Open CASCADE Technology

BUILDING OCCT WITH CMAKE

User's Guide

CONTENTS

1. INTRODUCTION **2**

2. OPEN CASCADE SOURCES **2**

3. THIRD-PARTY PRODUCTS **2**

3.1. Download precompiled binaries 2

3.2. Build from source package 2

4. THE WOK **3**

4.1. Installation and configuration 3
4.1.1. Windows 3
4.1.2. Linux 3
4.1.3. Mac OS X 4

4.2. How to create a WOK workbench 4

5. CMAKE **5**

5.1. Windows 5

5.2. Linux 6

5.3. Mac OS X 7

6. OCCT PROJECT FILES **8**

6.1. Generation of derived sources and Cmake meta-projects 8

6.2. Cmake meta-projects configuration 8
6.2.1. Set Cmake working directories 8
6.2.2. Configuration of the OCCT project 11
6.2.3. Configuration of the 3rdparty product paths 11
6.2.4. Install path configuration 13

6.3. OCCT project files generation 13
6.3.1. Windows 13
6.3.2. Linux 13
6.3.3. Mac OS X 14

7. OCCT PROJECT BUILDING **15**

7.1. Windows (Visual studio) 15

7.2. Linux (make) 15

7.3. Mac OS X (XCode) 15

8. OCCT PROJECT DEBUGGING FOR VISUAL STUDIO **15**

1. INTRODUCTION

This document provides practical guidelines for Building OCCT with CMake. CMake is free software released under a BSD license. It can create GNU Makefiles, KDevelop project files, XCode project files, and Visual Studio project files.

2. OPEN CASCADE SOURCES

Clone the OCCT Git repository from the server (e.g., **git clone ssh://gitolite@git.dev.opencascade.org/occt occt**) or unpack the source archive[1] (you can find it on the page).

3. THIRD-PARTY PRODUCTS

The OCCT uses the following third-party products:

- Tcl/Tk (used by testing tools)
- Freetype (for text rendering)
- Qt (used by demonstration tools)
- FreeImage (support of common graphic formats) ***[optional]***
- gl2ps (export of OCCT viewer contents to a vector graphic file) ***[optional]***
- TBB (support of multi-threading and parallel execution) ***[optional]***

You can get them in two ways:

3.1. Download precompiled binaries

Download archives with precompiled binaries and source files of third-party products for building the OCCT on the page[2].

3.2. Build from source package

The guides for third-party product compilation are available by links: **Windows**[3], **Linux**[4]

[1] http://www.opencascade.org/getocc/download/loadocc/
[2] http://www.opencascade.org/getocc/download/3rdparty/
[3] http://dev.opencascade.org/sites/default/files/documents/OCCT_Build3rdParty_Windows_V2.pdf
[4] http://dev.opencascade.org/sites/default/files/documents/OCCT_Build3rdParty_Linux_V2.pdf

4. THE WOK

4.1. Installation and configuration

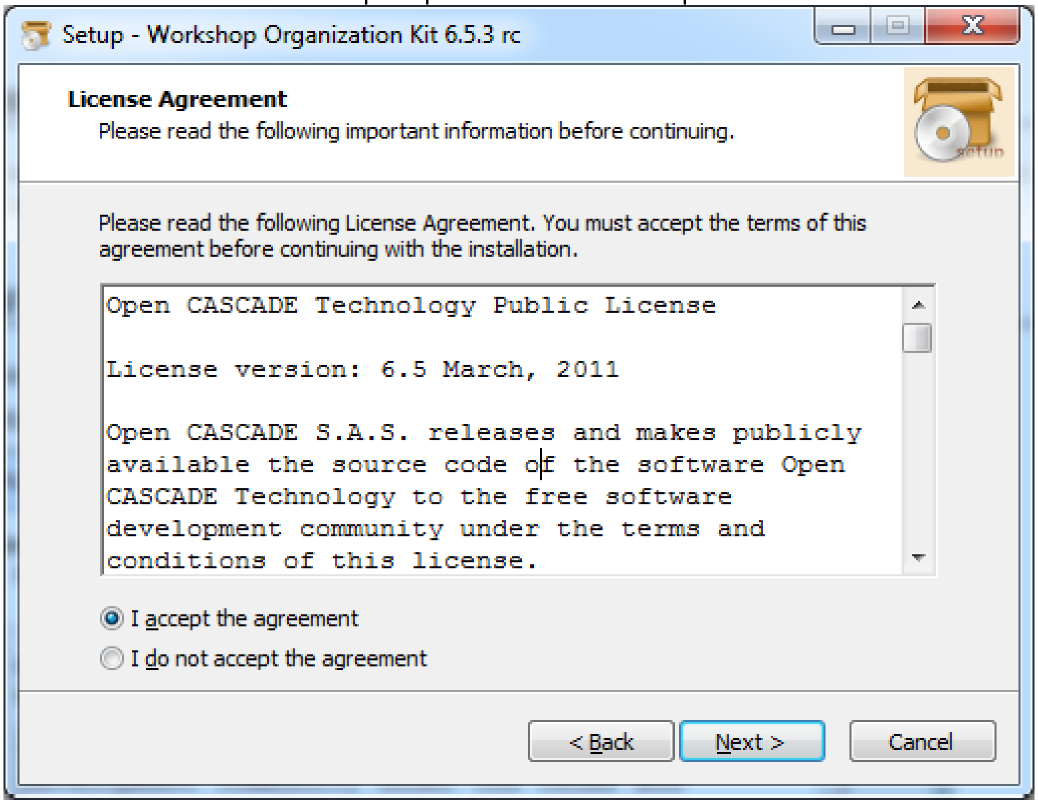
WOK (Workshop Organization Kit) is a Open CASCADE building environment. It is required if you plan to work with OCCT git repository since some source files could be generated only by WOK.

Download the latest version of WOK binary distribution on page http://dev.opencascade.org/index.php?q=home/resources - (Downloads)

WOK 6.6.0 beta5 or higher is required.

4.1.1. Windows

Run the installer. You will be prompted to read and accept the OCCT Public License:



Setup - Workshop Organization Kit 6.5.3 rc

License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

```
Open CASCADE Technology Public License

License version: 6.5 March, 2011

Open CASCADE S.A.S. releases and makes publicly
available the source code of the software Open
CASCADE Technology to the free software
development community under the terms and
conditions of this license.
```

● I accept the agreement
○ I do not accept the agreement

Click “Next” and proceed with the installation.

At the end of the installation specify the version and the location of Visual Studio to be used, and the location of third-party libraries. If you will use only cmake, you do not need to specify anything and can just click “close” button. You can change these settings at any time later. For this click the item “Customize environment (GUI tool)” in the WOK group from Windows Start menu.

The options from this group provide two ways to run WOK:

- In command prompt window (“WOK TCL shell”).
- In Emacs editor (“WOK Emacs”). Using Emacs is convenient if you need to work within WOK environment.

By default WOK installer creates a WOK factory with name “LOC” within workshop “dev” (WOK path :LOC:dev).

