



Qt Creator Plugin Development

Aurindam Jana
Tobias Hunger

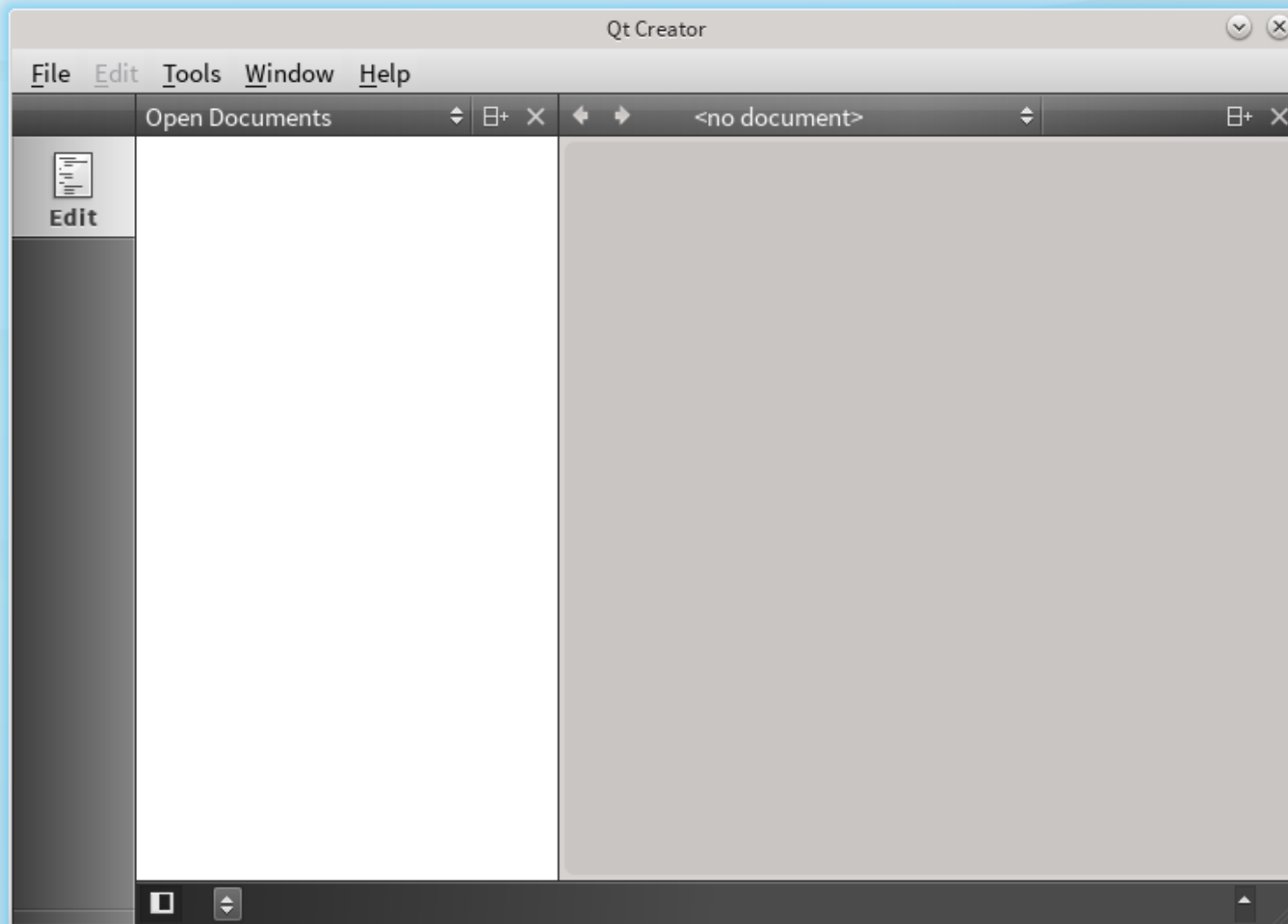
- Qt Creator is based on plugins

```
bin : zsh - Konsole
tobias@lea> ls -alhF qtcreator
-rwxr-xr-x 1 tobias tobias 78K 2. Sep 05:01 qtcreator*
tobias@lea> /home/code/installed/creator-current/bin
```

