Getting Started with Open Quark

Jennifer Chen

Last modified: Oct 26, 2007

Overview

This guide provides an overview of the Open Quark framework for new users. It provides a brief overview of the technology and points out the available resources for the platform and its tools.

Open Quark in a nutshell

Quark Platform is a suite of technologies developed by Business Objects. It is a complete environment for developing applications. Its main components include:

- **CAL**: a general purpose, lazily evaluated, strictly-typed functional language similar to Haskell. Examples of the features are:
    - ***Java integration*:** Easy and efficient integration with Java allows you to use functionally declared objects in an object oriented environment. Users are also able to use Java types and call regular Java logic in CAL.
    - ***Compiler and runtime:*** both fully multithreaded. The compiler is able to dynamically compile logic into Java bytecode and execute it on the Java Virtual Machine.
    - ***Debug and exception support***
    - ***CALDoc:*** source embedded documentation
    - Extensive ***library modules***
    - ***Metadata*** on CAL entities
- **Eclipse integration**: Eclipse plug-in for facilitating development with the CAL language
- **Interactive CAL Environment (ICE)**: an interactive text-based shell that enables the evaluation of CAL expressions
- **Gem Cutter**: a graphical tool that encapsulates pieces of CAL code into “Gems”. The user-friendly gems can be easily connected to create large and complicated logic. It is a great way to start developing in CAL.

First things first: installing Open Quark framework and tools

The platform is available in 2 forms: binary distribution and sources. The zip files are available for download from the Software section on the Quark website.

Simple install:

Unzip the binary distribution zip file. The unzipped folder contains executable batch files and shell scripts.

If you wish to create binary projects or source projects within Eclipse, download the corresponding zip file and read "Setting up Eclipse for CAL" section in Using CAL with Eclipse.

CAL Eclipse plug-in installation:

The handy Eclipse plug-in can be installed either by going to the Eclipse update site or downloading a zip file containing the archived update site for offline installation. Read Using CAL with Eclipse for detailed instructions and tutorial on the features. Source code for the plug-in is also available for download from the Quark website; however, this cannot be used to directly install the plug-in.

Recommended Resources

There are many resources in the Documents and Video section on the website to help you get familiarized with Open Quark. Depending on your background, you may find certain documents helpful. Videos can be streamed by following the links, or alternatively can be downloaded from the website in QuickTime or DivX formats.

If you are new to functional programming languages:

CAL and the Interactive CAL Environment (ICE) Shell
This 45 minute video demonstrates a number of features of the CAL language itself, as well as some development tool capabilities.

Sample code
Highlights syntactic features of CAL through examples. Familiarizing yourself with functional programming concepts will make subsequent demonstrations easier to grasp.

- ❖ Hint: It is tough to learn programming languages all by reading. Try using the Gem Cutter's code gems to dynamically evaluate snippets of code to practice.

CAL User's Guide
Everything you want to know about CAL, including syntax, standard library modules, language reference, and standard library reference. It is a great reference to keep around.

If you have some familiarity with functional languages:

CAL for Haskell Programmers
Summarizes some differences between CAL and Haskell in syntax, usage and performance. If you already have some knowledge of Haskell, this document will help you fill in the specifics for CAL.

Effective CAL
Highlights some good coding practices in CAL. This is for those who are comfortable with programming in CAL and want to take CAL to the next level.

About the Gem Cutter:

Introduction to Gems and the Gem Cutter (Part 1) and (Part 2)
Introduce the basics of the Gem Cutter and its graphical language. This is a nice place to start if you are new to functional programming concepts.

The Gem Cutter Manual
Explore the features of the graphical tool through examples. This is a beginner's guide to the Gem Cutter, and you can practice the tricks you have seen from the videos. The manual is also available from the Gem Cutter's menubar (**Help → Help Topics**).

The Gem Cutter - A Graphical Tool for Creating Functions in the Strongly-typed Lazy Functional Language CAL
This is a technical paper explaining the novel features in the Gem Cutter for people already knowledgeable about functional programming. It digs under the surface to explain the elements that make the Gem Cutter work as a simple and yet expressive visual programming language.

About ICE:

CAL and the Interactive CAL Environment (ICE) Shell
This video demonstrates a number of features of the CAL language itself, as well as some development tool capabilities.

Interactive CAL Environment (ICE) Manual
Describes the text-based interactive environment for evaluating CAL expressions, obtaining information about CAL entities, refactoring, debugging, etc

Development in CAL:

Walkthrough of some Developer Tool Features
Demonstrates some developer tools available, including tools for working with CALDoc and CAL archive files.

A Tour of the Eclipse Plug-in features
Showcases features of the Eclipse plug-in. It is essential if you are developing in Eclipse.

Using CAL with Eclipse

This document will guide you through installation and show you the latest and greatest features of the plug-in.

Developing Eclipse Plug-ins that make use of CAL Workspaces
This document provides instructions on how to setup the Quark Platform to use CAL workspaces within the context of an Eclipse plug-in, so that clients using CAL may continue to operate.

Integration of CAL with Java:

Java Meets Quark
Shows the Framework SDK and illustrates API use with examples. This document will continue to evolve to deal with more SDK concepts.

Using JFIT
Explains the use of the Java Foreign Import Tool. This allows the batch creation of CAL interfaces to Java code for convenience. Existing Java classes can be used in CAL to enhance code reusability.

Business Activity Monitor Overview
Discusses a complete sample Java application “Business Activity Monitor” (BAM), included in the Quark Framework download. It gives insight on some non-trivial uses of Java and CAL in developing applications.

Library and SDK documentation:

The current CAL library documentation is downloadable and browsable.

The current Java SDK library documentation is downloadable and browsable.

About CALDoc:

CALDoc Crib Sheet
Short reference document on the syntax and recommended usage guidelines for CALDoc.

Using the Command Line Interface for the Documentation Generator
Shows how to use the CALDocTool from the command line to generate HTML documentation from CALDoc and metadata for the various entities in a CAL workspace.

Deployment:

Using Quark with Standalone JARs

Describes how to create and run applications that are written in CAL and are packaged as *standalone JARs*.

Using CAL Archives (Car files)
Shows how a CAL program can be deployed as part of a Java application using CAL Archives (Car files). Demonstrates how to build and use Car files using different tools like ICE and Gem Cutter.

Implementation:

CAL and the Computer Language Benchmarks Game
Describes the results of implementing the Computer Language Benchmarks Game (http://shootout.alioth.debian.org/) in CAL. It demonstrates that the performance of CAL is comparable to that of Java for this benchmark suite.

CAL Global Optimizer
Describes the transformations performed by the CAL Global Optimizer during an optimization pass written in CAL.

CAL Runtime Internals
Describes the internals of CAL's LECC runtime. This document is a work in progress.

Generating DocBook Documentation
Describes how documents are generated from our DocBook source files.

How to Build Open Quark
Explains the build process for Open Quark and various collaterals (for example, generating PDFs, Docbook, antlr grammars etc).

Additional Help

The Open Quark Framework for Java and the CAL Language
The official Business Objects website for the Open Quark Framework. This site contains updates about Quark as well as links to all the resources.

Please visit our forum on Google groups for discussions.

More reading is available on Open Quark on Wikipedia, which gives a summary of the Quark platform.