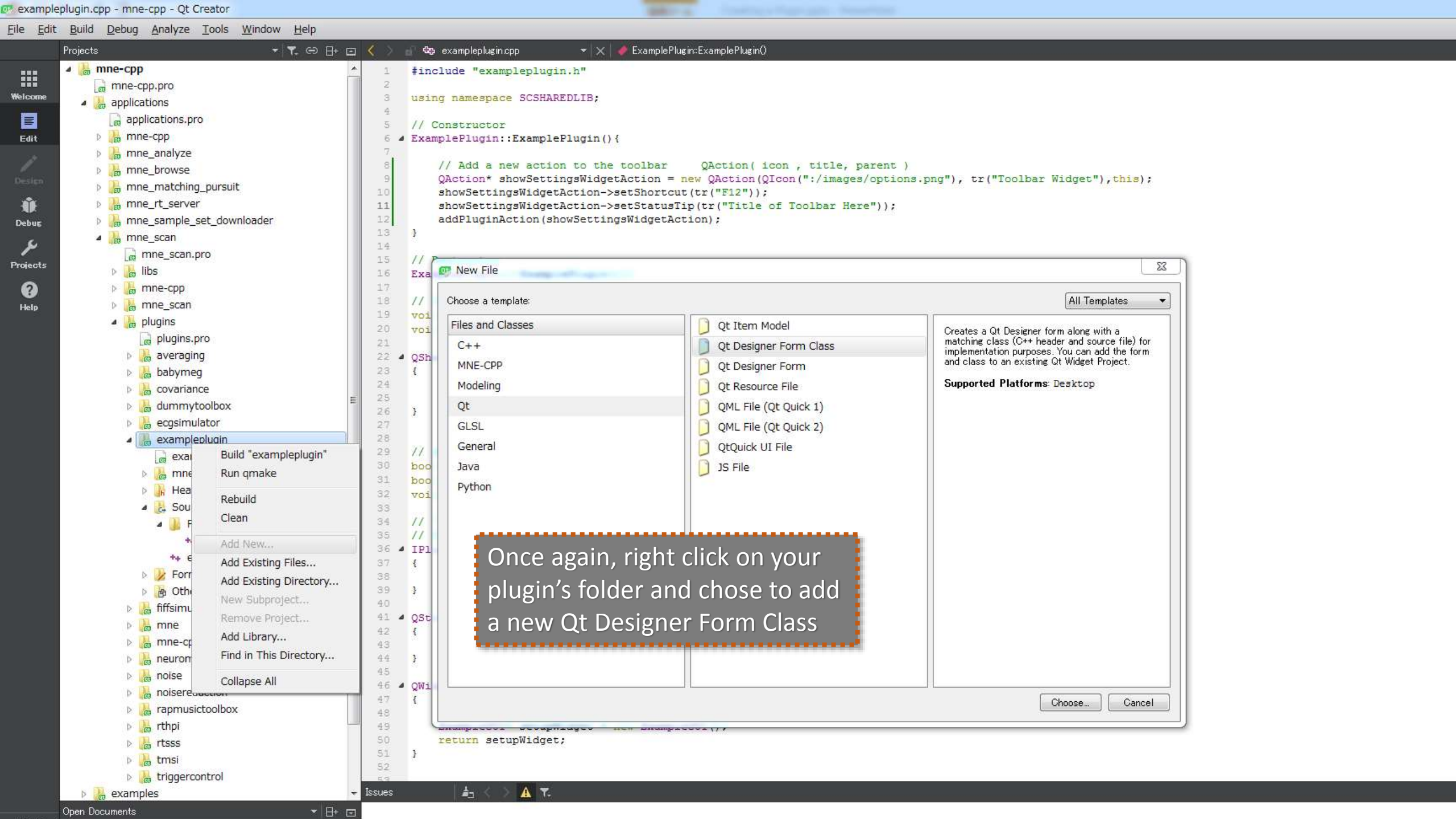
Issues

Tip:

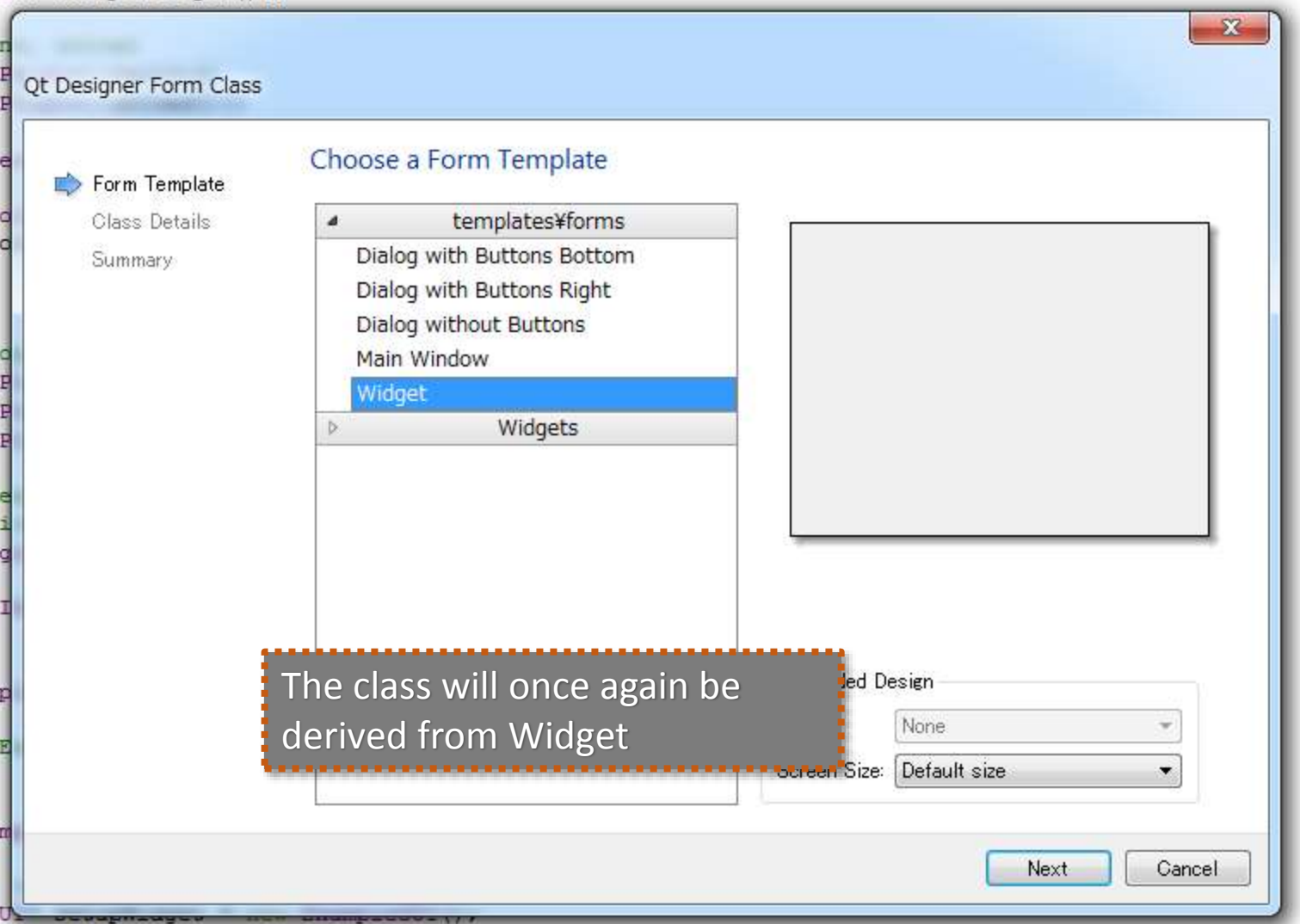
In Qt, strings that will be shown to the user should be wrapped in tr(). This makes the strings translatable if somebody ever needs to translate your plugin into another language.

Next:

Right now your toolbar button doesn't do anything. Next we'll create another widget for it to open.


exampleplugin.cpp - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
Build "exampleplugin"
Run qmake
Rebuild
Clean
Add New...
Add Existing Files...
Add Existing Directory...
New Subproject...
Remove Project...
Add Library...
Find in This Directory...
Collapse All
fiffsimu
mne
neuron
noise
rapmusictoolbox
rthpi
rtsss
tmsi
triggercontrol
examples
Open Documents
Welcome
Edit
Design
Debug
Projects
Help
exampleplugin.cpp
ExamplePlugin::ExamplePlugin()
#include "exampleplugin.h"
using namespace SCSHAREDLIB;
// Constructor
ExamplePlugin::ExamplePlugin(){
// Add a new action to the toolbar QAction(icon , title, parent)
QAction* showSettingsWidgetAction = new QAction(QIcon(":/images/options.png"), tr("Toolbar Widget"),this);
showSettingsWidgetAction->setShortcut(tr("F12"));
showSettingsWidgetAction->setStatusTip(tr("Title of Toolbar Here"));
addPluginAction(showSettingsWidgetAction);
}
return setupWidget;
}
New File
Choose a template:
All Templates
Files and Classes
C++
MNE-CPP
Modeling
Qt
GLSL
General
Java
Python
Qt Item Model
Qt Designer Form Class
Qt Designer Form
Qt Resource File
QML File (Qt Quick 1)
QML File (Qt Quick 2)
QtQuick UI File
JS File
Creates a Qt Designer form along with a matching class (C++ header and source file) for implementation purposes. You can add the form and class to an existing Qt Widget Project.
Supported Platforms: Desktop
Choose...
Cancel
Once again, right click on your plugin's folder and chose to add a new Qt Designer Form Class
Issues

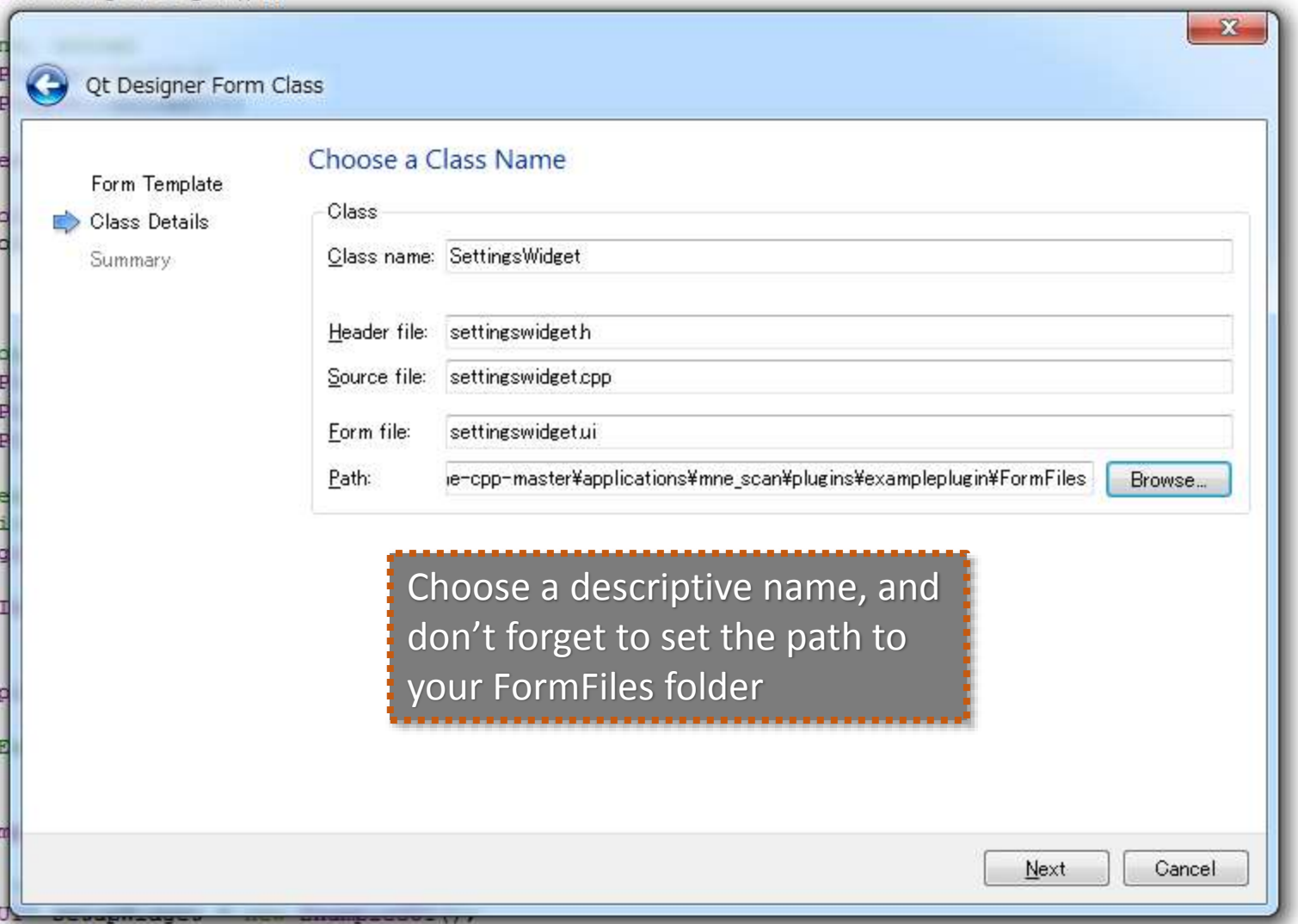
```
    showSettingsWidgetAction->setStatusTip(tr("Title of Toolbar Here"));
    addPluginAction(showSettingsWidgetAction);
}

// Destructor
ExamplePlugin::~ExamplePlugin(){}

// Init, clon
void ExampleP
void ExampleP

QSharedPointe
{
    QSharedPo
    return po
}


// start, sto
bool ExampleP
bool ExampleP
void ExampleP

// 'const' me
// Ctrl + Cli
IPlugin::Plug
{
    return _I
}

QString Examp
{
    return "E
}

QWidget* Exam
{
    // Setup
    ExampleGUI
    return setupWidget;
}
```



The class will once again be derived from Widget

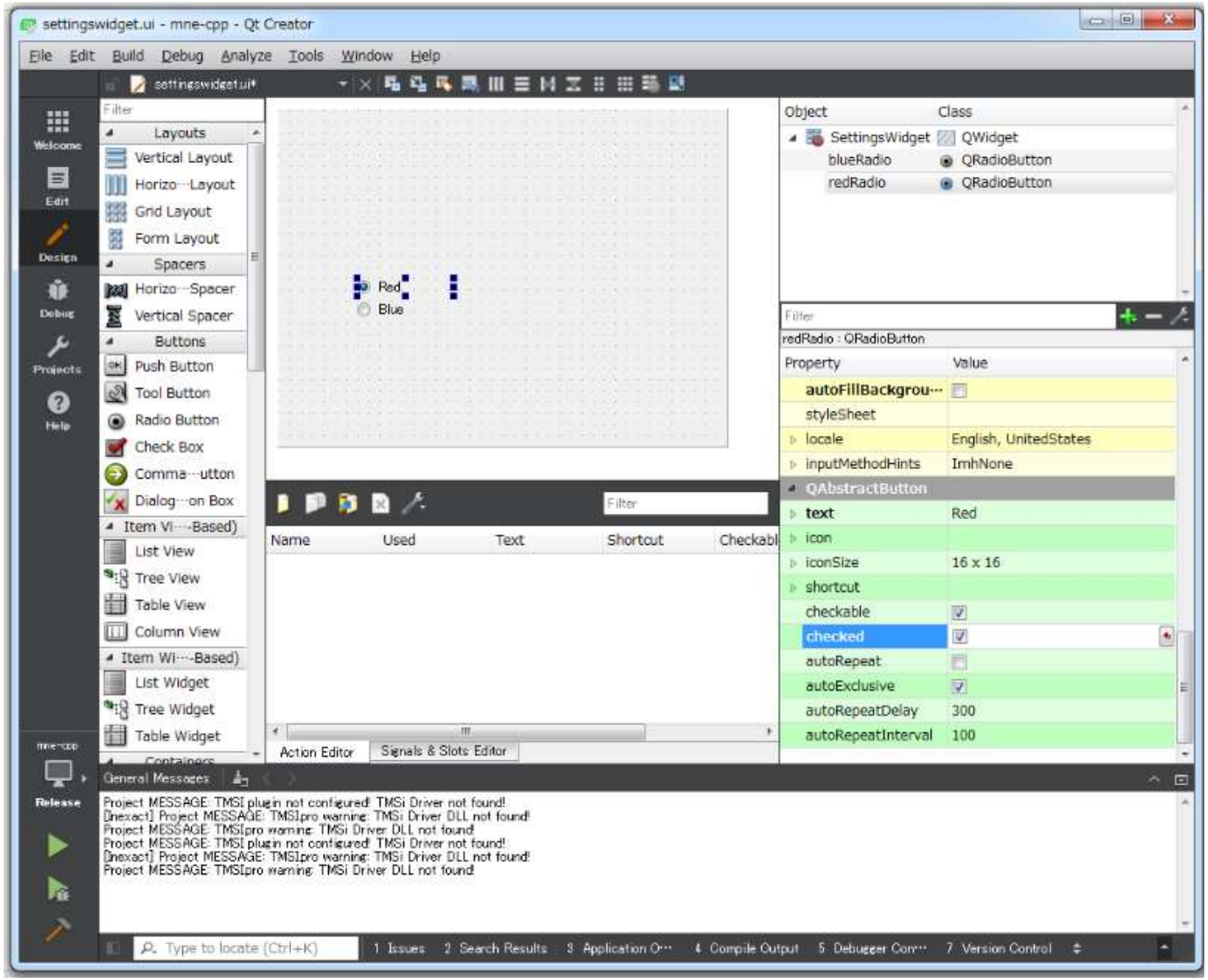
Issues

```
    showSettingsWidgetAction->setStatusTip(tr("Title of Toolbar Here"));
    addPluginAction(showSettingsWidgetAction);
}

// Destructor
ExamplePlugin::~ExamplePlugin(){}

// Init, clon
void ExampleP
void ExampleP

QSharedPointe
{
    QSharedPo
    return po
}


// start, sto
bool ExampleP
bool ExampleP
void ExampleP

// 'const' me
// Ctrl + Cli
IPlugin::Plug
{
    return _I
}

QString Examp
{
    return "E
}

QWidget* Exam
{
    // Setup
    ExampleGU
    return setupWidget;
}
```

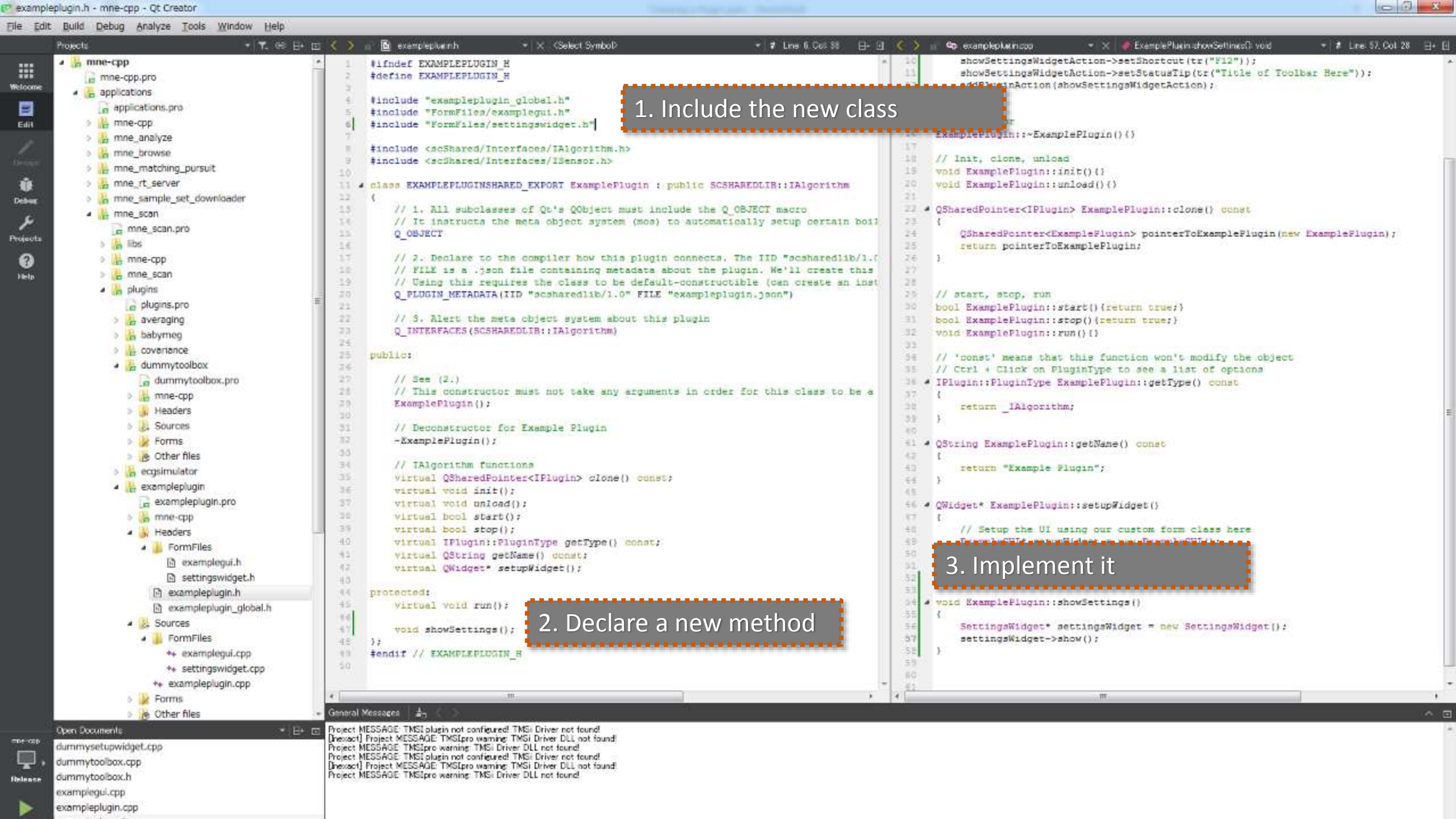


Choose a descriptive name, and don't forget to set the path to your FormFiles folder