- **No** generic Rich Client Platform

```
creator-20130902-042857 : zsh - Konsole
tobias@lea> bin/qtcreator -settingspath /tmp
Failed to load core: Could not find 'Core.pluginspec' in /home/code/i
nstalled/creator-20130902-042857/lib/qtcreator/plugins,/home/tobias/.
local/share/data///QtProject/qtcreator/plugins/2.8.81
tobias@lea> /home/code/installed/creator-current
```

- Qt Creator *is* platform for IDEs



ExtensionSystem

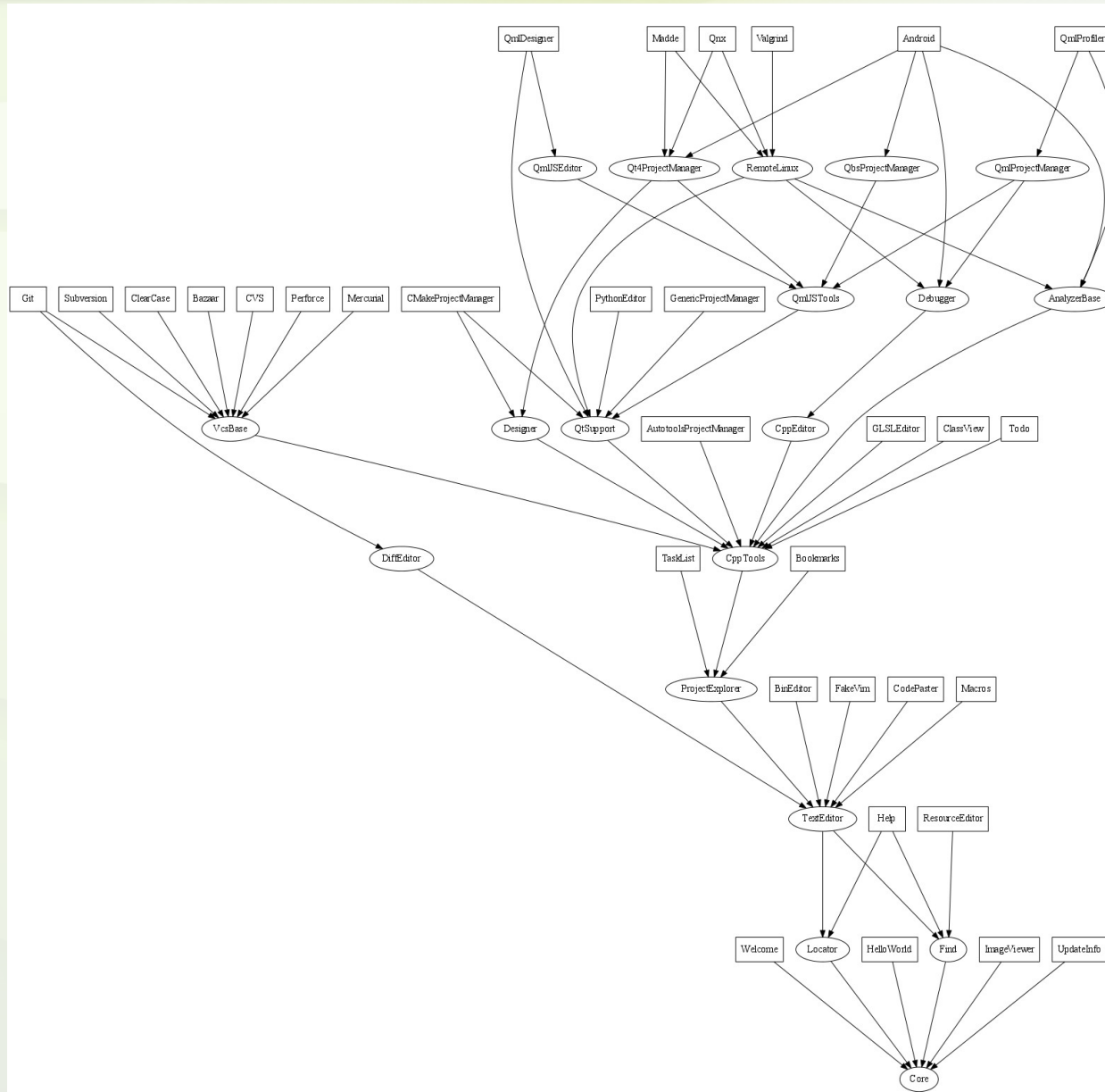
Library using namespace `ExtensionSystem` in
`src/libs/extensionsystem`

