



# BINARY WOK PACKAGE

# White Paper

<b>1. CONFIGURING THE ENVIRONMENT .....</b>	<b>3</b>
1.1. BINARY WOK PACKAGE .....	3
1.2. INSTALLATION OF BINARY WOK PACKAGE .....	3
1.3. CONFIGURATION AND LAUNCH .....	3
<b>2. INITIALIZING THE WORKBENCH .....</b>	<b>4</b>
<b>3. USING VISUAL STUDIO FOR BUILDING WORKBENCH.....</b>	<b>5</b>
<b>4. LAUNCHING DRAW TEST HARNESS .....</b>	<b>6</b>

Copyright © 2012, by OPEN CASCADE S.A.S.

PROPRIETARY RIGHTS NOTICE: All rights reserved. Verbatim copying and distribution of this entire document are permitted worldwide, without royalty, in any medium, provided the copyright notice and this permission notice are preserved.

The information in this document is subject to change without notice and should not be construed as a commitment by OPEN CASCADE S.A.S.

OPEN CASCADE S.A.S. assures no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such a license.

**CAS.CADE**, **Open CASCADE** and **Open CASCADE Technology** are registered trademarks of OPEN CASCADE S.A.S. Other brand or product names are trademarks or registered trademarks of their respective holders.

---

#### NOTICE FOR USERS:

This User Guide is a general instruction for Open CASCADE Technology study. It may be incomplete and even contain occasional mistakes, particularly in examples, samples, etc.

OPEN CASCADE S.A.S. bears no responsibility for such mistakes. If you find any mistakes or imperfections in this document, or if you have suggestions for improving this document, please, contact us and contribute your share to the development of Open CASCADE Technology: [bugmaster@opencascade.com](mailto:bugmaster@opencascade.com)



<http://www.opencascade.com/contact/>

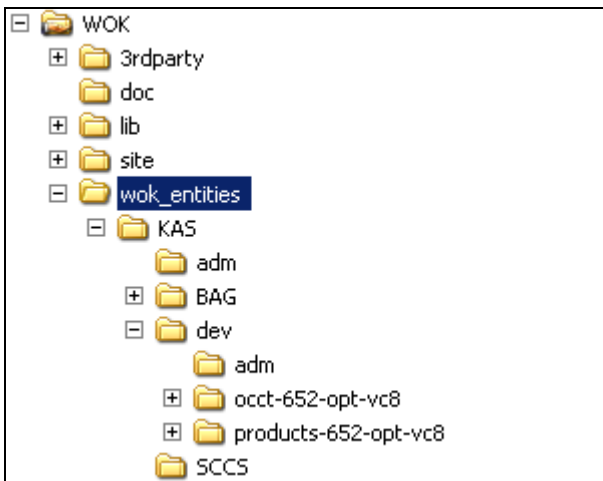
# 1. *Configuring the environment*

## 1.1. *Binary WOK package*

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

## 1.2. *Installation of Binary WOK package*

During the installation process the Install Wizard will ask you which VStudio you are going to use, where VStudio and 3rd-parties are located and will automatically create the default Factory (`%WOK_INSTALL_DIR%\wok_entities\KAS\dev`).



## 1.3. *Configuration and launch*

The Install Wizard will create a Workshop Organization Kit Shortcut Group. You can use Shortcuts from this group in order to:

- Customize the environment (edit file “custom.bat” evaluated when WOK starts)
- Run WOK inside Emacs editor (“WOK Emacs”). The short-cut starts empty Emacs; in order to open a WOK shell, press Esc-X and type “woksh”, then press Enter.
- Run WOK in command prompt window (“WOK TCL shell”).

---

## 2. *Initializing the workbench*

By default WOK installer creates factory KAS with one workshop dev (WOK path :KAS:dev).

To start working with OCCT, first create a WOK workbench (command **wcreate**). By default, it is created in the default location within the factory. It is possible to redirect it to a different location, e.g. to where OCCT code is checked out from your source repository. The only restriction is that the workbench should have the same name as that directory. For instance:

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

To work with OCCT code stored in the source repository, you need to make a checkout operation into the workbench (if not done before). Before this operation it is strictly recommended to remove **UDLIST** files from directory **adm** that is created by WOK in the workbench home subdirectory. For example:

```
> rm -rf D:/occt/adm D:/occt/src D:/occt/tools
> svn checkout http://svn/svn/occt/trunk D:/occt
```

Continue to work using normal WOK functionality (wprocess, umake, etc.) and source control system command.

If you need to make an isolated patch on OCCT do not forget to use WOK functionality to create inheritable workbenches by setting parent - child relationships between workbenches using wcreate option **-f**. For instance:

```
KAS:dev> wcreate -f occt occt-patches -DHome=D:/occt-patches
```

In the sample above **occt** means the existing father and **occt-patch** means a newly created child.

For patch creation and testing the child workbench can contain only a modified package and a toolkit corresponding to it. At the same time a package can contain modified files only.

---

### 3. *Using Visual Studio for building workbench*

It is possible to use Visual Studio environment for modifying, building, and debugging the code maintained by WOK. For that, MS project files should be generated (or updated each time any modification is made in CDL files or WOK units layout) with command **wgenproj**:

```
KAS:dev> wokcd occt
```

```
KAS:dev:occt> wgenproj
```

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

When 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):

```
KAS: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!

## 4. Launching DRAW Test Harness

It is possible to launch DRAW Test Harness (DRAWEXE) directly from WOK prompt.

Before launching DRAWEXE you have to configure the environment within `wenv` command:

```
KAS:dev> wokcd occt
KAS:dev:occt> wenv
KAS:dev:occt> DRAWEXE
KAS:dev:occt> pload ALL
```