4.1.2. Linux

Unpack the .tgz archive containing WOK distributive into the installation directory <WOK_INSTALL_DIR> and perform the following commands:

```
> cd <WOK_INSTALL_DIR>/site
> wok_confgui.sh
```

Define all necessary paths to third-party products in the dialog window if you are going to use other products additionally to cmake. If you will use only cmake – click “close” button.

Perform the following command to initialize WOK:

```
> wok_init.sh
```

WOK can be run in two ways:

- Using WOK TCL shell:

  ```
  > wok_tclsh.sh
  ```

- Using Emacs editor (“WOK Emacs”):

  ```
  > wok_emacs.sh
  ```

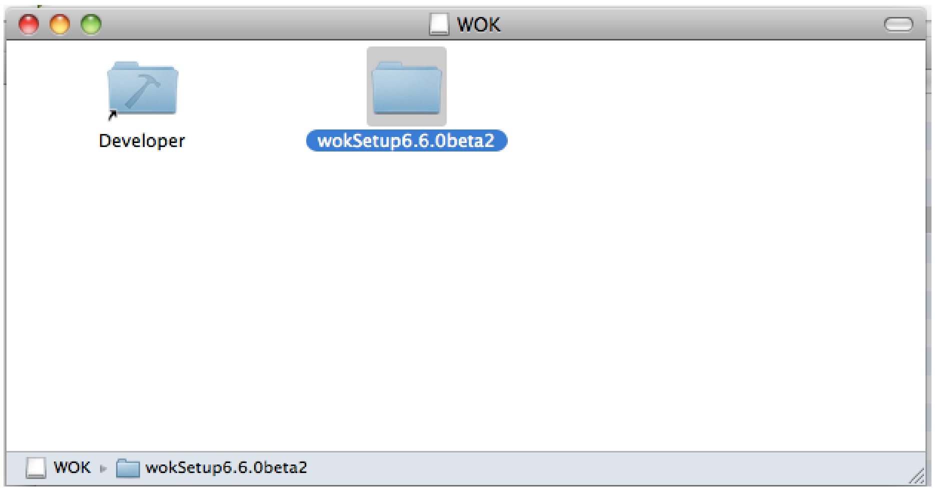
Using Emacs is convenient if you need to work within WOK environment.

4.1.3. Mac OS X

Mount downloaded disk image in DMG format and copy WOK folder in it to desired location <WOK_INSTALL_DIR>. Launch Terminal.app (Applications -> Utilities) and perform initialization of WOK factory:

```
> cd <WOK_INSTALL_DIR>/site
> wok_init.sh
```

Now you may open installed WOK in Finder.app and configure paths to third-party products using site/WokConfig.app if you are going to use other products additionally to cmake. If you will use only cmake – skip this step.



WOK can be launched within Emacs editor by site/WokEmacs.app.

4.2. How to create a WOK workbench

Create a WOK workbench (command `wcreate`) and set its Home to the directory, where the repository is created. The workbench should have the same name as that directory.

E.g., assuming that OCCT repository has been cloned into D:/occt folder:

```
LOC:dev> wcreate occt -DHome=D:/occt
```

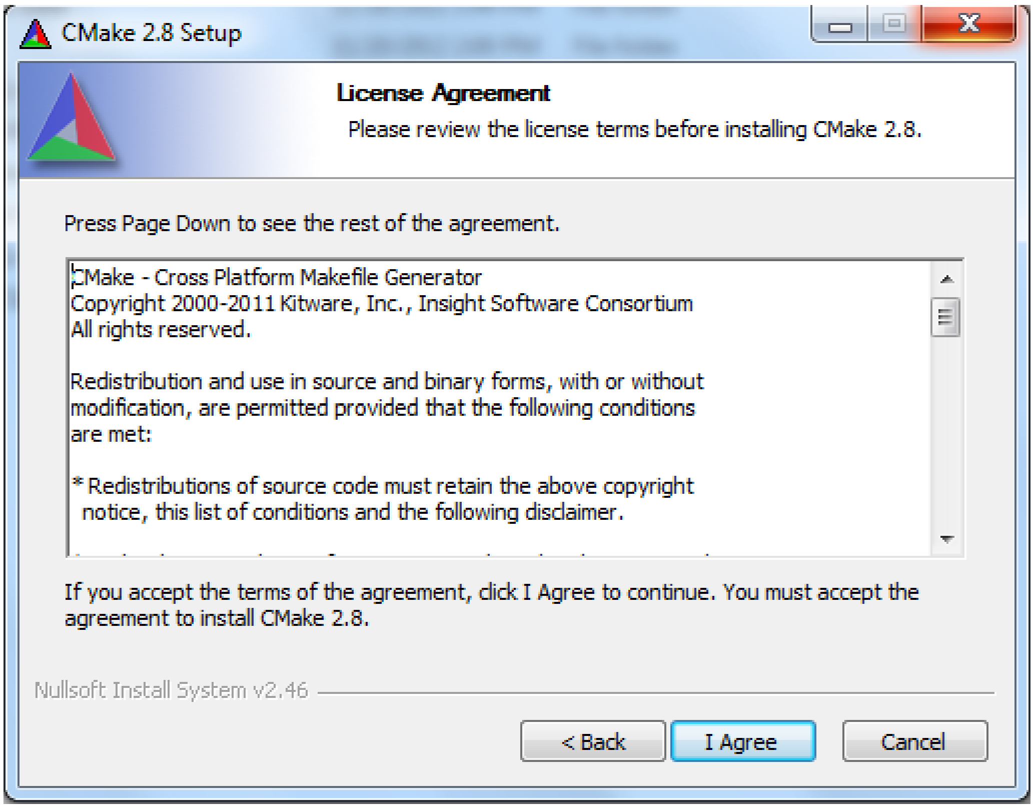
Then you can work with this workbench using normal WOK functionality (wprocess, umake, etc.; see WOK User’s Guide for details) or use it only for generation of derived sources and project files and build with Visual Studio on Windows or `make` command on Linux.

5. CMAKE

Download the latest version[5] (the minimal version is 2.6) of CMake distributive.

5.1. Windows

Run the installer. You will be prompted to read and accept the license terms to proceed:



License Agreement

Please review the license terms before installing CMake 2.8.

Press Page Down to see the rest of the agreement.

CMake - Cross Platform Makefile Generator
Copyright 2000-2011 Kitware, Inc., Insight Software Consortium
All rights reserved.

Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:

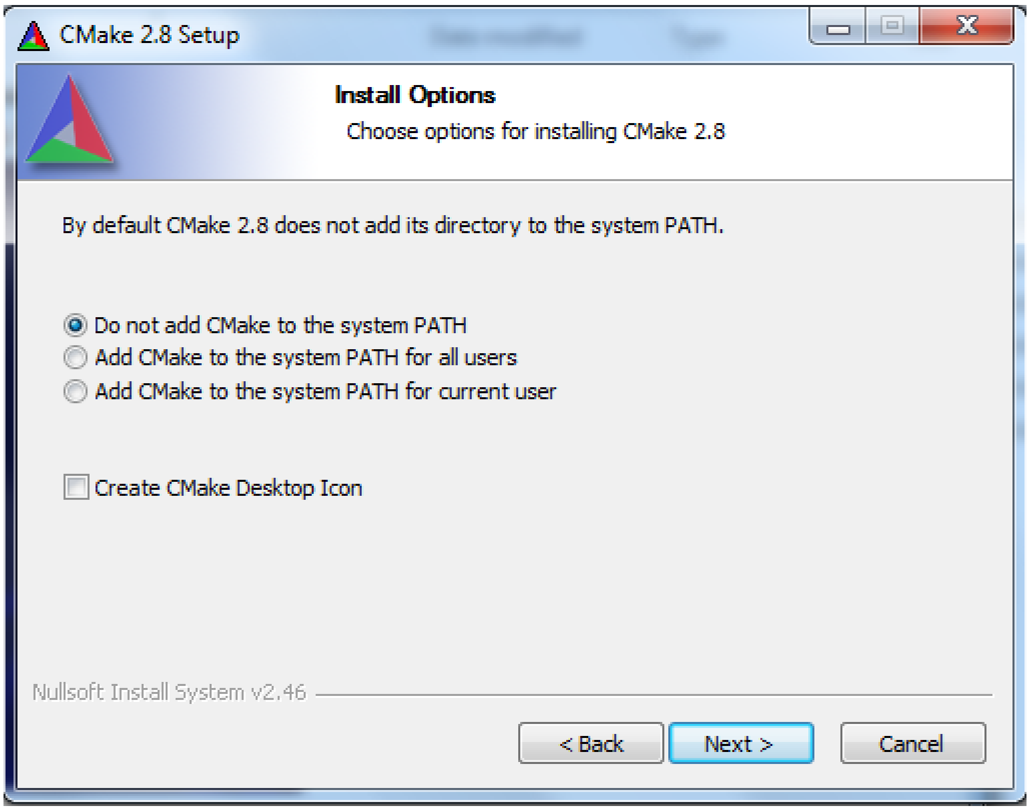
* Redistributions of source code must retain the above copyright
  notice, this list of conditions and the following disclaimer.

If you accept the terms of the agreement, click I Agree to continue. You must accept the agreement to install CMake 2.8.

Nullsoft Install System v2.46

Click “I Agree” and proceed with the installation.

5 http://www.cmake.org/cmake/resources/software.html#latest


CMake 2.8 Setup
Install Options
Choose options for installing CMake 2.8
Nullsoft Install System v2.46
< Back
Next >
Cancel

By default CMake 2.8 does not add its directory to the system PATH.

[✓ Do not add CMake to the system PATH]
Add CMake to the system PATH for all users
Add CMake to the system PATH for current user

☐ Create CMake Desktop Icon

If you do not know whether to add Cmake to the system PATH or not, set “do not add”.

5.2. Linux

Check if CMake exists in the system. If it does not exist, Download[6] precompiled binaries provided for many UNIX platforms. (Installing CMake Guide[7])

There are several possible approaches to building CMake from a source tree:

- If there is no existing CMake installation, a bootstrap script is provided:

```
> ./bootstrap
> make
> make install
```

(Note: the `make` install step is optional, cmake will run from the build directory.)

- An existing CMake installation can be used to build a new version:

```
> cmake .
> make
> make install
```

(Note: the `make` install step is optional, cmake will run from the build directory.)

- On UNIX, if you do not use GNU C++ compiler, you need to indicate which compiler you want to use. This is done by setting the environment variables CC and CXX before running `configure` script. For example on the SGI with the 7.3X compiler, you build like this:

```
(> setenv CXX CC; > setenv CC cc; > ./bootstrap)
```

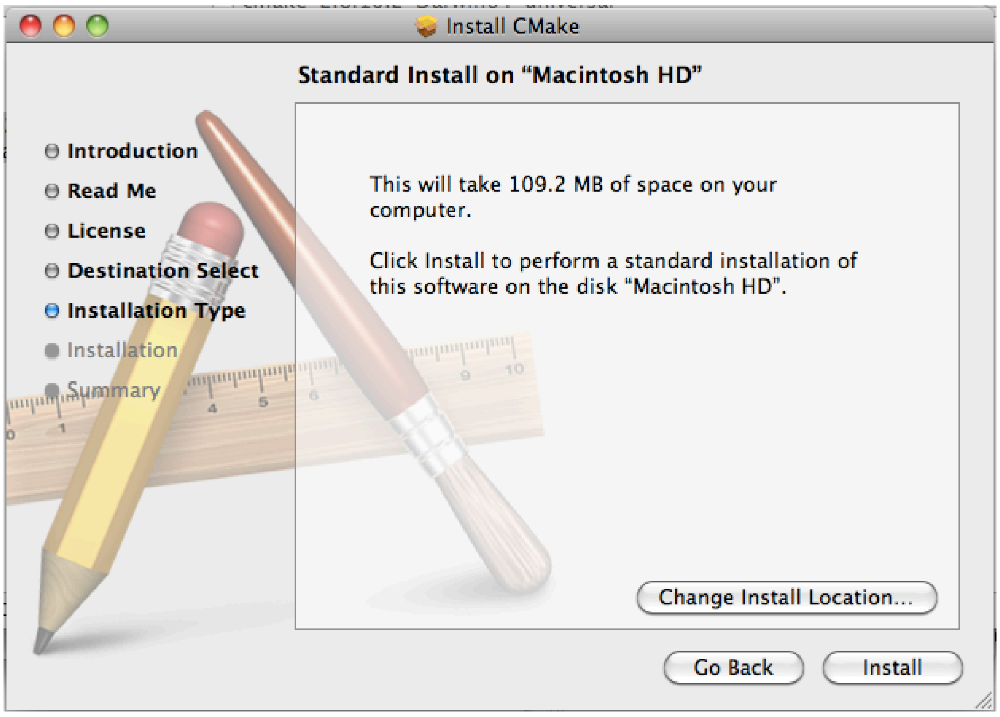
[6] http://www.cmake.org/cmake/resources/software.html
[7] http://www.cmake.org/cmake/help/install.html

```
> make
> make install
```

5.3. Mac OS X

Mount downloaded disk image in DMG format and run the installer with PKG extension. You will be prompted to read and accept the license terms to proceed.

Click “I Agree” and proceed with the installation. Installation may ask you to install symkinks into /usr/bin system folder. If you do not know what it is for set “do not add”.



This will take 109.2 MB of space on your computer.

Click Install to perform a standard installation of this software on the disk "Macintosh HD".

6. OCCT PROJECT FILES

6.1. Generation of derived sources and Cmake meta-projects

In the WOK prompt, in the created workbench, use wgenproj command with **-target=cmake** argument

```
:LOC:dev:occt> wgenproj –target=cmake
```

This command creates the "main" CMakelists.txt meta-project and CMakelists.txt meta-projects for each toolkit within OCCT. CMakelists.txt meta-project of the toolkit is located in ***<OCCT root>*/adm/cmake/<project name>** folder.

