



SciEngines
massively parallel computing

RIVYERA

reconfigure versatilyly your efficient raw architecture

ProjectCompass Guide

Version 1.16

SciEngines GmbH
Fraunhoferstrasse 13
24118 Kiel
Germany

www.sciengines.com

Table of Contents

1	Introduction	2
2	Launching the ProjectCompass	3
3	Creating a new Project	5
4	Using a generated project	7

The Information in this document is provided for use with SciEngines GmbH ('SciEngines') products. No license, express or implied, to any intellectual property associated with this document or such products is granted by this document.

All products described in this document whose name is prefaced by 'COPA-COBANA', 'RIVYERA', 'SciEngines' or 'SciEngines enhanced' ('SciEngines products') are owned by SciEngines GmbH (or those companies that have licensed technology to SciEngines) and are protected by patents, trade secrets, copyrights or other industrial property rights. The SciEngines products described in this document may still be in development. The final form of each product and release date thereof is at the sole and absolute discretion of SciEngines. Your purchase, license and/or use of SciEngines products shall be subject to SciEngines's then current sales terms and conditions.



1 Introduction

Developing software for any kind of computer architecture is a sophisticated process, especially when a set of tools from different vendors is involved. SciEngines designed a tool, that establishes all required settings to straight forward start a new project - the *ProjectCompass*. The SciEngines ProjectCompass is intended to provide best assistance at the beginning of every project.

The ProjectCompass builds structured frameworks, where the SciEngines Rivyera API is already included for both, the VHDL- and the Host-part. More precisely, it creates a prepared Xilinx ISE VHDL Project and a related Eclipse IDE Project. Custom code can instantly and easily be added.

Platform-independency is guaranteed, since the ProjectCompass is a SUN Java Application, so it should run on every operating system providing a SUN Java Runtime Environment 1.6 (JRE) or later.

This document is a detailed guide, explaining the range of functions of the SciEngines ProjectCompass.

2 Launching the ProjectCompass

To start a new design, the ProjectCompass GUI can be launched.

Microsoft Windows

Microsoft operating systems usually do not provide a Java Runtime Environment. Please make sure, that a JRE is installed. The ProjectCompass is launched by double-clicking it.

Mac OS X

Apple computers usually ship with preinstalled Mac OS X. All current Mac OS X versions support Java. To launch the ProjectCompass, double click onto the applications's icon.

Unix or Linux

Execute *ProjectCompass* in a shell of your choice.

RIVYERA, shipped Operating System

Execute *ProjectCompass* in a shell of your choice.

After the tool is initialized, the main window is displayed.

The screenshot shows the ProjectCompass GUI window. The title bar reads "ProjectCompass, Copyright (c) 2010-2012, SciEngines GmbH. All rights reserved." The main area contains several form fields: "project name" (PingPong), "target root directory" (/tmp/Projects/PingPong) with a "Browse" button, "author" (SciEngines), and "description" (Example Project). Below these are dropdown menus for "data registers" (1), "ctrl registers" (1), and "fpga type" (xc6slx150-3fgg676). There is a checkbox for "generate slave FPGA project, too" which is unchecked. At the bottom, there is a dropdown for "fpga project" (ISE 14), a checkbox for "overwrite existing files" which is unchecked, and "Quit" and "Create" buttons.

Note: If the ProjectCompass GUI should be run from a SSH-connected computer, make sure that a Display-Server (X-Server) is installed and running. The SSH-connection has to be established by using parameter -X or -Y!

The ProjectCompass also provides a command line interface. To run the command line interface, execute in a shell

ProjectCompass -h or *ProjectCompass --help*

The official help will be displayed.

