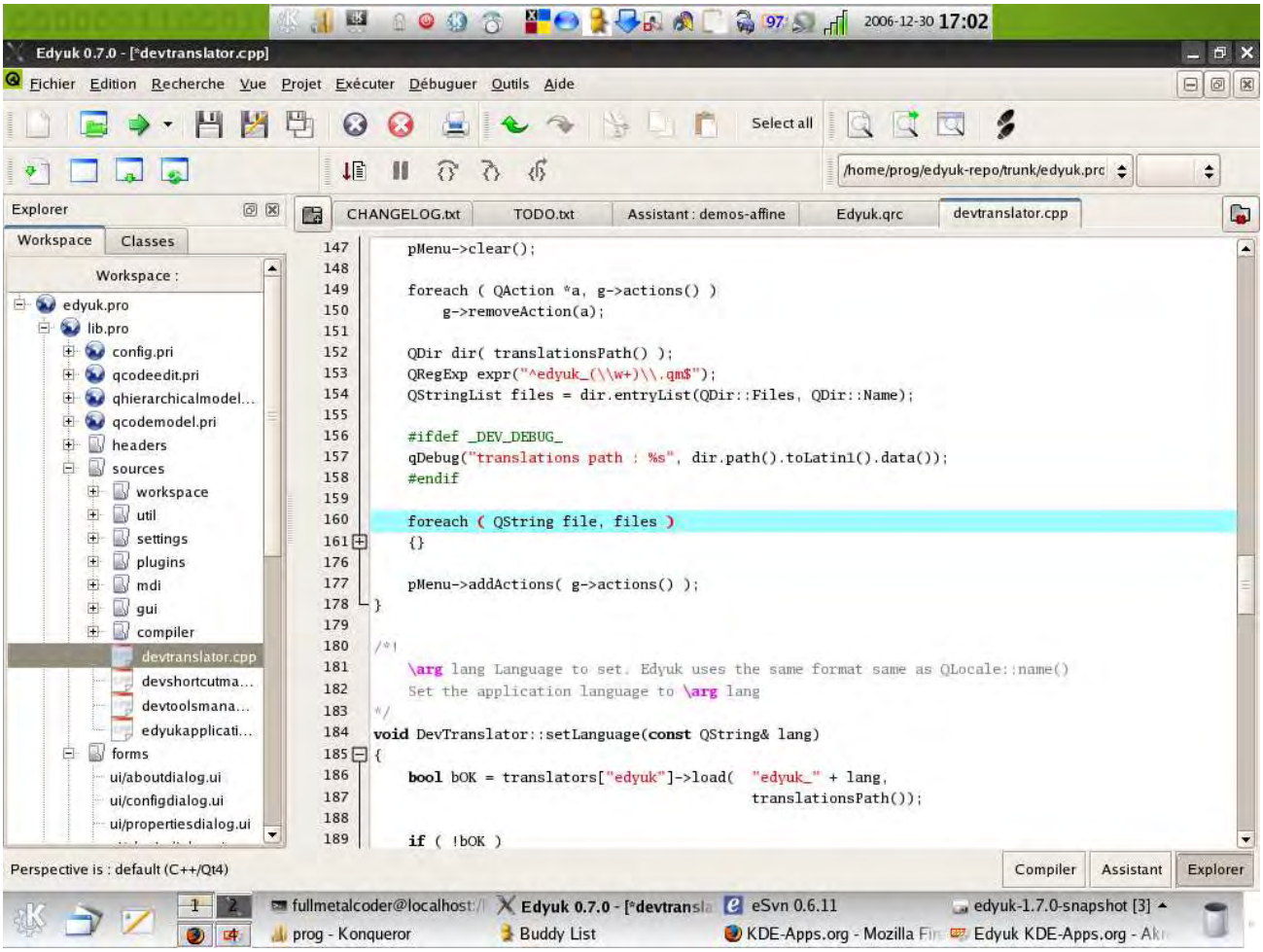
QtCentre Programming Contest

Application name: Edyuk
Category: Developpment tools
Author: fullmetalcoder

Overview :

Edyuk is meant to be a fully-featured Integrated Development Environment (IDE). Its primary target is, of course, C++/Qt software development, but it is designed to be easily extended through plugins which may change the GUI, add support for new build systems (i.e. Project files and makefile generators), new compilers and new debuggers.



Features :

Here is a, non-exhaustive, list of features which are useful to the user :

- Perspective-based GUI
- Extended edition framework that uses generic XML language definition files to perform syntax highlighting, auto indenting, parenthesis matching and text folding.
- Powerful project management which handles all specificities of the qmake project file format and allows easy modification without any data loss
- Compilation using GCC/MinGW with a progress dialog to show how the compilation is going and a table widget that holds parsed compiler output and jump to the location of the error in a click.
- Graphical debugging using GDB. Breakpoints managements directly in the editor widgets, various steping mode, backtrace view, registers display, a raw debug log and the possiblity of entering a custom command are among the possibilities offered by Edyuk.
- Class browsing and, later on, code completion
- A special widget is dedicated to the edition of Qt resource files (.qrc)
- Qt Assistant is integrated (a browser widget, a documentation tree, an alphabetical index and full text search)
- Qt Designer is integrated in a dedicated perspective
- Edyuk can be translated at run-time
- Mostly all shortcuts are configurable through a dedicated dialog, even those from plugins.

Architecture :

These technical considerations are of no use to a end-user but necessary if one wants to create new plugins. Edyuk is completely plugin-oriented. It basically means that without any plugins Edyuk can't do much, hopefully a default plugin is provided which handle C++/Qt development. The GUI of Edyuk is by essence dynamic because it is perspective based and also due to its use of dynamic menus which are “delocalized”. Indeed, every plugin, perspective, and editing widget (text or custom) can add new menu entries and toolbars thanks to a custom and highly flexible MDI layer.

To allow such a flexibility it has been required to split Edyuk into three basic components :

- A core library which holds the common base (mdi layer, editing framework, ...) and the implementation of all top level application objects

- A wrapper executable which only creates an application object and executes it.
- Plugins which are loaded and managed by the core components and provide a wide set of specific functionalities.

A wide range of algorithms and data structures are used throughout Edyuk. It includes :

- Model/View : project management and class browsing use custom models
- Factory pattern : the editing framework uses some custom implementation of that well-known object pattern to allow easy creation of new editor widgets
- Multi-threaded programming : all time-consuming tasks, such as compilation and class-browsing setup, are managed by separate thread to enhance the responsiveness of Edyuk
- DOM : XML files are used by various Edyuk's components, such as the shortcut manager, the tools manager and the generic editing engine
- and more...

Requirements :

Edyuk does not have any extra software requirements. Only Qt4 and a supported compiler are needed to build it. However, as with any IDE, some of its features need extra software at run-time, such as compilation and debugging which rely on the GNU suite (make, gcc and gdb). The presence of qmake (which is normally guaranteed by the dependency on Qt 4) is also required to build projects but not to deal with single file compilation.