



appliedinformatics

Applied Informatics POCO^{PRO} C++ Frameworks

Getting Started Guide

Version 1.11

Purpose of This Document

This document guides developers interested in the POCO^{PRO} C++ Frameworks through the evaluation process.

The document is targeted at developers and development/technical managers wanting to get an overview of the functionality and features offered by the POCO^{PRO} C++ Frameworks. Familiarity with the C++ programming language is assumed.

Validity of This Document

This document covers release 2018.1 and later releases of the POCO^{PRO} C++ Frameworks.

Copyright, Trademarks, Disclaimer

Copyright © 2008-2018, Applied Informatics Software Engineering GmbH. All rights reserved.

All trademarks or registered marks in this document belong to their respective owners.

Information in this document is subject to change without notice and does not represent a commitment on the part of Applied Informatics. This document is provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the particular purpose. Applied Informatics reserves the right to make improvements and/or changes to this document or the products described herein at any time.

Table of Contents

1	Welcome.....	4
2	Installing the Evaluation Release	5
2.1	Installing on Windows	5
2.2	Installing on Unix Platforms	8
2.2.1	Unpacking the Installer	8
2.2.2	Running the Install Script	9
2.2.3	Building the Samples	10
3	Working with the POCO ^{PRO} C++ Frameworks	12
3.1	Viewing the Documentation	12
3.2	Working with the Samples	12
3.3	Using Eclipse on Linux	13
3.3.1	Eclipse CDT Build System	13
3.3.2	External Build System	16
3.4	Sample Notes	21
3.4.1	XML Samples	21
3.4.2	Remoting NG Samples	21
3.4.3	Open Service Platform Samples	21
4	Support, Licensing and Contact.....	23

1 Welcome

Thank you for your interest in the POCO^{PRO} C++ Frameworks, and welcome to our evaluation program. This document will help you in getting a smooth ride while installing the POCO^{PRO} C++ Frameworks evaluation software and evaluating the software for potential use in your project or company.

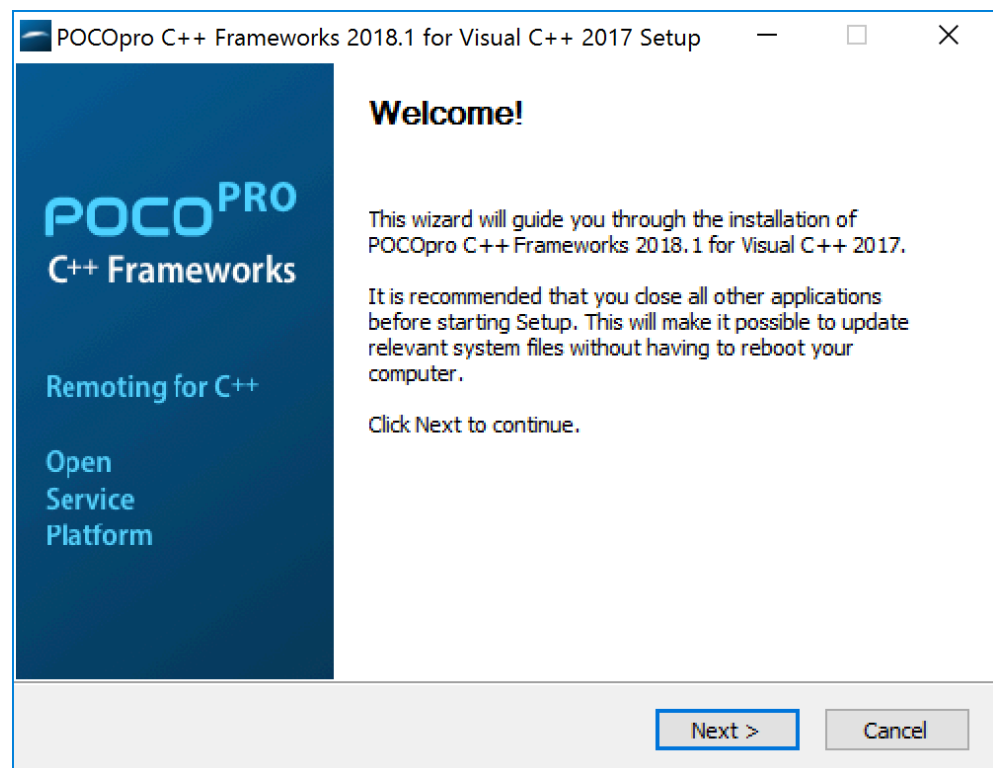
If at any point during the evaluation you have any questions or need support, please contact us via email at support@appinf.com.