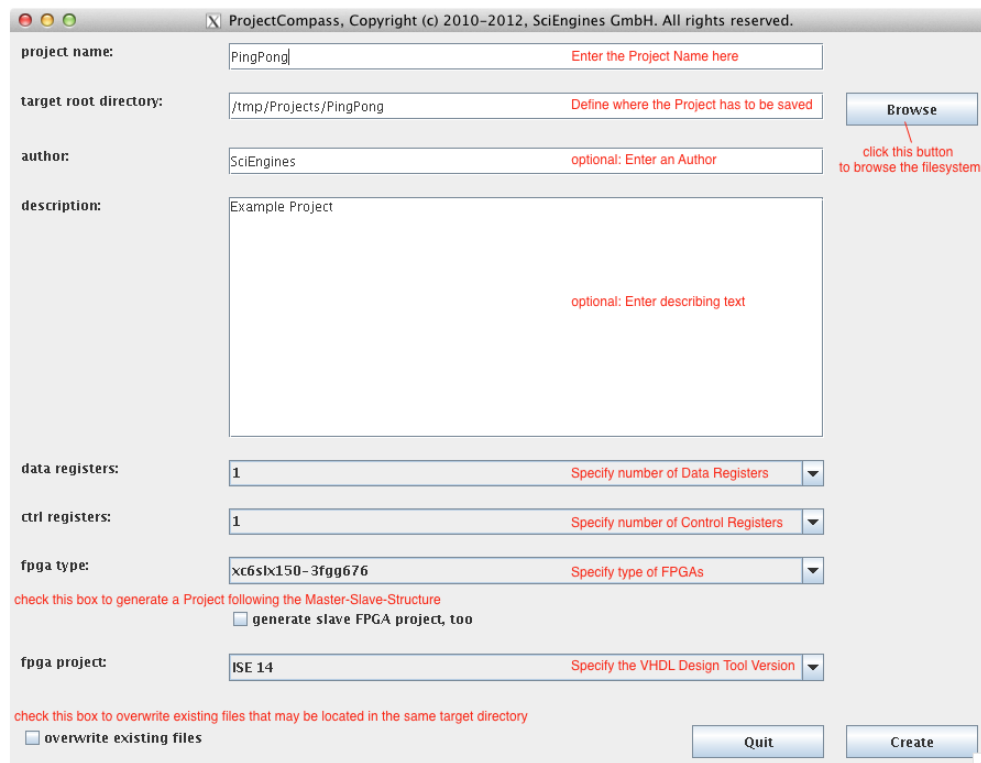
```
# :>ProjectCompass --help
Usage: ProjectCompass [OPTIONS]
to generate a project.
where options include:
-g --genexample PATH generates the PingPong example in path PATH
-c --creg NUM generates PingPong with NUM ctrl-registers
(only valid in conjunction with -g)
-d --dreg NUM generates PingPong with NUM data-registers
(only valid in conjunction with -g)
-t --type TYPE set FPGA type. TYPE has to be one of these:
"xc3s1000-4ft-256"
"xc3s1500-4fg676"
"xc3s5000-4fg676"
"xc6slx75-3fg484"
"xc6slx150-3fgg676"
"xc4vex35-10ff668"
-h --help print this help and exit
-v --version print product version and exit
```

Parameter		Description
-g	PATH	Generates an example project, including all structures and example code
-c	NUM	Related to -g, this option sets the number of control registers
-d	NUM	Related to -g, this option sets the number of data registers
-t	TYPE	Specify the type of the FPGA that is under development
-h		Prints the help menu
-v		Prints the application's version number

3 Creating a new Project

The main window of the ProjectCompass uses the chosen preferences to create the project files.

Option	Description
<i>project name</i>	Name of the Project
<i>target root directory</i>	Path of the Design Folder to store the Project to
<i>author</i>	Name of the Author (optional)
<i>description</i>	Project Description (optional)
<i>data registers</i>	Number of required API Data Registers
<i>ctrl registers</i>	Number of required Control Registers
<i>fpga type</i>	Destination FPGA Chip Device
<i>fpga project</i>	VHDL Development Tool Plattform



The filesystem can be browsed to determine the *target root directory*. For Master-Slave architectures, the ProjectCompass is able to conveniently generate an accordingly structured project. This is necessary for SciEngines Rivyera systems equipped with add-on boards where each regular FPGA has a connected Slave-FPGA it interferes with. Project files with the same name are only overwritten if the ProjectCompass is encouraged to do this.

In the example, a new project 'PingPong' is created and saved to the directory */tmp/Projects/PingPong*. The project is set up to use one data and one control register for a Xilinx Spartan6 and ISE14.