Issues



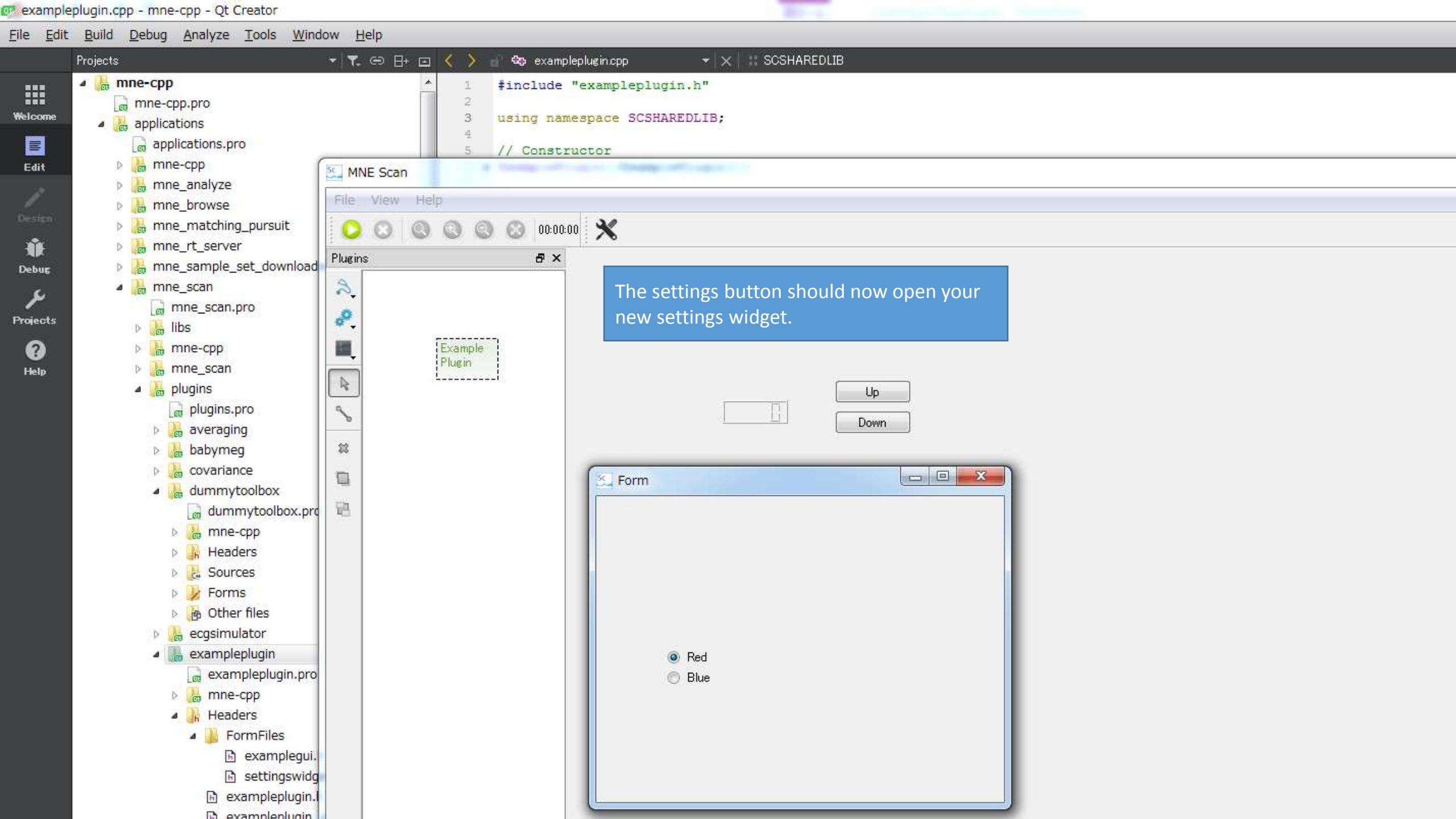
Add two radio buttons to the UI and change their names to something meaningful


exampleplugin.h - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
dummytoolbox.pro
mne-cpp
Headers
Sources
Forms
Other files
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
FormFiles
examplegui.h
settingswidget.h
exampleplugin.h
exampleplugin_global.h
Sources
FormFiles
examplegui.cpp
settingswidget.cpp
exampleplugin.cpp
Forms
Other files
Open Documents
dummysetupwidget.cpp
dummytoolbox.cpp
dummytoolbox.h
examplegui.cpp
exampleplugin.cpp
Welcome
Edit
Debug
Projects
Help
Release
exampleplugin.h
<Select Symbol>
Line: 6, Col: 38
#ifndef EXAMPLEPLUGIN_H
#define EXAMPLEPLUGIN_H
#include "exampleplugin_global.h"
#include "FormFiles/examplegui.h"
#include "FormFiles/settingswidget.h"
#include <scShared/Interfaces/IAlgorithm.h>
#include <scShared/Interfaces/ISensor.h>
class EXAMPLEPLUGINSHARED_EXPORT ExamplePlugin : public SCSHAREDLIB::IAlgorithm
{
// 1. All subclasses of Qt's QObject must include the Q_OBJECT macro
// It instructs the meta object system (moc) to automatically setup certain boil
Q_OBJECT
// 2. Declare to the compiler how this plugin connects. The IID "scsharedlib/1.0
// FILE is a .json file containing metadata about the plugin. We'll create this
// Using this requires the class to be default-constructible (can create an inst
Q_PLUGIN_METADATA(IID "scsharedlib/1.0" FILE "exampleplugin.json")
// 3. Alert the meta object system about this plugin
Q_INTERFACES(SCSHAREDLIB::IAlgorithm)
public:
// See (2.)
// This constructor must not take any arguments in order for this class to be a
ExamplePlugin();
// Deconstructor for Example Plugin
~ExamplePlugin();
// IAlgorithm functions
virtual QSharedPointer<IPlugin> clone() const;
virtual void init();
virtual void unload();
virtual bool start();
virtual bool stop();
virtual IPlugin::PluginType getType() const;
virtual QString getName() const;
virtual QWidget* setupWidget();
protected:
virtual void run();
void showSettings();
};
#endif // EXAMPLEPLUGIN_H
1. Include the new class
2. Declare a new method
examplepluginc.cpp
ExamplePlugin::showSettings() : void
Line: 57, Col: 28
showSettingsWidgetAction->setShortcut(tr("F12"));
showSettingsWidgetAction->setStatusTip(tr("Title of Toolbar Here"));
ExamplePlugin::~ExamplePlugin(){}
// Init, clone, unload
void ExamplePlugin::init(){}
void ExamplePlugin::unload(){}
QSharedPointer<IPlugin> ExamplePlugin::clone() const
{
QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
return pointerToExamplePlugin;
}
// start, stop, run
bool ExamplePlugin::start(){return true;}
bool ExamplePlugin::stop(){return true;}
void ExamplePlugin::run(){}
// 'const' means that this function won't modify the object
// Ctrl + Click on PluginType to see a list of options
IPlugin::PluginType ExamplePlugin::getType() const
{
return _IAlgorithm;
}
QString ExamplePlugin::getName() const
{
return "Example Plugin";
}
QWidget* ExamplePlugin::setupWidget()
{
// Setup the UI using our custom form class here
3. Implement it
void ExamplePlugin::showSettings()
{
SettingsWidget* settingsWidget = new SettingsWidget();
settingsWidget->show();
}
General Messages
Project MESSAGE: TMSI plugin not configured! TMSi Driver not found!
[inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSI plugin not configured! TMSi Driver not found!
[inexact] Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!
Project MESSAGE: TMSIpro warning: TMSi Driver DLL not found!

```
    showSettingsWidgetAction->setStatusTip(tr("Title of Toolbar Here"));
    addPluginAction(showSettingsWidgetAction);

    // connect the action to the class member function
    connect(showSettingsWidgetAction, &QAction::triggered, this, &ExamplePlugin::showSettings );
}

// D
Exam

// Init, clone, unload
void ExamplePlugin::init(){}
void ExamplePlugin::unload(){}

QSharedPointer<IPlugin> ExamplePlugin::clone() const
{
    QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
    return pointerToExamplePlugin;
}


// start, stop, run
bool ExamplePlugin::start(){return true;}
bool ExamplePlugin::stop(){return true;}
void ExamplePlugin::run(){}

// 'const' means that this function won't modify the object
// Ctrl + Click on PluginType to see a list of options
IPlugin::PluginType ExamplePlugin::getType() const
{
    return _IAlgorithm;
}

QString ExamplePlugin::getName() const
{
    return "Example Plugin";
}

QWidget* ExamplePlugin::setupWidget()
{
    // Setup the UI using our custom form class here
    ExampleGUI* setupWidget = new ExampleGUI();
```

4. Connect the settings button to the function we just created

General Messages


exampleplugin.cpp - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
exampleplugin.cpp
SCSHAREDLIB
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_download
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
dummytoolbox.pro
mne-cpp
Headers
Sources
Forms
Other files
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
FormFiles
examplegui.
settingswidg
exampleplugin.
Welcome
Edit
Design
Debug
Projects
Help
1 #include "exampleplugin.h"
2
3 using namespace SCSHAREDLIB;
4
5 // Constructor
MNE Scan
File View Help
00:00:00
Plugins
Example Plugin
The settings button should now open your new settings widget.
Up
Down
Form
Red
Blue

Accessing Measurement Data

```
#define EXAMPLEPLUGIN_H

#include "exampleplugin_global.h"
#include "FormFiles/examplegui.h"
#include "FormFiles/settingswidget.h"

#include <scShared/Interfaces/IAlgorithm.h>
#include <scShared/Interfaces/ISensor.h>

#include <scMeas/newrealtimemultisamplearray.h>
#include <generics/circularmatrixbuffer.h>

using namespace SCSHAREDLIB;

ugin : pub
ugin
ust include the Q_OBJECT macro
 (mos) to automatically setup certain boiler plate functions


 plugin connects. The IID "scsharedlib/1.0" is where all plugins connect in mne_scan.
adata about the plugin. We'll create this file next, for now it doesn't exist.
 default-constructible (can create an instance without any arguments)
    Q_PLUGIN_METADATA(IID "scsharedlib/1.0" FILE "exampleplugin.json")

    // 3. Alert the meta object system about this plugin
    Q_INTERFACES(SCSHAREDLIB::IAlgorithm)

public:

    // See (2.)
    // This constructor must not take any arguments in order for this class to be a plugin
    ExamplePlugin();

    // Deconstructor for Example Plugin
    ~ExamplePlugin();

    // IAlgorithm functions
    virtual QSharedPointer<IPlugin> clone() const;
    virtual void init();
    virtual void unload();
    virtual bool start();
    virtual bool stop();
    virtual IPlugin::PluginType getType() const;
```

If you haven't already, add using namespace SCSHAREDLIB to make accessing members of that namespace more convenient within this file

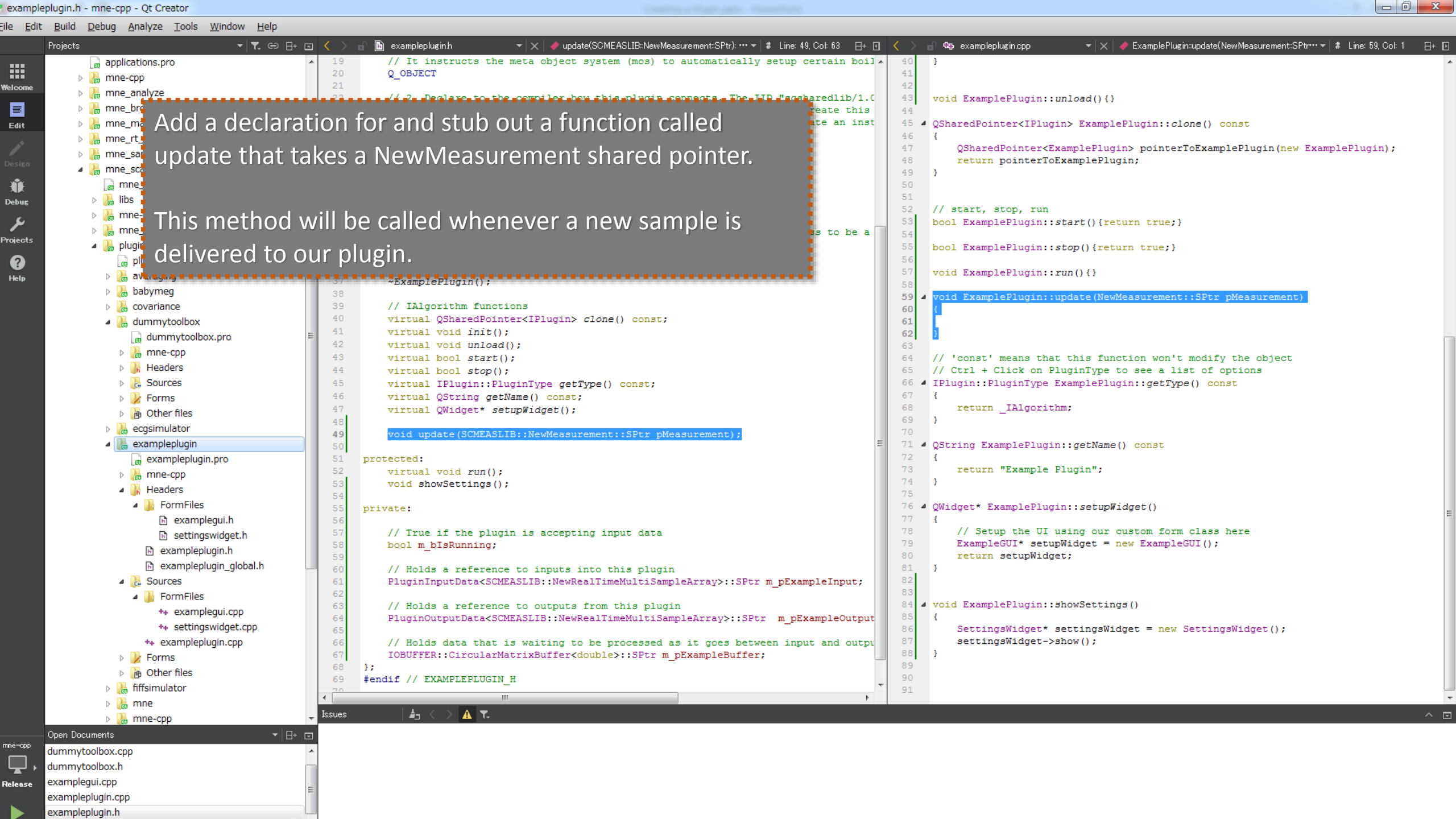
We're going to need access two classes from the measurement generics libraries, so import them into your plugin's header.

```
30  public:
31
32      // See (2.)
33      // This constructor must not take any arguments in order for this class to be a plugin
34      ExamplePlugin();
35
36      // Deconstructor for Example Plugin
37      ~ExamplePlugin();
38
39      // IAlgorithm functions
40      virtual QSharedPointer<IPlugin> clone() const;
41      virtual void init();
42      virtual void unload();
43      virtual bool start();
44      virtual bool stop();
45      virtual IPlugin::PluginType getType() const;
46      virtual QString getName() const;
47      virtual QWidget* setupWidget();
48
49      void update(SCMEASLIB::NewMeasurement::SPtr pMeasurement);
50
51  protected:
52
53      virtual void run();
54      void showSettings();
55
56  private:
57
58      // Pointer to info about number of channels, sensor positions, sampling rate, etc.
59      FIFFLIB::FiffInfo::SPtr m_pFiffInfo;
60
61      // Pointer to inputs into this plugin
62      PluginInputData<SCMEASLIB::NewRealTimeMultiSampleArray>::SPtr m_pExampleInput;
63
64      // Pointer to outputs from this plugin
65      PluginOutputData<SCMEASLIB::NewRealTimeMultiSampleArray>::SPtr  m_pExampleOutput;
66
67      // Pointer to data that is waiting to be processed as it goes between input and output
68      IOBUFFER::CircularMatrixBuffer<double>::SPtr m_pExampleBuffer;
69
70      // True if the plugin is processing data in the buffer above
71      bool m_bIsRunning;
72  };
73  #endif // EXAMPLEPLUGIN_H
```

Next up we need to add a few private variable declarations

These variable names follow MNE-CPP coding convention

p - Qt Creator
nalyze Tools Window Help
exampleplugin.cpp*
ExamplePlugin::ExamplePlugin()
ations.pro
pp
analyze
browse
matching_pursuit
t_server
sample_set_downloader
scan
e_scan.pro
e-cpp
e_scan
gins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
dummytoolbox.pro
mne-cpp
Headers
Sources
Forms
Other files
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
FormFiles
examplegui.h
settingswidget.h
#include "exampleplugin.h"
using namespace SCSHAREDLIB;
using namespace SCMEASLIB;
using namespace IOBUFFER;
// Constructor
ExamplePlugin::ExamplePlugin()
: m_bIsRunning(false)
, m_pExampleInput(NULL)
, m_pExampleOutput(NULL)
, m_pExampleBuffer(CircularMatrixBuffer<double>::SPtr())
{
, title, parent)
(":/images/options.png"), tr("Toolbar Widget"),this);
oolbar Here"));
connect(showSettingsWidgetAction, &QAction::triggered, this, &ExamplePlugin::showSettings);
}
QSharedPointer<IPlugin> ExamplePlugin::clone() const
{
QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
return pointerToExamplePlugin;
}
// start, stop, run
bool ExamplePlugin::start(){return true;}
Use namespaces for SCMEASLIB and IOBUFFER so you don't have to type too much
Add initialization for each of the variables we just defined to the class Constructor.
Variables can also be initialized within the body of the function, but MNE's coding conventions prefer the syntax used here.
TIP:
Now is also a good time to perform a build just to make sure you haven't broken anything.


examplepluginh - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
examplepluginh
update(SCMEASLIB::NewMeasurement::SPtr): ···
Line: 49, Col: 63
examplepluginc
ExamplePlugin::update(NewMeasurement::SPtr···
Line: 59, Col: 1
Add a declaration for and stub out a function called update that takes a NewMeasurement shared pointer.
This method will be called whenever a new sample is delivered to our plugin.
Welcome
Edit
Design
Debug
Projects
Help
applications.pro
mne-cpp
mne_analyze
libs
babymeg
covariance
dummytoolbox
dummytoolbox.pro
Headers
Sources
Forms
Other files
ecgsimulator
exampleplugin
exampleplugin.pro
examplegui.h
settingswidget.h
exampleplugin.h
exampleplugin_global.h
examplegui.cpp
settingswidget.cpp
exampleplugin.cpp
fiffsimulator
mne
Issues
Open Documents
dummytoolbox.cpp
dummytoolbox.h
examplegui.cpp
exampleplugin.cpp
exampleplugin.h
Release

```
21      // connect the action to the class member function
22      connect(showSettingsWidgetAction, &QAction::triggered, this, &ExamplePlugin::showSettings );
23  }
24
25  // Destructor
26  ExamplePlugin::~ExamplePlugin(){}
27
28  // Init, clone, unload
29  void ExamplePlugin::init(){
30
31      // Add an input
32      m_pExampleInput = PluginInputData<NewRealTimeMultiSampleArray>::create(this, "ExampleInput", "Example Plugin's input data");
33      m_inputConnectors.append(m_pExampleInput);
34
35      // Add an ouput
36      m_pExampleOutput = PluginOutputData<NewRealTimeMultiSampleArray>::create(this, "ExampleOut", "Example Plugin's output data");
37      m_outputConnectors.append(m_pExampleOutput);
38
39      // Register for updates
40      connect(m_pExampleInput.data(), &PluginInputConnector::notify, this, &ExamplePlugin::update, Qt::DirectConnection);
41  }
42
43
44  void ExamplePlugin::unload(){}
45
46  QSharedPointer<IPlugin> ExampleP
47  {
48      QSharedPointer<ExamplePlugin
49      return pointerToExamplePlugi
50  }
51
52
53  // start, stop, run
```

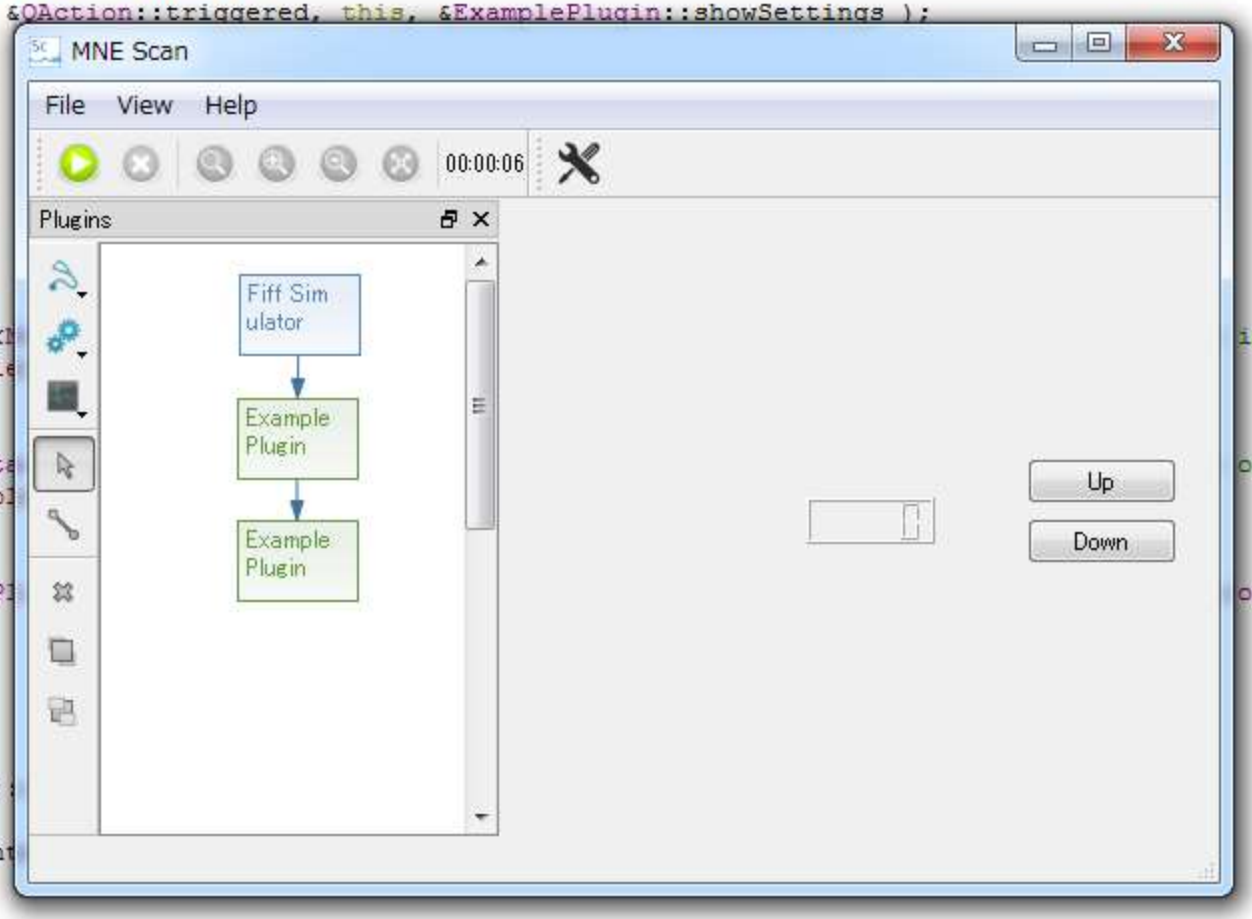
init() is called when the plugin is dropped onto the stage. In this method we need to add a new input, an output, and set it up so that our update function is called whenever we are notified of new data

Note:
We're going to feed the input data straight into the output. This plugin will analyze the data without modifying it. If you wish to create a plugin that does modify the data (such as a filter), you'll need to make changes to this code.