2 Installing the Evaluation Release

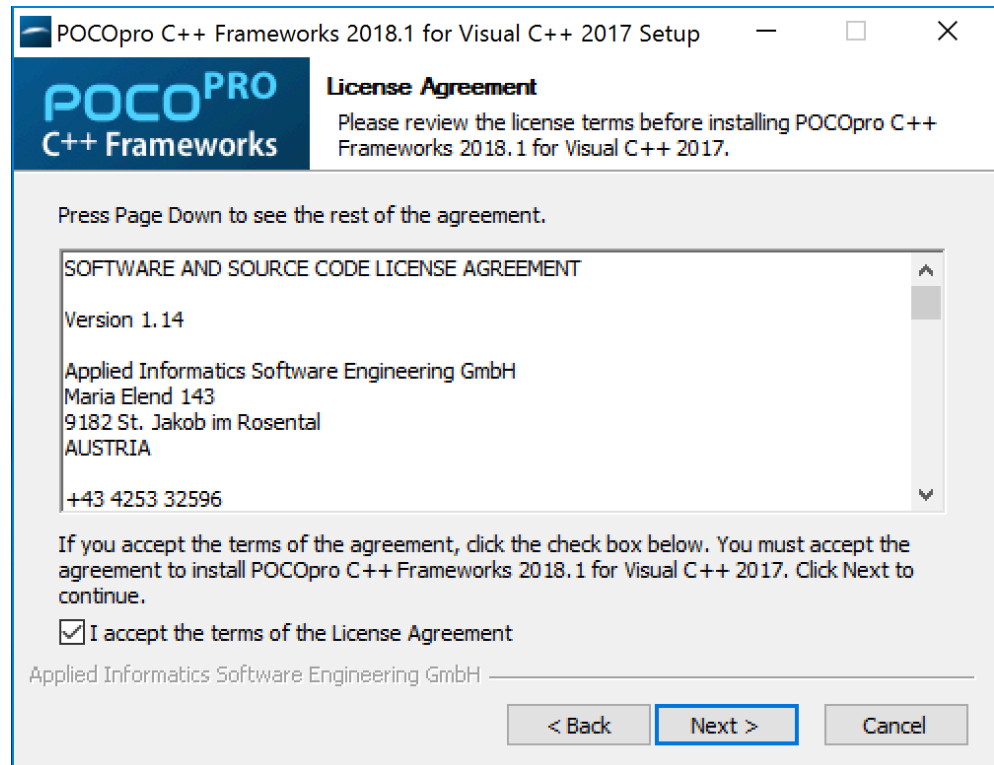
Applied Informatics offers evaluation versions of the POCO^{PRO} C++ Frameworks for Windows (Visual Studio), Linux and macOS platforms. Installers can be downloaded from the following address: <https://pocoproject.org/pro>. Multiple versions of the Windows installer are available, supporting Visual Studio versions 2013 to 2017. Installers for Linux distributions and macOS are available as well. Note that the Windows evaluation installers only contain 32-bit libraries and executables. The full product supports building 64-bit libraries and executables as well.

