Open CASCADE Technology

Building OCCT with WOK

CONTENTS

1. INTRODUCTION

This document presents a short guide for installing binary WOK package and for building OCCT from sources. Note that WOK is required to generate HXX and CXX files for OCCT classes defined using CDL language, and to project files.

For the time being Binary WOK Package is available in the form of Install Wizard on Windows platform only.

2. PRE-REQUISITES

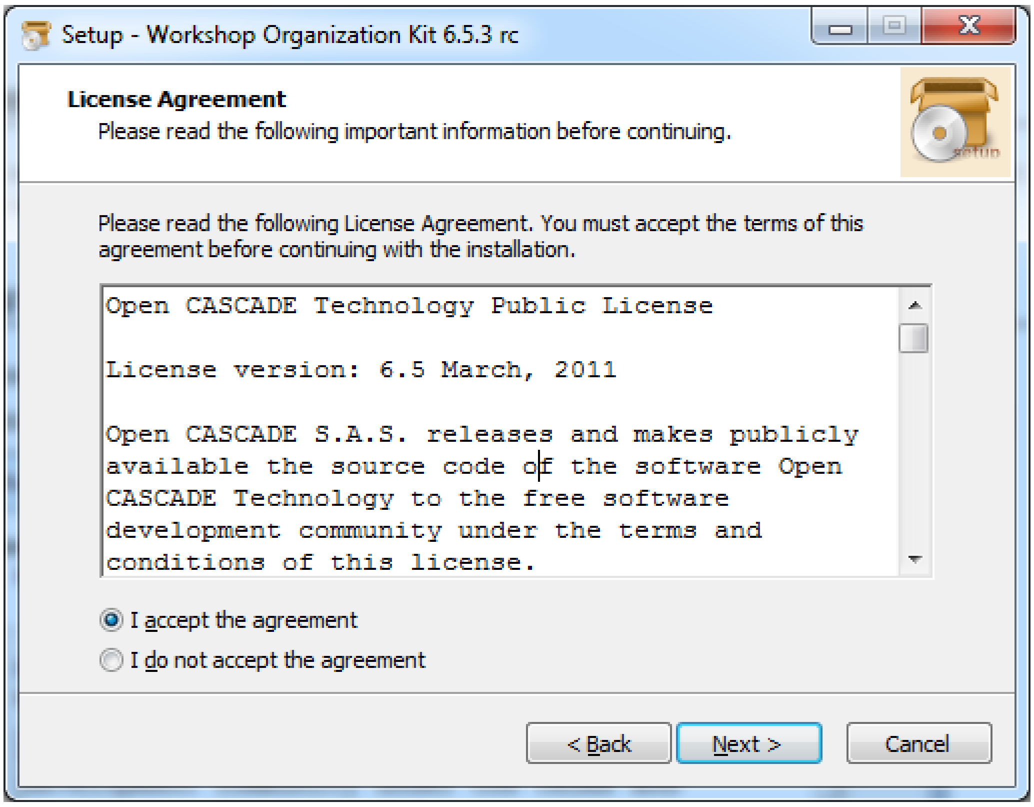
Before installing and using WOK, please make sure that you have installed a compiler (it is assumed that it is Visual Studio on Windows or gcc on Linux) and third-party components required for building OCCT.

3. INSTALL BINARY WOK PACKAGE

Download the latest version of WOK from http://dev.opencascade.org/index.php?q=home/resources.

3.1. Windows

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


Setup - Workshop Organization Kit 6.5.3 rc

License Agreement
Please read the following important information before continuing.