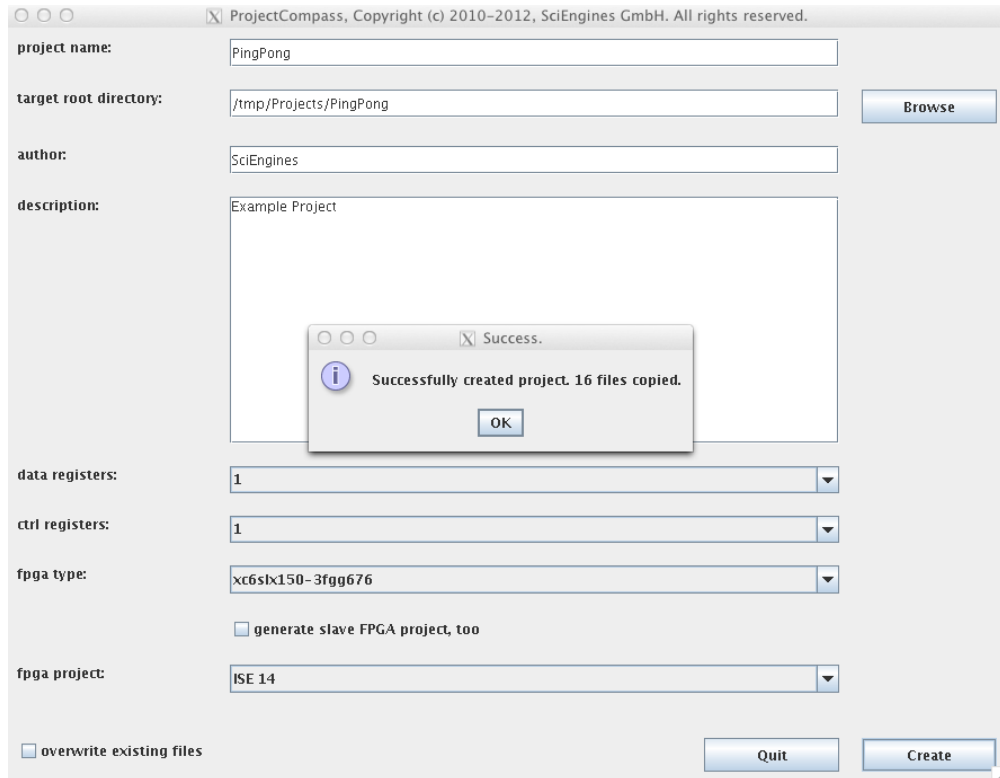
All necessary options can also be specified by command line parameters. Additionally, an example project can be generated: the simple Rivyera API communication-example *PingPong*. This project already has user code filled into the arranged API framework. The idea of the implemented VHDL Core is to echo each incoming data word back to the source.

Note: A configuration file stores the given preferences. These settings are recognized at the next start of the ProjectCompass.

For example:

```
# :>ProjectCompass -g /tmp/Projects/PingPong -c 1 -d 1 -t xc6lx150-3fgg676
```

When the project framework is completely set up, the ProjectCompass notifies the successive generation of the project files. Confirm this notification and quit ther ProjectCompass by clicking *Quit*.