2.1 Installing on Windows

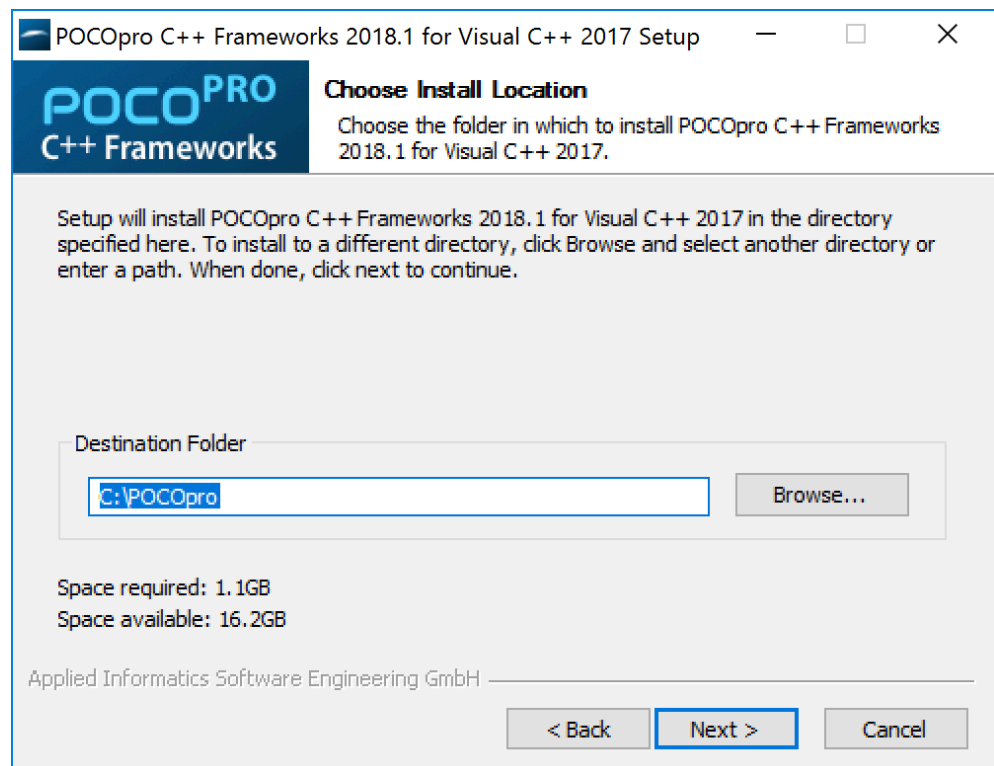
Installing the evaluation package for Windows platforms is straightforward. If you have not done so yet, please download the appropriate installer application for the version of Visual Studio you are using from <https://pocoproject.org/pro>. All installers contain the same software, only compiled with different versions of the compiler. Also, the project and solution files for the samples are different. After downloading has completed, run the installer application.