Issues

```
, m_pExampleOutput(NULL)
, m_pExampleBuffer(CircularMatrixBuffer<double>::SPtr())
{

    // Add a new action to the toolbar     QAction( icon , title, parent )
    QAction* showSettingsWidgetAction = new QAction(QIcon(":/images/options.png"), tr("Toolbar Widget"),this);
    showSettingsWidgetAction->setShortcut(tr("F12"));
    showSettingsWidgetAction->setStatusTip(tr("Title of Toolbar Here"));
    addPluginAction(showSettingsWidgetAction);

    // connect the action to the class member function
    connect(showSettingsWidgetAction, &QAction::triggered, this, &ExamplePlugin::showSettings );
```

Rebuild, and then check to ensure that you are now link input and outputs for your plugin

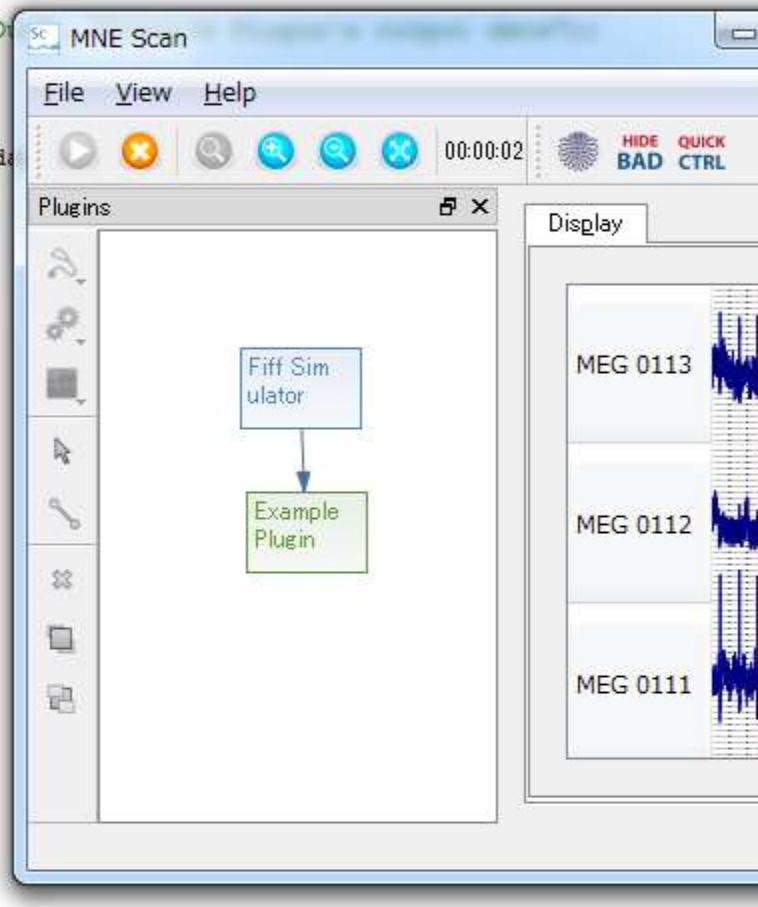



```
void ExamplePlugin::init(){

    // Add an input
    m_pExampleInput = PluginInputData<
    m_inputConnectors.append(m_pExample

    // Add an ouput
    m_pExampleOutput = PluginOutputData
    m_outputConnectors.append(m_pExampl

    // Register for updates
    connect(m_pExampleInput.data(), &P
}


void ExamplePlugin::unload(){}

QSharedPointer<IPlugin> ExamplePlugin::
{
    QSharedPointer<ExamplePlugin> point
    return pointerToExamplePlugin;
}


// start, stop, run
```

Issues

```
; tradius: 9.02422
QTextStream: No device
QTextStream: No device
RealTimeMultiSampleArrayModel::updateProjection - New projection calculated.
update
update
update
update
update
update
update
update
update
update
update
update
update
update
update
update
update
update
```

Add a print statement to your function and run the Fiff Simulator to check if the update function is being called as expected




```
"ExampleInput", "Example Plugin's input data");
```

```
"ExampleO
```

```
Plugin::upda
```

```
58  void ExamplePlugin::run(){}
59
60  void ExamplePlugin::update(NewMeasurement::SPtr pMeasurement)
61  {
62      printf("update\n");
63  }
64
65  // 'const' means that this function won't modify the object
66  // Ctrl + Click on PluginType to see a list of options
67  IPlugin::PluginType ExamplePlugin::getType() const
68  {
69      return _IAlgorithm;
70  }
71
72  QString ExamplePlugin::getName() const
73  {
74      return "Example Plugin";
```

Warning:
You will not be able to run the Fiff Simulator if you have not downloaded the sample data and placed it in the correct directory. See the Wiki for full instructions.

Issues

Window Help

exampleplugin.cpp | ExamplePlugin::update(NewMeasurement::SPtr): void

```
99  }
100
101 // Each time new data is received, place it into the buffer to be processed
102 void ExamplePlugin::update(NewMeasurement::SPtr pMeasurement)
103 {
104     // The measurement data that comes in is generic measurement data
105     // We need to tell the program to treat it as a NewRealTimeMultiSampleArray
106     QSharedPointer<NewRealTimeMultiSampleArray> pRTMSA = pMeasurement.dynamicCast<NewRealTimeMultiSampleArray>();
107
108     // The first time through, the buffer will not have been setup yet. Take care of that here
109     if (!m_pExampleBuffer){
110         unsigned int maxTimeSamples = 64;
111         unsigned int numRows        = pRTMSA->getNumChannels();
112         unsigned int numColumns     = pRTMSA->getMultiSampleArray()[0].cols();
113         CircularMatrixBuffer<double>* buffer = new CircularMatrixBuffer<double>(maxTimeSamples, numRows, numColumns);
114         m_pExampleBuffer = CircularMatrixBuffer<double>::SPtr(buffer);
115     }
116
117     // Check if fiff info has been set yet
118     if (!m_pFiffInfo){
119         m_pFiffInfo = pRTMSA->info();
120
121         // Set the size and other details of the output based on the information in m_pFiffInfo
122         m_pExampleOutput->data()->initFromFiffInfo(m_pFiffInfo);
123         m_pExampleOutput->data()->setMultiArraySize(1); // The number of samples to append before which observers will be notified
124         m_pExampleOutput->data()->setVisibility(true);  // Sets if the data is displayed in the UI or not
125     }
126
127     // A matrix with dynamic dimensions
128     MatrixXd t_mat;
129
130     // Determine how many time samples have been collected since the last update
131     // For each time sample, append the sample array to the buffer so it gets processed
132     for (unsigned char i = 0; i < pRTMSA->getMultiArraySize(); ++i) {
133         t_mat = pRTMSA->getMultiSampleArray()[i];
134         m_pExampleBuffer->push(&t_mat);
135     }
136 }
137
138
```

Add the code shown here to your plugins update method.

We are now filling up the buffer with data as it flows in, but we are not doing anything to process it yet

IAlgorithm.h - mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

- mne-cpp
  - mne-cpp.pro
  - applications
    - applications.pro
    - mne-cpp
    - mne_analyze
    - mne_browse
    - mne_matching_pursuit
    - mne_rt_server
    - mne_sample_set_downloader
    - mne_scan
      - mne_scan.pro
      - libs
      - mne-cpp
      - mne_scan
      - plugins
        - plugins.pro
        - averaging
        - babymeg
        - covariance
        - dummytoolbox
          - dummytoolbox.pro
          - mne-cpp
          - Headers
          - Sources
            - FormFiles
            - dummytoolbox.cpp
          - Forms
          - Other files
        - ecgsimulator
        - exampleplugin
          - exampleplugin.pro
          - mne-cpp
          - Headers
          - Sources
            - FormFiles
            - exampleplugin.cpp
          - Forms
          - Other files
        - fiffsimulator
        - mne

IAlgorithm.h | setupWidget(): QWidget *

```
111     /**
112     * Stops the IAlgorithm.
113     * Pure virtual method inherited by IPlugin.
114     *
115     * @return true if success, false otherwise
116     */
117     virtual bool stop() = 0;
118
119     //=============================================================
120     /**
121     * Returns the plugin type.
122     * Pure virtual method inherited by IPlugin.
123     *
124     * @return type of the IAlgorithm
125     */
126     virtual PluginType getType() const = 0;
127
128     //=============================================================
129     /**
130     * Returns the plugin name.
131     * Pure virtual method inherited by IPlugin.
132     *
133     * @return the name of the IAlgorithm.
134     */
135     virtual QString getName() const = 0;
136
137     //=============================================================
138     /**
139     * True if multi instantiation of plugin is allowed.
140     *
141     * @return true if multi instantiation of plugin is al
142     */
143     virtual inline bool multiInstanceAllowed() const;
144
145     //=============================================================
146     /**
147     * Returns the set up widget for configuration of IAlg
148     * Pure virtual method inherited by IPlugin.
149     *
150     * @return the setup widget.
151     */
152     virtual QWidget* setupWidget() = 0; //setup();
153
154 protected:
155     //=============================================================================================
156     /**
157     * The starting point for the thread. After calling start(), the newly created thread calls this function.
158     * Returning from this method will end the execution of the thread.
159     * Pure virtual method inherited by QThread
160     */
161     virtual void run() = 0;
```

IAlgorithm.h

We see in the header for IAlgorithm that MNE plugins are subclassed off QThread. There are a few important points that we can glean from this header.

1. run() is automatically called after start()
2. Processing of data in the buffer should be performed in run()
3. We should not return from run() while we expect work to be done
4. The plugin thread needs to be stopped somewhere

```
39      // Register for updates
40      connect(m_pExampleInput.data(), &PluginInputConnector::notify, this, &ExamplePlugin::update, Qt::DirectConnection);
41  }
42
43
44  void ExamplePlugin::unload(){}
45
46  QSharedPointer<IPlugin> ExamplePlugin::clone() const
47  {
48      QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
49      return pointerToExamplePlugin;
50  }
51
52  bool ExamplePlugin::start()
53  {
54      // Set isRunning to true and then call run
55      // The order will be important!
56      m_bIsRunning = true;
57      QThread::start();
58      return true;
59  }
60
61  bool ExamplePlugin::stop()
62  {
63      return true;
64  }
65
66  // Each time new data is rece
67  void ExamplePlugin::update(Ne
68  {
69      // The measurement data t
70      // We need to tell the pr
71      QSharedPointer<NewRealTim                                    RealTimeMultiSampleArray>();
72
73      // Check if fiff info has been set yet
74      if (!m_pFiffInfo){
75          m_pFiffInfo = pRTMSA->info();
76
77          // Set the size and other details of the output based on the information in m_pFiffInfo
78          m_pExampleOutput->data()->initFromFiffInfo(m_pFiffInfo);
79          m_pExampleOutput->data()->setMultiArraySize(1); // The number of samples to append before which observers will be notified
80          m_pExampleOutput->data()->setVisibility(true);  // Sets if the data is displayed in the UI or not
81      }
82
83      // A matrix with dynamic dimensions
```

1. Add the code shown here to your plugin's start method. Starting the thread will automatically trigger the run() method.

Warning:
It is important that we use the QThread:: namespace to specify that we mean the start() method inherited from QThread as opposed to the method we're currently creating.

```
        // Set the size and other details of the output based on the information in m_pFiffInfo
        m_pExampleOutput->data()->initFromFiffInfo(m_pFiffInfo);
        m_pExampleOutput->data()->setMultiArraySize(1); // The number of samples to append before which observers will be notified
        m_pExampleOutput->data()->setVisibility(true);  // Sets if the data is displayed in the UI or not
    }

    // A matrix with dynamic dimensions
    MatrixXd t_mat;

    // Determine how many time samples have been collected since the last update
    // For each time sample, append the sample array to the buffer so it gets processed
    for (unsigned char i = 0; i < pRTMSA->getMultiArraySize(); ++i) {
        t_mat = pRTMSA->getMultiSampleArray()[i];
        m_pExampleBuffer->push(&t_mat);
    }
}

void ExamplePlugin::run()
{

    // Check every 10ms until the fiff info has been set
    while(!m_pFiffInfo)
        msleep(10);

    // run() will return as soon as m_bIsRunning is set to false
    // Otherwise it will continually process any data in the buffer
    while (m_bIsRunning)
    {
        // TODO: Process data and place in output
    }
}

// 'const' means that this function won't modify the object
// Ctrl + Click on PluginType to see a list of options
IPlugin::PluginType ExamplePlugin::getType() const
{
    return _IAlgorithm;
}

QString ExamplePlugin::getName() const
{
    return "Example Plugin";
}
```

2. Add the code shown here to your plugins update method.

3. Note that we never return from run() unless isRunning is set to false

```
43      // Register for updates
44      connect(m_pExampleInput.data(), &PluginInputConnector::notify, this, &ExamplePlugin::update, Qt::DirectConnection);
45  }
46
47
48  void ExamplePlugin::unload(){}
49
50  QSharedPointer<IPlugin> ExamplePlugin::clone() const
51  {
52      QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
53      return pointerToExamplePlugin;
54  }
55
56  bool ExamplePlugin::start()
57  {
58      // Set isRunning to true and then call run
59      // The order will be important!
60      m_bIsRunning = true;
61      QThread::start();
62      return true;
63  }
64
65  bool ExamplePlugin::stop()
66  {
67      // Make the run loop exit by setting isRunning to false
68      m_bIsRunning = false;
69
70      // Abort any push push or pop operations that are in progress and clear the buffer
71      m_pExampleBuffer->releaseFromPop();
72      m_pExampleBuffer->releaseFromPush();
73      m_pExampleBuffer->clear();
74
75      return true;
76  }
77
78  // Each time new data is received, place it into the buffer to be processed
79  void ExamplePlugin::update(NewMeasurement::SPtr pMeasurement)
80  {
81      // The measurement data that comes in is generic measurement data
82      // We need to tell the program to treat it as a NewRealTimeMultiSampleArray
83      QSharedPointer<NewRealTimeMultiSampleArray> pRTMSA = pMeasurement.dynamicCast<NewRealTimeMultiSampleArray>();
84
85      // Check if fiff info has been set yet
86      if (!m_pFiffInfo){
```

4. Use the stop() function as a place to stop the thread by forcing run() to return.

```
11      , m_pExampleOutput(NULL)
12      , m_pExampleBuffer(CircularMatrixBuffer<double>::SPtr())
13  {
14
15      // Add a new action to the toolbar     QAction( icon , title, parent )
16      QAction* showSettingsWidgetAction = new QAction(QIcon(":/images/options.png"), tr("Toolbar Widget"),this);
17      showSettingsWidgetAction->setShortcut(tr("F12"));
18      showSettingsWidgetAction->setStatusTip(tr("Title of Toolbar Here"));
19      addPluginAction(showSettingsWidgetAction);
20
21      // connect the action to the class member function
22      connect(showSettingsWidgetAction, &QAction::triggered, this, &ExamplePlugin::showSettings );
23  }
24
25  // Destructor
26  ExamplePlugin::~ExamplePlugin()
27  {
28      if (this->isRunning())
29          stop();
30  }
31
32  // Init, clone, unload
33  void ExamplePlugin::init(){
34
35      // Add an input
36      m_pExampleInput = PluginInputData<NewRealTimeMultiSampleArray>::create(this, "ExampleInput", "Example Plugin's input data");
37      m_inputConnectors.append(m_pExampleInput);
38
39      // Add an ouput
40      m_pExampleOutput = Plug                                    e(this, "ExampleOut", "Example Plugin's output data");
41      m_outputConnectors.appe
42
43      // Register for updates
44      connect(m_pExampleInput                                    xamplePlugin::update, Qt::DirectConnection);
45  }
46
47
48  void ExamplePlugin::unload(
49
50  QSharedPointer<IPlugin> Exa
51  {
52      QSharedPointer<ExampleP
53      return pointerToExamplePlugin;
54  }
55
```

Call the stop function from the class destructor

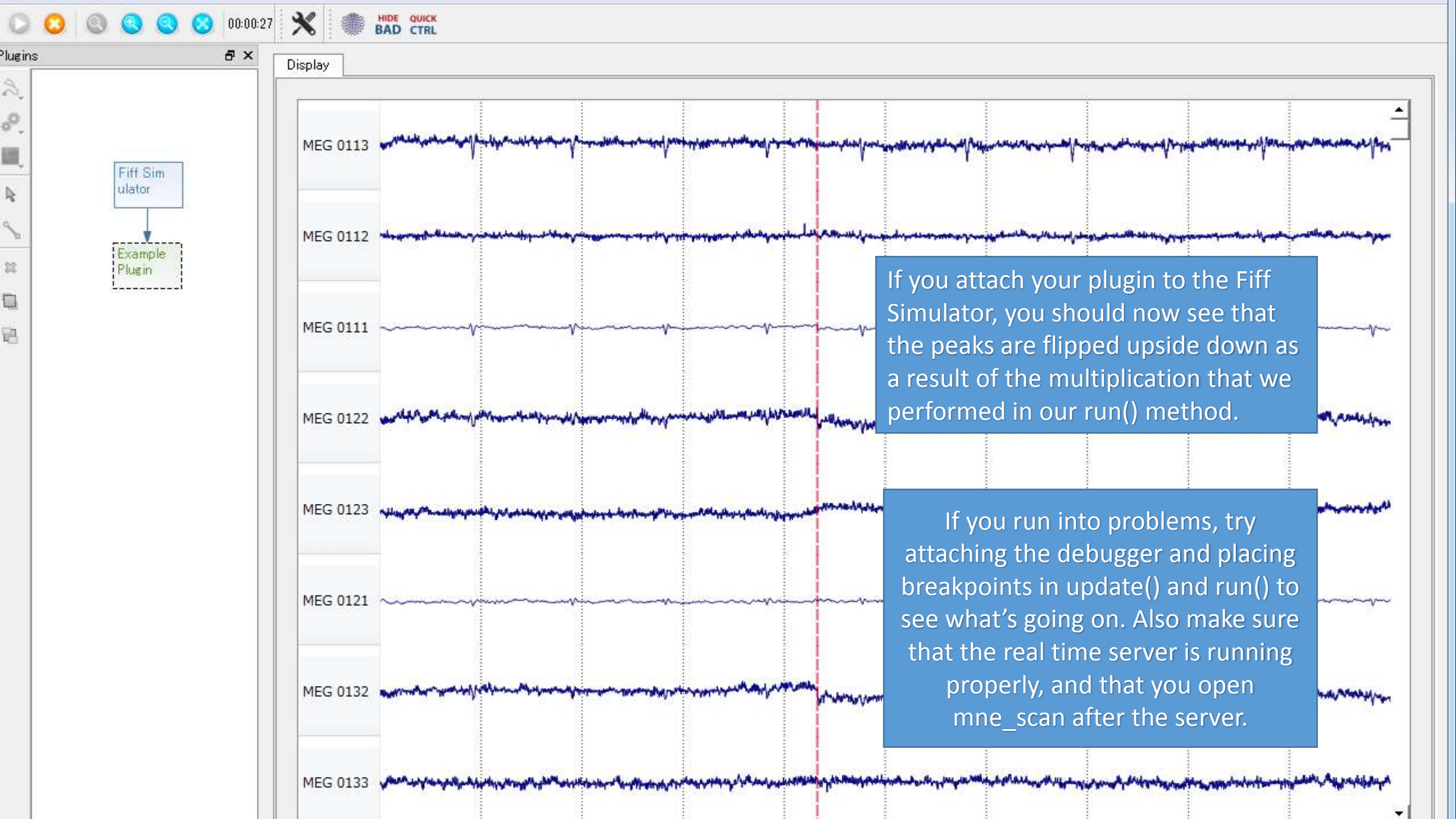
If the user closes the window without pressing the stop button, the plugin could be released from memory before the thread is stopped. This will ensure stop is always called first.

We now have the plug setup to begin running, push fresh measurements onto a buffer, and stop when commanded to.