- Plugins
 - with dependencies
 - command line arguments
- (Q)ObjectPool
 - Lifecycle management for Objects
 - Retrieve Object(s) based on type or name
- UI elements to manage plugins

PluginSpec

```
<plugin name="Something" version="3.0.1" compatVersion="3.0.0">
  <vendor>Somebody</vendor>
  <copyright>(C) 2013 Somebody</copyright>
  <license>License Text</license>
  <category>Version Control</category>
  <description>Integration of something.</description>
  <url>http://www.qt-project.org</url>
  <dependencyList>
    <dependency name="Core" version="3.0.0"/>
  </dependencyList>
  <argumentList>
    <argument name="-special">Do something special</argument>
  </argumentList>
</plugin>
```

Dependencies



Extension System

1. Read all `.pluginspec`-files
2. Order Plugins
3. Load Plugins (Core -> Welcome)
4. `Initialize(...)` (Core -> Welcome)
5. `extensionsInitialized()` (Welcome -> Core)
6. `DelayedInitialize()` (Welcome->Core)

IPlugin

```
class EXTENSIONSYSTEM_EXPORT IPlugin : public QObject
{
    Q_OBJECT
public:
    [...]
    virtual bool initialize(const QStringList &arguments,
                           QString *errorString) = 0;
    virtual void extensionsInitialized() = 0;
    virtual bool delayedInitialize();
    [...]
};
```

Profiling

Profiling of plugin loading and start-up:

```
qtcreeator -profile
```

Object Pool

```
class EXTENSIONSYSTEM_EXPORT PluginManager : public QObject
{
    [...]
public
    // Object pool operations
    static void addObject(QObject *obj);
    static void removeObject(QObject *obj);
    static QList<QObject *> allObjects();
    template <typename T> static QList<T *> getObjects()
    { [...] }
    template <typename T> static T *getObject()
    { [...] }

    static QObject *getObjectByName(const QString &name);
    static QObject *getObjectByClassName(const QString &className);
    [...]
};
```

Object Pool II

```
class EXTENSIONSYSTEM_EXPORT IPlugin
    : public QObject
{
public:
    [...]
    void addObject (QObject *obj);
    void addAutoReleasedObject (QObject *obj);
    void removeObject (QObject *obj);
    [...]
};
```

Qt Creator Plugins



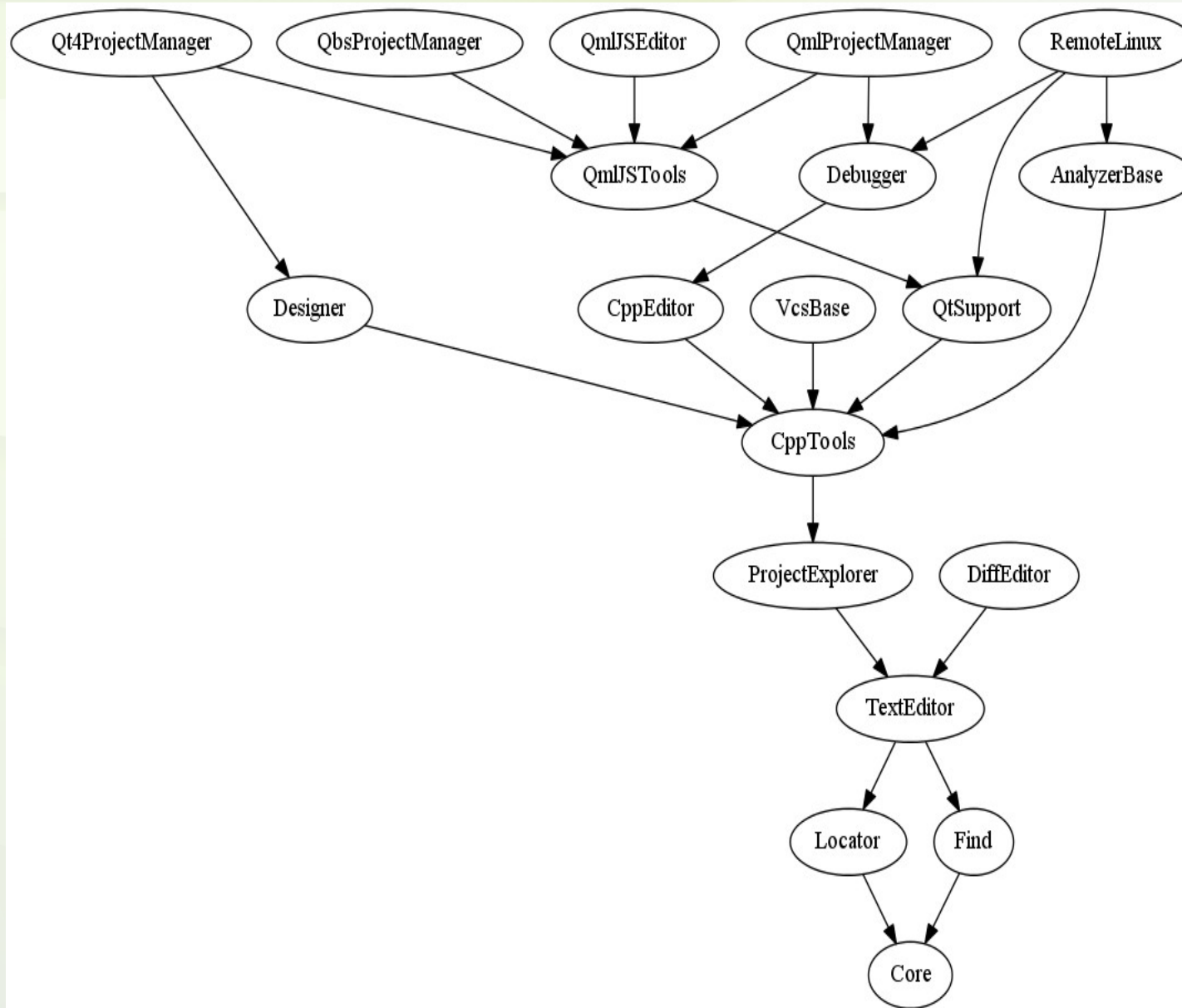
Qt Creator 3.0 promises:

