22.06.2011

What is the moc

Ingmar Gergel

What is the moc?

QT Meta-Object System

Meta-Object System provides:

- the signals and slots mechanism for inter-object communication
- run-time type information
- dynamic property system

Meta-Object System is based on:

1. `QObject` class
2. `Q_Object` macro
3. Meta-Object Compiler (moc) 

Meta-Object Compiler (moc)

Meta-Object Compiler:

"provides a clean way to go beyond the compiled language's facilities"

- Reads a C++ source file
  - if the file contains `Q_Object` macro
    - produces *.c++ file containing the meta-object code

Usage:

- `moc –o<file> -f[<file>] –i -D<macro> -h –v –Fdir`
- ~~usually automatically called by the build system (qmake)~~

MITK build system and the moc

Source directory

files.cmake

```
1   SET(SRC_CPP_FILES
2     berrySingleNodeSelection.cpp
3     QmitkDataManagerView.cpp
4     QmitkDataManagerPreferencePage.cpp
5     QmitkDataManagerHotkeysPrefPage.cpp
6   )
7
8   SET(INTERNAL_CPP_FILES
9     mitkPluginActivator.cpp
10    QmitkPropertyListView.cpp
11    QmitkNodeTableViewKeyFilter.cpp
12    QmitkInfoDialog.cpp
13  )
14
15  SET(MOC_H_FILES
16    src/QmitkDataManagerView.h
17    src/QmitkDataManagerPreferencePage.h
18    src/QmitkDataManagerHotkeysPrefPage.h
19    src/internal/QmitkNodeTableViewKeyFilter.h
20    src/internal/QmitkPropertyListView.h
21    src/internal/QmitkInfoDialog.h
22    src/internal/mitkPluginActivator.h
23  )
24
25  SET(CPP_FILES )
26
27  SET(CACHED_RESOURCE_FILES
```

Binary directory

Name	Elementtyp
internal	Dateiordner
moc_QmitkDataManagerHotkeysPrefPage.cxx	C++ Source
moc_QmitkDataManagerPreferencePage.cxx	C++ Source
moc_QmitkDataManagerView.cxx	C++ Source
moc_QmitkDataManagerHotkeysPrefPage.cxx...	CXX_PARAMETERS...
moc_QmitkDataManagerPreferencePage.cxx_...	CXX_PARAMETERS...
moc_QmitkDataManagerView.cxx_parameters	CXX_PARAMETERS...

Name	Elementtyp
moc_mitkPluginActivator.cxx	C++ Source
moc_QmitkInfoDialog.cxx	C++ Source
moc_QmitkNodeTableViewKeyFilter.cxx	C++ Source
moc_QmitkPropertyListView.cxx	C++ Source
moc_mitkPluginActivator.cxx_parameters	CXX_PARAMETERS...
moc_QmitkInfoDialog.cxx_parameters	CXX_PARAMETERS...
moc_QmitkNodeTableViewKeyFilter.cxx_pa...	CXX_PARAMETERS...
moc_QmitkPropertyListView.cxx_parameters	CXX_PARAMETERS...