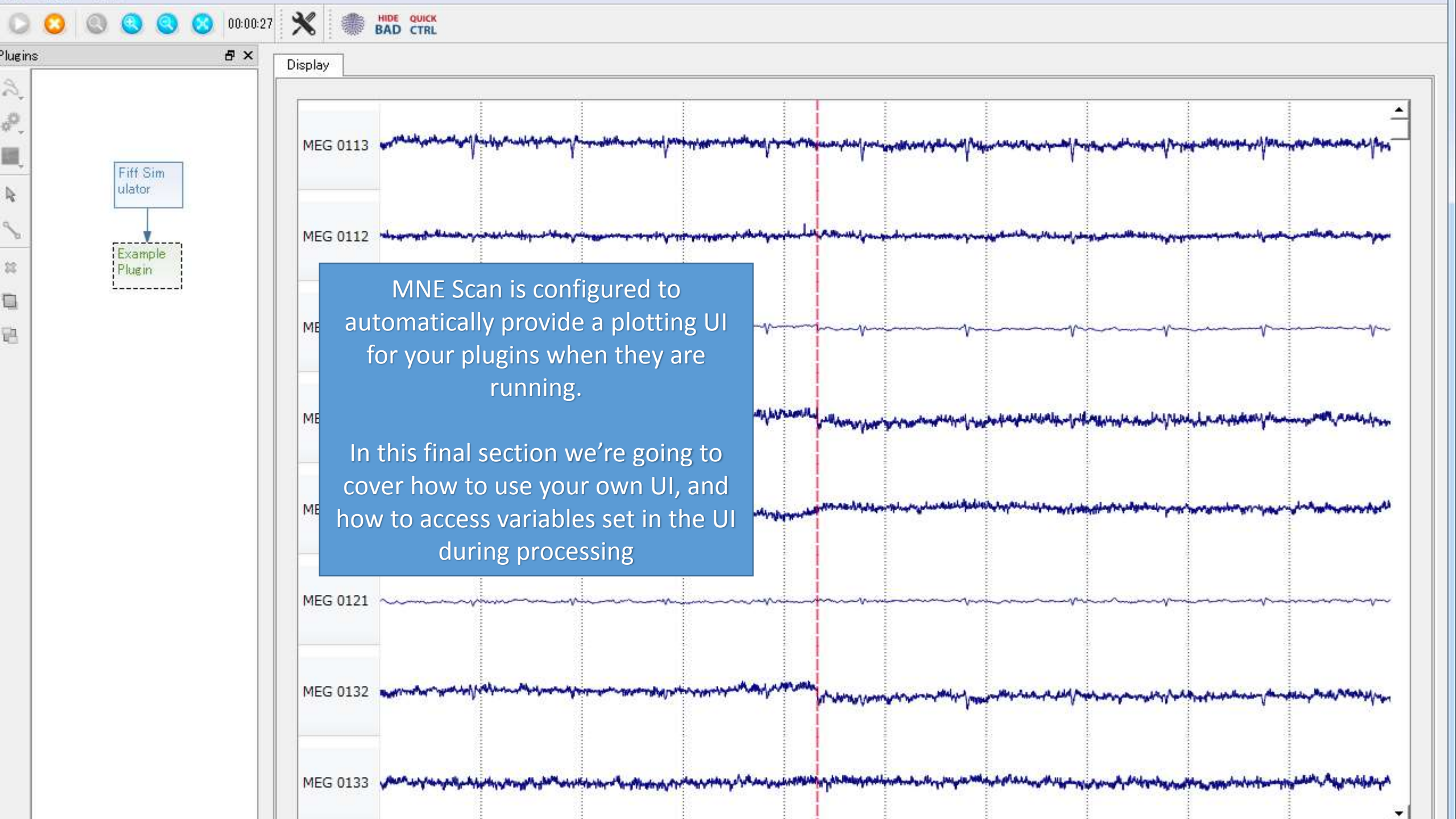
The final steps are getting data out of the buffer, processing it, and sending the output to other plugins.

```
96      MatrixXd t_mat;
97
98      // Determine how many time samples have been collected since the last update
99      // For each time sample, append the sample array to the buffer so it gets processed
100     for (unsigned char i = 0; i < pRTMSA->getMultiArraySize(); ++i) {
101         t_mat = pRTMSA->getMultiSampleArray()[i];
102         m_pExampleBuffer->push(&t_mat);
103     }
104 }
105
106 void ExamplePlugin::run()
107 {
108
109     // Check every 10ms until the fiff info has been set
110     while(!m_pFiffInfo)
111         msleep(10);
112
113     // run() will return as soon as m_bIsRunning is set to false
114     // Otherwise it will continually process any data in the buffer
115     while (m_bIsRunning)
116     {
117         // 1. Get data from the buffer
118         MatrixXd t_mat = m_pExampleBuffer->pop();
119
120         // 2. Do some kind of processing
121         t_mat = t_mat * (-3);
122
123         // 3. Send the data out to any connected plugins
124         // This will also update the display
125         m_pExampleOutput->data()->setValue(t_mat);
126     }
127 }
128
129 // 'const' means that this function won't modify the object
130 // Ctrl + Click on PluginType to see a list of options
131 IPlugin::PluginType ExamplePlugin::getType() const
132 {
133     return _IAlgorithm;
134 }
135
136 QString ExamplePlugin::getName() const
137 {
138     return "Example Plugin";
139 }
```

Go back to our // TODO: comment and add in the following


00:00:27
HIDE BAD
QUICK CTRL
Plugins
Fiff Sim ulator
Example Plugin
Display
MEG 0113
MEG 0112
MEG 0111
MEG 0122
MEG 0123
MEG 0121
MEG 0132
MEG 0133
If you attach your plugin to the Fiff Simulator, you should now see that the peaks are flipped upside down as a result of the multiplication that we performed in our run() method.
If you run into problems, try attaching the debugger and placing breakpoints in update() and run() to see what's going on. Also make sure that the real time server is running properly, and that you open mne_scan after the server.

Output Window

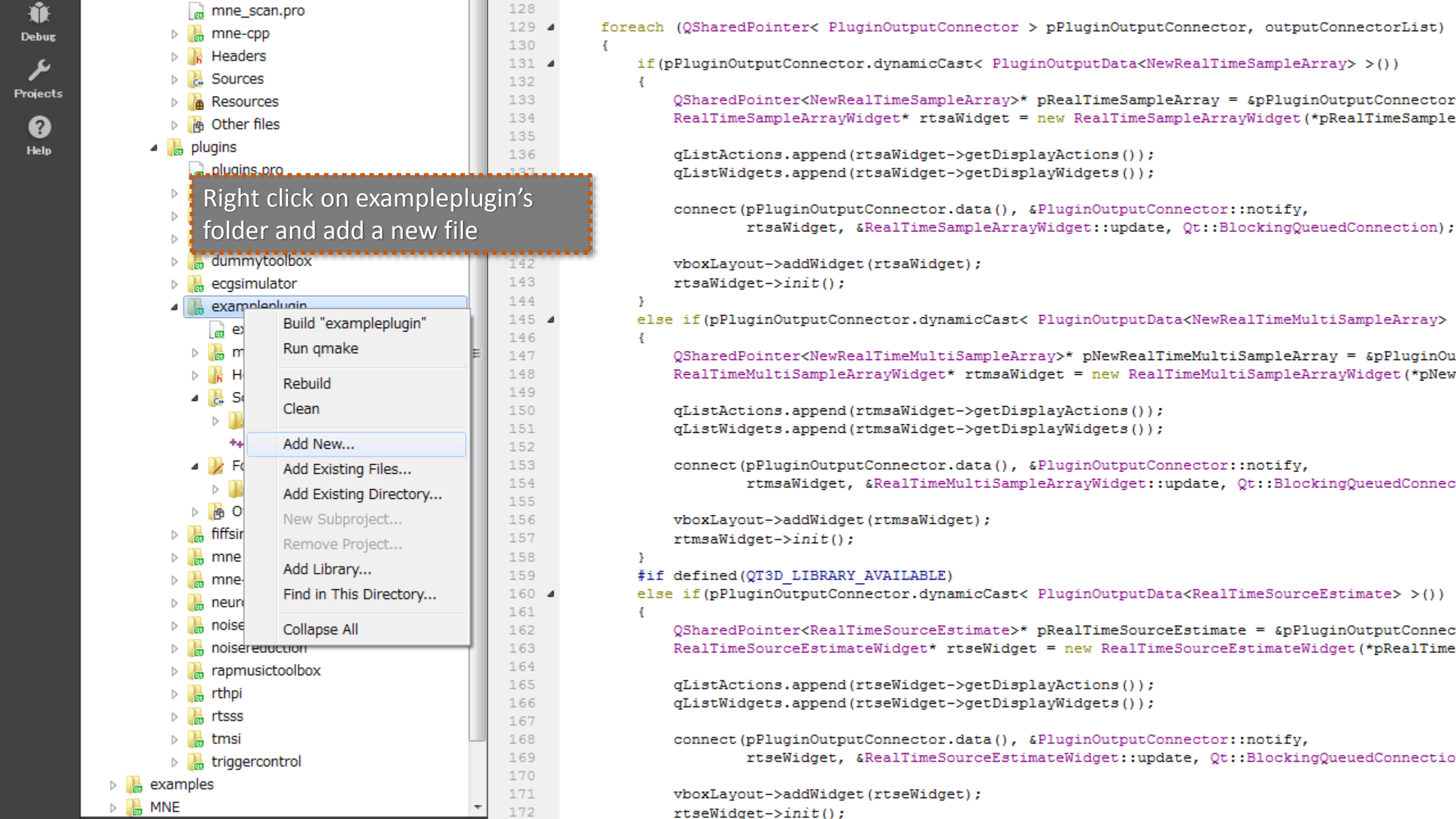

00:00:27
HIDE BAD
QUICK CTRL
Plugins
Fiff Sim ulator
Example Plugin
Display
MEG 0113
MEG 0112
MEG 0121
MEG 0132
MEG 0133
MNE Scan is configured to automatically provide a plotting UI for your plugins when they are running.
In this final section we're going to cover how to use your own UI, and how to access variables set in the UI during processing

```
    QWidget* newDisp = new QWidget;
    QVBoxLayout* vboxLayout = new QVBoxLayout;
    QHBoxLayout* hboxLayout = new QHBoxLayout;

    qListActions.clear();

    foreach (QSharedPointer< PluginOutputConnector > pPluginOutputConnector, outputConnectorList)
    {
        if(pPluginOutputConnector.dynamicCast< PluginOutputData<NewRealTimeSampleArray> >())
        {
            QSharedPointer<NewRealTimeSampleArray>* pRealTimeSampleArray = &pPluginOutputConnector.dynamicCast< PluginOutputData<NewRealTimeSampleArray> >
            RealTimeSampleArrayWidget* rtsaWidget = new RealTimeSampleArrayWidget(*pRealTimeSampleArray, pT, newDisp);

            qListActions.append(rtsaWidget->getDisplayActions());
            qListWidgets.append(rtsaWidget->getDisplayWidgets());

            connect(pPluginOutputConnector.data(), &PluginOutputConnector::notify,
                    rtsaWidget, &RealTimeSampleArrayWidget::update, Qt::BlockingQueuedConnection);

            vboxLayout->addWidget(rtsaWidget);
            rtsaWidget->init();
        }
        else if(pPluginOutputConnector.dynamicCast< PluginOutputData<NewRea
        {
            QSharedPointer<NewRealTimeMultiSampleArray>* pNewRealTimeMultiS
            RealTimeMultiSampleArrayWidget* rtmsaWidget = new RealTimeMulti

            qListActions.append(rtmsaWidget->getDisplayActions());
            qListWidgets.append(rtmsaWidget->getDisplayWidgets());

            connect(pPluginOutputConnector.data(), &PluginOutputConnector::
                    rtmsaWidget, &RealTimeMultiSampleArrayWidget::update, Q

            vboxLayout->addWidget(rtmsaWidget);
            rtmsaWidget->init();
        }
        #if defined(QT3D_LIBRARY_AVAILABLE)
        else if(pPluginOutputConnector.dynamicCast< PluginOutputData<RealTi
        {
            QSharedPointer<RealTimeSourceEstimate>* pRealTimeSourceEstimate
            RealTimeSourceEstimateWidget* rtseWidget = new RealTimeSourceEs

            qListActions.append(rtseWidget->getDisplayActions());
            qListWidgets.append(rtseWidget->getDisplayWidgets());
```

In this function from applications/mne_scan/libs/scShared/Sources/Management/displaymanager.cpp we can see that when the run button is pressed, mne_scan cycles through all the outputs and creates a real time widget for that plugin based on the type of data that the output is

If your plugin absolutely must use the main widow, you'll have to modify this function.

For most cases, it's probably appropriate to simply create an extra widow and show what your plugin is doing there. That's what we'll do here.


Debug
Projects
Help
mne_scan.pro
mne-cpp
Headers
Sources
Resources
Other files
plugins
plugins.pro
dummytoolbox
ecgsimulator
exampleplugin
Build "exampleplugin"
Run qmake
Rebuild
Clean
Add New...
Add Existing Files...
Add Existing Directory...
New Subproject...
Remove Project...
Add Library...
Find in This Directory...
Collapse All
fiffsi
mne
mne-
neur
noise
noisereduction
rapmusictoolbox
rthpi
rtsss
tmsi
triggercontrol
examples
MNE
Right click on exampleplugin's folder and add a new file
128
129 foreach (QSharedPointer< PluginOutputConnector > pPluginOutputConnector, outputConnectorList)
130 {
131 if(pPluginOutputConnector.dynamicCast< PluginOutputData<NewRealTimeSampleArray> >())
132 {
133 QSharedPointer<NewRealTimeSampleArray>* pRealTimeSampleArray = &pPluginOutputConnector
134 RealTimeSampleArrayWidget* rtsaWidget = new RealTimeSampleArrayWidget(*pRealTimeSample
135
136 qListActions.append(rtsaWidget->getDisplayActions());
137 qListWidgets.append(rtsaWidget->getDisplayWidgets());
138
139 connect(pPluginOutputConnector.data(), &PluginOutputConnector::notify,
140 rtsaWidget, &RealTimeSampleArrayWidget::update, Qt::BlockingQueuedConnection);
141
142 vboxLayout->addWidget(rtsaWidget);
143 rtsaWidget->init();
144 }
145 else if(pPluginOutputConnector.dynamicCast< PluginOutputData<NewRealTimeMultiSampleArray>
146 {
147 QSharedPointer<NewRealTimeMultiSampleArray>* pNewRealTimeMultiSampleArray = &pPluginOu
148 RealTimeMultiSampleArrayWidget* rtmsaWidget = new RealTimeMultiSampleArrayWidget(*pNew
149
150 qListActions.append(rtmsaWidget->getDisplayActions());
151 qListWidgets.append(rtmsaWidget->getDisplayWidgets());
152
153 connect(pPluginOutputConnector.data(), &PluginOutputConnector::notify,
154 rtmsaWidget, &RealTimeMultiSampleArrayWidget::update, Qt::BlockingQueuedConnec
155
156 vboxLayout->addWidget(rtmsaWidget);
157 rtmsaWidget->init();
158 }
159 #if defined(QT3D_LIBRARY_AVAILABLE)
160 else if(pPluginOutputConnector.dynamicCast< PluginOutputData<RealTimeSourceEstimate> >())
161 {
162 QSharedPointer<RealTimeSourceEstimate>* pRealTimeSourceEstimate = &pPluginOutputConnec
163 RealTimeSourceEstimateWidget* rtseWidget = new RealTimeSourceEstimateWidget(*pRealTime
164
165 qListActions.append(rtseWidget->getDisplayActions());
166 qListWidgets.append(rtseWidget->getDisplayWidgets());
167
168 connect(pPluginOutputConnector.data(), &PluginOutputConnector::notify,
169 rtseWidget, &RealTimeSourceEstimateWidget::update, Qt::BlockingQueuedConnectio
170
171 vboxLayout->addWidget(rtseWidget);
172 rtseWidget->init();

displaymanager.cpp - mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

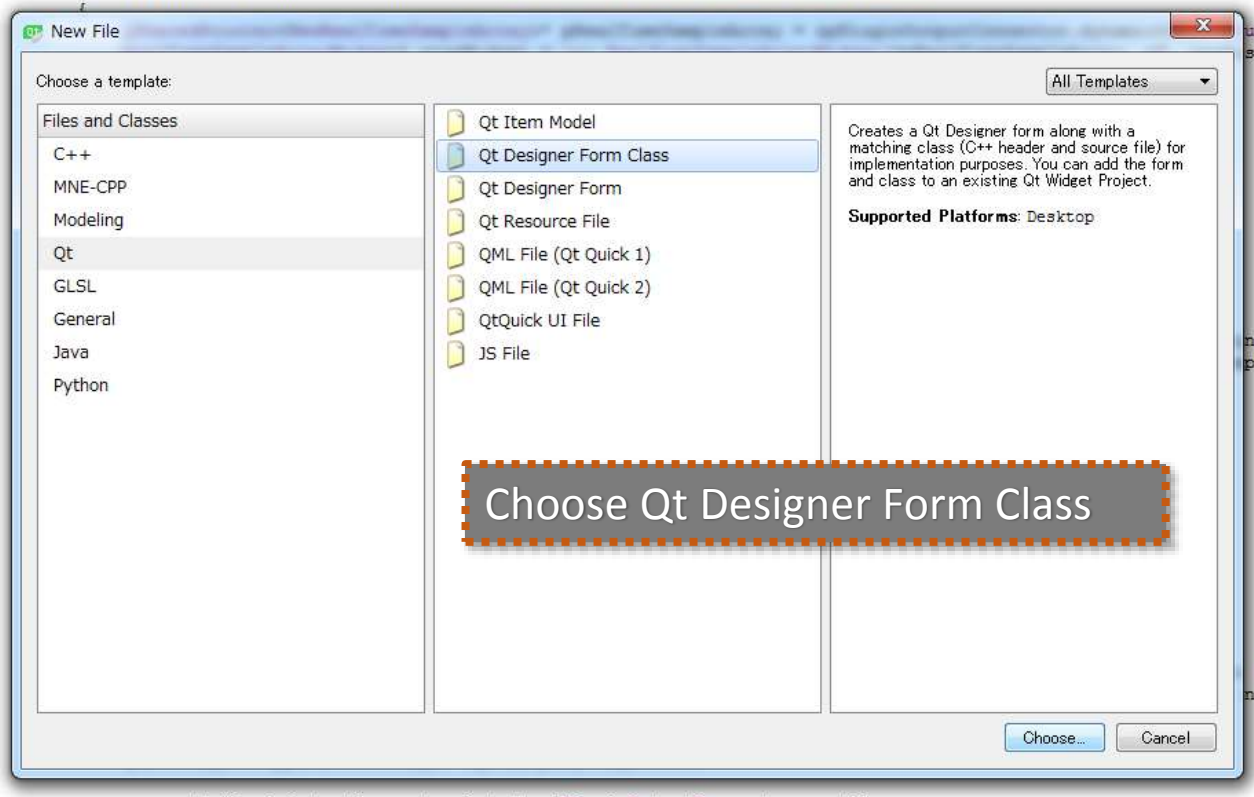
- mne_matching_pursuit
- mne_rt_server
- mne_sample_set_downloader
- mne_scan
  - mne_scan.pro
  - libs
  - mne-cpp
  - mne_scan
    - mne_scan.pro
    - mne-cpp
    - Headers
    - Sources
    - Resources
    - Other files
  - plugins
    - plugins.pro
    - averaging
    - babymeg
    - covariance
    - dummytoolbox
    - ecgsimulator
    - exampleplugin
      - exampleplugin.pro
      - mne-cpp
      - Headers
      - Sources
        - FormFiles
        - exampleplugin.cpp
      - Forms
        - FormFiles
      - Other files
    - fiffsimulator
    - mne
    - mne-cpp
    - neuromag
    - noise
    - noisereduction
    - rapmusictoolbox
    - rthpi
    - rtsss
    - tmsi
    - triggercontrol
- examples
- MNE

Open Documents

displaymanager.cpp

displaymanager.cpp | DisplayManager::show(IPlugin::OutputConnectorList &, QSharedPointer<QTime> &, QList<QAction *> &, QList<QWidget *> &): QWidget * | Line: 125, C

```
118
119 //*************************************************************************************************************
120
121 QWidget* DisplayManager::show(IPlugin::OutputConnectorList &outputConnectorList, QSharedPointer<QTime>& pT, QList< QAction* >& qListActions, QList< QWidget* >& qListWidget
122 {
123     QWidget* newDisp = new QWidget;
124     QVBoxLayout* vboxLayout = new QVBoxLayout;
125     QHBoxLayout* hboxLayout = new QHBoxLayout;
126
127     qListActions.clear();
128
129     foreach (QSharedPointer< PluginOutputConnector > pPluginOutputConnector, outputConnectorList)
130     {
131         if(pPluginOutputConnector.dynamicCast< PluginOutputData<NewRealTimeSampleArray> >())
132         {
133                                                                                                    uginOutputData<NewRealTimeSampleArray> >()->data();
134                                                                                                    sp);
135
136
137
138
139
140
141
142
143
144
145
146
147                                                                                                    namicCast< PluginOutputData<NewRealTimeMultiSampleArray>
148                                                                                                    pleArray, pT, newDisp);
149
150
151
152
153
154
155
156
157
158
159
160
161
162                                                                                                    PluginOutputData<RealTimeSourceEstimate> >()->data();
163                                                                                                    newDisp);//new RealTimeSourceEstimateWidget(*pRealTimeSou
164
165
166
167
168             connect(pPluginOutputConnector.data(), &PluginOutputConnector::notify,
169                     rtseWidget, &RealTimeSourceEstimateWidget::update, Qt::BlockingQueuedConnection);
170
171             vboxLayout->addWidget(rtseWidget);
172             rtseWidget->init();
173         }
174         #endif
175         else if(pPluginOutputConnector.dynamicCast< PluginOutputData<RealTimeEvoked> >())
176         {
177             QSharedPointer<RealTimeEvoked>* pRealTimeEvoked = &pPluginOutputConnector.dynamicCast< PluginOutputData<RealTimeEvoked> >()->data();
178
```