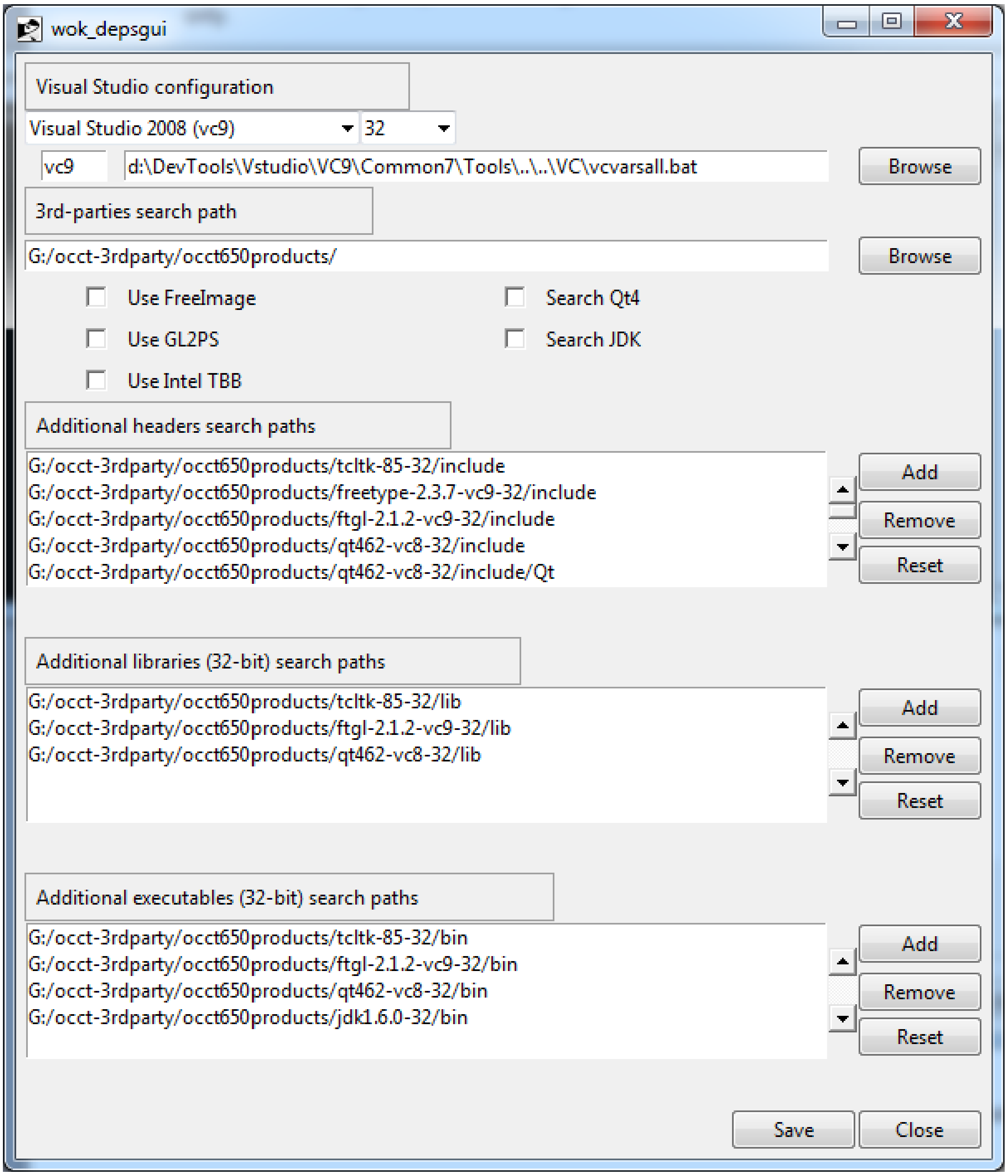
Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

```
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 you will be prompted to specify the version and the location of Visual Studio to be used, and the location of third-party libraries:



Visual Studio configuration

Visual Studio 2008 (vc9) 32

vc9 d:\DevTools\Vstudio\VC9\Common7\Tools\..\..\VC\vcvarsall.bat

3rd-parties search path

G:/occt-3rdparty/occt650products/

☐ Use FreeImage
☐ Search Qt4
☐ Use GL2PS
☐ Search JDK
☐ Use Intel TBB

Additional headers search paths

G:/occt-3rdparty/occt650products/tcltk-85-32/include
G:/occt-3rdparty/occt650products/freetype-2.3.7-vc9-32/include
G:/occt-3rdparty/occt650products/ftgl-2.1.2-vc9-32/include
G:/occt-3rdparty/occt650products/qt462-vc8-32/include
G:/occt-3rdparty/occt650products/qt462-vc8-32/include/Qt

Additional libraries (32-bit) search paths

G:/occt-3rdparty/occt650products/tcltk-85-32/lib
G:/occt-3rdparty/occt650products/ftgl-2.1.2-vc9-32/lib
G:/occt-3rdparty/occt650products/qt462-vc8-32/lib

Additional executables (32-bit) search paths

G:/occt-3rdparty/occt650products/tcltk-85-32/bin
