

# Directory Structures

---

## Overview

This page describes preferred directory structures for *openEHR* Subversion repositories. It should be treated as a guide rather than an absolute prescription, since some languages and development frameworks require their own structures. Nevertheless, the structure near the top of the tree should be respected as closely as possible, in order to make sure all repositories are as consistent as possible.

## Subversion "control" directories

To manage tags and branches, Subversion does not provide any inbuilt features in the repository directory structure, but its lazy copy methodology does support rerepresentation of tags and branches once created. For this reason, Subversion repository administrators normally include a top level directory hierarchy like this in the repository:

- trunk
- tags
- branches

In *openEHR*, we have opted to capitalise these names, to distinguish them from normal directories, and add a directory for release branches. There are basic rules for how to work in these top-level directories, which are shown against the directories used in most *openEHR* repositories.

- TRUNK - normal development
- BRANCHES - lazy copy of some existing revision in TRUNK - any modifications allowed
- RELEASES - lazy copy of some existing revision in TRUNK - bugfix modifications only allowed
- TAGS - lazy copy of some existing revision in TRUNK - not to be modified!

Most users will just work by checking out the contents of the TRUNK directory from the server, making changes, and committing them.

## Mainline Directory Structure

There are two kinds of repositories: source and distribution. Multiple source (including documentation) repositories create software, the binaries of which are released via a distribution repository. To date, *openEHR* uses only a single distribution repository, known as *oe\_distrib*.

The directory structure in use for software source repositories is as follows:

```

|           |      +-- doc           # specific documentation for appX
(if any)   |
|           |      +-- appY
|           |      +-- appZ
|           |      +-- components     # source area for binary reusable
components (.so, .dll, etc)
|           |      +-- compX
|           |           |      +-- lib           # build area for compX
|           |           |      +-- src           # source code for compX (over and
above source in 'libraries' area)
|           |           |           |      +-- ...
|           |           |           |      +-- etc           # config files for compX
|           |           |           |      +-- doc           # specific documentation for compX
(if any)   |
|           |           |      +-- test          # test cases for compX
|           |           |      +-- compY
|           |           |      +-- compZ
|           |           |      +-- libraries     # main source code for apps,
components
|           |           |      +-- libX
|           |           |           |      +-- src           # source code for libX
|           |           |           |           |      +-- ...
|           |           |           |           |      +-- doc           # specific documentation for libX
(if any)   |
|           |           |           |      +-- test          # test cases for libX
|           |           |           |      +-- libY
|           |           |           |      +-- libZ
|           |           |           |      +-- distrib     # any config & install files needed
for installation of apps & components
|           |           |           |      +-- doc           # repository-wide documentation
|           |           |           |      +-- scripts      # scripts required for maintaining
the repository

```

In the above, the three main top level directories are apps, components, libraries. These directories are all source directories, of which the last contains most of the re-usable source code. The source code (and configuration files, documentation etc) in the other two directory trees include files particular to the relevant applications and components, e.g. GUI code, relevant build files (e.g. to specify a dll build). The actual generated binaries will of course appear in these directories, due to the developer working, but should not be uploaded to the source Subversion repository; instead, they should be put into an appropriate place in the distribution repository (see [http://www.openehr.org/wsvn/oe\\_distrib/windows/?rev=0&sc=0](http://www.openehr.org/wsvn/oe_distrib/windows/?rev=0&sc=0) for the structure of this).