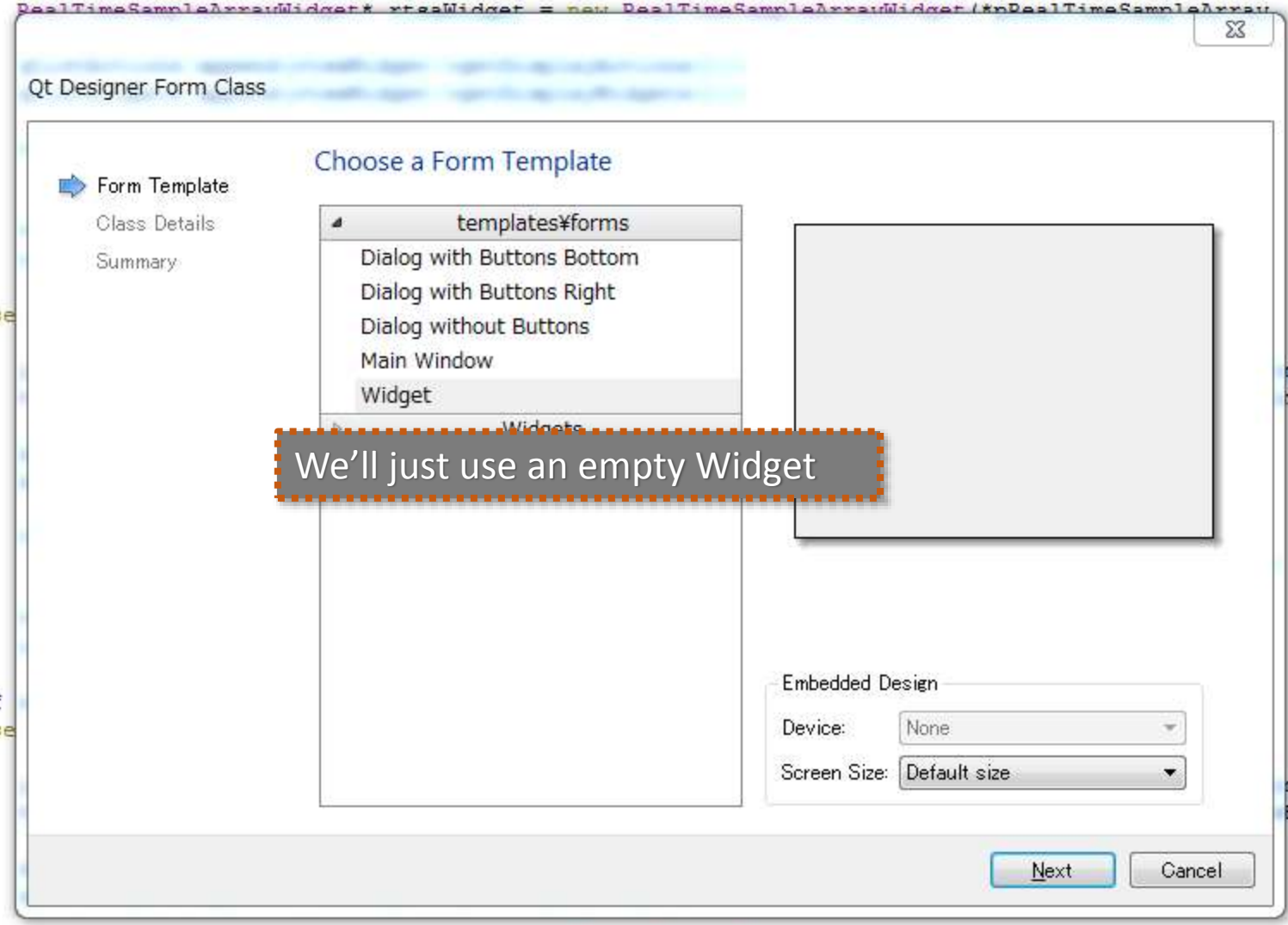


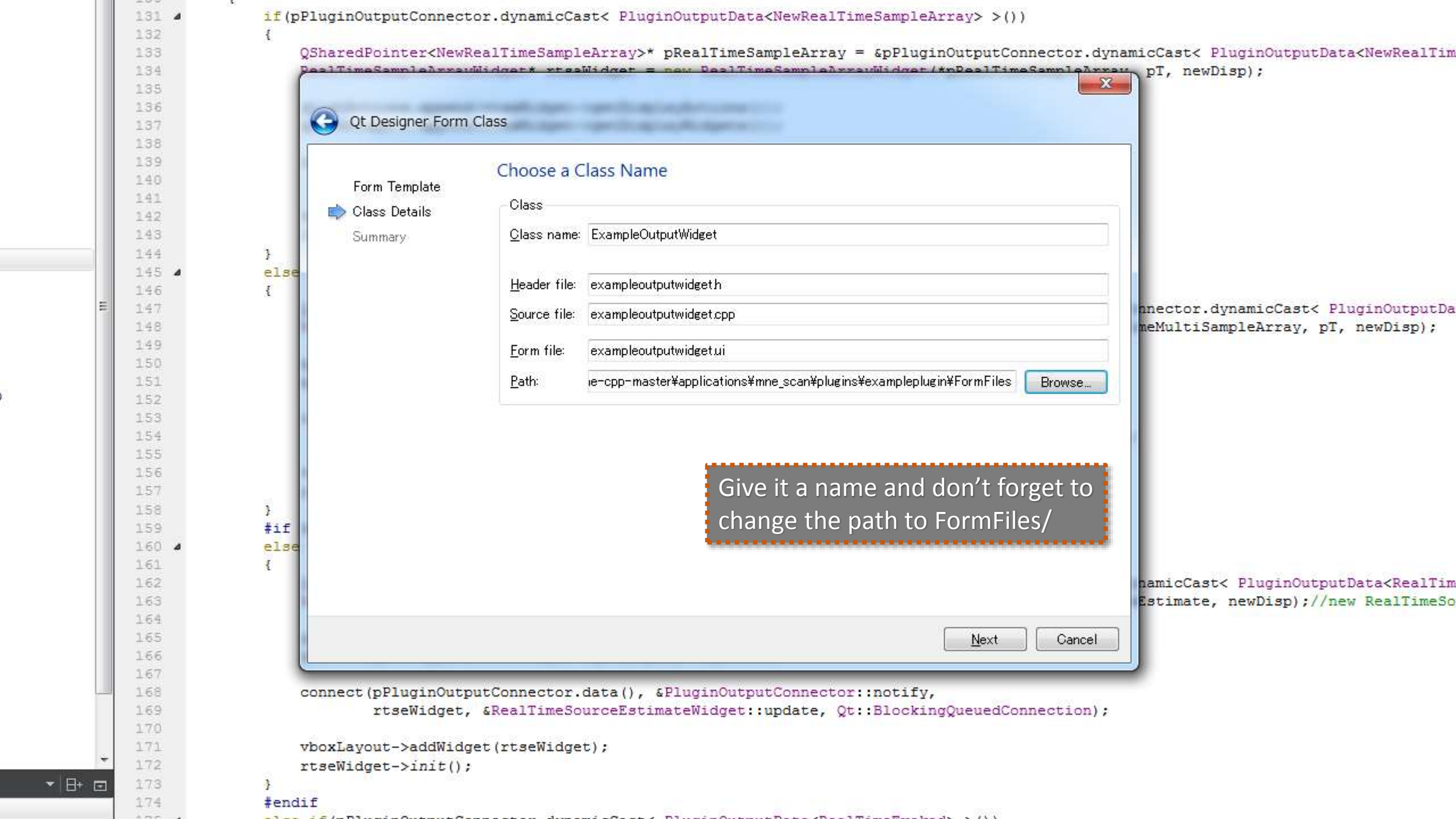
Choose Qt Designer Form Class

Welcome Edit Design Debug Projects Help

mne-cpp Release

```
129     foreach (QSharedPointer< PluginOutputConnector > pPluginOutputConnector, outputConnectorList)
130     {
131         if(pPluginOutputConnector.dynamicCast< PluginOutputData<NewRealTimeSampleArray> >())
132         {
133             QSharedPointer<NewRealTimeSampleArray>* pRealTimeSampleArray = &pPluginOutputConnector.dynamicCast< PluginOutputData<NewRealTi
134             RealTimeSampleArrayWidget* rtseaWidget = new RealTimeSampleArrayWidget(*pRealTimeSampleArray, pT, newDisp);
135
136
137
138
139
140
141
142
143
144         }
145         else
146         {
147                                                                                                    nnector.dynamicCast< PluginOutputD
148                                                                                                    meMultiSampleArray, pT, newDisp);
149
150
151
152
153
154
155
156
157
158         }
159         #if
160         else
161         {
162                                                                                                    namicCast< PluginOutputData<RealTi
163                                                                                                    Estimate, newDisp);//new RealTimeS
164
165
166
167
168             connect(pPluginOutputConnector.data(), &PluginOutputConnector::notify,
169                     rtseWidget, &RealTimeSourceEstimateWidget::update, Qt::BlockingQueuedConnection);
170
171             vboxLayout->addWidget(rtseWidget);
172             rtseWidget->init();
```




Qt Designer Form Class
Form Template
Class Details
Summary
Choose a Class Name
Class
Class name: ExampleOutputWidget
Header file: exampleoutputwidget.h
Source file: exampleoutputwidget.cpp
Form file: exampleoutputwidget.ui
Path: e-cpp-master¥applications¥mne_scan¥plugins¥exampleplugin¥FormFiles
Browse...
Give it a name and don't forget to change the path to FormFiles/
Next
Cancel

exampleoutputwidget.ui - mne-cpp - Qt Creator

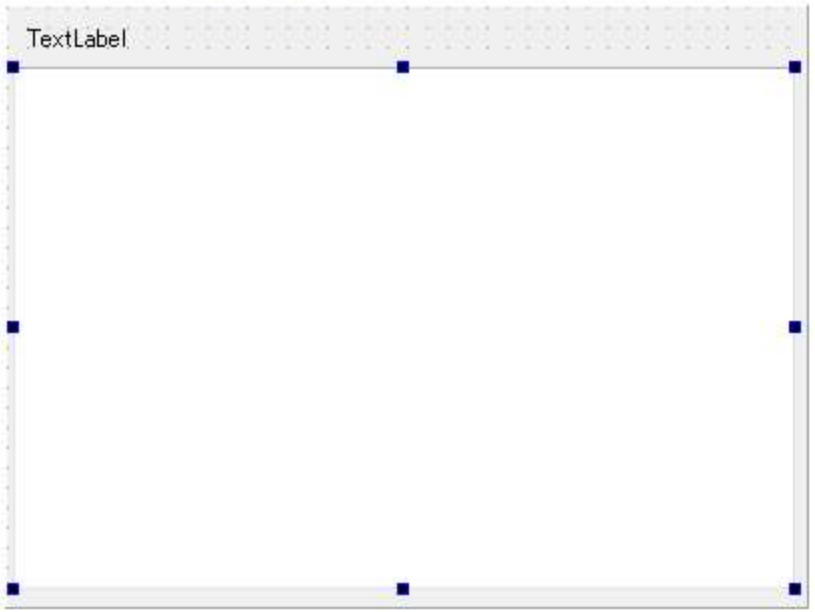
File Edit Build Debug Analyze Tools Window Help

exampleoutputwidget.ui*

Welcome
Edit
Design
Debug
Projects
Help

Filter

- Table View
- Column View
- Item Wi···-Based)
- List Widget
- Tree Widget
- Table Widget
- Containers
- Group Box
- Scroll Area
- Tool Box
- Tab Widget
- Stacked Widget
- Frame
- Widget
- MDI Area
- Dock Widget
- QAxWidget
- Input Widgets
- Combo Box
- Font C···bo Box
- Line Edit
- Text Edit
- Plain Text Edit
- Spin Box
- Double Spin Box
- Time Edit



Add a text label and a text edit to the new form file.

The label will show information about the selected channel and the text edit will show measurement values from the selected channel

exampleplugin.h - mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

- mne_matching_pursuit
- mne_rt_server
- mne_sample_set_downloader
- mne-cpp
  - Headers
  - Sources
  - Resources
  - Other files
- plugins
  - plugins.pro
  - averaging
  - babymeg
  - covariance
  - dummytoolbox
  - ecgsimulator
  - exampleplugin
    - exampleplugin.pro
    - mne-cpp
    - Headers
      - FormFiles
      - exampleplugin.h
      - exampleplugin_global.h
    - Sources
      - FormFiles
      - exampleplugin.cpp
    - Forms

Welcome
Debug
Projects
Help

The first step is to make this new window appear when the plugin starts.

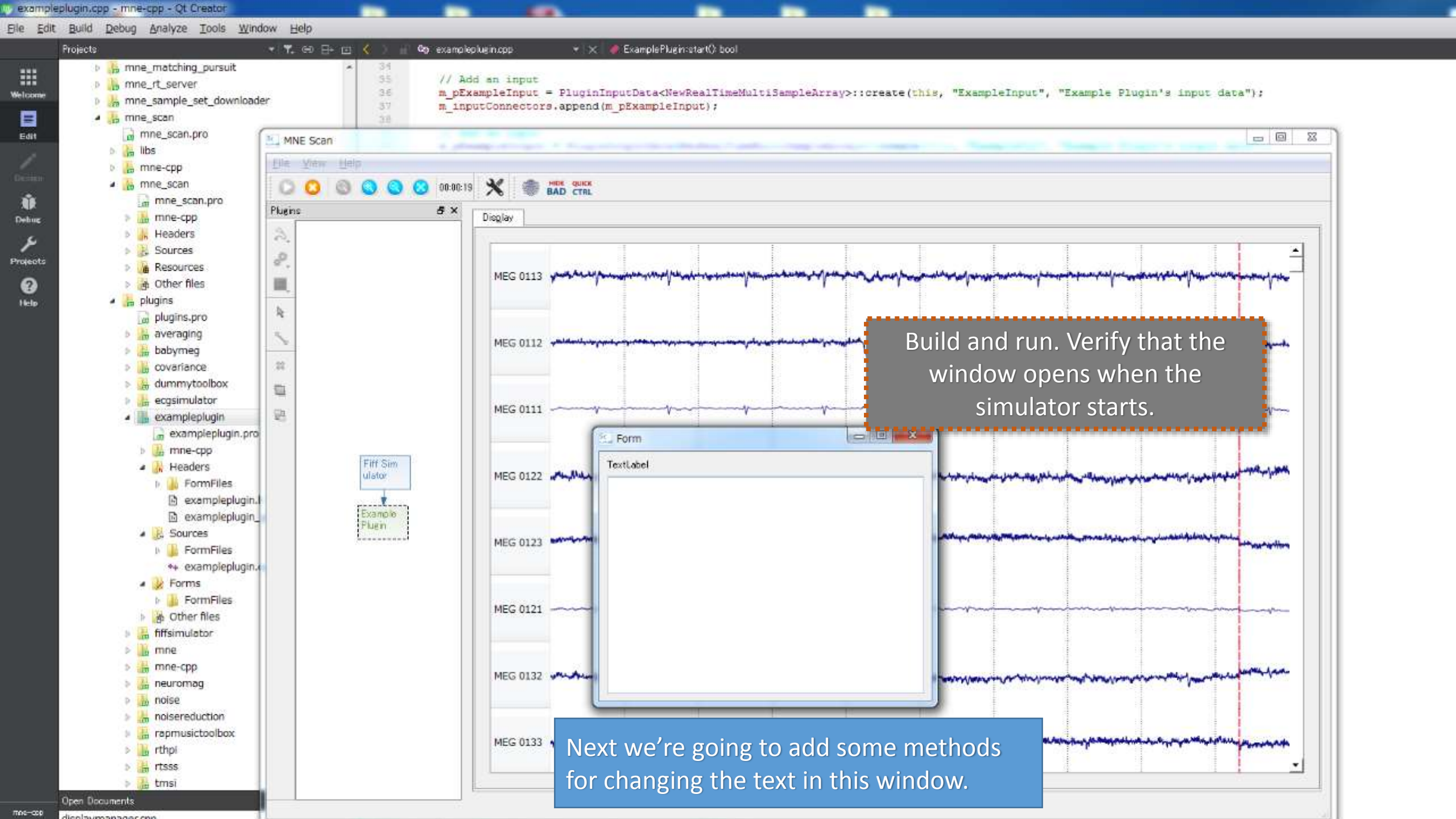
exampleplugin.h | <Select Symbol>

```
#ifndef EXAMPLEPLUGIN_H
#define EXAMPLEPLUGIN_H

#include "exampleplugin_global.h"
#include "FormFiles/examplegui.h"
#include "FormFiles/settingswidget.h"
#include "FormFiles/exampleoutputwidget.h"

#include <scShared/Interfaces/IAlgorithm.h>
#include <scShared/Interfaces/ISensor.h>

#include <scMeas/newrealtimemultisamplearray.h>
#include <generics/circularmatrixbuffer.h>

using namespace SCSHAREDLIB;

class EXAMPLEPLUGINSHARED_EXPORT ExamplePlugin : public SCSHAREDLIB::IAlgorithm
{
    // 1. All subclasses of Qt's QObject must include the Q_OBJECT macro
    // It instructs the meta object system (mos) to automatically setup certain boiler plate functi
    Q_OBJECT

    // 2. Declare to the compiler how this plugin connects. The IID "scsharedlib/1.0" is where all
    // FILE is a .json file containing metadata about the plugin. We'll create this file next, for
    // Using this requires the class to be default-constructible (can create an instance without an
    Q_PLUGIN_METADATA(IID "scsharedlib/1.0" FILE "exampleplugin.json")

    // 3. Alert the meta object system about this plugin
    Q_INTERFACES(SCSHAREDLIB::IAlgorithm)

public:

    // See (2.)
    // This constructor must not take any arguments in order for this class to be a plugin
    ExamplePlugin();

    // Deconstructor for Example Plugin
    ~ExamplePlugin();

    // IAlgorithm functions
```

Import the new file into your plugin's header file.

```
45  }
46
47
48  void ExamplePlugin::unload(){}
49
50  QSharedPointer<IPlugin> ExamplePlugin::clone() const
51  {
52      QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
53      return pointerToExamplePlugin;
54  }
55
56  bool ExamplePlugin::start()
57  {
58      // Set isRunning to true and then call run
59      // The order will be important!
60      m_bIsRunning = true;
61      QThread::start();
62
63      // Show the output window
64      // Making the parent NULL creates the widget in a new window
65      ExampleOutputWidget* outputWidget = new ExampleOutputWidget(NULL);
66      outputWidget->show();
67
68      return true;
69  }
70
71  void ExamplePlugin::run()
72  {
73
74      // Check every 10ms until the fiff info has been set
75      while(!m_pFiffInfo)
76          msleep(10);
77
78      // run() will return as soon as m_bIsRunning is set to false
79      // Otherwise it will continually process any data in the buffer
80      while (m_bIsRunning)
81      {
82          // 1. Get data from the buffer
83          MatrixXd t_mat = m_pExampleBuffer->pop();
84
85          // 2. Do some kind of processing
86          t_mat = - t_mat;
87
88          // 3. Send the data out to any connected plugins
```

In the source file's start() method, add code to create and show the output window we just created.


exampleplugin.cpp - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
mne_scan.pro
mne-cpp
Headers
Sources
Resources
Other files
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
FormFiles
Sources
FormFiles
Forms
FormFiles
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
rthpi
rtsss
tmsi
Open Documents
exampleplugin.cpp
ExamplePlugin::start() bool
// Add an input
m_pExampleInput = PluginInputData<NewRealTimeMultiSampleArray>::create(this, "ExampleInput", "Example Plugin's input data");
m_inputConnectors.append(m_pExampleInput);
MNE Scan
File View Help
00:00:19
Plugins
Display
Fiff Simulator
Example Plugin
MEG 0113
MEG 0112
MEG 0111
MEG 0122
MEG 0123
MEG 0121
MEG 0132
MEG 0133
Form
TextLabel
Build and run. Verify that the window opens when the simulator starts.
Next we're going to add some methods for changing the text in this window.

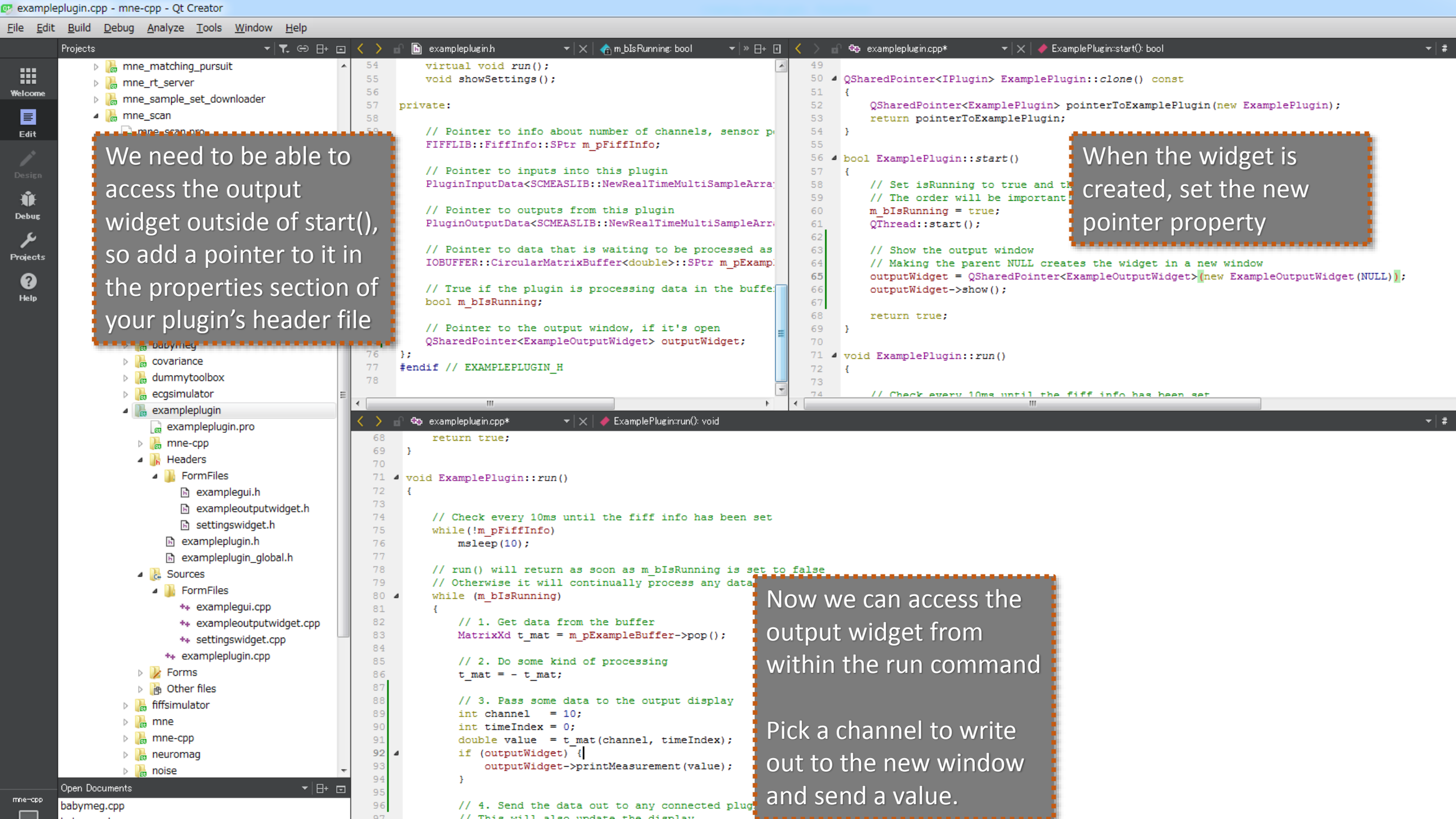
```
    Q_OBJECT

public:
    explicit ExampleOutputWidget(QWidget *parent = 0);
    ~ExampleOutputWidget();

    // Set the label show what sensor is selected
    void writeLabel(QString title);

    // Print out the value of the measurement to the text browser
    void printMeasurement(double value);

private:
    Ui::ExampleOutputWidget *ui;
};

#endif // EXAMPLEOUTPUTWIDGET_H
```

