

Building of Open CASCADE Technology (OCCT) 6.5.2 on Linux

1. BUILDING OF THIRD-PARTY PRODUCTS

There are two types of third-party products, which are necessary for building OCCT:

- Mandatory products: Tcl 8.5.8, Tk 8.5.8, TclX 8.4.0, FreeType 2.3.7, Ftl 2.1.2.
- Optional products: gl2ps 1.3.5, FreeImage 3.14.1, TBB 30-018.

1.1. BUILDING OF MANDATORY THIRD-PARTY PRODUCTS

1.1.1. Tcl 8.5.8

- Enter the directory where source files of Tcl are located (<TCL_SRC_DIR>).

```
cd <TCL_SRC_DIR>/unix
```

- Run configure command.

```
./configure [--enable-64bit] --enable-gcc --enable-shared  
--enable-threads --prefix=<TCL_INSTALL_DIR>
```

- If configure command is finished successfully, perform building process.

```
make
```

- If building is finished successfully, perform installation of Tcl. All binary and other files of the Tcl product will be put into the directory specified by <TCL_INSTALL_DIR>.

```
make install
```

1.1.2. Tk 8.5.8

- Enter the directory where source files of Tk are located (<TK_SRC_DIR>).

```
cd <TK_SRC_DIR>/unix
```

- Run configure command, where <TCL_LIB_DIR> should be specified as <TCL_INSTALL_DIR>/lib.

```
./configure [--enable-64bit] --enable-gcc --enable-shared  
--enable-threads --with-tcl=<TCL_LIB_DIR>  
--prefix=<TK_INSTALL_DIR>
```

- If configure command is finished successfully, perform building process.

```
make
```

- If building is finished successfully, perform installation of Tk. All binary and other files of Tk product will be put into the directory specified by <TK_INSTALL_DIR> (usual approach is to specify <TK_INSTALL_DIR> same as <TCL_INSTALL_DIR>).

```
make install
```

1.1.3.TclX 8.4.0

- Enter the directory where source files of TclX are located (<TCLX_SRC_DIR>).

```
cd <TCLX_SRC_DIR>
```

- Run configure command, where <TCL_LIB_DIR> should be specified as <TCL_INSTALL_DIR>/lib and <TK_LIB_DIR> should be specified as <TK_INSTALL_DIR>/lib.

```
./configure [--enable-64bit] --enable-gcc --enable-shared  
            --enable-threads --with-tcl=<TCL_LIB_DIR>  
            --with-tk=<TK_LIB_DIR> --prefix=<TCLX_INSTALL_DIR>
```

- If configure command is finished successfully, perform building process.

```
make
```

- If building is finished successfully, perform installation of TclX. All binary and other files of TclX product will be put into the directory specified by <TCLX_INSTALL_DIR> (usual approach is to specify <TCLX_INSTALL_DIR> same as <TCL_INSTALL_DIR>).

```
make install
```

1.1.4.FreeType 2.3.7

Note, that FreeType is built and used as a static library in OCCT.

- Enter the directory where source files of FreeType are located (<FREETYPE_SRC_DIR>).

```
cd <FREETYPE_SRC_DIR>
```

- Run configure command. Note, that CFLAGS option shown below is required on 64bit platforms only.

```
./configure --enable-shared=no --prefix=<FREETYPE_INSTALL_DIR>  
[CFLAGS="-fPIC -m64"]
```

- If configure command is finished successfully, perform building process.

```
make
```

- If building is finished successfully, perform installation of FreeType. All binary and other files of FreeType product will be put into the directory specified by <FREETYPE_INSTALL_DIR>.

```
make install
```

1.1.5.Ftgl 2.1.2

- Enter the directory where source files of Ftgl are located (<FTGL_SRC_DIR>).

```
cd <FTGL_SRC_DIR>/unix
```

- Run configure command.

```
./configure --enable-shared=yes  
--with-freetype-prefix=<FREETYPE_INSTALL_DIR>  
--prefix=<FTGL_INSTALL_DIR>
```

- If configure command is finished successfully, perform building process.

```
make
```

- If building is finished successfully, perform installation of Ftgl. All binary and other files of Ftgl product will be put into the directory specified by <FTGL_INSTALL_DIR>.

```
make install
```

1.2. BUILDING OF OPTIONAL THIRD-PARTY PRODUCTS

1.2.1.gl2ps 1.3.5

Note: installation of gl2ps requires cmake.