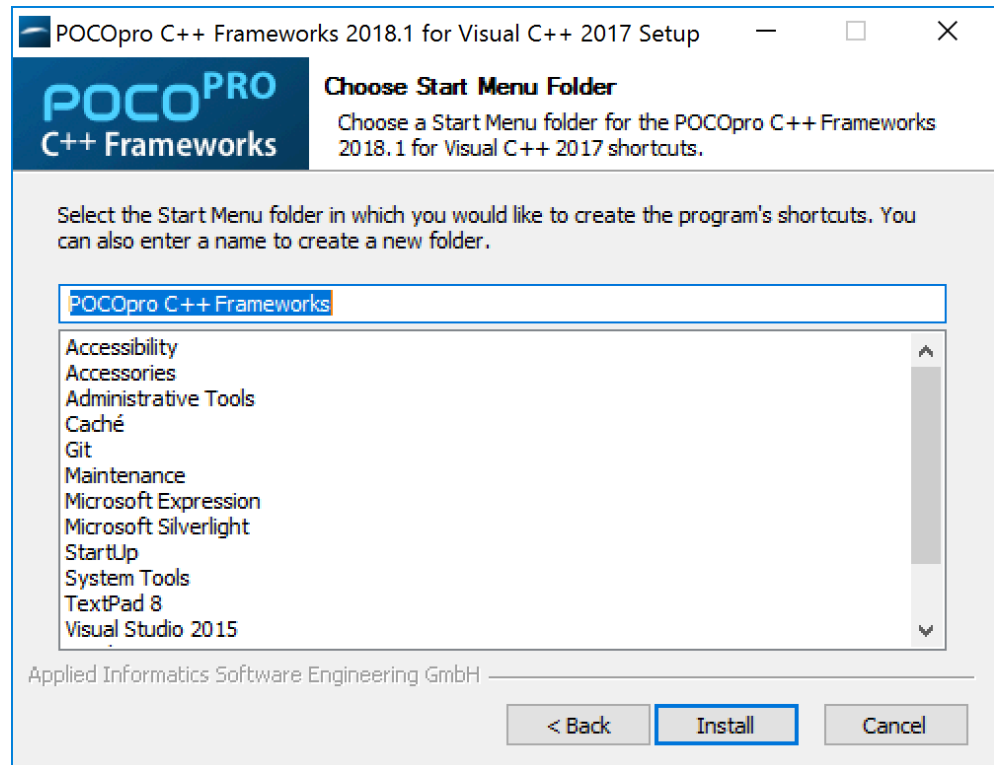
Click **Next** to begin the installation.



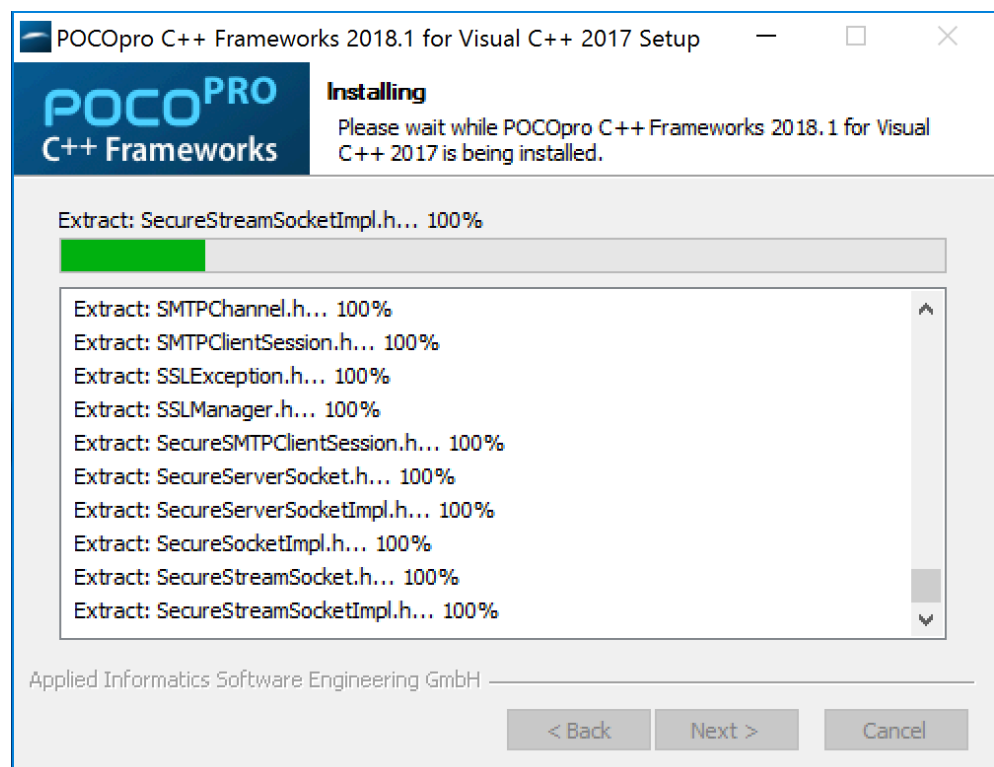
Review the license agreement and click the checkbox to accept the terms in the license agreement. Then click **Next** to continue.