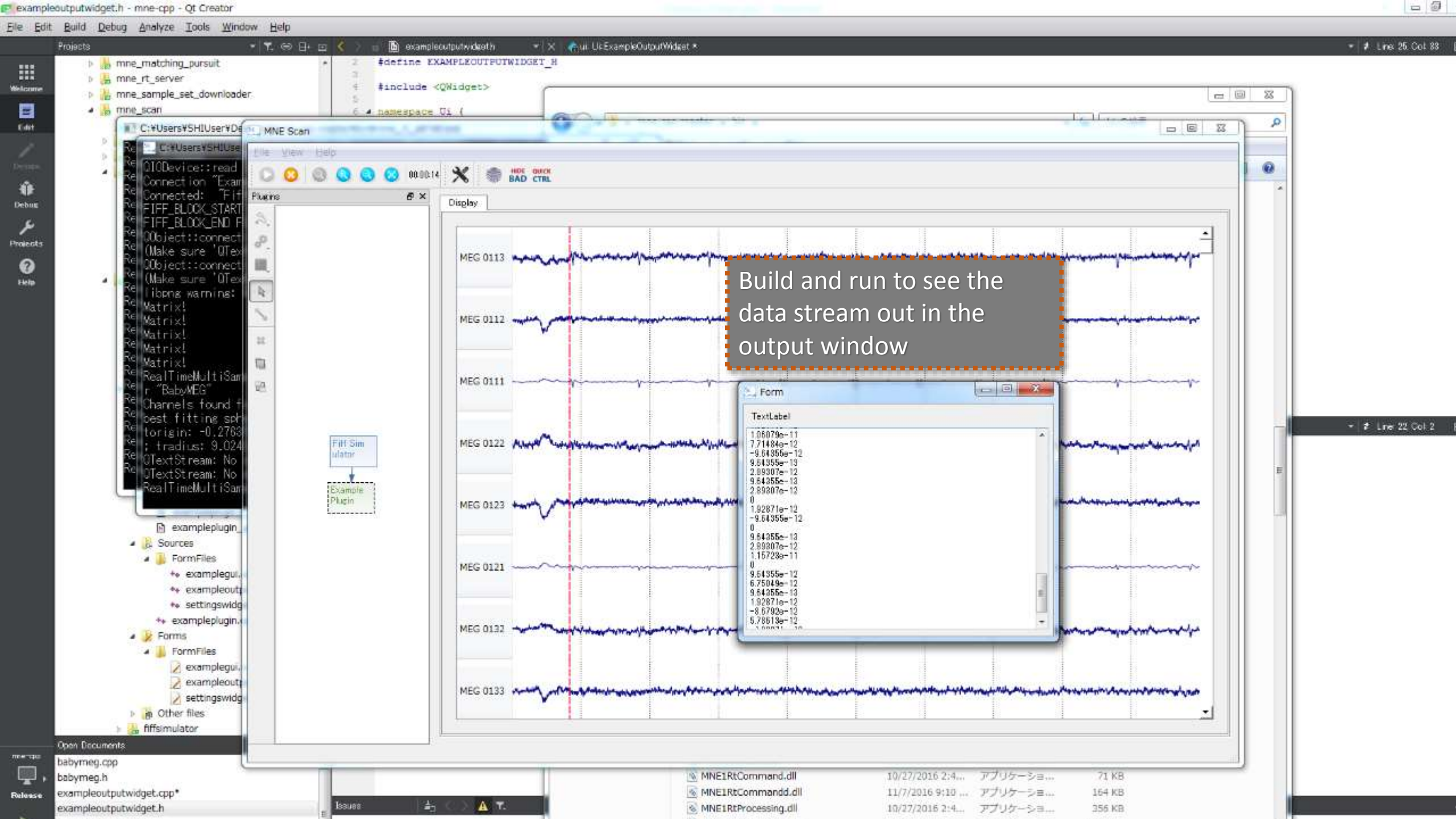
Add declarations in the header file

exampleoutputwidget.cpp* | ExampleOutputWidget::printMeasurement(double): void

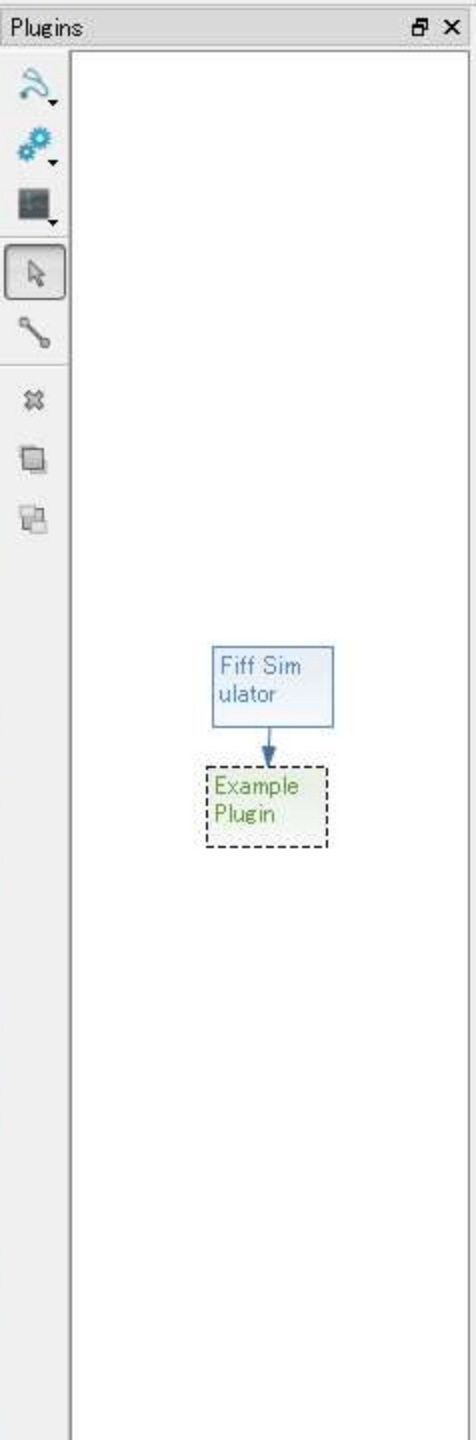
```
#include "exampleoutputwidget.h"
#include "ui_exampleoutputwidget.h"

ExampleOutputWidget::ExampleOutputWidget(QWidget *parent) :
    QWidget(parent),
    ui(new Ui::ExampleOutputWidget)
{
    ui->setupUi(this);
}

ExampleOutputWidget::~ExampleOutputWidget()
{
    delete ui;
}

void ExampleOutputWidget::writeLabel(QString title)
{
    ui->label->setText(title);
}

void ExampleOutputWidget::printMeasurement(double value)
{
    ui->textEdit->append(QString::number(value));
}
```

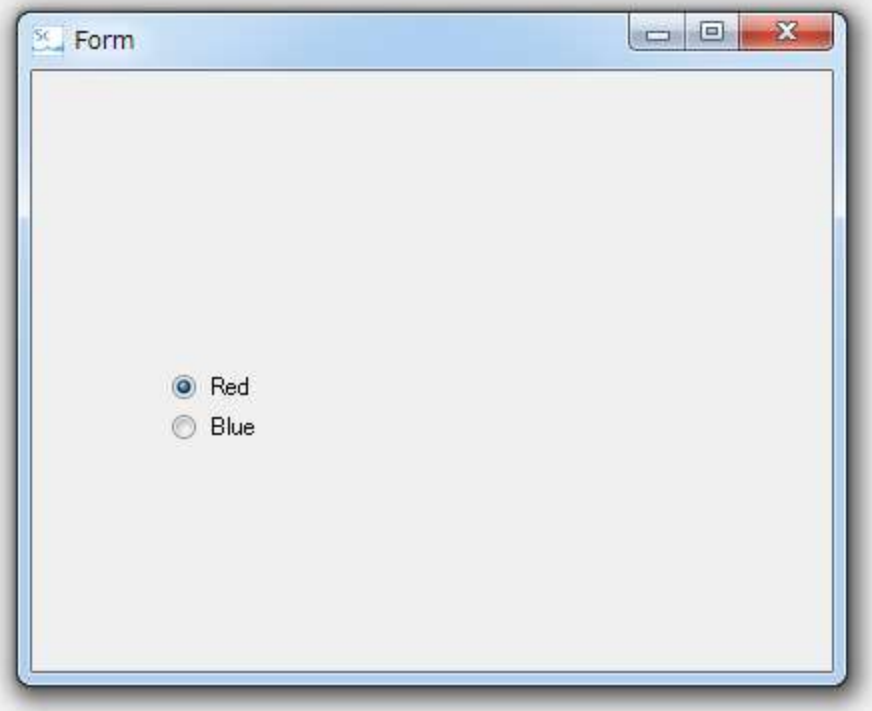
Flesh out the functions in the source file


examplepluginplugin.cpp - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
FormFiles
examplegui.h
exampleoutputwidget.h
settingswidget.h
exampleplugin.h
exampleplugin_global.h
Sources
FormFiles
examplegui.cpp
exampleoutputwidget.cpp
settingswidget.cpp
exampleplugin.cpp
Forms
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
Open Documents
babymeg.cpp
Welcome
Edit
Design
Debug
Projects
Help
mne-cpp
exampleplugin.h
m_bIsRunning: bool
virtual void run();
void showSettings();
private:
// Pointer to info about number of channels, sensor p
FIFFLIB::FiffInfo::SPtr m_pFiffInfo;
// Pointer to inputs into this plugin
PluginInputData<SCMEASLIB::NewRealTimeMultiSampleArra
// Pointer to outputs from this plugin
PluginOutputData<SCMEASLIB::NewRealTimeMultiSampleArr
// Pointer to data that is waiting to be processed as
IOBUFFER::CircularMatrixBuffer<double>::SPtr m_pExampl
// True if the plugin is processing data in the buffe
bool m_bIsRunning;
// Pointer to the output window, if it's open
QSharedPointer<ExampleOutputWidget> outputWidget;
};
#endif // EXAMPLEPLUGIN_H
We need to be able to access the output widget outside of start(), so add a pointer to it in the properties section of your plugin's header file
exampleplugin.cpp*
ExamplePlugin::start(): bool
QSharedPointer<IPlugin> ExamplePlugin::clone() const
{
QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
return pointerToExamplePlugin;
}
bool ExamplePlugin::start()
{
// Set isRunning to true and th
// The order will be important
m_bIsRunning = true;
QThread::start();
// Show the output window
// Making the parent NULL creates the widget in a new window
outputWidget = QSharedPointer<ExampleOutputWidget>(new ExampleOutputWidget(NULL));
outputWidget->show();
return true;
}
void ExamplePlugin::run()
{
When the widget is created, set the new pointer property
exampleplugin.cpp*
ExamplePlugin::run(): void
return true;
}
void ExamplePlugin::run()
{
// Check every 10ms until the fiff info has been set
while(!m_pFiffInfo)
msleep(10);
// run() will return as soon as m_bIsRunning is set to false
// Otherwise it will continually process any data
while (m_bIsRunning)
{
// 1. Get data from the buffer
MatrixXd t_mat = m_pExampleBuffer->pop();
// 2. Do some kind of processing
t_mat = - t_mat;
// 3. Pass some data to the output display
int channel = 10;
int timeIndex = 0;
double value = t_mat(channel, timeIndex);
if (outputWidget) {
outputWidget->printMeasurement(value);
}
// 4. Send the data out to any connected plug
Now we can access the output widget from within the run command
Pick a channel to write out to the new window and send a value.


exampleoutputwidget.h - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
#define EXAMPLEOUTPUTWIDGET_H
#include <QWidget>
namespace Ui {
MNE Scan
File View Help
Plugins
Display
MEG 0113
MEG 0112
MEG 0111
MEG 0122
MEG 0123
MEG 0121
MEG 0132
MEG 0133
Build and run to see the data stream out in the output window
Form
TextLabel
Fiff Simulator
Example Plugin
Sources
FormFiles
Forms
FormFiles
Other files
fiffsimulator
Open Documents
babymeg.cpp
babymeg.h
exampleoutputwidget.cpp*
exampleoutputwidget.h
Release
Issues
MNE1RtCommand.dll
MNE1RtCommandd.dll
MNE1RtProcessing.dll

Settings





The next step is to make the buttons in the settings window affect the output.

We're going to make the output text print in the color of the checked radio button.

```
protected:

    virtual void run();

    void showSettings();

private:

    // The color that output is printed in
    QColor outputColor;

    // Pointer to info about number of channels, sensor positions, sampling rate, etc
    FIFFLIB::FiffInfo::SPtr m_pFiffInfo;

    // Pointer to inputs into this plugin
    PluginInputData<SCMEASLIB::NewRealTimeMultiSampleArray>::SPtr m_pExampleInput;

    // Pointer to outputs from this plugin
```

Add a private variable for the output color and a public function for changing it in the header

Sidenote:
Why not just make the variable public and not a function at all?

With this setup, we can freely modify our internal variables and what happens within the changeColor() method, and it will not cause problems for any other classes. If other classes were allowed direct access to the variable, then modifications to exampleplugin.cpp could reverberate across other classes.

This also allows us to take an action when the variable changes. If we just set the variable directly, it's more difficult to trigger events.

exampleplugin.cpp* | ExamplePlugin::showSettings(): void

```
QString ExamplePlugin::getName() const
{
    return "Example Plugin";
}

QWidget* ExamplePlugin::setupWidget()
{
    // Setup the UI using our custom form class here
    ExampleGUI* setupWidget = new ExampleGUI();
    return setupWidget;
}

void ExamplePlugin::changeColor(QColor color)
{
    outputColor = color;
}

void ExamplePlugin::showSettings()
{
    SettingsWidget* settingsWidget = new SettingsWidget();
    settingsWidget->show();
}
```

Add an implementation for the function in the source file

```
15      explicit ExampleOutputWidget(QWidget *parent = 0);
16      ~ExampleOutputWidget();
17
18      // Set the label show what sensor is selected
19      void writeLabel(QString title);
20
21      // Print out the value of the measurement to the text browser
22      void printMeasurement(double value, QColor color);
23
24  private:
25      Ui::ExampleOutputWidget *ui;
26  };
27
28  #endif // EXAMPLEOUTPUTWIDGET_H
29
```

Edit both the header and source file of the output widget to include a QColor argument in the print measurement method.

exampleoutputwidget.cpp | ExampleOutputWidget::printMeasurement(double, QColor): void

```
2   #include "ui_exampleoutputwidget.h"
3
4   ExampleOutputWidget::ExampleOutputWidget(QWidget *parent) :
5       QWidget(parent),
6       ui(new Ui::ExampleOutputWidget)
7   {
8       ui->setupUi(this);
9   }
10
11  ExampleOutputWidget::~ExampleOutputWidget()
12  {
13      delete ui;
14  }
15
16  void ExampleOutputWidget::writeLabel(QString title)
17  {
18      ui->label->setText(title);
19  }
20
21  void ExampleOutputWidget::printMeasurement(double value, QColor color)
22  {
23      ui->textEdit->setTextColor(color);
24      ui->textEdit->append(QString::number(value));
25  }
26
```

This will allow us to pass the color into the output widget easily and make it different for each measurement

Add a line to set the text color

Find:

File Edit Build Debug Analyze Tools Window Help

Projects

- mne-cpp
- mne_scan
  - mne_scan.pro
  - mne-cpp
  - Headers
  - Sources
  - Resources
  - Other files
- plugins
  - plugins.pro
  - averaging
  - babymeg
  - covariance
  - dummytoolbox
  - ecgsimulator
  - exampleplugin
    - exampleplugin.pro
    - mne-cpp
    - Headers
      - FormFiles
        - examplegui.h
        - exampleoutputwidget.h
        - settingswidget.h
      - exampleplugin.h
      - exampleplugin_global.h
    - Sources
      - FormFiles
        - examplegui.cpp
        - exampleoutputwidget.cpp
        - settingswidget.cpp
      - exampleplugin.cpp
    - Forms
      - FormFiles
        - examplegui.ui
        - exampleoutputwidget.ui
        - settingswidget.ui
      - Other files
  - fiffsimulator
  - mne
  - mne-cpp
  - neuromag

exampleplugin.cpp | ExamplePlugin::run(): void

```
72  void ExamplePlugin::run()
73  {
74
75      // Check every 10ms until the fiff info has been
76      while(!m_pFiffInfo)
77          msleep(10);
78
79      // run() will return as soon as m_bIsRunning is
80      // Otherwise it will continually process any data
81      while (m_bIsRunning)
82      {
83          // 1. Get data from the buffer
84          MatrixXd t_mat = m_pExampleBuffer->pop();
85
86          // 2. Do some kind of processing
87          t_mat = - t_mat;
88
89          // 3. Pass some data to the output display
90          int channel   = 10;
91          int timeIndex = 0;
92          double value  = t_mat(channel, timeIndex);
93          if (outputWidget) {
94              outputWidget->printMeasurement(value, outputColor);
95          }
96
97          // 4. Send the data out to any connected plugins
98          // This will also update the display
99          m_pExampleOutput->data()->setValue(t_mat);
100     }
101 }
102
103 bool ExamplePlugin::stop()
```

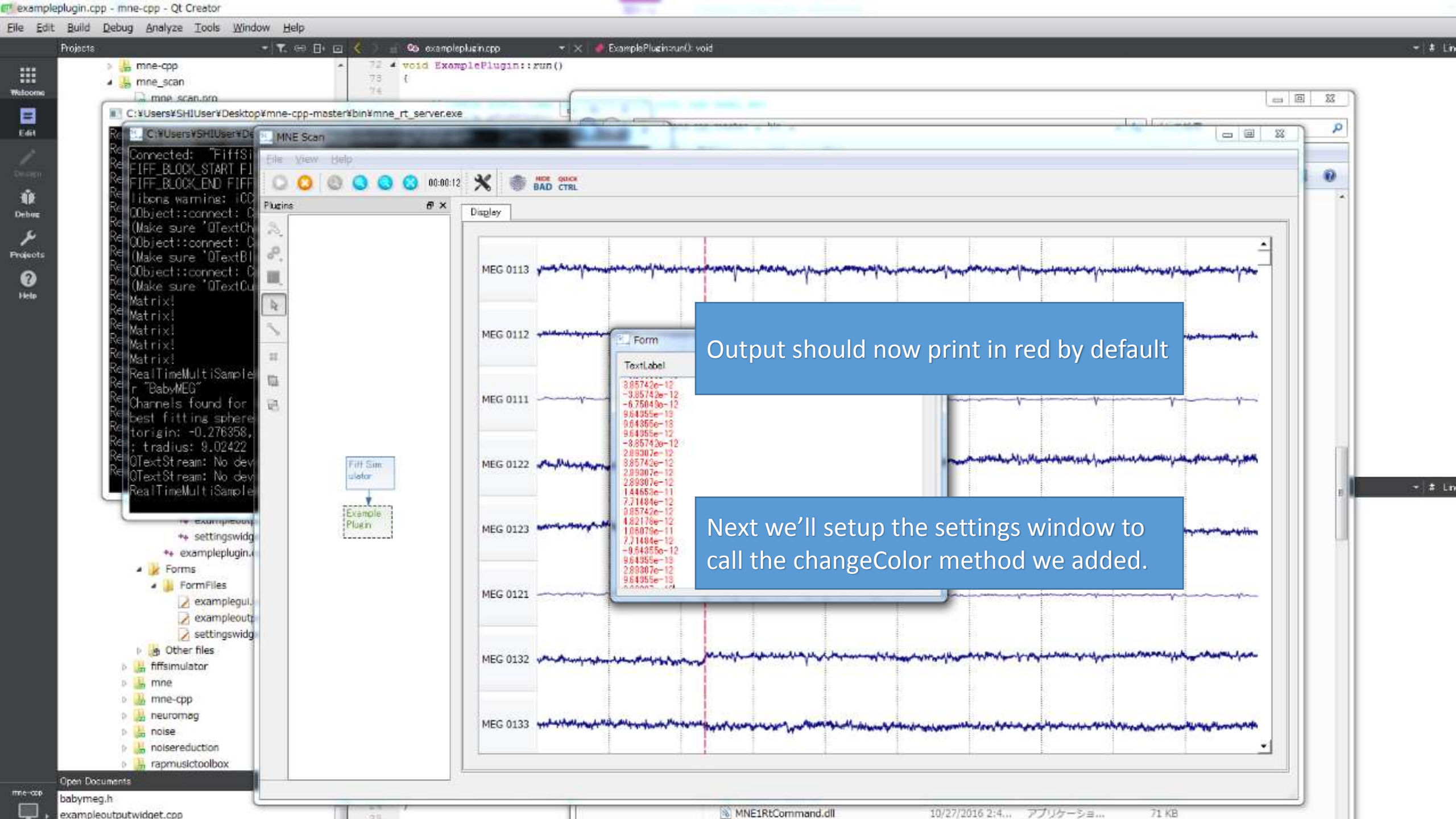
If you try to build now you will get an error because we haven't updated the spot where we call the function we just modified

Hop back to your plugin's source file and add a color argument to print

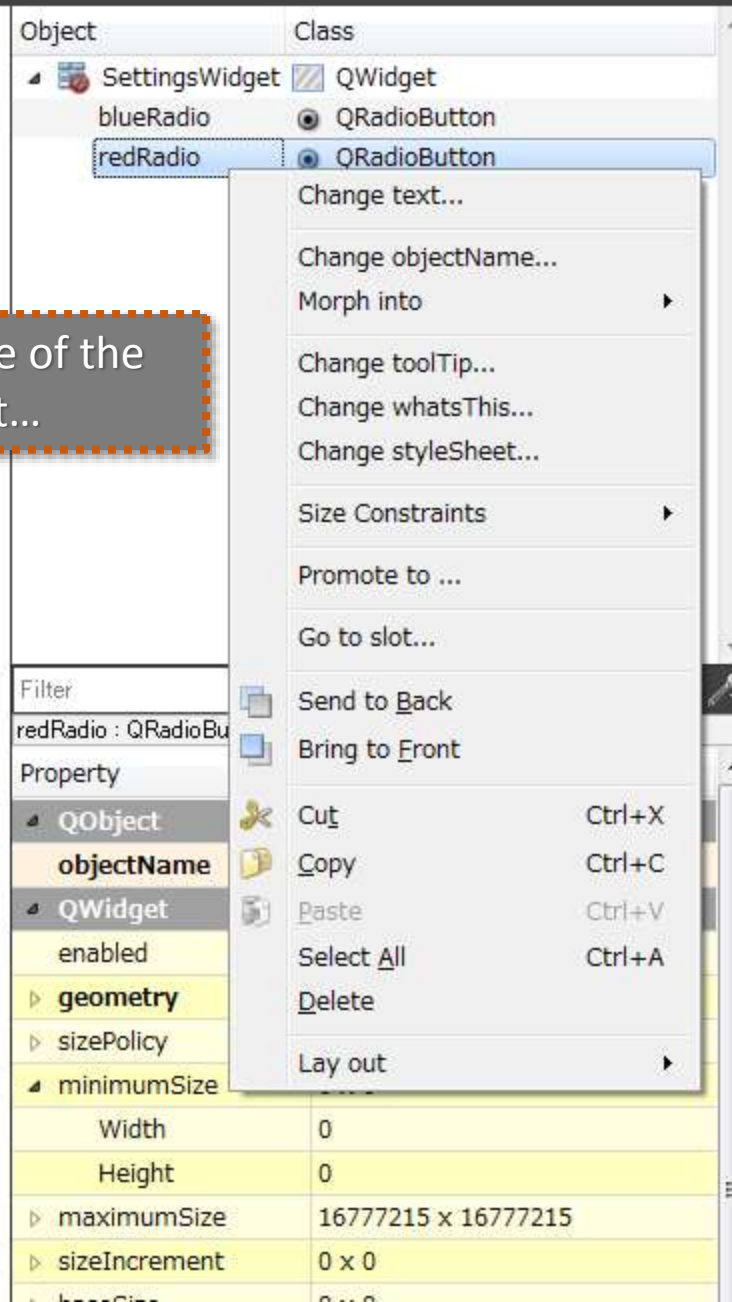
exampleplugin.cpp | ExamplePlugin::ExamplePlugin()

