# 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 six 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, a.k.a. 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 others with generally useful information.

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

---

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

The other 4 source code directories have a structure that is 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 source code for the ‘driver application’ known as the Raquel GUI or Raquel Teaching Tool is held in the ‘RaquelGUI’ directory. For the time being at least, this application is completed and is subject only to minor revisions and debugging. (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). Thus the ‘RaquelGUI’ directory only has a ‘Trunk’ subdirectory, which holds the latest version of the source code, and a ‘Tags’ subdirectory which itself contains a subdirectory for each release. This allows scope for a ‘Branches’ subdirectory in future should the need arise.

The ‘RaquelDBMS’ directory is to support the development of the core modules of the RAQUEL DBMS. Consequently it is split into the three subdirectories ‘Trunk’, ‘Branches’ and ‘Tags’. ‘Trunk’ and ‘Tags’ start with identical copies of the prototype DBMS, since in the former case this is the starting point for future DBMS development, and in the latter case constitutes the first release of the DBMS.

The ‘StorageStacks’ directory will be split into subdirectories, one for each kind of storage stack. Likewise the ‘ScalarTypes’ directory will be split into subdirectories, one for each kind of scalar type. A portfolio of stacks and types is to be built up, many of which will be Open Source contributions or based on Open Source contributions. It is better if the source code for each is managed independently of the rest, and so they are kept separate from the core of the DBMS (held in ‘RaquelDBMS’), each in their own subdirectory. In Subversion terminology, each such subdirectory may be called a ‘Vendor Branch’ although their software will not be from commercial vendors. In each subdirectory, ‘Trunk’, ‘Branches’ and ‘Tags’ subdirectories will be used as appropriate.

Future ‘driver applications’ may require additional top-level directories within the repository.

The ‘Information’ directory does not hold source code but information files. The directory holds those files which are generally applicable, plus a DBMS subdirectory holding those files specific to the DBMS, and a Teaching Tool subdirectory holding those specific to the GUI/Teaching Tool. Additional subdirectories could be added in future for information files about specific pieces of 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”}
```