G:/occt-3rdparty/occt650products/ftgl-2.1.2-vc9-32/bin
G:/occt-3rdparty/occt650products/qt462-vc8-32/bin
G:/occt-3rdparty/occt650products/jdk1.6.0-32/bin

You can change these settings at any time later. For this click on the item “Customize environment (GUI tool)” in the WOK group within the Windows Start menu.

The shortcuts 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.

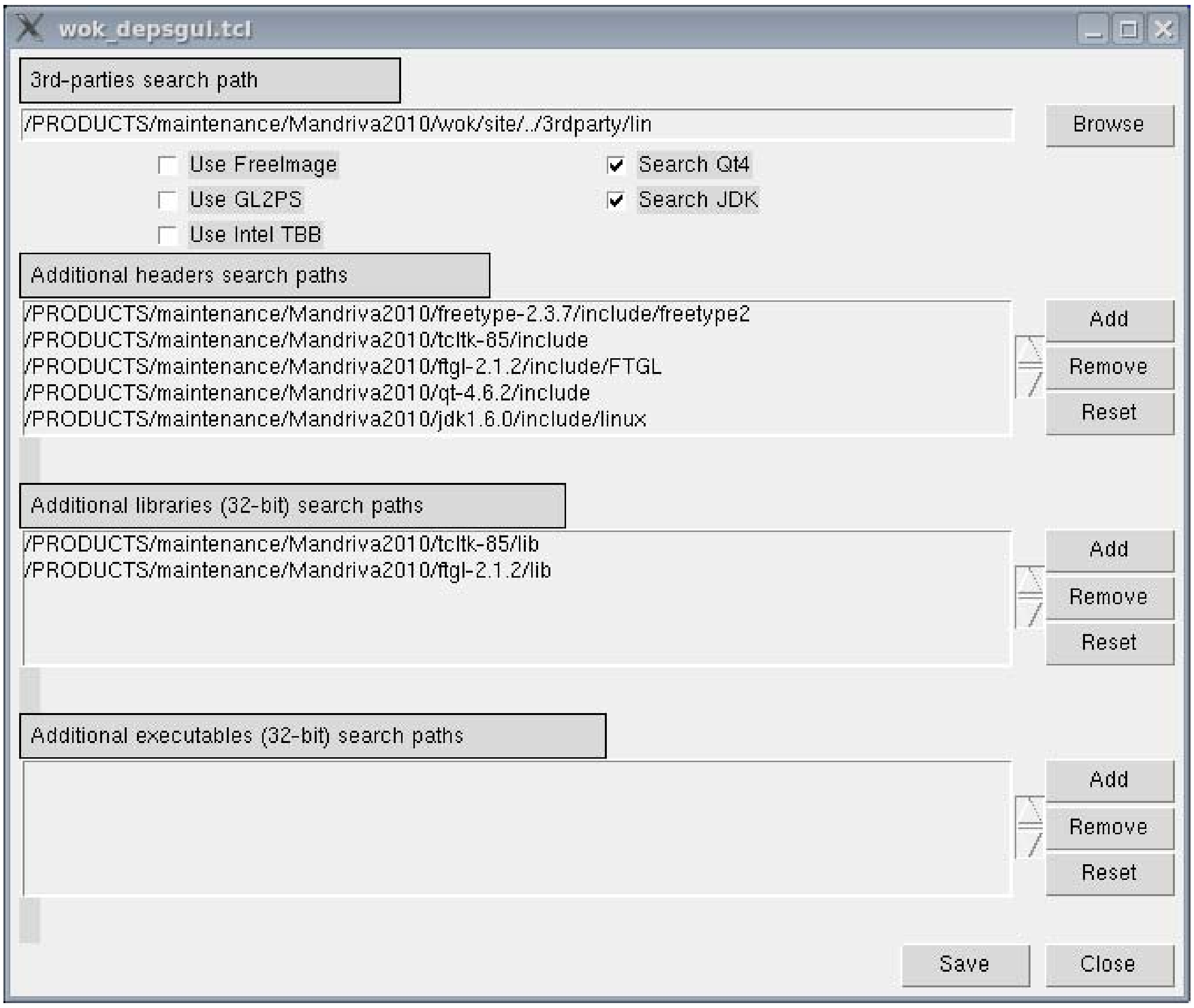
By default WOK installer creates a WOK factory with name “LOC” within workshop “dev” (WOK path :LOC:dev).

3.2. Linux

1. Unpack the .tgz archive containing WOK distributive into an installation directory <WOK_INSTALL_DIR>.
2. Perform the following commands assuming that you have unpacked WOK distributive archive into <WOK_INSTALL_DIR>:

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

   Define all necessary paths to third-party products in the dialog window:



wok_depsgui.tcl

3rd-parties search path

/PRODUCTS/maintenance/Mandriva2010/wok/site/../3rdparty/lin

- [ ] Use FreeImage
- [ ] Use GL2PS
- [ ] Use Intel TBB
- [x] Search Qt4
- [x] Search JDK

Additional headers search paths