If generation of OCCT derived sources is not required, use command `wgenproj` with additional **-no_wprocess** argument:

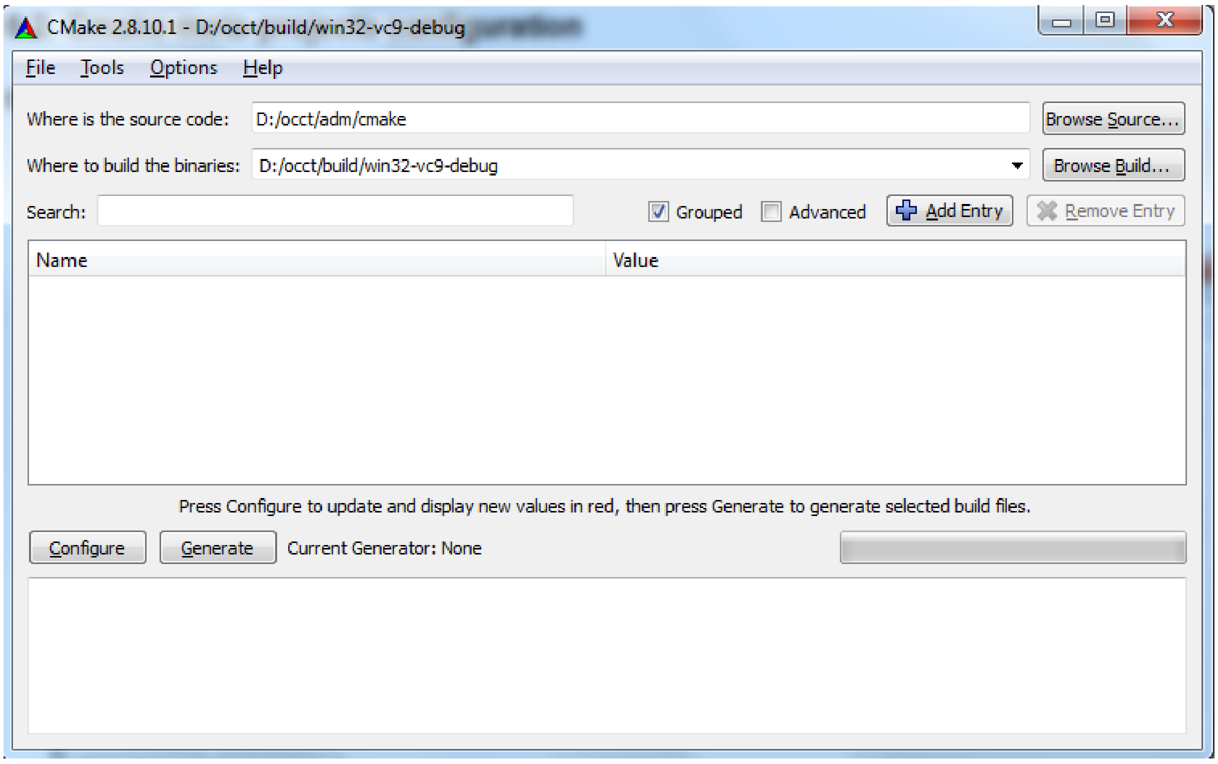
```
:LOC:dev:occt> wgenproj -target=cmake -no_wprocess
```

6.2. Cmake meta-projects configuration

6.2.1. Set Cmake working directories

- **Windows**

Use cmake-gui (Start - Programs – CMake 2.8 – CMake(cmake-gui)) to generate project files for the chosen build environment (e.g., Visual Studio 2008):


CMake 2.8.10.1 - D:/occt/build/win32-vc9-debug

Where is the source code: D:/occt/adm/cmake

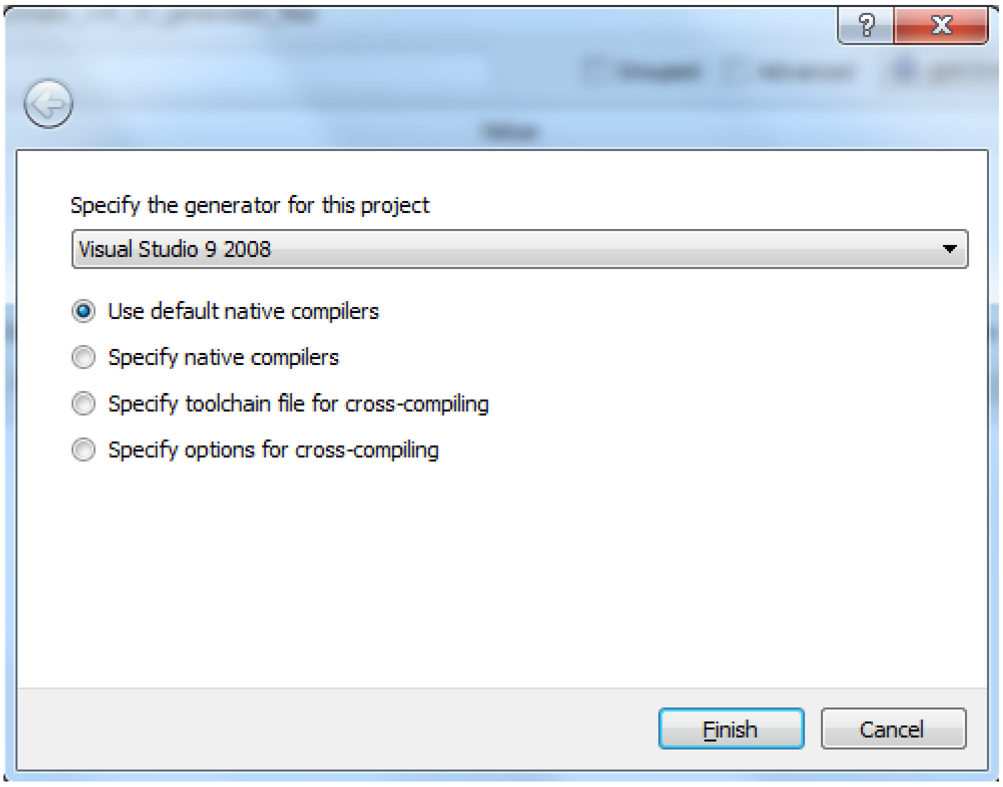
Where to build the binaries: D:/occt/build/win32-vc9-debug

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: None

1. Specify "main" CMakelists.txt meta-project location by clicking **Browse Source** (e.g., **d:/occt/adm/cmake**)
2. Specify location (build folder) for Cmake generated project files by clicking **Browse Build** (e.g., **d:/occt/build/win32-vc9-debug**) (each cmake configuration of the project uses a specific build directory and a specific directory for installed files. It is recommended to compose names of the binary and install directory from system, bitness, compiler and build type.)