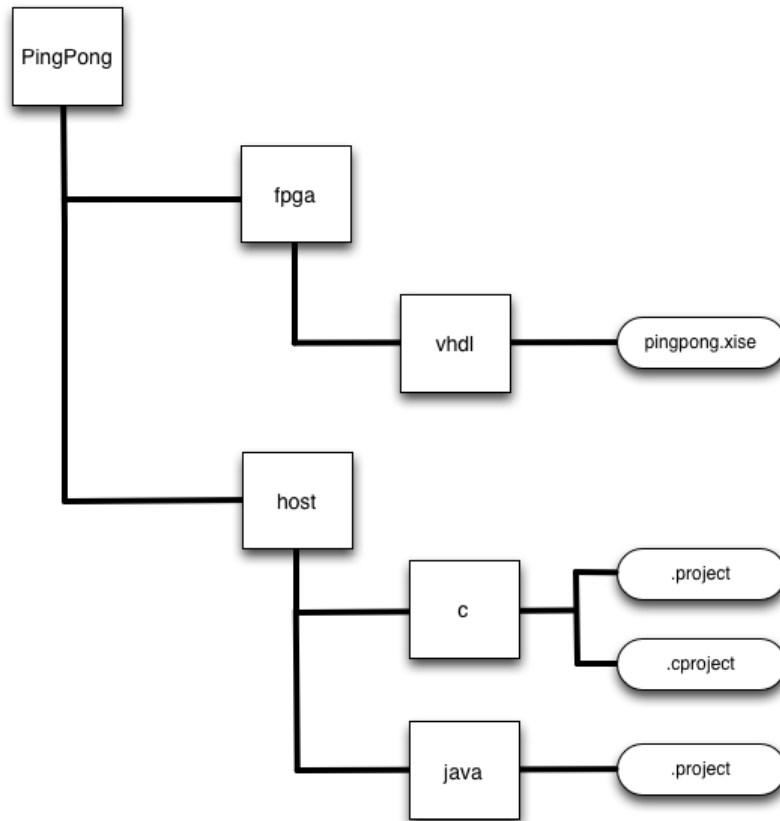


Note: If the ProjectCompass GUI is not used, the command line interface will notify the successful creation of the project files, too.

```
# :>ProjectCompass -g /tmp/Projects/PingPong -c 1 -d 1 -t xc6lx150-3fgg676  
# :>Successfully created project, 16 files copied
```


4 Using a generated project

The SciEngines ProjectCompass generated two main folders into the target directory: *fpga* and *host*.



The VHDL Project can directly be opened with the installed Xilinx Development Tool ISE. Choose:

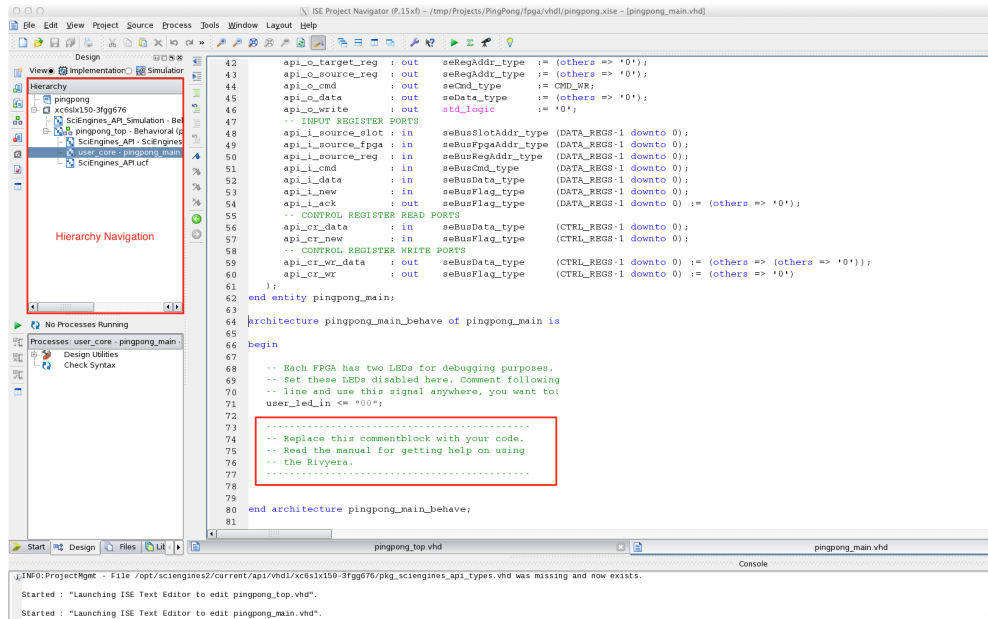
File - Open Project - *Browse to Project Directory* - select *.xise - Open

Note: The Xilinx ISE version determined in the ProjectCompass settings and the installed version should match!