```
/PRODUCTS/maintenance/Mandriva2010/freetype-2.3.7/include/freetype2
/PRODUCTS/maintenance/Mandriva2010/tcltk-85/include
/PRODUCTS/maintenance/Mandriva2010/ftgl-2.1.2/include/FTGL
/PRODUCTS/maintenance/Mandriva2010/qt-4.6.2/include
/PRODUCTS/maintenance/Mandriva2010/jdk1.6.0/include/linux
```

Additional libraries (32-bit) search paths

```
/PRODUCTS/maintenance/Mandriva2010/tcltk-85/lib
/PRODUCTS/maintenance/Mandriva2010/ftgl-2.1.2/lib
```

Additional executables (32-bit) search paths

3. Run the following commands to create WOK LOC factory:

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

4. Your installation procedure is over. To run WOK please use one the following commands:

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

   or

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

4. INITIALIZE A WORKBENCH

To start working with OCCT, clone the OCCT Git repository from the server (see http://dev.opencascade.org/index.php?q=home/resources for details) or unpack the source archive.

Then create a WOK workbench (command **wcreate**) setting its Home to the directory, where the repository is created. The workbench should have the same name as that directory.

For example, 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, as described below.

5. BUILD A WORKBENCH USING STANDARD TOOLS

Use command **wgenproj** in WOK to generate derived headers and source files with help of Visual Studio project files on Windows or automake (.am) files on Linux, as follows:

```
LOC:dev> wokcd occt
LOC:dev:occt> wgenproj
```

Note that this command takes several minutes to complete at the first call.

Re-execute this step if some CDL files in OCCT have been modified (either by you directly, or due to updates in the repository) Close Visual Studio before this operation. Note that WOK may fail to update correctly; in such case remove sub-directories drv and .adm and repeat the command.

5.1. Compilation with Visual Studio

When the projects are generated, start Visual Studio by script "`msvc.bat`" generated in the workbench home directory:

```
> D:/occt/msvc.bat
```

or directly from WOK prompt (note that WOK sets the current directory to the workbench home when switching to the workbench):

```
LOC:dev:occt> msvc.bat
```

Note that object files and binaries are built by VS projects and by WOK in different locations, hence binaries and executable environments are different!

5.2. Compilation with make

When the projects are generated, execute "build_configure", "configure" generated in the workbench home directory and "make install" command after that:

- ./build_configure
- ./configure <FLAGS>

  Where <FLAGS> is a set of the following directives:

--with-tcl=	- define location of tclConfig.sh
--with-tk=	- define location of tkConfig.sh
--with-freetype=	- define location of installed FreeType product
--with-ftgl=	- define location of installed Ftgl product
--with-gl2ps=	- define location of installed gl2ps product
--with-freeimage=	- define location of installed FreeImage product

--with-tbb-include=	- define location of tbb.h
--with-tbb-library=	- define location of libtbb.so
--enable-debug=	- yes: include debug information
	- no: not include debug information
--enable-production=	- yes: switch on code optimization
	- no: switch off code optimization
--prefix=	- define location for installation of OCCT binaries

For example:

```
./configure –prefix=/PRODUCTS/occt-6.5.3 --with-tcl=/PRODUCTS/tcltk-8.5.8/lib --with-
tk=/PRODUCTS/tcltk-8.5.8/lib --with-freetype=/PRODUCTS/freetype-2.3.7 --with-
ftgl=/PRODUCTS/ftgl-2.1.2 --with-gl2ps=/PRODUCTS/gl2ps-1.3.5 --with-
freeimage=/PRODUCTS/freeimage-3.14.1 --with-tbb-
include=/PRODUCTS/tbb30_018oss/include --with-tbb-
library=/PRODUCTS/tbb30_018oss/lib/ia32/cc4.1.0_libc2.4_kernel2.6.16.21
```

- make –j8 install

6. LAUNCH DRAW TEST HARNESS

Launch batch file **draw.bat** on Windows or **draw.sh** on Linux generated by command wgenproj to start DRAW:

```
D:\occt> draw.bat
Hint: use "pload ALL" command to load standard commands
Draw[1]> pload ALL
Draw[2]>
```