3. **Configure** opens the window with a drop-down list of generators supported by CMake project. Select the required generator (e.g., Visual Studio 2008) and click **Finish**)


Specify the generator for this project

Visual Studio 9 2008

[✓ Use default native compilers]

Specify native compilers

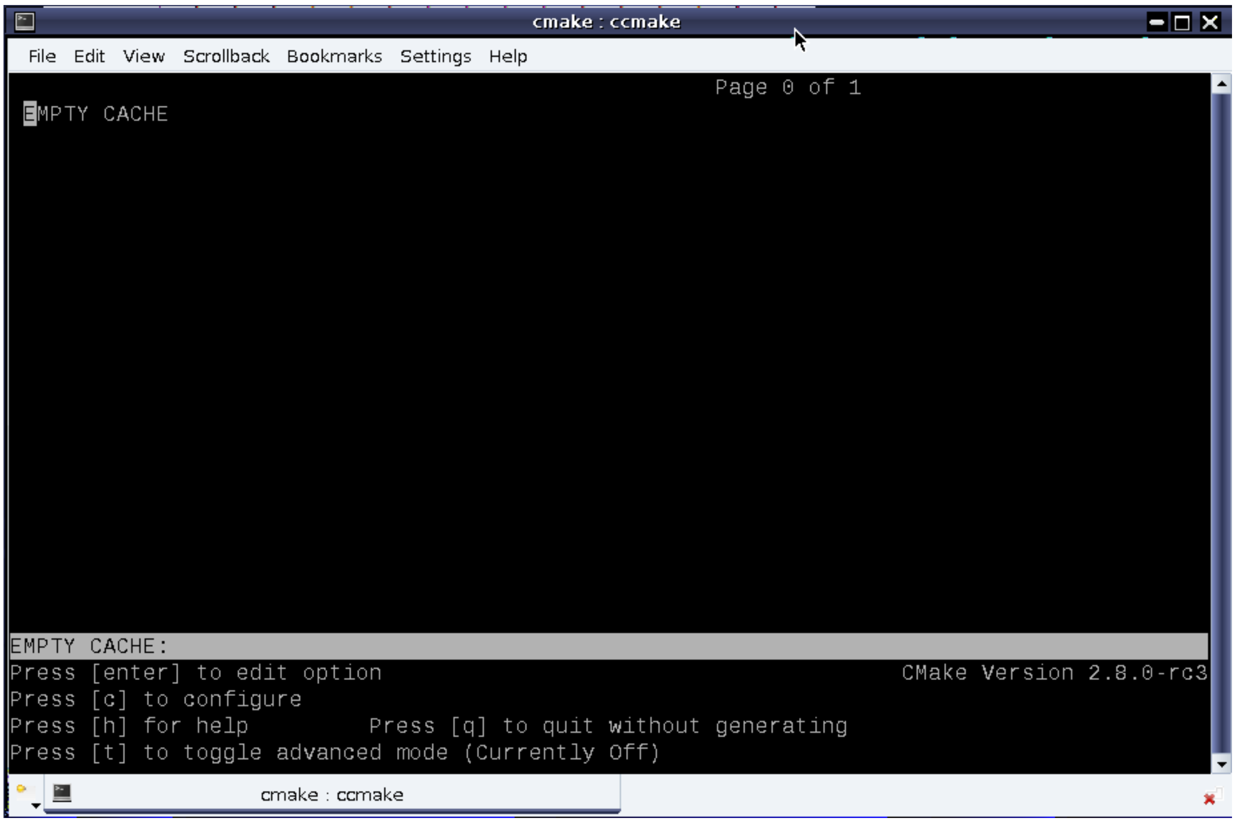
Specify toolchain file for cross-compiling

Specify options for cross-compiling

- **Linux**

In the console, change to the build directory and call ccmake with the path to the source directory of the project:

```
> cd ~/occt/build/debug
> ccmake ~/occt/adm/cmake
```

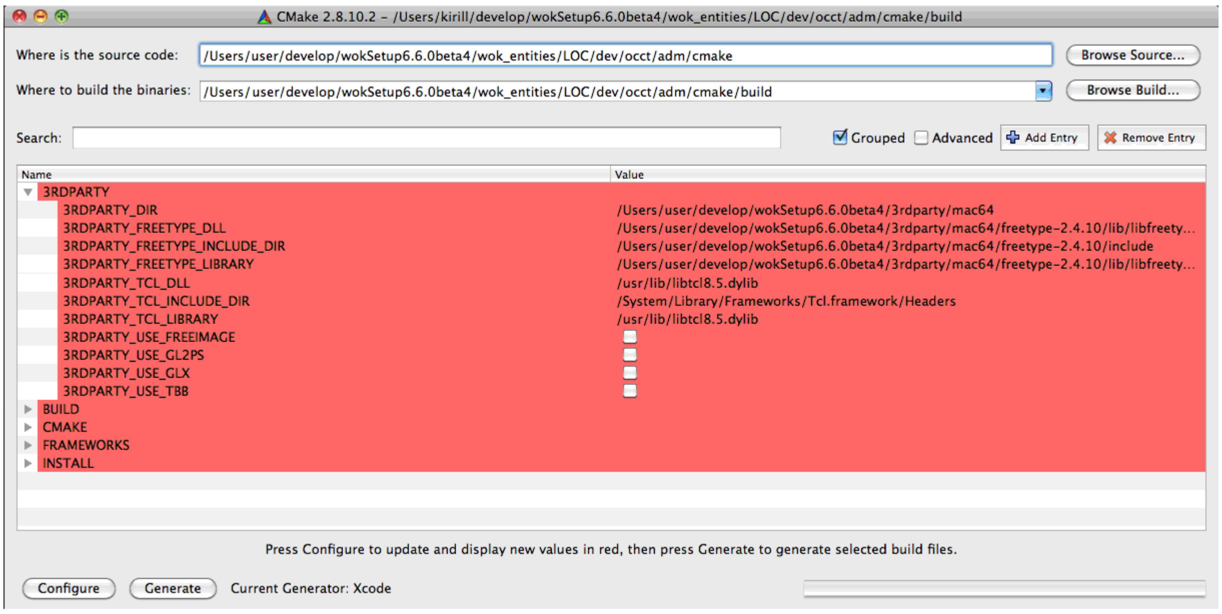


```
                                                          Page 0 of 1
 EMPTY CACHE










EMPTY CACHE:
Press [enter] to edit option                                          CMake Version 2.8.0-rc3
Press [c] to configure
Press [h] for help           Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

Press “c” to configure.

- **Mac OS X**

Use cmake-gui (Applications -> CMake 2.8-10.app) to generate project files for the chosen build environment (e.g., XCode).


CMake 2.8.10.2 – /Users/kirill/develop/wokSetup6.6.0beta4/wok_entities/LOC/dev/occt/adm/cmake/build

Where is the source code: /Users/user/develop/wokSetup6.6.0beta4/wok_entities/LOC/dev/occt/adm/cmake

Where to build the binaries: /Users/user/develop/wokSetup6.6.0beta4/wok_entities/LOC/dev/occt/adm/cmake/build

Search: ✓ Grouped ☐ Advanced

Name	Value
3RDPARTY	
3RDPARTY_DIR	/Users/user/develop/wokSetup6.6.0beta4/3rdparty/mac64
3RDPARTY_FREETYPE_DLL	/Users/user/develop/wokSetup6.6.0beta4/3rdparty/mac64/freetype-2.4.10/lib/libfreety...
3RDPARTY_FREETYPE_INCLUDE_DIR	/Users/user/develop/wokSetup6.6.0beta4/3rdparty/mac64/freetype-2.4.10/include
3RDPARTY_FREETYPE_LIBRARY	/Users/user/develop/wokSetup6.6.0beta4/3rdparty/mac64/freetype-2.4.10/lib/libfreety...
3RDPARTY_TCL_DLL	/usr/lib/libtcl8.5.dylib
3RDPARTY_TCL_INCLUDE_DIR	/System/Library/Frameworks/Tcl.framework/Headers
3RDPARTY_TCL_LIBRARY	/usr/lib/libtcl8.5.dylib
3RDPARTY_USE_FREEIMAGE	☐
3RDPARTY_USE_GL2PS	☐
3RDPARTY_USE_GLX	☐
3RDPARTY_USE_TBB	☐
BUILD	
CMAKE	
FRAMEWORKS	
INSTALL	

Press Configure to update and display new values in red, then press Generate to generate selected build files.

Current Generator: Xcode

6.2.2. Configuration of the OCCT project

The error message, which appears at the end of configuration process, informs you about the required variables, which need to be defined. This error will appear until all required variables are defined correctly.

Note: In cmake-gui there is "grouped" option, which groups variables with a common prefix.

The variables with BUILD_ prefix:

- BUILD_BITNESS – defines bitness of the future project (32 by default)
- BUILD_TYPE – defines build configuration of the future project (Release by default)
- BUILD_<MODULE> - allows including the toolkit set of the specified module to the future project or excluding it from the project.
- BUILD_TOOLKITS – allows including additional specified toolkits (list of items separated by a space or a semicolon) to the common set of the future project.

Check USE_<PRODUCT> variable (USE_FREEIMAGE, USE_GL2PS and USE_TBB) if you want to use this 3rd-party product in the future project.

6.2.3. Configuration of the 3rdparty product paths

If you have 3rd-party libraries in a non-default location (e.g., on Windows, binaries downloaded from OCCT site[8] into a certain folder), specify 3RDPARTY_DIR variable that points to the folders of 3rdparty products (some or all). At the next configuration 3rd-party product paths stored in 3RDPARTY_<PRODUCT>_DIR variable will be searched for in 3RDPARTY_DIR directory. If the structure of 3RDPARTY_DIR directory is the same as adopted in the OCCT, the directory will contain product dir, lib and header files.

Press "configure" ("c" key for ccmake)

Important: The names of searched libraries and header files are hardcoded.

The result of the 3rdparty product search will be recorded in the corresponding variables:

- 3RDPARTY_<PRODUCT>_DIR – path to the product directory (with directory name) (e.g., D:/3rdparty/Tcl-8.5.12.0-32)

- 3RDPARTY_<PRODUCT>_LIBRARY – path to the .lib libraries (with the library name) (e.g., D:/3rdparty/Tcl-8.5.12.0-32/lib/tcl85.lib). In non-windows case, this variable is the same as 3RDPARTY_<PRODUCT>_DLL.

- 3RDPARTY_<PRODUCT>_INCLUDE – path to the include directory that contains the required header file (with "include" name) (e.g., D:/3rdparty/Tcl-8.5.12.0-32/include including tcl.h)

- 3RDPARTY_<PRODUCT>_DLL – path to the .dll/.so/.dylib library (with the library name) (e.g., D:/3rdparty/Tcl-8.5.12.0-32/bin/tcl85.dll)

[8] http://www.opencascade.org/getocc/download/3rdparty/

The search process is as follows:

```
1 level:. 3RDPARTY_DIR
        2 level: 3RDPARTY_<PRODUCT>_DIR
                3 level: 3RDPARTY_<PRODUCT>_LIBRARY
                3 level: 3RDPARTY_<PRODUCT>_INCLUDE
                3 level: 3RDPARTY_<PRODUCT>_DLL
```

If a variable of any level is not defined (empty or <variable name>-NOTFOUND) and the upper level variable is defined, the content of the non-defined variable will be searched for at the next configuration step. <u>If search process in level 3 doesn't find the required files, it searches in default places also</u>.

Note: Freetype search process tries to find ft2build.h file into 3RDPARTY_FREETYPE INCLUDE dir and after that adds "3RDPARTY_FREETYPE_INCLUDE /freetype2" path to common includes if it exists.

Important: If BUILD_TYPE or BITNESS variable is changed – at the next configuration 3RDPARTY_ variables will be replaced by the search process result, except for the 3RDPARTY_DIR variable.

Note: CMake will produce an error after the configuration step until all required variables are defined correctly.

If the search result (include path, or library path, or dll path) does not meet your expectations - you can change 3RDPARTY_<PRODUCT>_DIR variable, clear (if they are not empty) 3RDPARTY_<PRODUCT>_DLL, 3RDPARTY_<PRODUCT>_INCLUDE_DIR and 3RDPARTY_<PRODUCT>_LIBRARY variables (or clear one of them) and run the configuration process again. At this time the search will be performed in the new identified directory and the result will be recorded to empty variables (non-empty variables will not be replaced).

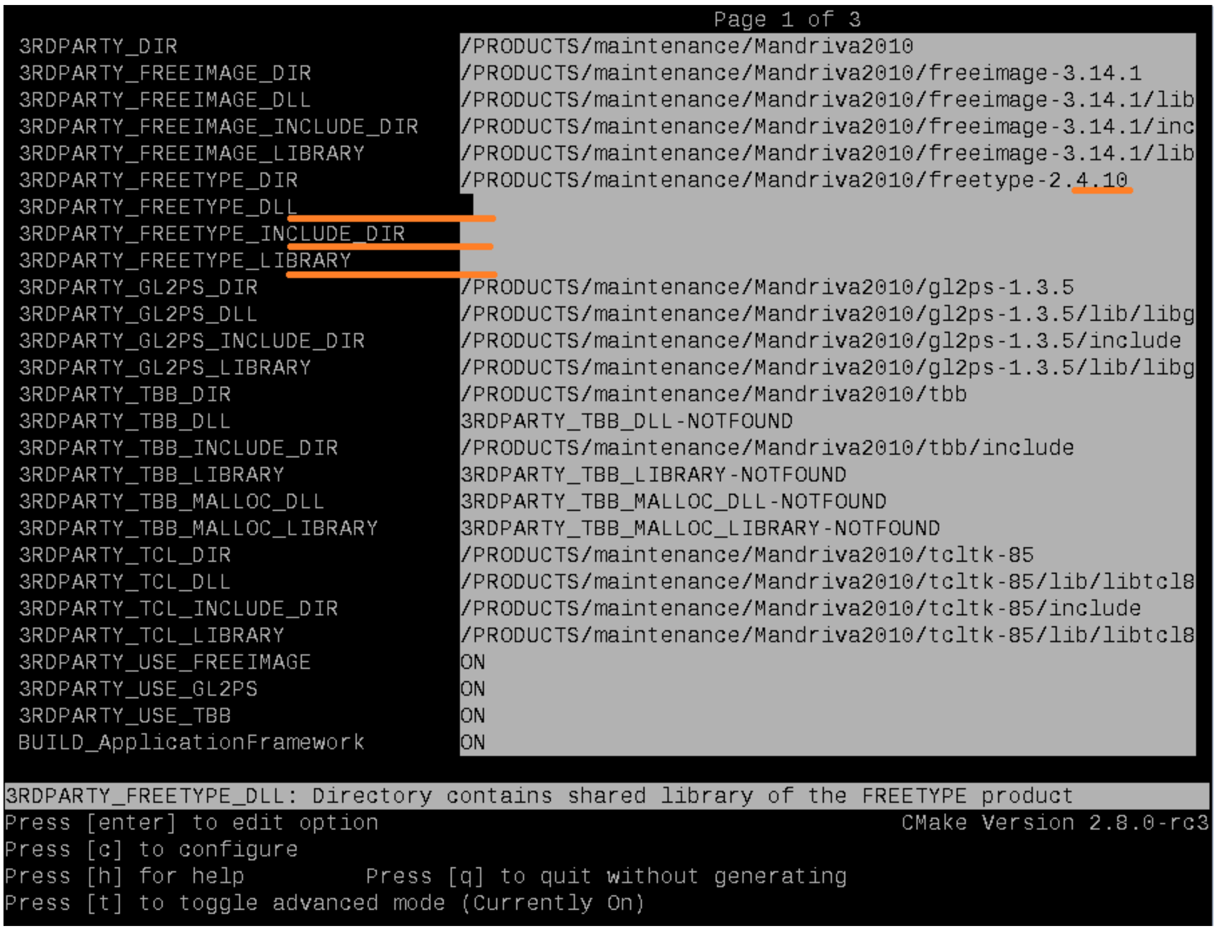
For example, (Linux case) 3RDPARTY_FREETYPE_DIR variable