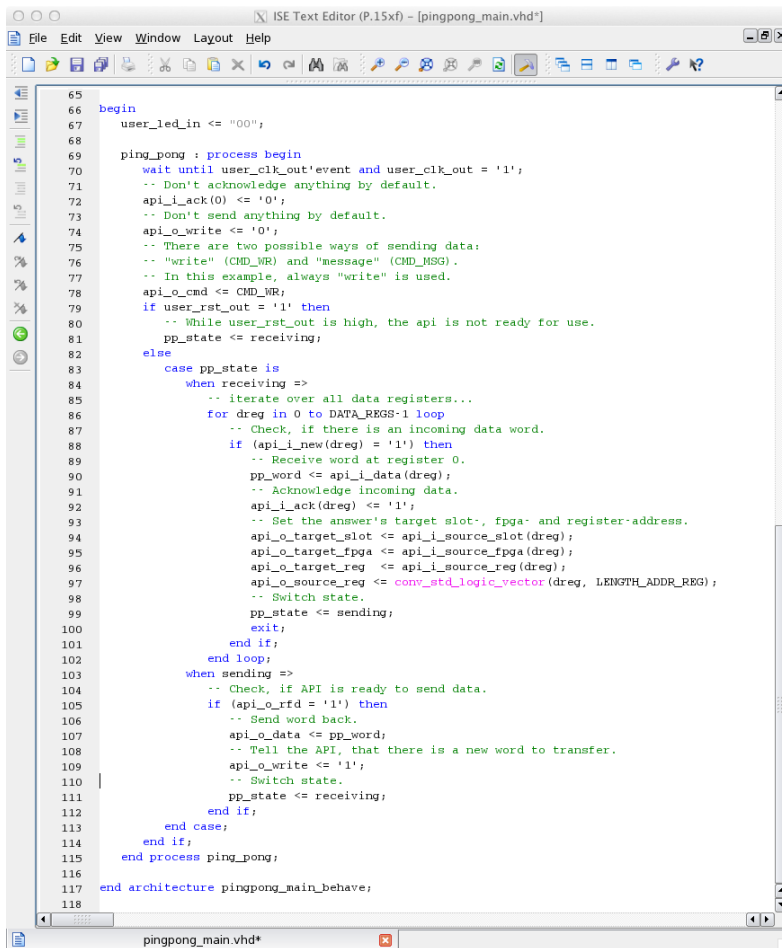
The project files for the host application can also directly included into the Eclipse IDE. Choose:

File - Import - General - Existing Projects into Workspace
- select the *host* directory according to your preferred programming language
- Finish

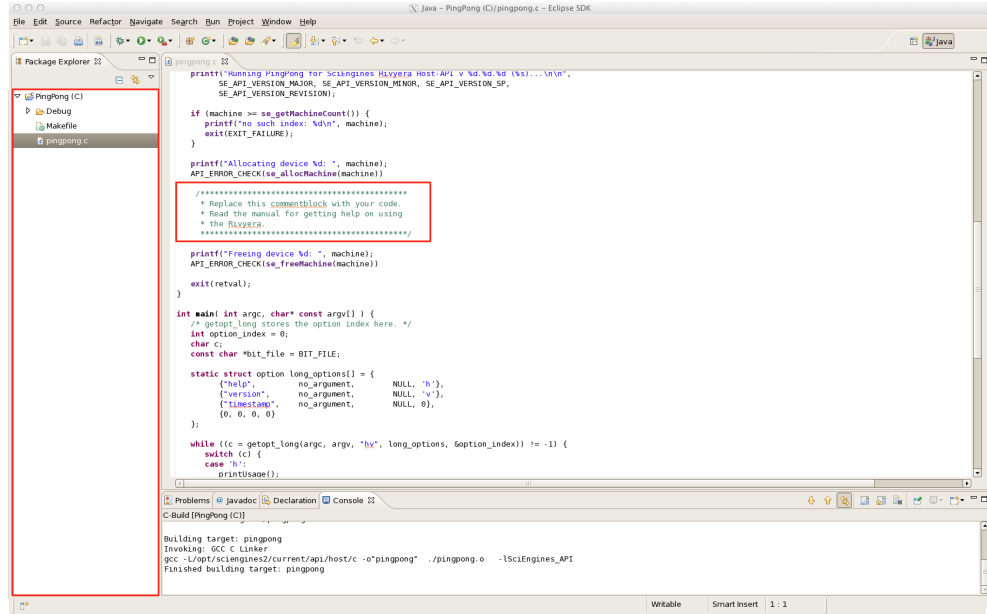
All related project files are directly included into ISE. The *Hierarchy* is useful to navigate through the files. *user_core* contains the section, where the custom code should be inserted. The correct location is marked by a commented area.