- + API freeze starting with Beta releases
- + Source and binary compatibility in patch-level releases

Qt Creator does ***not*** promise:

- Compatibility in major or minor releases

Existing Plugins



Core Plugin

- `ActionManager`
- `DocumentManager` + `EditorManager`
- `DocumentModel`
- `HelpManager`
- `VcsManager`
- `MainWindow`
- ... many more

Core Plugin: ActionManager



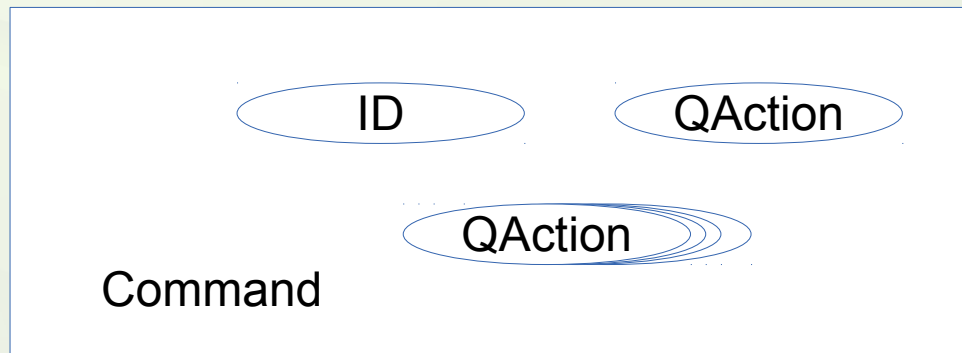
Developer
Days
2013

- Register QAction based on
 - Id
 - Context
- Get a Command back
 - Proxy QAction to use in UI.
 - Triggers (one of) registered QAction based on current Context