```
/PRODUCTS/maintenance/Mandriva2010/freetype-2.3.7
```

can be changed to

```
/PRODUCTS/maintenance/Mandriva2010/freetype-2.4.10
```

and the related variables: 3RDPARTY_FREETYPE_DLL, 3RDPARTY_FREETYPE_INCLUDE_DIR and 3RDPARTY_FREETYPE_LIBRARY will be cleared.


```
                                                     Page 1 of 3
 3RDPARTY_DIR                     /PRODUCTS/maintenance/Mandriva2010
 3RDPARTY_FREEIMAGE_DIR           /PRODUCTS/maintenance/Mandriva2010/freeimage-3.14.1
 3RDPARTY_FREEIMAGE_DLL           /PRODUCTS/maintenance/Mandriva2010/freeimage-3.14.1/lib
 3RDPARTY_FREEIMAGE_INCLUDE_DIR   /PRODUCTS/maintenance/Mandriva2010/freeimage-3.14.1/inc
 3RDPARTY_FREEIMAGE_LIBRARY       /PRODUCTS/maintenance/Mandriva2010/freeimage-3.14.1/lib
 3RDPARTY_FREETYPE_DIR            /PRODUCTS/maintenance/Mandriva2010/freetype-2.4.10
 3RDPARTY_FREETYPE_DLL
 3RDPARTY_FREETYPE_INCLUDE_DIR
 3RDPARTY_FREETYPE_LIBRARY
 3RDPARTY_GL2PS_DIR               /PRODUCTS/maintenance/Mandriva2010/gl2ps-1.3.5
 3RDPARTY_GL2PS_DLL               /PRODUCTS/maintenance/Mandriva2010/gl2ps-1.3.5/lib/libg
 3RDPARTY_GL2PS_INCLUDE_DIR       /PRODUCTS/maintenance/Mandriva2010/gl2ps-1.3.5/include
 3RDPARTY_GL2PS_LIBRARY           /PRODUCTS/maintenance/Mandriva2010/gl2ps-1.3.5/lib/libg
 3RDPARTY_TBB_DIR                 /PRODUCTS/maintenance/Mandriva2010/tbb
 3RDPARTY_TBB_DLL                 3RDPARTY_TBB_DLL-NOTFOUND
 3RDPARTY_TBB_INCLUDE_DIR         /PRODUCTS/maintenance/Mandriva2010/tbb/include
 3RDPARTY_TBB_LIBRARY             3RDPARTY_TBB_LIBRARY-NOTFOUND
 3RDPARTY_TBB_MALLOC_DLL          3RDPARTY_TBB_MALLOC_DLL-NOTFOUND
 3RDPARTY_TBB_MALLOC_LIBRARY      3RDPARTY_TBB_MALLOC_LIBRARY-NOTFOUND
 3RDPARTY_TCL_DIR                 /PRODUCTS/maintenance/Mandriva2010/tcltk-85
 3RDPARTY_TCL_DLL                 /PRODUCTS/maintenance/Mandriva2010/tcltk-85/lib/libtcl8
 3RDPARTY_TCL_INCLUDE_DIR         /PRODUCTS/maintenance/Mandriva2010/tcltk-85/include
 3RDPARTY_TCL_LIBRARY             /PRODUCTS/maintenance/Mandriva2010/tcltk-85/lib/libtcl8
 3RDPARTY_USE_FREEIMAGE           ON
 3RDPARTY_USE_GL2PS               ON
 3RDPARTY_USE_TBB                 ON
 BUILD_ApplicationFramework       ON

