CQL Project Layout

# Overview

The CQL project holds the CQL language, tools associated with it, a Java bean API for working with the language programmatically, and assorted CQL implementations against various data sources (e.g. Hibernate via HQL). This project is maintained as a caGrid Community Project under the caGrid Knowledge Center, which provides for open access, a consistent build / test system, and a readily available Ivy repository into which artifacts generated by the project can be published.

# Repository Layout

The CQL project repository should be structured so as to allow future growth and addition of new sub-projects without interfering with existing projects or requiring major changes to the build and publish process.

/cql

/resources

/schema

/cql1

/cql2

/software

/beans

/cqlToHQL3.2

/cql2toHQL3.2

/utils

/docs

The *resources* directory contains libraries and other external artifacts on which the CQL project’s software depends. Ideally this should be kept to an absolute minimum so as to avoid conflicts with libraries in projects which depend on the CQL project.

The *schemas* directory houses the official CQL and CQL 2 XML schema documents.

The *software* directory is the base directory of any and all software tooling provided by the CQL project. Each subdirectory of this one is dedicated to a single project. Some examples have been listed in the above source tree diagram.

The *docs* directory contains any documentation pertaining to CQL or the software projects in the repository. Examples include design documentation, user’s guides, requirements, and notes.