Actions & Commands

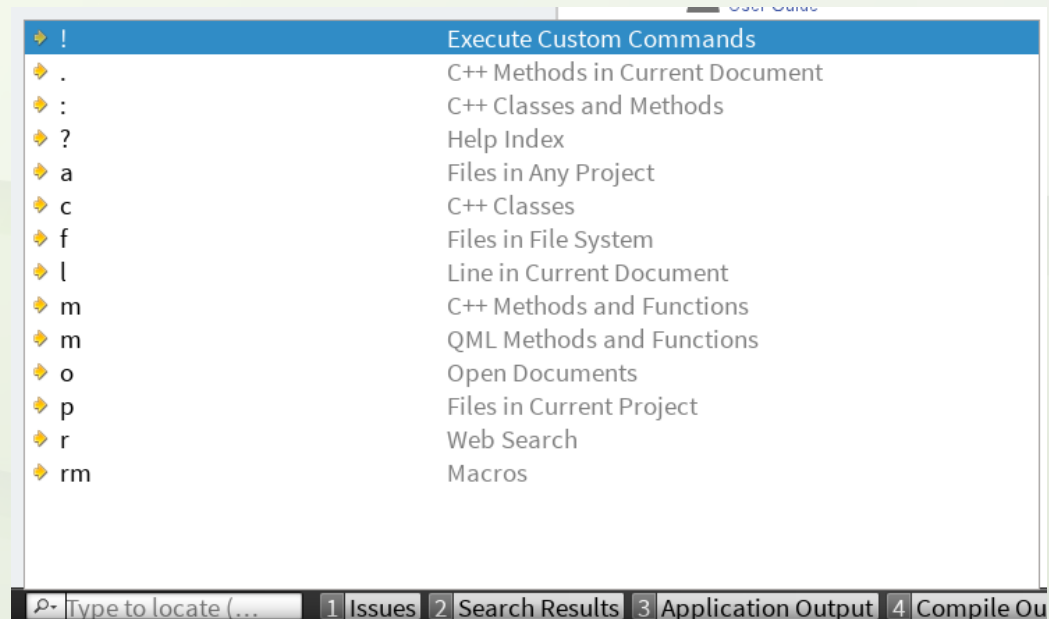
```
// Create an action to be triggered by a menu entry
QAction *helloWorldAction
    = new QAction(tr("Say \"&Hello World!\", this);
connect(helloWorldAction, SIGNAL(triggered()),
        SLOT(sayHelloWorld()));

// Register the action with the action manager
Core::Command *command =
    Core::ActionManager::registerAction(helloWorldAction,
        "HelloWorld.HelloWorldAction", context);
```