3RDPARTY_FREETYPE_DLL: Directory contains shared library of the FREETYPE product
Press [enter] to edit option                                     CMake Version 2.8.0-rc3
Press [c] to configure
Press [h] for help          Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently On)
```

During configuration process the cleaned variables will be filled with new found values.

6.2.4. Install path configuration

Define in INSTALL_DIR variable the path to the installed OCCT files (libraries, executables and headers).

If INSTALL_<PRODUCT> variable is checked – 3rd-party products will be copied to the install directory.

At the end of the configuration process “configuring done” message will be shown and the generation process can be started.

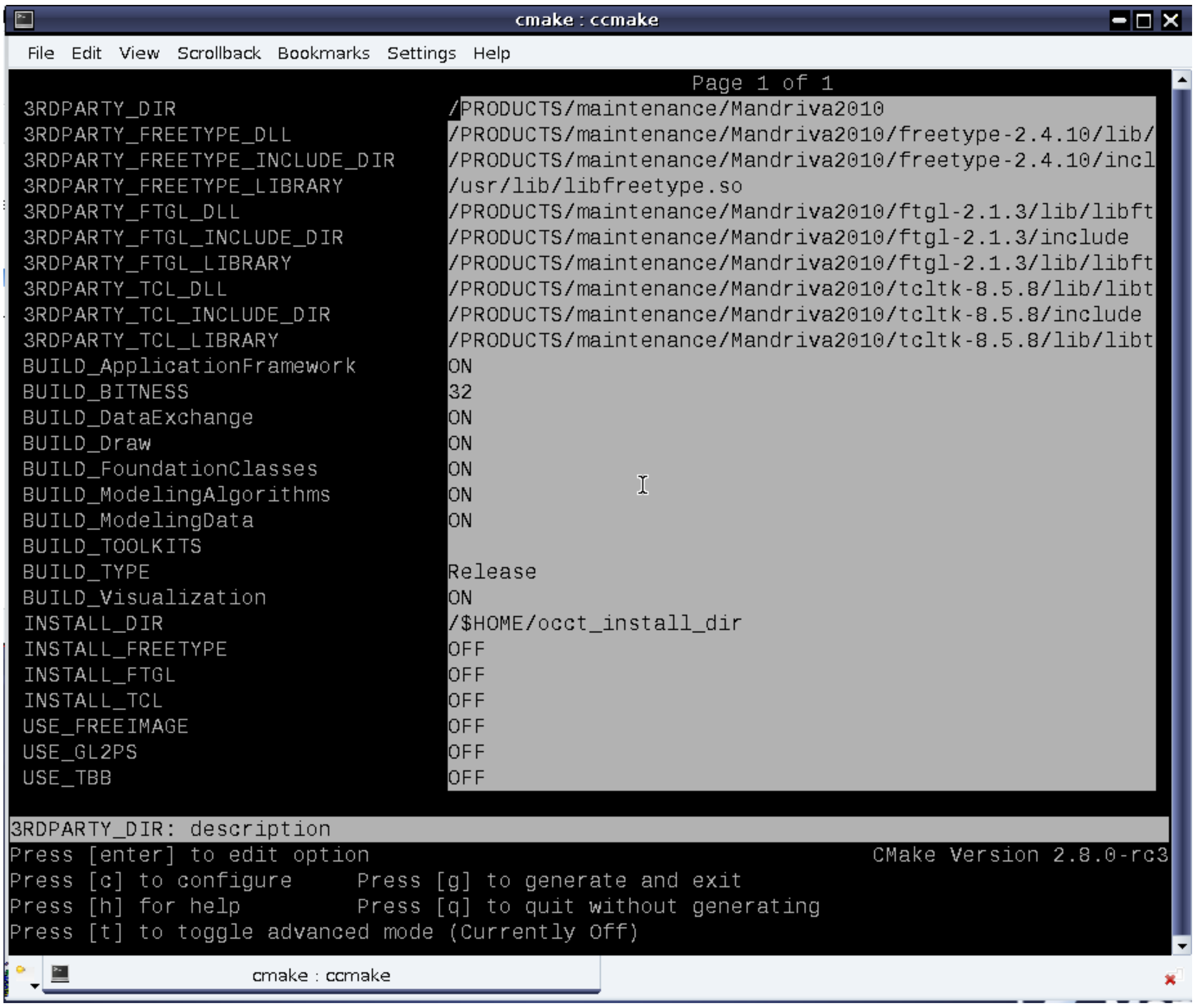
6.3. OCCT project files generation

6.3.1. Windows

Click **Generate** button and wait until the generation process is finished. Then the project files will appear in the build folder (e.g., **d:/occt/build/win32-vc9-release***)*.

6.3.2. Linux

When the configuration is complete, start the generation process by pressing “g”.



```
                                                  Page 1 of 1
 3RDPARTY_DIR                     /PRODUCTS/maintenance/Mandriva2010
 3RDPARTY_FREETYPE_DLL            /PRODUCTS/maintenance/Mandriva2010/freetype-2.4.10/lib/
 3RDPARTY_FREETYPE_INCLUDE_DIR    /PRODUCTS/maintenance/Mandriva2010/freetype-2.4.10/incl
 3RDPARTY_FREETYPE_LIBRARY        /usr/lib/libfreetype.so
 3RDPARTY_FTGL_DLL                /PRODUCTS/maintenance/Mandriva2010/ftgl-2.1.3/lib/libft
 3RDPARTY_FTGL_INCLUDE_DIR        /PRODUCTS/maintenance/Mandriva2010/ftgl-2.1.3/include
 3RDPARTY_FTGL_LIBRARY            /PRODUCTS/maintenance/Mandriva2010/ftgl-2.1.3/lib/libft
 3RDPARTY_TCL_DLL                 /PRODUCTS/maintenance/Mandriva2010/tcltk-8.5.8/lib/libt
 3RDPARTY_TCL_INCLUDE_DIR         /PRODUCTS/maintenance/Mandriva2010/tcltk-8.5.8/include
 3RDPARTY_TCL_LIBRARY             /PRODUCTS/maintenance/Mandriva2010/tcltk-8.5.8/lib/libt
 BUILD_ApplicationFramework       ON
 BUILD_BITNESS                    32
 BUILD_DataExchange               ON
 BUILD_Draw                       ON
 BUILD_FoundationClasses          ON
 BUILD_ModelingAlgorithms         ON
 BUILD_ModelingData               ON
 BUILD_TOOLKITS
 BUILD_TYPE                       Release
 BUILD_Visualization              ON
 INSTALL_DIR                      /$HOME/occt_install_dir
 INSTALL_FREETYPE                 OFF
 INSTALL_FTGL                     OFF
 INSTALL_TCL                      OFF
 USE_FREEIMAGE                    OFF
 USE_GL2PS                        OFF
 USE_TBB                          OFF

3RDPARTY_DIR: description
Press [enter] to edit option                                     CMake Version 2.8.0-rc3
Press [c] to configure     Press [g] to generate and exit
Press [h] for help         Press [q] to quit without generating
Press [t] to toggle advanced mode (Currently Off)
```

6.3.3. Mac OS X

Click **Generate** button and wait until the generation process is finished. Then the project files will appear in the build folder (e.g., **/Developer/occt/build/XCode**).

7. OCCT PROJECT BUILDING

The install folder contains bin, inc, lib and res folders and a script to run DRAWEXE (draw.bat or draw.sh).

- “bin” contains executables, DLL (Windows) style shared libraries and pdb-files in OCCT debug version,.
- “lib” contains the import parts of DLL libraries.
- “inc” contains header files.
- “res” contains all required source files for OCCT.

7.1. Windows (Visual studio)

Go to the build folder, start the Visual Studio solution (OCCT.sln) and build it by clicking Build - Build Solution.

When the building process finished, build the INSTALL project (by default the build solution process skips the building of the INSTALL project) to move the above files to INSTALL_DIR. For this in the solution explorer right click on the INSTALL project and select Project Only - Build Only INSTALL.

7.2. Linux (make)

Change directory to binary dir and run `make` command

```
> make
```

To copy all libraries, executables and chosen 3rd-party libraries run “make” command with “install” argument

```
> make install
```

This command will move the above files to INSTALL_DIR.

7.3. Mac OS X (XCode)

Go to the build folder, start the XCode solution (OCCT.xcodeproj) and build it by clicking Build -> Build. Please notice that XCode may have worst responsibility to user actions due to sources processing at first start.

When the building process finished, build the INSTALL project (by default the build solution process skips the building of the INSTALL project) to move the above files to INSTALL_DIR. Notice that env.sh (configure PATH and DYLD_LIBRARY_PATH environment variables as well as Draw Harness extra variables) and draw.sh (to launch DRAWEXE) will be created in target directory.

8. OCCT PROJECT DEBUGGING FOR VISUAL STUDIO

Run OCCT.bat from the build directory to start Visual Studio with required environment for debugging.