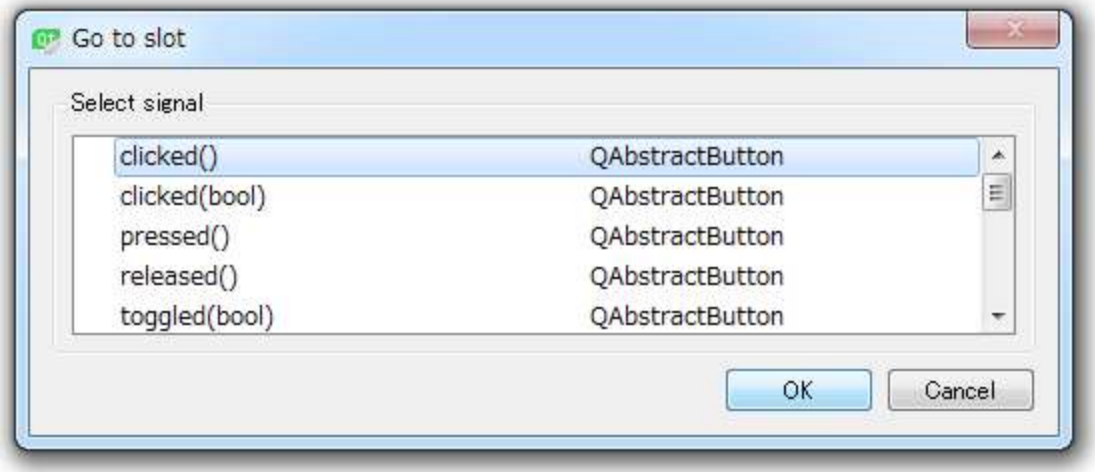
```
1   #include "exampleplugin.h"
2
3   using namespace SCSHAREDLIB;
4   using namespace SCMEASLIB;
5   using namespace IOBUFFER;
6
7   // Constructor
8   ExamplePlugin::ExamplePlugin()
9   : m_bIsRunning(false)
10  , outputColor(QColor(255,0,0)) // output color initializes to red
11  , m_pExampleInput(NULL)
12  , m_pExampleOutput(NULL)
13  , m_pExampleBuffer(CircularMatrixBuffer<double>::SPtr())
14  {
15
16      // Add a new action to the toolbar     QAction( icon , title, parent )
17      QAction* showSettingsWidgetAction = new QAction(QIcon(":/images/options.png"), tr("Toolbar Widget"),this);
18      showSettingsWidgetAction->setShortcut(tr("F12"));
```

We also need to add an initializer for the color variable added previously


exampleplugin.cpp - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne_scan
mne_scan.pro
exampleplugin.cpp
ExamplePlugin::run(): void
void ExamplePlugin::run()
{
C:\Users\SHIUser\Desktop\mne-cpp-master\bin\mne_rt_server.exe
Connected: "FiffSi
FIFF_BLOCK_START Fl
FIFF_BLOCK_END FIFF
libcng warning: iCC
QObject::connect: C
(Make sure 'QTextCh
QObject::connect: C
(Make sure 'QTextBl
QObject::connect: C
(Make sure 'QTextCu
Matrix!
Matrix!
Matrix!
Matrix!
Matrix!
RealTimeMultiSample
r "BabyMEG"
Channels found for
best fitting sphere
torigin: -0.276358,
; tradius: 9.02422
QTextStream: No dev
QTextStream: No dev
RealTimeMultiSample
MNE Scan
File View Help
00:00:12
HIDE BAD QUICK CTRL
Plugins
Display
Fiff Simulator
Example Plugin
MEG 0113
MEG 0112
MEG 0111
MEG 0122
MEG 0123
MEG 0121
MEG 0132
MEG 0133
Form
TextLabel
Output should now print in red by default
Next we'll setup the settings window to call the changeColor method we added.
settingswidg
exampleplugin.
Forms
FormFiles
examplegui.
exampleoutp
settingswidg
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
Open Documents
babymeg.h
exampleoutputwidget.cpp
MNE1RtCommand.dll
10/27/2016 2:4...
71 KB

In the Form Editor, right click on one of the radio buttons and choose Go to slot…





Select clicked()

Filter

Used | Text | Shortcut | Checkable | ToolTip

```
11  SettingsWidget::~SettingsWidget()
12  {
13      delete ui;
14  }
15
16  void SettingsWidget::on_redRadio_clicked()
17  {
18
19  }
20
```

Confirm that a new method was automatically added to your settings widget.

This is an alternative to connecting a signal and slot in the constructor method.

utwidget.h
t.h

lobal.h

op
utwidget.cpp
t.cpp
op

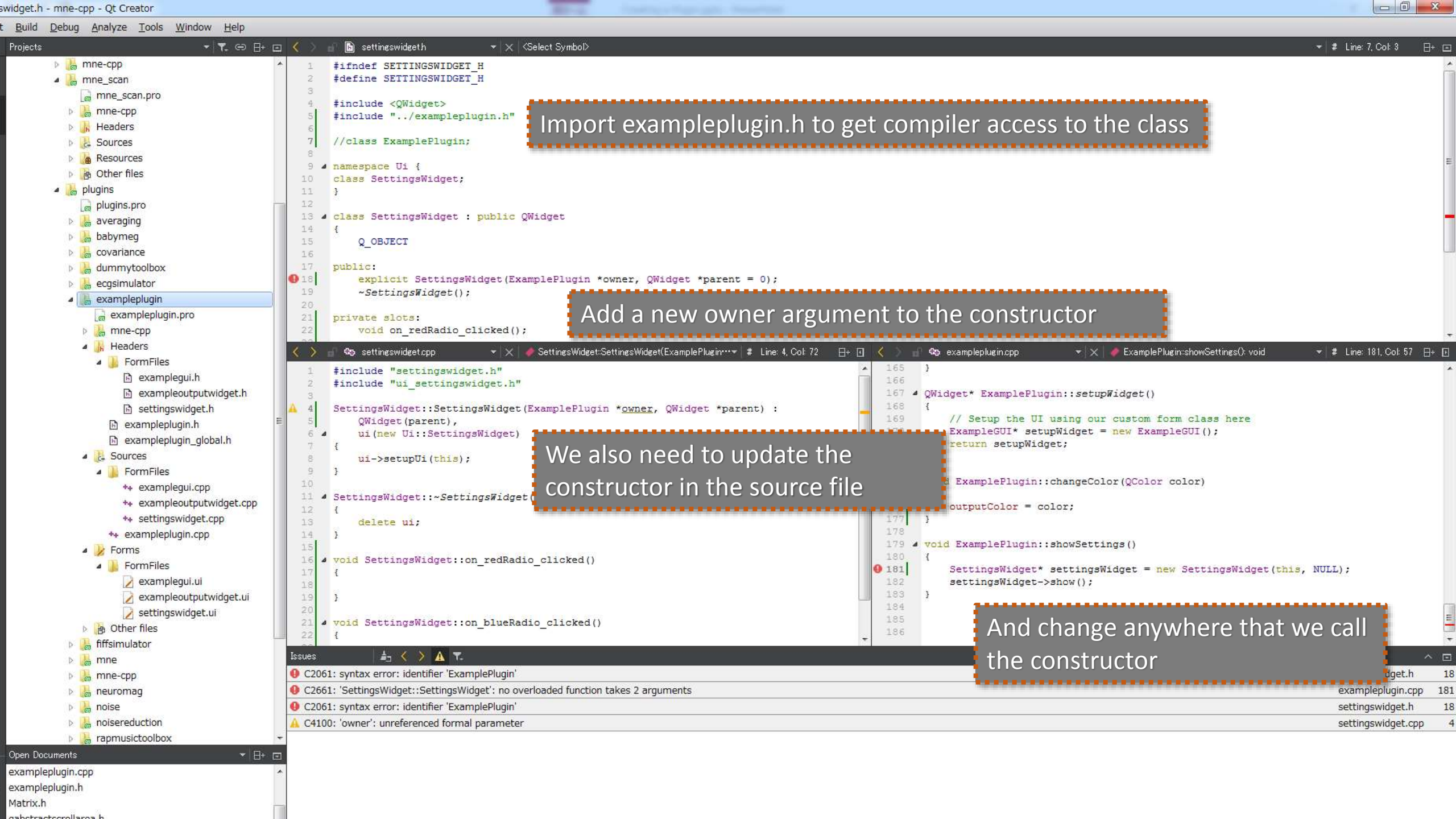
utwidget.ui
t.ui

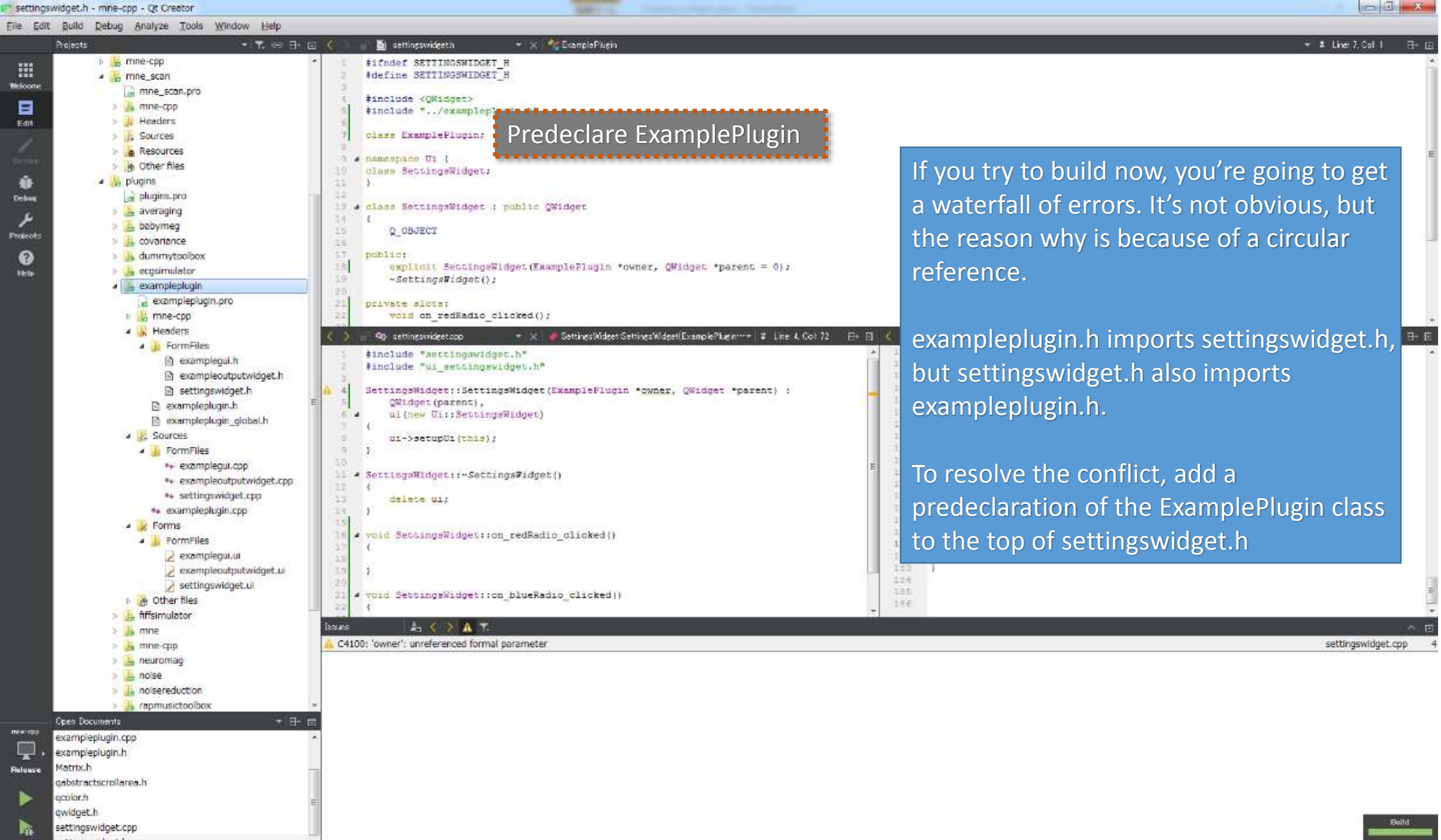
```
11  SettingsWidget::~SettingsWidget()
12  {
13      delete ui;
14  }
15
16  void SettingsWidget::on_redRadio_clicked()
17  {
18
19  }
20
21  void SettingsWidget::on_blueRadio_clicked()
22  {
23
24  }
25
```

Repeat for the other radio button as well

We want to call changeColor in these methods, but to do that we need a reference to our plugin.

The way we'll accomplish that is by adding an owner argument to the class constructor and saving it in a property.


swidget.h - mne-cpp - Qt Creator
Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne_scan
mne_scan.pro
mne-cpp
Headers
Sources
Resources
Other files
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
FormFiles
examplegui.h
exampleoutputwidget.h
settingswidget.h
exampleplugin.h
exampleplugin_global.h
Sources
FormFiles
examplegui.cpp
exampleoutputwidget.cpp
settingswidget.cpp
exampleplugin.cpp
Forms
FormFiles
examplegui.ui
exampleoutputwidget.ui
settingswidget.ui
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
Open Documents
exampleplugin.cpp
exampleplugin.h
Matrix.h
settingswidget.h
<Select Symbol>
Line: 7, Col: 3
#ifndef SETTINGSWIDGET_H
#define SETTINGSWIDGET_H
#include <QWidget>
#include "../exampleplugin.h"
//class ExamplePlugin;
namespace Ui {
class SettingsWidget;
}
class SettingsWidget : public QWidget
{
Q_OBJECT
public:
explicit SettingsWidget(ExamplePlugin *owner, QWidget *parent = 0);
~SettingsWidget();
private slots:
void on_redRadio_clicked();
Import exampleplugin.h to get compiler access to the class
Add a new owner argument to the constructor
settingswidget.cpp
SettingsWidget::SettingsWidget(ExamplePlugin*...
Line: 4, Col: 72
#include "settingswidget.h"
#include "ui_settingswidget.h"
SettingsWidget::SettingsWidget(ExamplePlugin *owner, QWidget *parent) :
QWidget(parent),
ui(new Ui::SettingsWidget)
{
ui->setupUi(this);
}
SettingsWidget::~SettingsWidget()
{
delete ui;
}
void SettingsWidget::on_redRadio_clicked()
{
}
void SettingsWidget::on_blueRadio_clicked()
{
We also need to update the constructor in the source file
exampleplugin.cpp
ExamplePlugin::showSettings(): void
Line: 181, Col: 57
}
QWidget* ExamplePlugin::setupWidget()
{
// Setup the UI using our custom form class here
ExampleGUI* setupWidget = new ExampleGUI();
return setupWidget;
ExamplePlugin::changeColor(QColor color)
outputColor = color;
}
void ExamplePlugin::showSettings()
{
SettingsWidget* settingsWidget = new SettingsWidget(this, NULL);
settingsWidget->show();
}
And change anywhere that we call the constructor
Issues
C2061: syntax error: identifier 'ExamplePlugin' dget.h 18
C2661: 'SettingsWidget::SettingsWidget': no overloaded function takes 2 arguments exampleplugin.cpp 181
C2061: syntax error: identifier 'ExamplePlugin' settingswidget.h 18
C4100: 'owner': unreferenced formal parameter settingswidget.cpp 4



If you try to build now, you're going to get a waterfall of errors. It's not obvious, but the reason why is because of a circular reference.

exampleplugin.h imports settingswidget.h, but settingswidget.h also imports exampleplugin.h.

To resolve the conflict, add a predeclaration of the ExamplePlugin class to the top of settingswidget.h

settingswidget.cpp - mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

Projects

- mne-cpp
- mne_scan
  - mne_scan.pro
  - mne-cpp
  - Headers
  - Sources
  - Resources
  - Other files
- plugins
  - plugins.pro
  - averaging
  - babymeg
  - covariance
  - dummytoolbox
  - ecgsimulator
  - exampleplugin
    - exampleplugin.pro
    - mne-cpp
    - Headers
      - FormFiles
        - examplegui.h
        - exampleoutputwidget.h
        - settingswidget.h
      - exampleplugin.h
      - exampleplugin_global.h
    - Sources
      - FormFiles
        - examplegui.cpp
        - exampleoutputwidget.cpp
        - settingswidget.cpp
      - exampleplugin.cpp
    - Forms
      - FormFiles
        - examplegui.ui
        - exampleoutputwidget.ui
        - settingswidget.ui
    - Other files
  - fiffsimulator
  - mne
  - mne-cpp
  - neuromag
  - noise
  - noisereduction
  - rapmusictoolbox

Open Documents

exampleplugin.cpp

settingswidget.h — owner: ExamplePlugin * — Line: 30, Col: 26

```
#ifndef SETTINGSWIDGET_H
#define SETTINGSWIDGET_H

#include <QWidget>
#include "../exampleplugin.h"

class ExamplePlugin;

namespace Ui {
class SettingsWidget;
}

class SettingsWidget : public QWidget
{
    Q_OBJECT

public:
    explicit SettingsWidget(ExamplePlugin *owner, QWidget *parent = 0);
    ~SettingsWidget();

private slots:
    void on_redRadio_clicked();

    void on_blueRadio_clicked();

private:
    Ui::SettingsWidget *ui;

    // A pointer to the plugin that created this widget
    ExamplePlugin *owner;
};

#endif // SETTINGSWIDGET_H
```

Add a private property that points to ExamplePlugin

settingswidget.cpp — SettingsWidget::SettingsWidget(ExamplePlugin…

```
#include "settingswidget.h"
#include "ui_settingswidget.h"

SettingsWidget::SettingsWidget(ExamplePlugin *owner, QWidget *par
    QWidget(parent),
    ui(new Ui::SettingsWidget)
{
    ui->setupUi(this);
    this->owner = owner;
}

Sett
{

}

void
{

}

void SettingsWidget::on_blueRadio_clicked()
{

}
```

Inside of the constructor, set the new variable to the owner passed in when the window is created.

Issues

```
11
12  SettingsWidget::~SettingsWidget()
13  {
14      delete ui;
15  }
16
17  void SettingsWidget::on_redRadio_clicked()
18  {
19      owner->changeColor(QColor(255,0,0));
20  }
21
22  void SettingsWidget::on_blueRadio_clicked()
23  {
24      owner->changeColor(QColor(0,0,255));
25  }
26
```

Finally, fill in the button functions to change the color that gets passed to the output window.