Locator Plugin

- ILocatorFilter

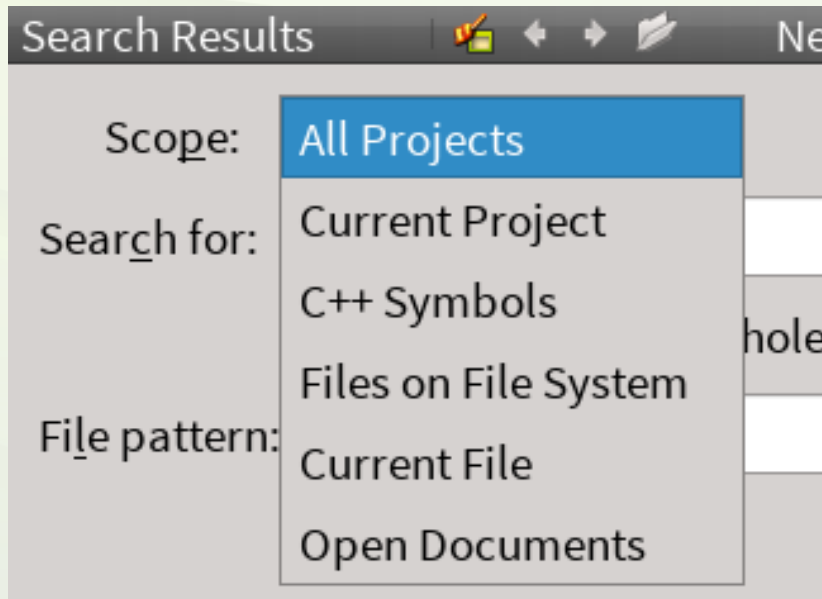


Find Plugin

- `IFindSupport`

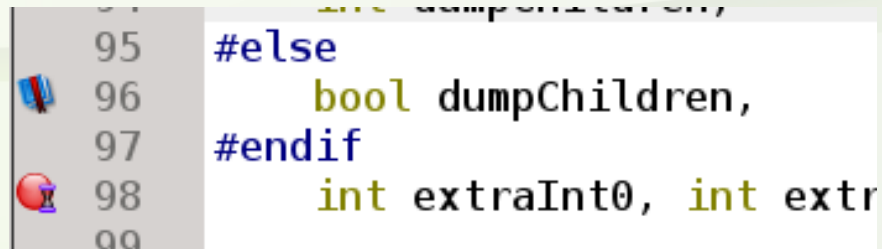


- `IFindFilter`



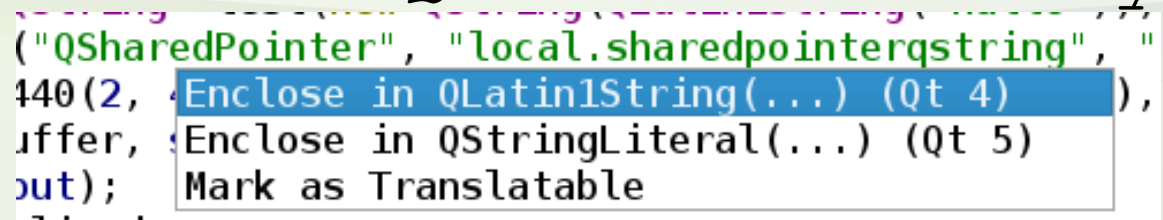
TextEditor

- ITextEditor + BaseTextEditor
- ITextMark



```
95 #else
96     bool dumpChildren,
97 #endif
98     int extraInt0, int extr
99
```

- QuickFixOperation + QuickFixFactory

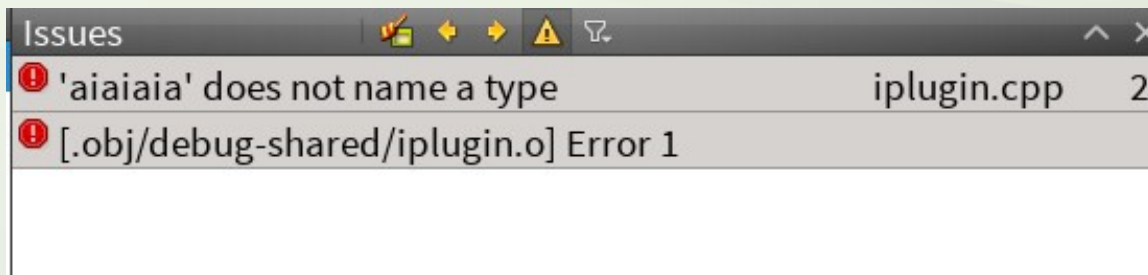
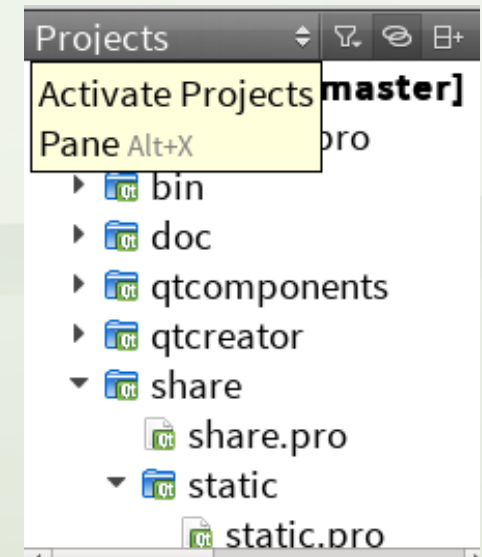


```
("QSharedPointer", "local.sharedpointerqstring", "
140(2, "Enclose in QLatin1String(...) (Qt 4)",
offer, "Enclose in QStringLiteral(...) (Qt 5)",
out); Mark as Translatable
```