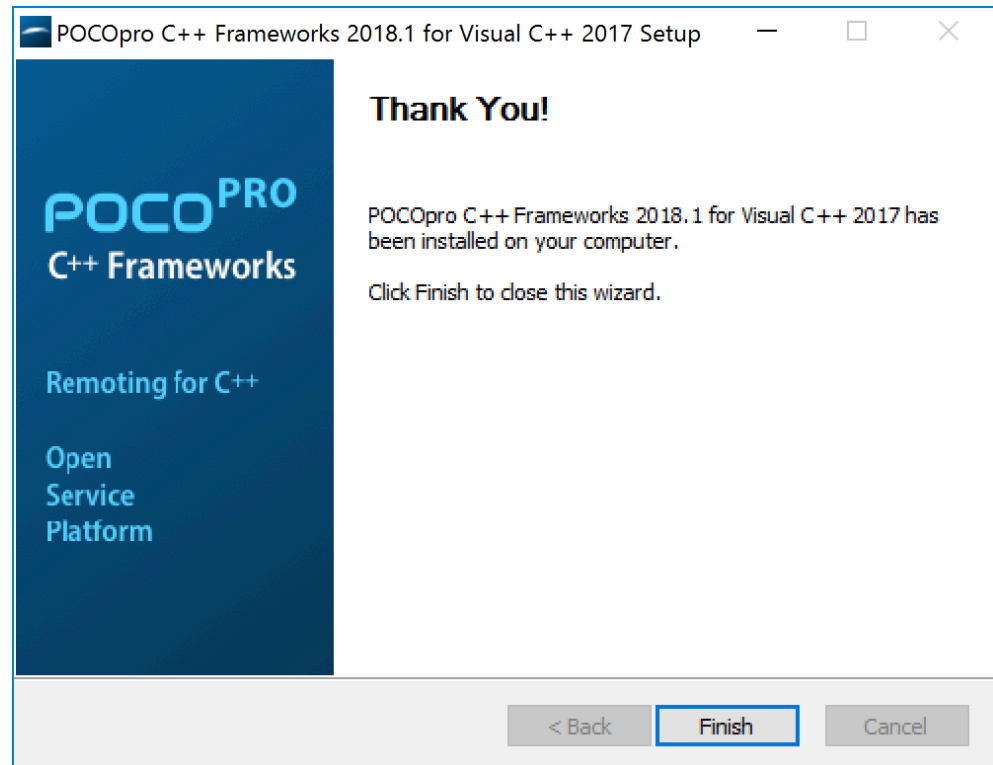
Choose a directory where to install the software (the default is C:\POCOpro). Then click **Next** to continue.



Then select the Start menu folder, where you would like to have the shortcuts to the documentation, samples, and the uninstall application. The default is to create a **POCopro C++ Frameworks** entry in the Programs menu. Click **Install** to finish the installation.



The installer is now copying the files to your computer. This may take a few minutes.



When the installation is completed, click **Finish** or close the window to exit the installer application. You are now almost ready to explore the POCO^{PRO} C++ Frameworks.

As the final step, you must install the license file which you received from Applied Informatics. The file is called `poco.license`, and you must place the file either in the Windows directory (usually, `C:\WINDOWS`), or in your Windows home directory (on Windows 7 this is `C:\Users\<YourName>`).

If you have not yet received the license file yet, please go to <https://pocoproject.org/pro> and fill out the form. The license file will be emailed to you automatically, usually within a few minutes.

2.2 Installing on Unix Platforms

Installing the POCO^{PRO} C++ Frameworks evaluation software on a Unix or Linux system is almost as easy as running a Windows installer.

