RAQUEL Project Policy Regarding the Source Code Repository

Introduction

This document summarises RAQUEL project policy with regard to use of the RAQUEL source code repository. The repository uses Subversion on the SourceForge site, but although this inevitably influences the policy, the policy decisions described here are largely independent of Subversion and SourceForge.

See the book "*Version Control with Subversion for Subversion 1.7*" or the earlier book "*Version Control with Subversion for Subversion 1.5*" for details of how to use Subversion. Different editions and versions are available from http://svnbook.red-bean.com/. PDF versions of versions 1.7 and 1.5 can be obtained from http://svnbook.red-bean.com/en/1.7/svn-book.pdf and http://svnbook.red-bean.com/en/1.5/svn-book.pdf respectively.

See the introductory 'Help' facilities on https://sourceforge.net/p/forge/documentation/svn/ for details of how to use Subversion on SourceForge.

Repository Structure

The repository has a hierarchical structure. Currently there are 7 directories[1] below the repository root :

1. **RaquelPrototype**. This holds the code of the Proof of Concept prototype of the RAQUEL DBMS.
2. **RaquelGUI**. This is for the development of the RAQUEL GUI, also known as the Raquel Teaching Tool.
3. **RaquelDBMS**. This is for the development of the DBMS itself.
4. **StorageStacks**. Each subdirectory here is for a specific kind of storage stack.
5. **ScalarTypes**. Each subdirectory here is for a specific scalar data type.
6. **Information**. This holds standard Open Source 'Boilerplate' files plus other files containing generally useful information.
7. **ZipFiles**. This holds copies of the zip files used to download software from SourceForge.

The directories fall into 2 categories :

- Software Development directories : **RaquelDBMS**, **RaquelGUI**, **StorageStacks**, and **ScalarTypes**.
- Support directories : **RaquelPrototype**, **Information**, and **ZipFiles**.

---

1 The 'Information' directory was added on 16th January 2017, and the 'ZipFiles' directory on 12th September 2018. Prior to this, there were only the other 5 directories.

Software Development Directories

These directories have a structure based on what is considered 'good practice' for a software repository. The structure consists of three subdirectories called :

1. Trunk,
2. Branches,
3. Tags.

The 'Trunk' holds the main line of software development. Typically the software in it evolves either as a result of debugging (bugs being found by user tests, in Beta, Alpha or released versions) or by having developments created elsewhere 'merged' into it.

'Branches' is where developments are created for later merging into the 'Trunk' version. 'Branches' will itself have many subdirectories, one for each development.

'Tags' is where software releases to users are held, each release in its own subdirectory. Each release is normally a specific copy of the main development trunk at a point in time when the release was published. For further details see chapter 4 of "*Version Control with Subversion for Subversion*", either versions 1.5 or 1.7 - this chapter explains 'good practice' and how to implement it in Subversion.

The **RaquelDBMS** directory is to support the development of the core modules of the RAQUEL DBMS, and so is split into the three subdirectories 'Trunk', 'Branches' and 'Tags'. Core modules are those required by the DBMS regardless of the storage stacks and scalar types made available.

The **RaquelGUI** directory is for the 'driver application' known as the Raquel GUI or Raquel Teaching Tool, and is also split into the three subdirectories 'Trunk', 'Branches' and 'Tags'.

For the time being at least, this application is essentially complete. (The original Raquel Teaching Tool was completed some years ago. It was incorporated wholesale into the Proof of Concept prototype, and since then has undergone minor revision, extension and debugging). Future developments are expected to be developments of the menus to support additional DBMS functionality (and possibly support for an animator).

Other future kinds of 'driver applications' are anticipated to be independent of **RaquelGUI** and have their own (top-level) directories in the repository.

The **StorageStacks** directory is expected to have a subdirectory for each kind of storage stack.

Likewise the **ScalarTypes** directory will have one subdirectory for each kind of scalar type.

It is intended to build up a portfolio of stacks and types. It is better if the source code for each is managed independently and kept separate, each in their own subdirectory. Each subdirectory would have a 'Trunk', 'Branches' and 'Tags' subdirectory structure, as appropriate.

Support Directories
These directories have a structure appropriate to their role.

The **RaquelPrototype** directory has two subdirectories.

The subdirectory called 'Original Code' holds the prototype DBMS software as delivered by the developers of the code in August 2009.

The subdirectory called 'CodeOrganisedByDocumentation' contains the same code but re-organised so as to be consistent with the documentation about the prototype produced by the code developers.

RaquelPrototype is effectively an archive, and it will not be used for further development apart from reference purposes.

The **Information** directory holds information files. It holds those files relevant to all software, plus a DBMS subdirectory for those relevant just to the DBMS, and a Teaching Tool subdirectory for those relevant just to the GUI/Teaching Tool.

Additional subdirectories could be added in future for information files relevant just to other software.

The **ZipFiles** directory has two subdirectories, called 'DBMS' and 'TeachingTool'. Each contains an archive of the zip files used to download the DBMS and GUI/Teaching Tool respectively, together with the currently used zip files.

Additional subdirectories could be added in future for zip files used for other software.

End-Of-Line Handling

Different development platforms have different conventions as to how the end of a line of code is represented. In particular, Linux/Unix systems use a linefeed (= LF) character and Microsoft Windows systems use a carriage return character followed by a linefeed (= CRLF). Since it is possible that the RAQUEL software may be developed on both Linux and Windows systems, the two different line endings could cause confusion as to what genuine code changes have been made when updated code is committed to the repository.

Therefore the Subversion property `svn:eol-style` must be attached to all source code files, with the property value `native`. This enables the following. While source code is always stored in the repository with LF end-of-lines, when users check out working copies of the code onto Windows platforms they will have CRLF end-of-lines, and users checking out working copies of the code onto Linux platforms will have LF end-of-lines; when the two sets of users commit revised code back to the repository, it will be stored with LF end-of-lines.

To achieve this requires that every Subversion client has its configuration file edited to have the appropriate contents. Configuration files are :

- `~/.subversion/config` on Linux/Unix systems,
- `%APPDATA%\Subversion\config` on MS Windows systems.

The configuration should be opened with a text editor and its contents changed and stored as required.

Revision Dates

Since the revision numbers associated with source code have no temporal significance, it is useful to assign a date and time to a software release to make its management easier. The date and time will be assigned at check out time. As a number of different date and time formats are provided, the project will standardise on the one illustrated by the following checkout example :

```
svn checkout -r {"2010-02-08 15:30"}
```