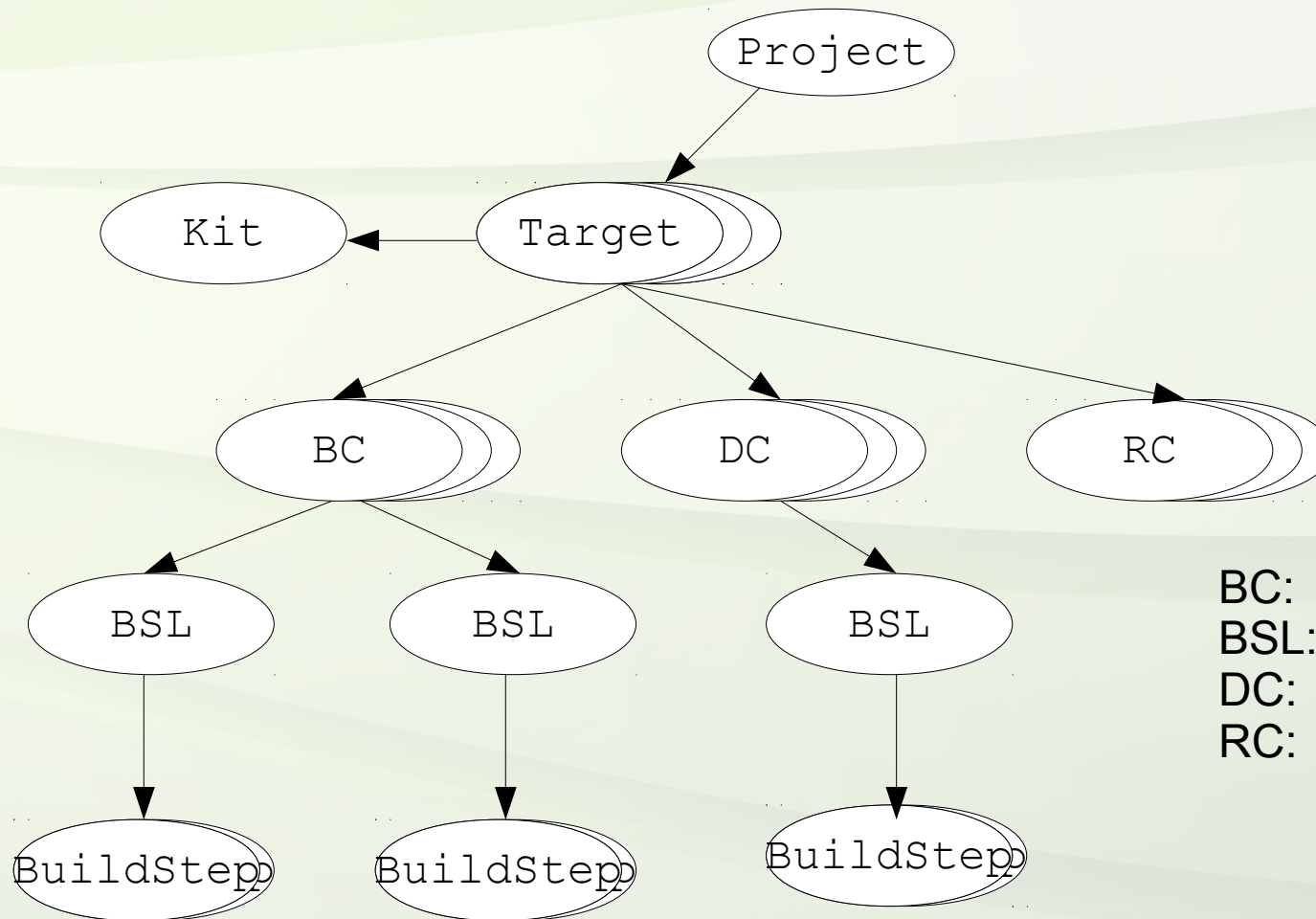
- SyntaxHighlighter

ProjectExplorer

- Project + related classes
 - Target, Kit + KitInformation
 - Build-/Deploy-/RunConfiguration
- ToolChain + ToolChainManager
- IDevice + DeviceManager
- ProjectNodes
- Task + TaskHub



Projects in Qt Creator



BC: BuildConfiguration
BSL: BuildStepList
DC: DeployConfiguration
RC: RunConfiguration

- `BaseQtVersion + QtVersionManager`

- `CppModelManager`
- **Code related to Cpp coding style:**
`CppCodeStylePreferences`, etc.

VcsBase

- `VcsBaseClient`
- `VcsBaseSubmitEditor`

Advice for Plugin Writers



- Do not load unstable plugins into your work environment
- Qt Creator plugin wizard: Directory + project name must match!
- Use Namespaces (check `.ui`-files!)
- Keep your Symbols private
- Check other plugins for inspiration
- Communicate with us

Documentation

- Extending Qt Creator Manual:

<http://doc-snapshot.qt-project.org/qtcreator-extending/extending-index.html>

- Qt Creator mailing list:

<http://lists.qt-project.org/mailman/listinfo/qt-creator>

- IRC:

#qt-creator on freenode.net

Thank you!

aurindam.jana@digia.com

tobias.hunger@digia.com

auri_, hunger on IRC (freenode.net)