How to Build Open Quark

Edward Lam

Last modified: November 9, 2007

Building the Open Quark distribution

This section contains instructions on how to build the Open Quark distribution from the provided source distributions.

Requirements:

- Java 5 SDK update 8 (or later).
- Ant 1.6.5 (or later).

  JUnit 3.8 (or later) as a library dependency (see http://ant.apache.org/manual/install.html#librarydependencies)

  a good JUnit .jar to use is here:
  `(quark.src.folder)/import/Titan_Research/win32_x86/release/bin/External/java/`
  For instance, copy this into `(path to ant root)/lib/`

- Eclipse 3.3.0 Platform - Windows build (exact version required)
  Java Development Tools
  Plug-in Development Environment

  Note that the build scripts build against a specific version of the Eclipse platform since they contain references to the specific plugin .jars which are part of the Eclipse platform. The names of these .jars change from release to release.

  Note also that the build references classes which exist in OS platform-specific .jars. It is however possible to build on other platforms (such as Linux) against the Windows platform Eclipse build.

Steps:

1) Unzip the Quark Platform and Plug-in source distributions to the *same* folder.

   A few files are duplicated between the two distributions - these do not have to be unzipped the second time.

2) Run ant (with appropriate properties) using the build file: `OpenQuark_Build/Build_Scripts/build.xml`

   Probably the only property you will need to set is "`eclipse.plugins.dir`".

   For example:

```
> (path to ant root)/bin/ant
     -Declipse.plugins.dir=(eclipse plugins folder)
     -f (quark.src.folder)/export/OpenQuark_Build/Build_Scripts/build.xml
```

By default the output build files will end up in a folder named "`QuarkBuild`" in the parent of `quark.src.folder`.

Properties used by the build script:

`eclipse.plugins.dir`	The plugins folder the required Eclipse distribution. Eclipse 3.3.0 is currently required.
`quark.src.folder`	The folder containing the unzipped platform and plugin sources. This folder is typically the "src" folder whose parent is named "Quark".
`build.outputfolder`	The folder to which build artifacts will be written.

Building other Open Quark collaterals

This section contains instructions on how to build various collaterals provided with Open Quark, which are not built by the main build target in the build scripts.

Building the Lexer/Parser from the ANTLR Grammar

The CAL Lexer and Parser are generated from an ANTLR v2 grammar (http://antlr.org/).
The ANTLR tool to generate these can be invoked either via the build scripts or manually.

Building the Lexer/Parser using the build scripts

The CAL lexer and parser can be generated by calling the "`gen-parser`" target in the Open Quark build script.

For instance, follow all the steps described in building the Open Quark distribution, but adding the target name to the call to ant:

```
> (path to ant root)/bin/ant
    -Declipse.plugins.dir=(eclipse plugins folder)
    -f (quark.src.folder)/export/OpenQuark_Build/Build_Scripts/build.xml
    gen-parser
```

This generates the lexer and parser files directly to the directory in which they will be used,
ie. `(quark.src.folder)/CAL_Platform/src/org/openquark/cal/compiler/`

Building the Lexer/Parser manually

The ANTLR grammar files for CAL are found under:
`(quark source folder)/CAL_Platform/antlr grammar/`

The CAL lexer and parser can be generated from these by invoking the ANTLR tool on the following grammar files in turn: `CALDoc.g`, `CAL.g`, `CALTreeParser.g`

For instance change to the folder above and perform the following invocations:

```
> java -cp (path to antlr.jar) antlr.Tool CALDoc.g
> java -cp (path to antlr.jar) antlr.Tool CAL.g
> java -cp (path to antlr.jar) antlr.Tool CALTreeParser.g
```

where `(path to antlr.jar)` is:
`(quark.src.folder)/import/Titan_Research/win32_x86/release/bin/External/java/antlr.jar`

Note that the above invocations will generate the files to the current directory, so to incorporate any changes these files should be copied to the proper location,
ie. `(quark source folder)/CAL_Platform/src/org/openquark/cal/compiler/`

Generating DocBook-based Documentation

Information about this is described in the document: Generating DocBook Documentation.pdf.

Generating PDFs from Microsoft Word Documents

This may be done via the following methods:
1) Print to PDF using PrimoPDF (http://www.primopdf.com/)
2) Export to PDF using MS Office 12 with the Save as PDF or XPS Add-in.

The latter method has the advantage of preserving hyperlinks, however it requires a licensed copy of Microsoft Office 12.

Generating the Gem Cutter Technical Paper PDF from LaTeX

The LaTeX sources for the Gem Cutter Technical Paper are found under:
`(quark.src.folder)/Quark_Gems/Docs/latex/Gem_Cutter_Technical_Paper/`

This paper is designed to be used with the LaTeX class file for SIGPLAN conference papers.
As such, the class file and the 9pt template should be downloaded from
http://www.sigplan.org/authorInformation.htm and placed these in the folder above.

Generation of the Gem Cutter Technical Paper can be done via the following series of invocations:

```
> pdflatex gemcutter-techpaper
> bibtex gemcutter-techpaper
> pdflatex gemcutter-techpaper
> pdflatex gemcutter-techpaper
```

A Unix-style Makefile demonstrating this can be found in the file: `Makefile`

A Windows batch file demonstrating this with MiKTeX can be found in the file: `makePDF.bat`
This may be used after updating to point to the local installation of MiKTeX.
With MiKTeX 2.6, this has been confirmed to work with a basic install augmented with the natbib package.