- Install cmake product or build it from sources (refer to the cmake documentation for more details: <http://www.cmake.org/cmake/help/install.html>).
- Enter the directory where source files of gl2ps are located (<GL2PS_SRC_DIR>).

```
cd <GL2PS_SRC_DIR>
```

- Start cmake to perform configuration of the source directory.

```
cmake -DCMAKE_INSTALL_PREFIX=<GL2PS_INSTALL_DIR>
      -DCMAKE_BUILD_TYPE=Release
```

- If cmake command is finished successfully, perform building process.

```
make
```

- If building is finished successfully, perform installation of gl2ps. All binary and other files of gl2ps product will be put into the directory specified by <GL2PS_INSTALL_DIR>.

```
make install
```

On some platforms the building of gl2ps can fail. In that case, try to modify the file CMakeList.txt located in the gl2ps source directory as follows:

- Comment line 106:
before: `add_library(lib STATIC gl2ps.c gl2ps.h)`
after: `#add_librrary(lib STATIC gl2ps.c gl2ps.h)`
- Comment line 107:
before: `set_target_properties(lib PROPERTIES OUTPUT_NAME gl2ps)`
after: `#set_target_properties(lib PROPERTIES OUTPUT_NAME gl2ps)`
- Change line 116:
before: `install(TARGETS lib shared DESTINATION lib)`
after: `install(TARGETS shared DESTINATION lib)`

Then, restart installation of the gl2ps from the cmake step.

1.2.2.FreelImage 3.14.1

- Enter the directory where source files of FreelImage are located (<FREEIMAGE_SRC_DIR>).

```
cd <FREEIMAGE_SRC_DIR>
```

- Run building process.

```
make -f Makefile.gnu
```

- Run installation process. All files will be put into the directory specified by <FREEIMAGE_INSTALL_DIR>.

```
make -f Makefile.gnu DESTDIR=<FREEIMAGE_INSTALL_DIR> install
```

- Clean temporary files.

```
make -f Makefile.gnu clean
```

If FreeImage library is built successfully, build also its wrapper for C++ for (FreeImagePlus library).

- Enter the directory where source files of FreeImage are located (<FREEIMAGE_SRC_DIR>).

```
cd <FREEIMAGE_SRC_DIR>
```

- Start building of FreeImagePlus.

```
make -f Makefile.fip
```

- Start installation of FreeImagePlus. All files will be put into the directory specified by <FREEIMAGE_INSTALL_DIR>.

```
make -f Makefile.fip DESTDIR=<FREEIMAGE_INSTALL_DIR> install
```

- Clean temporary files.

```
make -f Makefile.fip clean
```

2. BUILDING OF OPENCASCADE TECHNOLOGY

- Enter the directory where source files of OCCT are located (<OCCT_SRC_DIR>).

```
cd <OCCT_SRC_DIR>
```

- Launch **build_configure** script in order to generate **configure** and **Makefile.in** files before launching configure step.

```
build_configure
```

- Start configuration of OCCT.

```
./configure <FLAGS>
```

Where <FLAGS> is set of the following directives:

--with-tcl=	- define location of libtcl.so
--with-tk=	- define location of libtk.so
--with-gl-include=	- define location of gl.h
--with-gl-library=	- define location of libGL.so
--with-xmu-include=	- define location of Xmu.h
--with-xmu-library=	- define location of libXmu.so
--with-tbb-include=	- define location of tbb.h
--with-tbb-library=	- define location of libtbb.so

<code>--with-freetype=</code>	- define location of installed FreeType product
<code>--with-ftgl=</code>	- define location of installed Ftgl product
<code>--with-gl2ps=</code>	- define location of installed gl2ps product
<code>--with-freeimage=</code>	- define location of installed FreeImage product
<code>--enable-static=yes</code>	- build static library (default = no)
<code>--enable-shared=no</code>	- build shared library (default = yes)
<code>--enable-debug=</code>	- yes: include debug information - no: not include debug information
<code>--enable-production=</code>	- yes: switch on code optimization - no: switch off code optimization
<code>--disable-draw</code>	- build OCCT without Draw Test Harness (not recommended)
<code>--prefix=</code>	- define location for installation of OCCT binaries

For example, in order to build OCCT without optional third-party products, the following options should not be used:

```
--with-tbb-include
--with-tbb-library
--with-gl2ps
--with-freeimage.
```

- Build OCCT libraries.

```
make
```

- 4. Perform installation of OCCT libraries.

```
make install
```