

# jxCAL CVS Repository Layout

by Cédric VIDAL

*This document details the layout of the cvs repository.*

## 1. Introduction

The CVS repository is a lot componentized. When one wants to work with the project, one will have to checkout different chunks of the repository. This is indeed complicated for a little project such as jxCAL, but as I don't really know where the project is going on, I prefer to have an over modularized repository which will get easier to work with when checkout scripts become available rather than an over simplistic repository which would have to be redone for future expansion. The focus is to make things simpler as the time goes on though.

## 2. Modules

- documentation/

That module contains the documentation for the project. The preferred documentation format is docbook xml as it will be included directly in the web site building process.

- core/

This module contains the heart of the project. There is only interpreted material, no material related to cocoon. The idea is to separate what depends on cocoon from what does not in order to be able to work with jxCAL without cocoon.

- core/views/

The views transform jxCAL calendars into different flavours, they are organized by categories such as week, month etc ...

These are mainly XSLT transformations that would occasionally go along with other material such as Javascripts, XML documents.

- `core/schemas/`

Schemas of the different types of documents used by jxCAL. They are organized by type. For now, the preferred schema is DTD but that might change, and more schema types might be supported later.
- `core/schemas/{schema-type}/{schema-name}.{schema-type-extension}`

Schema {schema-name} specified in the schema type {schema-type}.
- `core/views/{output-type}/{view-category}/`

{output-type} could be svg, ical or any other future format. {view-category} is any category supported by the output type. Views are not cocoon view components.
- `cocoon/`

jxCAL has been created from the beginning to work with Cocoon. This module contains all cocoon related material.
- `cocoon/features/`

Each jxCAL cocoon specific feature is put in this folder, for example, the aggregation feature is located in `cocoon/features/aggregation/`
- `cocoon/features/{feature-name}/src/java/`

Java sources related to the feature {feature-name}. The resulting compiled classes should be embedded in a jar `jxcalfeature-name-version.jar`
- `cocoon/features/{feature-name}/src/{source-type}/`

{source-type} being stylesheets, xsp, or any other type of source required by the feature.
- `examples/`

Here we will put example materials, often sample XML documents and XSLT transformations which are not general enough to be usefull in the core, showing only the usage of jxCAL. (well examples ... :o)
- `website/forrest/`

The site will probably be powered by Forrest (which is powered by cocoon of course ...). The jxCAL documentation which is not web specific will be in the module `documentation/` though. Documentation in docbook format will be directly included in the forrest transformation process, but must not contain any forrest dependencies.

*jxCAL CVS Repository Layout*