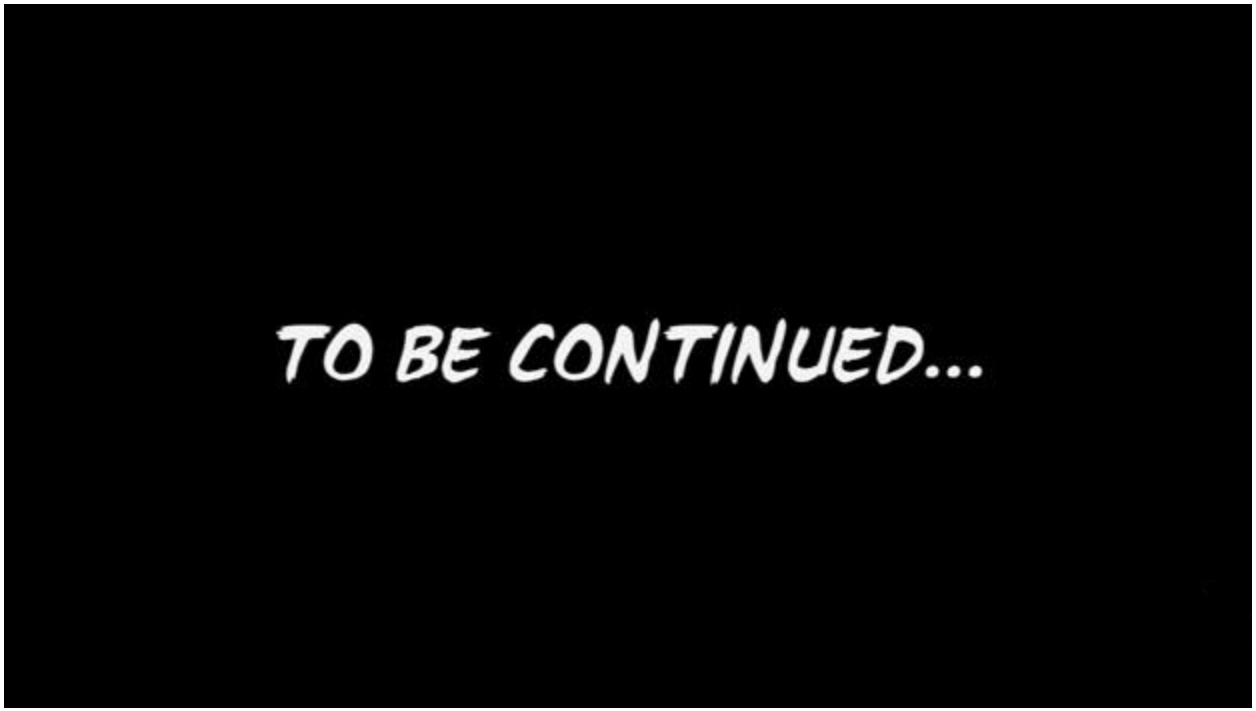
Example moc_QmitkDataManagerView.cxx

files.cmake | moc_QmitkDataManagerView.cxx

```
130
131 int QmitkDataManagerView::qt_metacall(QMetaObject::Call _c, int _id, void **_a)
132 {
133     typedef berry::QtViewPart QMocSuperClass;
134     _id = QMocSuperClass::qt_metacall(_c, _id, _a);
135     if (_id < 0)
136         return _id;
137     if (_c == QMetaObject::InvokeMetaMethod) {
138         switch (_id) {
139         case 0: OpacityChanged((*reinterpret_cast< int(*)>(_a[1]))); break;
140         case 1: OpacityActionChanged(); break;
141         case 2: ColorChanged(); break;
142         case 3: ColorActionChanged(); break;
143         case 4: TextureInterpolationChanged(); break;
144         case 5: TextureInterpolationToggled((*reinterpret_cast< bool(*)>(_a[1]))); break;
145         case 6: SurfaceRepresentationMenuAboutToShow(); break;
146         case 7: SurfaceRepresentationActionToggled((*reinterpret_cast< bool(*)>(_a[1]))); break;
147         case 8: NodeTableViewContextMenuRequested((*reinterpret_cast< const QPoint(*)>(_a[1]))); b
148         case 9: SaveSelectedNodes((*reinterpret_cast< bool(*)>(_a[1]))); break;
149         case 10: SaveSelectedNodes(); break;
150         case 11: RemoveSelectedNodes((*reinterpret_cast< bool(*)>(_a[1]))); break;
151         case 12: RemoveSelectedNodes(); break;
152         case 13: ReinitSelectedNodes((*reinterpret_cast< bool(*)>(_a[1]))); break;
153         case 14: ReinitSelectedNodes(); break;
154         case 15: MakeAllNodesInvisible((*reinterpret_cast< bool(*)>(_a[1]))); break;
155         case 16: MakeAllNodesInvisible(); break;
156         case 17: ShowOnlySelectedNodes((*reinterpret_cast< bool(*)>(_a[1]))); break;
157         case 18: ShowOnlySelectedNodes(); break;
158         case 19: ToggleVisibilityOfSelectedNodes((*reinterpret_cast< bool(*)>(_a[1]))); break;
159         case 20: ToggleVisibilityOfSelectedNodes(); break;
160         case 21: ShowInfoDialogForSelectedNodes((*reinterpret_cast< bool(*)>(_a[1]))); break;
```

Used by the
signals and slots mechanism

QT: Signals and slots mechanism



…upcoming bug squashing seminar

Take Home Message

Code generators (like the moc) are good

- no restriction to proprietary languages
- no one way wizards that generate obscure code
- no hacked C++ compliers needed
- no need to add generated files to your source

- “cleaner, safer and more in the spirit of UNIX”

Qt Reference Documentation

http://doc.qt.nokia.com/latest/moc.html