Development hints and tips

How to use make in a build directory ?

The command bv_maker configure generates the makefiles from the *CMakeLists.txt* files in the source files. The makefiles are created in the build directory, so it is possible to use make directly in the build directory to build a specific target.

For example:

```
cd <build_directory>
make anatomist
make AimsFileInfo
make axon-usrdocbook
```

In the following example, the anatomist and AimsFileInfo targets are associated to the rules that enables to create the matching executables. The target axon-usrdocbook runs the generation of the docbook documentation for the project axon.

To get the list of available targets: make help.

It is also possible to compile a specific project by executing make in the corresponding directory <build_directory>/build_files/<project>.

How to remove dead links in the build directory ?

On Linux, bv_maker creates symlinks in the build directory instead of copying the sources when it is possible. It is the case for python files for example, as they doesn't need to be compiled, the symlink in the build directory enables to test modifications of a python file without having to run again bv_maker build.
But when source files are moved, the symlink in the build directory can point to nothing. To remove these deadlinks, you can run the script **bv_nobadlink** in the build directory. This script is in the project *development/brainvisa_cmake*.

```
cd <build_directory>
bv_nobadlink
```