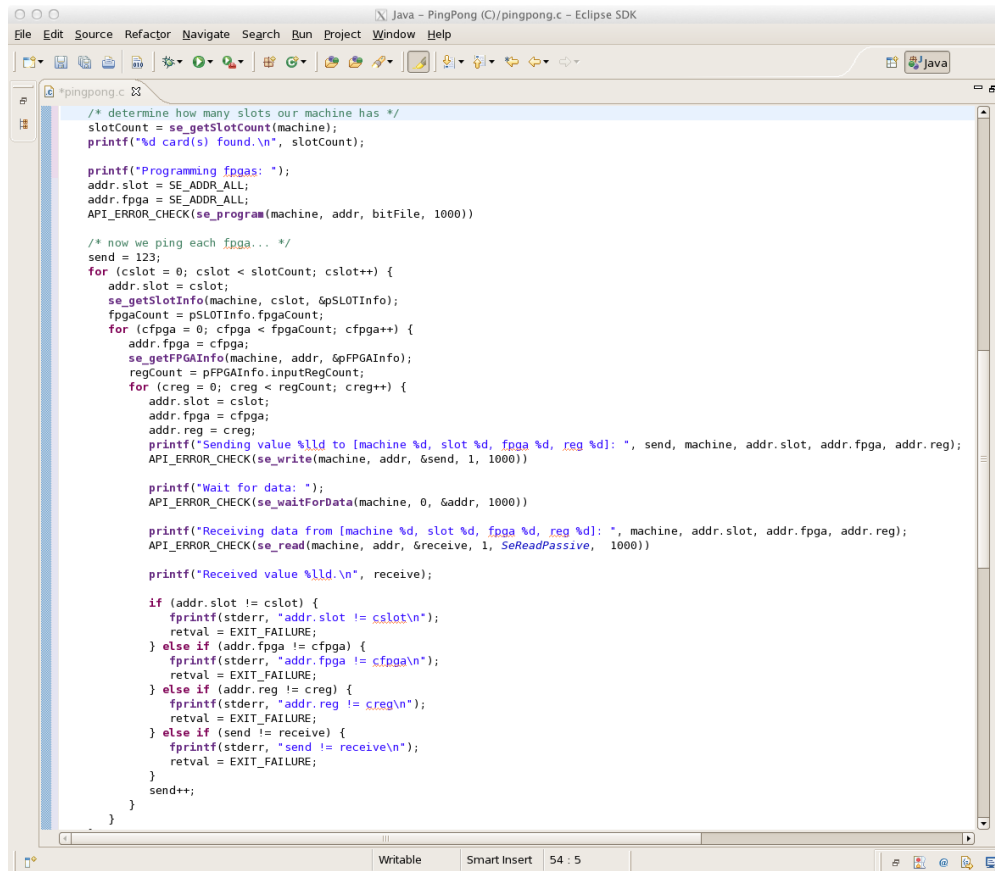
The *PingPong* example already has the user code filled in.



The host application is developed in the same way. The user code also has to be added to the API framework.



The PingPong example already provides the user application.



Thank you for choosing an original SciEngines product.

Imprint

Responsible for content:

Firm SciEngines GmbH

Street Fraunhoferstr 13

ZIP D-24118

City Kiel

Country Germany

Phone +49 431 5302 482

Email info@sciengines.com

WWW <http://www.sciengines.com>

CEO Gerd Pfeiffer

Commercial Register Amtsgericht Kiel

Commercial Register No. HR B 9565 KI

VAT- Identification Number DE 814955925

Disclaimer: Any information contained in this document is confidential, and only intended for reception and use by the specified person who bought the SciEngines product. Drawings, pictures, illustration and estimations are non binding and for illustration purposes only. If you are not the intended recipient, please return the document to the sender and delete any copies afterwards. In this case any copying, forwarding, printing, disclosure and use is strictly prohibited.