



OPP - Create a new Plug-in

Denis SAUNIER, Geoffrey BERGE,
Thibaud LAMARCHE, Thomas BERTHOME

June 04, 2014

Referring professor: Agnès ARNOULD

catalogue ouvert du cinéma



Summary

Create a new Plug-in.....	3
Create a new Interface in opp (optional).....	3
Generate opp library.....	3
Create a new project.....	4
Create the class to plug to the interface.....	4
Build.....	5
Annexes.....	5

Create a new Plug-in

Create a new Interface in opp (optional)

In order to have a working plug-in you need to implement an interface into opp. The interface can be declared in the file plugin.h.

The class must follow this pattern:

```
class NAME_OF_THE_INTERFACE{
public:
    virtual ~NAME_OF_THE_INTERFACE() { }
    virtual RETURN_TYPE METHOD1(PARAMS)=0;
    virtual RETURN_TYPE METHOD2(PARAMS)=0;
    ...
};
```

```
Q_DECLARE_INTERFACE(NAME, "opp.NAME");
```

All methods must be virtual and finish by “=0”.

The constructor must not be declared.

In order to charge an other plugin interface that has not been declared, you need add it in the method : `void MainWindow::loadPlugins();`

Generate opp library

In order to have access to the code of opp, you need to generate a static lib of opp. To do that, you need to go in to the file “opp.pro” and

- comment the lines (add #):

```
TEMPLATE = app
TARGET = opp
```

- uncomment the lines (remove #):

```
#TEMPLATE = lib
#CONFIG += staticlib
#TARGET = GeneratedLib/opp
```

This will generate a static library (libopp.a) that you need to copy in to the folder “pathToPluginProject/lib”.

Create a new project

Use QtCreator to create a new project. Then modify the file NAME.pro:

- add line `TEMPLATE = lib` #in order to generate a library
- add line `CONFIG += debug plugin` #to generate a debug plugin (change to release if needed)
- add line `DESTDIR = ./pluginsOPP` # in order to directly copy the folder
- add line `LIBS += -L$PWD/../lib/ -lopp` # in order to link the opp library
- add line `INCLUDEPATH += $PWD/../include` # in order to link the headers

Create the class to plug to the interface

Declare the main class of your plug-in as followed. The declaration must respect some conditions :

- extend the previously declared interface
- implement the virtual methods
- Create the wanted methods

in Plugin_NAME.h :

```
class Plugin_NAME : public NAME_OF_THE_INTERFACE
{
    Q_OBJECT
    Q_INTERFACES(NAME_OF_THE_INTERFACE)

    public:
        explicit Plugin_NAME();
        ~Plugin_NAME();
        RETURN_TYPE METHOD1(PARAMS)=0;
        RETURN_TYPE METHOD2(PARAMS)=0;
}
```

in Plugin_NAME.cpp :

```
Q_EXPORT_PLUGIN2(Plugin_NAME, Plugin_NAME)
```

Build

Build the project using QtCreator. At this time QtCreator ask you to choose an executable to launch. You can either cancel and then take the folder pluginsOPP and copy it next to your opp executable, or realize a small script that copy the folder and launch the executable.

Script sample :

```
#!/bin/bash
cp -R -f "/PATH_TO_PLUGIN_DEBUG/pluginsOPP" "/PATH_TO_OPP_DEBUG/"
/PATH_TO_OPP_DEBUG/opp
```

Annexes

A French tutorial:

<http://fr.openclassrooms.com/informatique/cours/qt-creation-de-plugins>

The repository for the plug-in:

https://github.com/cinemaouvert/opp_plugins