Unix or Linux installers come as a gzip'ed TAR file. It will be named like `POCopro-2018_1-lxeval-Ubuntu-16_04-x86_64.tar.gz`.

2.2.1 Unpacking the Installer

First, uncompress the TAR file on your target system:

```
$ tar -xzf POCopro-2018_1-lxeval-Ubuntu-16_04-x86_64.tar.gz
```


This will create a directory named like the TAR file in the current directory. Change into that directory, and run the installer script (`install.sh`) contained therein.

Note that during the installation you will be asked for a directory where to install the software. If you want to install the software outside of your home directory (e.g. in `/opt` or `/usr/local`), you must run the installer script with superuser privileges (i.e. as `root`).

2.2.2 Running the Install Script

```
$ cd POCOpro-2018_1-lxeval-Ubuntu-16_04-x86_64
$ ls
install.sh  license.txt  postinstall.sh  product.tar  product.txt  README
$ ./install.sh

***
*** Welcome to the
***
***     POCOpro C++ Frameworks 2018.1
***
*** installer program!
***

This program will guide you through the installation of POCOpro C++
Frameworks 2018.1.

Copyright (c) 2007-2018 by Applied Informatics Software Engineering GmbH.

Please review and accept the license terms before the installation begins.
Press RETURN to review the License Agreement.
```

Follow the instructions and press **RETURN** to review the license agreement. Use the **SPACE** bar to move through the pages. Enter **Yes** (with an uppercase Y) to begin the installation.

```
...
partnership to find an arrangement that approximates as nearly as
possible the inoperative terms.

===== END OF LICENSE AGREEMENT =====

Do you accept the terms in the License Agreement (please answer with 'Yes'
to accept)?
==> Yes
```

Next, you will be asked where to install the software. Enter the path to a directory where the software should be installed. If the directory you specify does not exist, it will be created.

```
Please specify the directory where POCOpro C++ Frameworks 2018.1 shall be
installed (default: '/home/guenter/pocopro ').
==> /opt/pocopro

The directory '/opt/pocopro' does not exist and will be created.
```

Press RETURN to continue, or CTRL-C to cancel.

Press RETURN to begin the installation, or CTRL-C if you wish to cancel.

The software is now being installed. This may take some time.

**** IMPORTANT NOTE ****

The POCOpro C++ Frameworks have been installed into
'/Users/guenter/pocopro'.

To compile the sample applications located in
'/Users/guenter/pocopro/samples',
you have to set the environment variable POCO_BASE to the
path given above before invoking GNU make:

```
> export POCO_BASE=/Users/guenter/pocopro
```

You can do that by sourcing the script
'/Users/guenter/pocopro/etc/poco.env'.

To build the samples for Remoting:

```
> source /Users/guenter/pocopro/etc/poco.env  
> cd $POCO_BASE/samples/RemotingNG  
> make -s
```

To build the samples for OSP:

```
> source /Users/guenter/pocopro/etc/poco.env  
> cd $POCO_BASE/samples/OSP  
> make -s
```

To view the documentation, point your web browser to
'/Users/guenter/pocopro/doc/index.html'.

If you have received a license file ('poco.license'),
please put this file into your home directory.

The software has been successfully installed.
Thank you for installing POCOpro C++ Frameworks 2018.1.

```
$ _
```

You are now almost ready to explore the POCO^{PRO} C++ Frameworks.

As the final step, you must install the license file which you received from Applied Informatics. The file is called `poco.license`, and you must place the file either in your home directory, or in the `/etc` directory.

2.2.3 Building the Samples

To reduce the size of the downloads, the installer packages do not contain pre-built executables for the sample applications. However, building the samples is easy.

First, ensure that the environment is set up for developing with the POCO^{PRO} C++ Frameworks and its build system. To do this, source the `poco.env` script shell contained in the `etc` directory. The script will set the `$POCO_BASE` environment variable and add `$POCO_BASE/bin` to the `$PATH` environment variable.