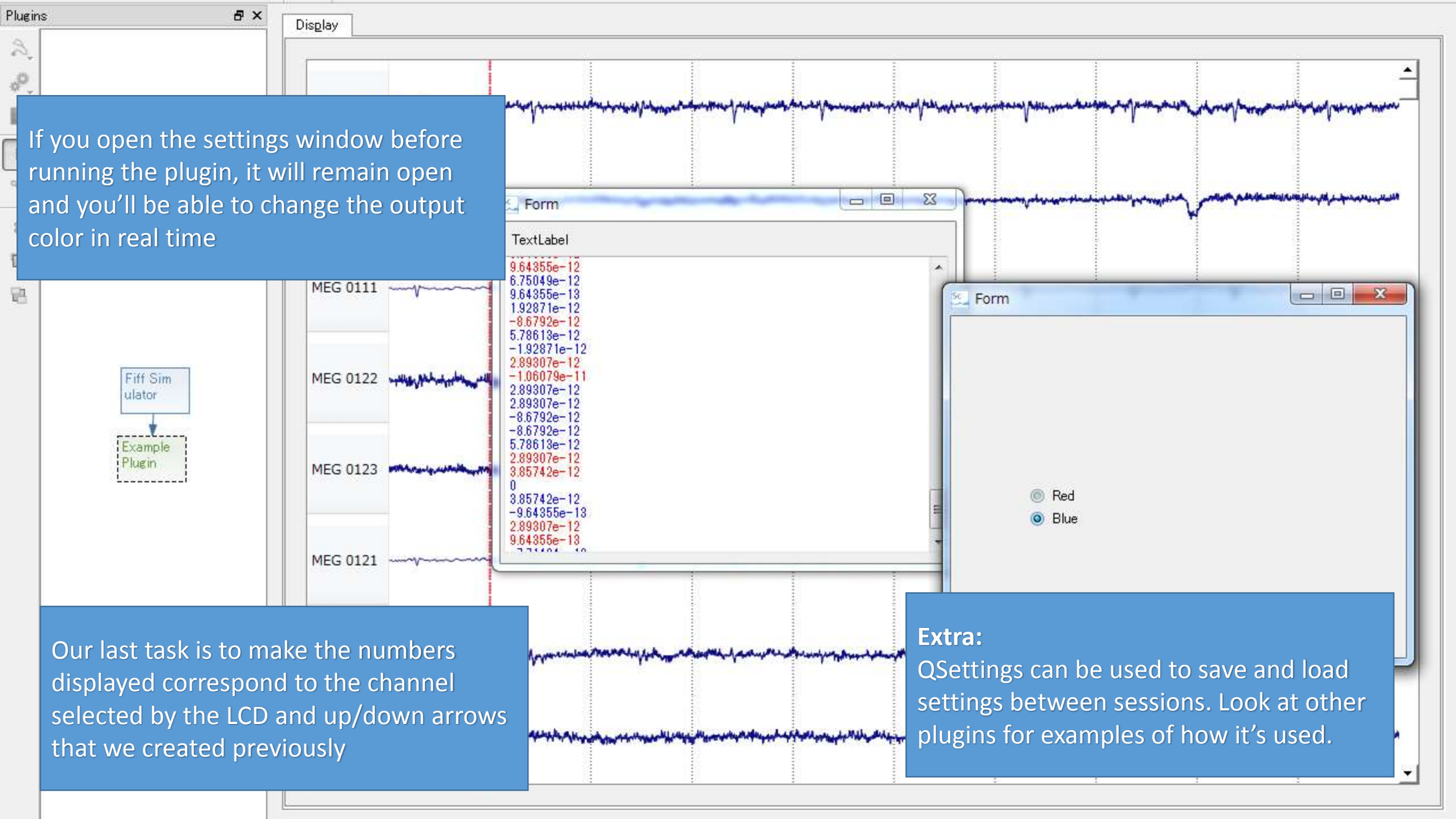
utwidget.h
t.h

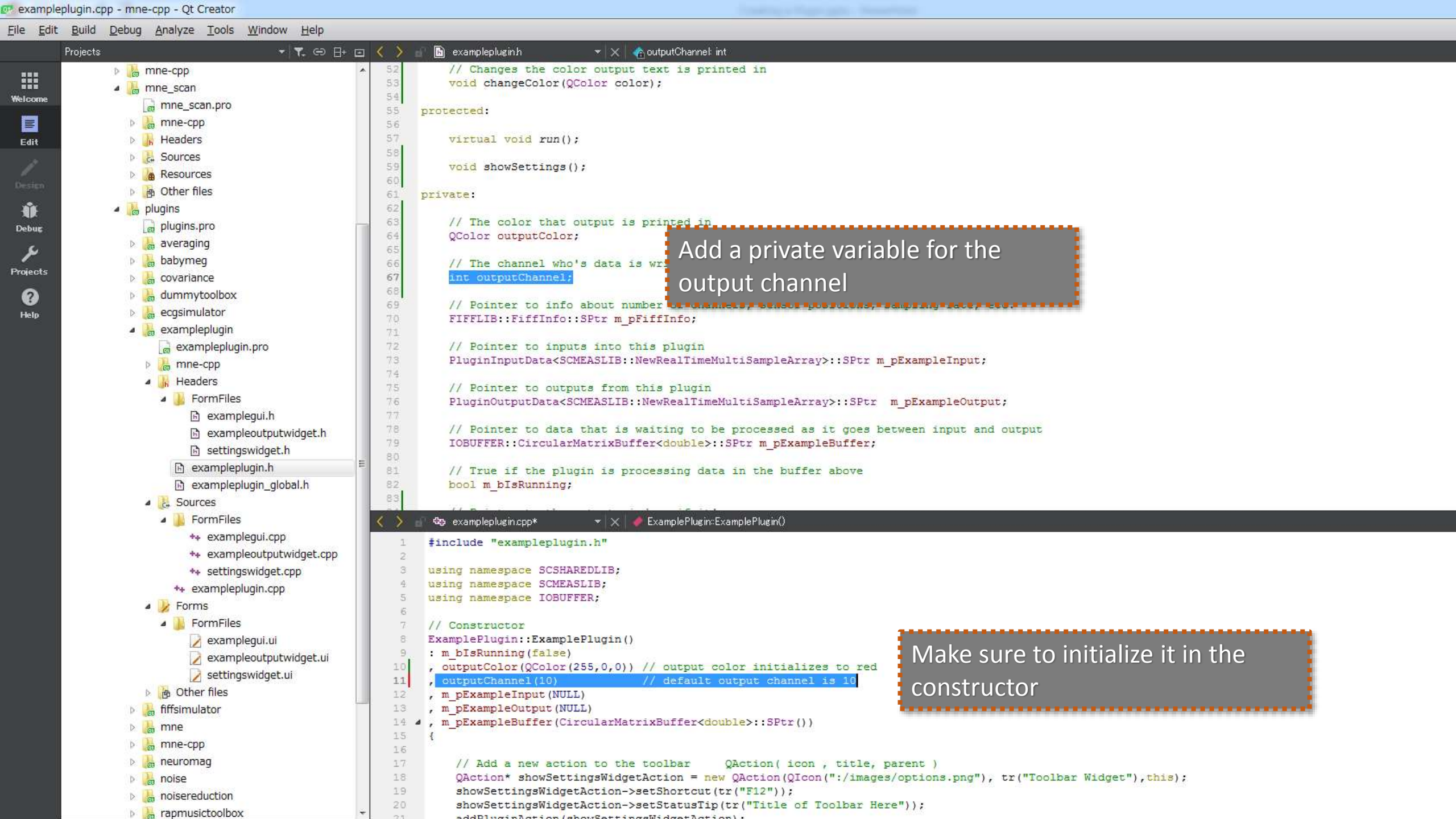
lobal.h

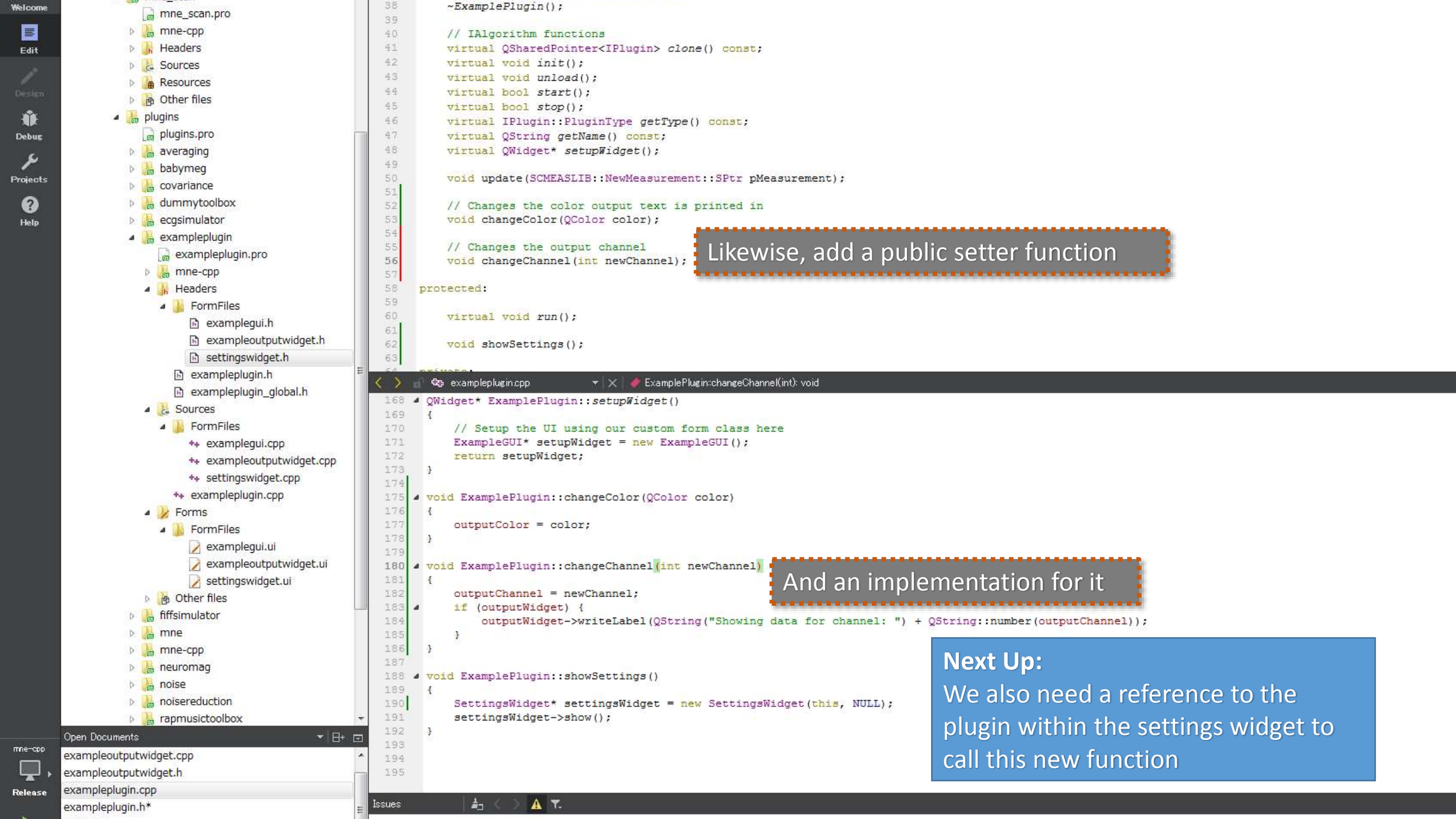
op
utwidget.cpp
t.cpp
op

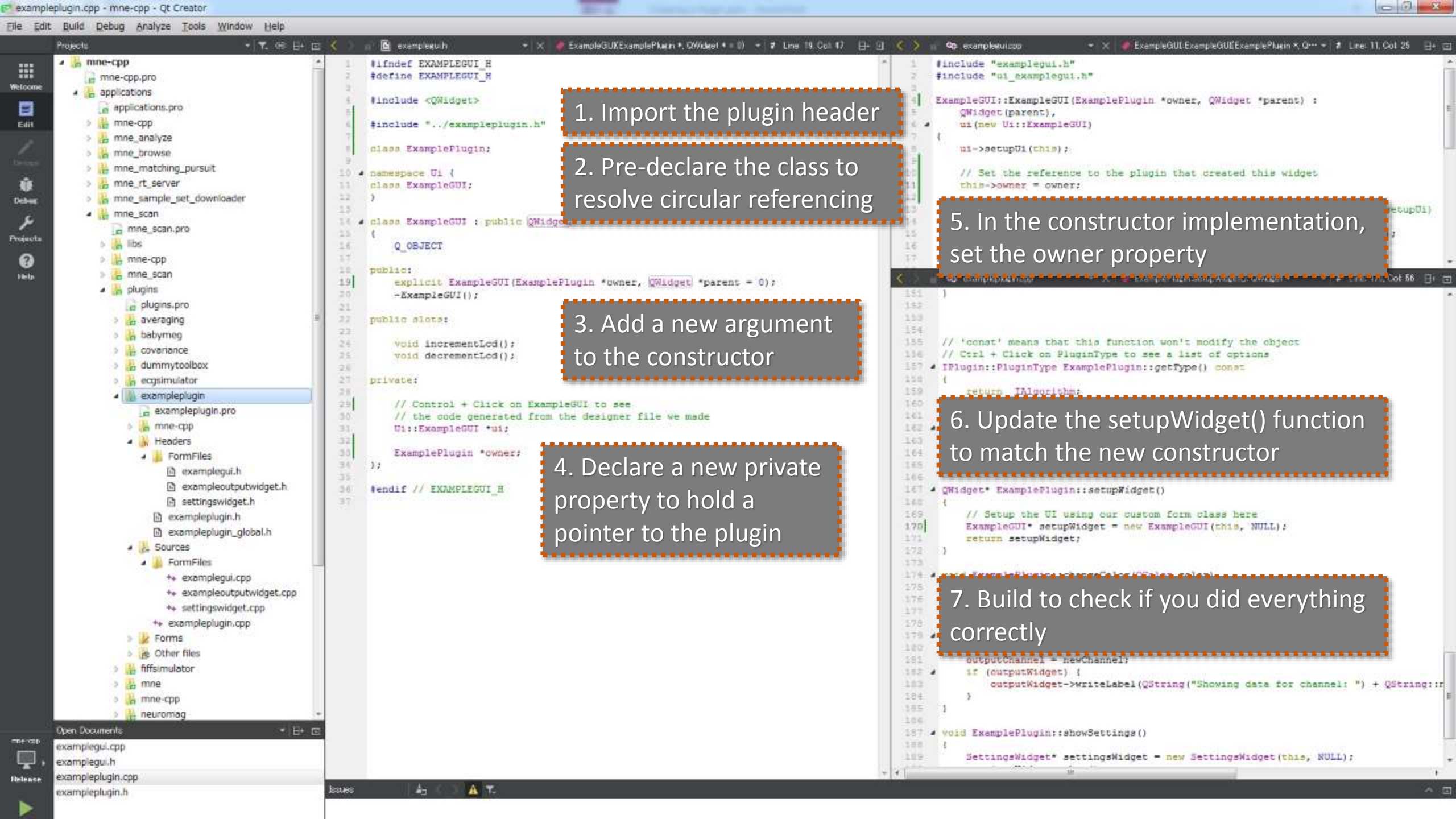
utwidget.ui
t.ui

Issues


Plugins
Display
If you open the settings window before running the plugin, it will remain open and you'll be able to change the output color in real time
Fiff Sim ulator
Example Plugin
MEG 0111
MEG 0122
MEG 0123
MEG 0121
Form
TextLabel
9.64355e-12
6.75049e-12
9.64355e-13
1.92871e-12
-8.6792e-12
5.78613e-12
-1.92871e-12
2.89307e-12
-1.06079e-11
2.89307e-12
2.89307e-12
-8.6792e-12
-8.6792e-12
5.78613e-12
2.89307e-12
3.85742e-12
0
3.85742e-12
-9.64355e-13
2.89307e-12
9.64355e-13
Form
Red
Blue
Our last task is to make the numbers displayed correspond to the channel selected by the LCD and up/down arrows that we created previously
Extra:
QSettings can be used to save and load settings between sessions. Look at other plugins for examples of how it's used.


exampleplugin.cpp - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
Welcome
Edit
Design
Debug
Projects
Help
mne-cpp
mne_scan
mne_scan.pro
mne-cpp
Headers
Sources
Resources
Other files
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
FormFiles
examplegui.h
exampleoutputwidget.h
settingswidget.h
exampleplugin.h
exampleplugin_global.h
Sources
FormFiles
examplegui.cpp
exampleoutputwidget.cpp
settingswidget.cpp
exampleplugin.cpp
Forms
FormFiles
examplegui.ui
exampleoutputwidget.ui
settingswidget.ui
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
exampleplugin.h
outputChannel: int
// Changes the color output text is printed in
void changeColor(QColor color);
protected:
virtual void run();
void showSettings();
private:
// The color that output is printed in
QColor outputColor;
// The channel who's data is wr
int outputChannel;
// Pointer to info about number
FIFFLIB::FiffInfo::SPtr m_pFiffInfo;
// Pointer to inputs into this plugin
PluginInputData<SCMEASLIB::NewRealTimeMultiSampleArray>::SPtr m_pExampleInput;
// Pointer to outputs from this plugin
PluginOutputData<SCMEASLIB::NewRealTimeMultiSampleArray>::SPtr m_pExampleOutput;
// Pointer to data that is waiting to be processed as it goes between input and output
IOBUFFER::CircularMatrixBuffer<double>::SPtr m_pExampleBuffer;
// True if the plugin is processing data in the buffer above
bool m_bIsRunning;
Add a private variable for the output channel
exampleplugin.cpp*
ExamplePlugin::ExamplePlugin()
#include "exampleplugin.h"
using namespace SCSHAREDLIB;
using namespace SCMEASLIB;
using namespace IOBUFFER;
// Constructor
ExamplePlugin::ExamplePlugin()
: m_bIsRunning(false)
, outputColor(QColor(255,0,0)) // output color initializes to red
, outputChannel(10) // default output channel is 10
, m_pExampleInput(NULL)
, m_pExampleOutput(NULL)
, m_pExampleBuffer(CircularMatrixBuffer<double>::SPtr())
{
// Add a new action to the toolbar QAction(icon , title, parent)
QAction* showSettingsWidgetAction = new QAction(QIcon(":/images/options.png"), tr("Toolbar Widget"),this);
showSettingsWidgetAction->setShortcut(tr("F12"));
showSettingsWidgetAction->setStatusTip(tr("Title of Toolbar Here"));
Make sure to initialize it in the constructor


Welcome
Edit
Design
Debug
Projects
Help
mne_scan.pro
mne-cpp
Headers
Sources
Resources
Other files
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
FormFiles
examplegui.h
exampleoutputwidget.h
settingswidget.h
exampleplugin.h
exampleplugin_global.h
Sources
FormFiles
examplegui.cpp
exampleoutputwidget.cpp
settingswidget.cpp
exampleplugin.cpp
Forms
FormFiles
examplegui.ui
exampleoutputwidget.ui
settingswidget.ui
Other files
fiffsimulator
mne
mne-cpp
neuromag
noise
noisereduction
rapmusictoolbox
Open Documents
exampleoutputwidget.cpp
exampleoutputwidget.h
exampleplugin.cpp
exampleplugin.h*
mne-cpp
Release
38 ~ExamplePlugin();
40 // IAlgorithm functions
41 virtual QSharedPointer<IPlugin> clone() const;
42 virtual void init();
43 virtual void unload();
44 virtual bool start();
45 virtual bool stop();
46 virtual IPlugin::PluginType getType() const;
47 virtual QString getName() const;
48 virtual QWidget* setupWidget();
50 void update(SCMEASLIB::NewMeasurement::SPtr pMeasurement);
52 // Changes the color output text is printed in
53 void changeColor(QColor color);
55 // Changes the output channel
56 void changeChannel(int newChannel);
58 protected:
60 virtual void run();
62 void showSettings();
Likewise, add a public setter function
exampleplugin.cpp
ExamplePlugin::changeChannel(int): void
168 QWidget* ExamplePlugin::setupWidget()
169 {
170 // Setup the UI using our custom form class here
171 ExampleGUI* setupWidget = new ExampleGUI();
172 return setupWidget;
173 }
175 void ExamplePlugin::changeColor(QColor color)
176 {
177 outputColor = color;
178 }
180 void ExamplePlugin::changeChannel(int newChannel)
181 {
182 outputChannel = newChannel;
183 if (outputWidget) {
184 outputWidget->writeLabel(QString("Showing data for channel: ") + QString::number(outputChannel));
185 }
186 }
188 void ExamplePlugin::showSettings()
189 {
190 SettingsWidget* settingsWidget = new SettingsWidget(this, NULL);
191 settingsWidget->show();
192 }
And an implementation for it
Next Up:
We also need a reference to the plugin within the settings widget to call this new function
Issues


exampleplugin.cpp - mne-cpp - Qt Creator
File Edit Build Debug Analyze Tools Window Help
Projects
mne-cpp
mne-cpp.pro
applications
applications.pro
mne-cpp
mne_analyze
mne_browse
mne_matching_pursuit
mne_rt_server
mne_sample_set_downloader
mne_scan
mne_scan.pro
libs
mne-cpp
mne_scan
plugins
plugins.pro
averaging
babymeg
covariance
dummytoolbox
ecgsimulator
exampleplugin
exampleplugin.pro
mne-cpp
Headers
FormFiles
examplegui.h
exampleoutputwidget.h
settingswidget.h
exampleplugin.h
exampleplugin_global.h
Sources
FormFiles
examplegui.cpp
exampleoutputwidget.cpp
settingswidget.cpp
exampleplugin.cpp
Forms
Other files
fiffsimulator
mne
mne-cpp
neuromag
Open Documents
examplegui.cpp
examplegui.h
exampleplugin.cpp
exampleplugin.h
Welcome
Edit
Debug
Projects
Help
Release
examplegui.h
#ifndef EXAMPLEGUI_H
#define EXAMPLEGUI_H
#include <QWidget>
#include "../exampleplugin.h"
class ExamplePlugin;
namespace Ui {
class ExampleGUI;
}
class ExampleGUI : public QWidget
{
Q_OBJECT
public:
explicit ExampleGUI(ExamplePlugin *owner, QWidget *parent = 0);
~ExampleGUI();
public slots:
void incrementLcd();
void decrementLcd();
private:
// Control + Click on ExampleGUI to see
// the code generated from the designer file we made
Ui::ExampleGUI *ui;
ExamplePlugin *owner;
};
#endif // EXAMPLEGUI_H
1. Import the plugin header
2. Pre-declare the class to resolve circular referencing
3. Add a new argument to the constructor
4. Declare a new private property to hold a pointer to the plugin
examplegui.cpp
#include "examplegui.h"
#include "ui_examplegui.h"
ExampleGUI::ExampleGUI(ExamplePlugin *owner, QWidget *parent) :
QWidget(parent),
ui(new Ui::ExampleGUI)
{
ui->setupUi(this);
// Set the reference to the plugin that created this widget
this->owner = owner;
}
5. In the constructor implementation, set the owner property
// 'const' means that this function won't modify the object
// Ctrl + Click on PluginType to see a list of options
IPlugin::PluginType ExamplePlugin::getType() const
{
return IAlgorithm;
}
6. Update the setupWidget() function to match the new constructor
QWidget* ExamplePlugin::setupWidget()
{
// Setup the UI using our custom form class here
ExampleGUI* setupWidget = new ExampleGUI(this, NULL);
return setupWidget;
}
7. Build to check if you did everything correctly
outputChannel = newChannel;
if (outputWidget) {
outputWidget->writeLabel(QString("Showing data for channel: ") + QString::
}
}
void ExamplePlugin::showSettings()
{
SettingsWidget* settingsWidget = new SettingsWidget(this, NULL);
Issues

```
11      this->owner = owner;
12
13      // After setup has completed, connect the buttons (MUST BE DONE AFTER setupUi)
14      connect(ui->upButton, SIGNAL(released()), this, SLOT(incrementLcd()));
15      connect(ui->downButton, SIGNAL(released()), this, SLOT(decrementLcd()));
16
17  }
18
19  ExampleGUI::~ExampleGUI()
20  {
21      delete ui;
22  }
23
24  void ExampleGUI::incrementLcd(){
25      // Get the current value from the lcd
26      int currentValue = ui->lcdNumber->intValue();
27
28      // Set the new value as one greater
29      ui->lcdNumber->display(currentValue + 1);
30
31      // Change the selected channel of the plugin
32      owner->changeChannel(currentValue + 1);
33  }
34
35  void ExampleGUI::decrementLcd(){
36      // Get the current value from the lcd
37      int currentValue = ui->lcdNumber->intValue();
38
39      // Set the new value as one less
40      ui->lcdNumber->display(currentValue - 1);
41
42      // Change the selected channel of the plugin
43      owner->changeChannel(currentValue -1);
44  }
45
```

In the increment and decrement methods, change the channel of the plugin.

```
47
48
49  void ExamplePlugin::unload(){}
50
51  QSharedPointer<IPlugin> ExamplePlugin::clone() const
52  {
53      QSharedPointer<ExamplePlugin> pointerToExamplePlugin(new ExamplePlugin);
54      return pointerToExamplePlugin;
55  }
56
57  bool ExamplePlugin::start()
58  {
59      // Set isRunning to true and then call run
60      // The order will be important!
61      m_bIsRunning = true;
62      QThread::start();
63
64      // Show the output window
65      // Making the parent NULL creates the widget in a new window
66      outputWidget = QSharedPointer<ExampleOutputWidget>(new ExampleOutputWidget(NULL));
67      outputWidget->show();
68
69      // Call change channel, but pass in the current channel
70      // This is to make sure the channel lablel is set
71      changeChannel(outputChannel);
72
73      return true;
74  }
75
76  void ExamplePlugin::run()
77  {
78
79      // Check every 10ms until the fiff info has been set
80      while(!m_pFiffInfo)
```

Also, add a call to changeChannel at the same time we create the widget. This will ensure the channel label is set when the window first appears

Issues

exampleoutputwidget.ui - mne-cpp - Qt Creator

File Edit Build Debug Analyze Tools Window Help

exampleoutputwidget.ui

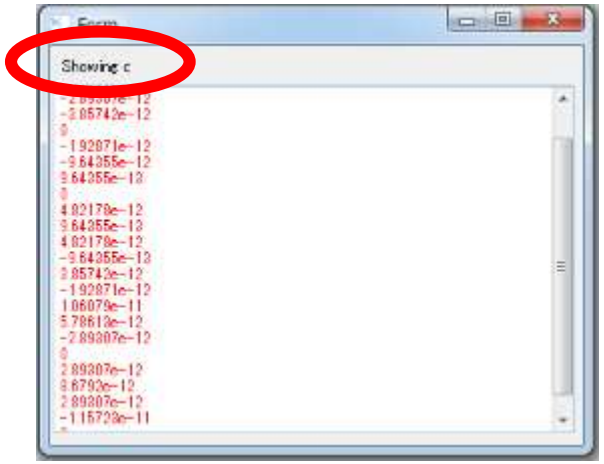
Welcome
Edit
Design
Debug
Projects
Help

Filter

- Layouts
  - Vertical Layout
  - Horizo···Layout
  - Grid Layout
  - Form Layout
- Spacers
  - Horizo···Spacer
  - Vertical Spacer
- Buttons
  - Push Button
  - Tool Button
  - Radio Button
  - Check Box
  - Comma···utton
  - Dialog···on Box
- Item Vi···-Based)
  - List View
  - Tree View
  - Table View
  - Column View
- Item Wi···-Based)
  - List Widget
  - Tree Widget
  - Table Widget
- Containers
  - Group Box
  - Scroll Area

TextLabel

If your text label is truncated, you may need to increase its bounds in the form file



Congratulations!

You’ve completed your first MNE Scan plugin