Note: In the following it is assumed that you have installed the POCO^{PRO} C++ Frameworks into a directory named `pocopro` in your home directory.

```
$ cd pocopro
$ source etc/poco.env
```

Then, change to the `samples` directory, and build the Foundation samples.

```
$ cd samples
$ cd Foundation
$ make -s -j4
```

Note: be sure to use **GNU make** (and not BSD make) to build the samples. Use the GNU Make `-jn` option to parallelize the build on a multicore machine.

Build the samples for all other libraries as required.

```
[guenter@localhost Foundation]$ cd ../XML
[guenter@localhost XML]$ make -s -j4
```

3 Working with the POCO^{PRO} C++ Frameworks

Now that you have successfully installed the POCO^{PRO} C++ Frameworks evaluation version, it's time to try it out.

3.1 Viewing the Documentation

Extensive reference documentation in HTML format is installed along with the libraries and tools by the installer. On Windows platforms, you can view the documentation with your default Web browser by opening the **Start** menu and selecting **Programs > POCOpro C++ Frameworks > Documentation**. On Unix platforms, open the file **doc/index.html** using your favorite Web browser.

Alternatively, the reference documentation is available online at <https://pocoproject.org/pro/docs/>.

3.2 Working with the Samples

On Windows platforms, Visual Studio solution and project files for all samples have been installed. To open all samples for the Foundation library in Visual Studio, open the *samples* directory using the Windows Explorer. You can do that using the **Start** menu, by choosing **Programs > POCOpro C++ Frameworks > Samples**, or by navigating to the samples directory (e.g., `C:\POCOpro\samples`) using the Windows Explorer. Then locate and open the **Foundation** directory, which contains a Visual Studio solution file for all Foundation library samples. For Visual Studio 2013 the solution file is named `samples_vs120.sln`. For Visual Studio 2015 the solution file is named `samples_vs140.sln`. For Visual Studio 2017, the solution file is named `samples_vs150.sln`. Double click the solution file to open it in Visual Studio.

*Depending on your Visual Studio version, it may be necessary to add the directories containing the POCO^{PRO} C++ Frameworks headers and libraries to the global header file and library search paths. In Visual Studio, select **Tools > Options**, then select **Projects and Solutions > VC++ Directories**. Make sure the directory containing the POCO^{PRO} C++ Frameworks header files (e.g., `C:\POCOpro\include`) is in the search list for Include files, and the library directory (e.g., `C:\POCOpro\lib`) is in the search list for Library files.*

*Please make sure that the **PATH** environment variable includes the directory containing the POCO^{PRO} C++ Frameworks shared libraries (usually `C:\POCOpro\bin`). The installer normally takes care of this.*

On Unix platforms, GNU Makefiles are provided for every sample. There is also a top-level Makefile invoking all sample Makefiles in every sample subdirectory.

Look at the sample code to see what it does and run the samples. Since almost all samples output something to the console, it is best to run the samples from a command shell.

Feel free to modify the samples to try out different features of the POCO^{PRO} C++ Frameworks.

Do not forget to set the `LD_LIBRARY_PATH` (`DYLD_LIBRARY_PATH` on macOS) environment variable accordingly (so that it includes `$POCO_BASE/lib`), before running a sample application or invoking one of the tools. Normally, this is done by sourcing the `etc/poco.env` script.

3.3 Using Eclipse on Linux

On Linux platforms, it is possible to use Eclipse (or other integrated development environments) to build applications with the POCO^{PRO} C++ Frameworks. There are two ways to use Eclipse – using Eclipse’s CDT build system, or using the POCO C++ Libraries build system used by the POCO^{PRO} C++ Frameworks.

Please note that the following steps require some familiarity with Eclipse CDT.

3.3.1 Eclipse CDT Build System

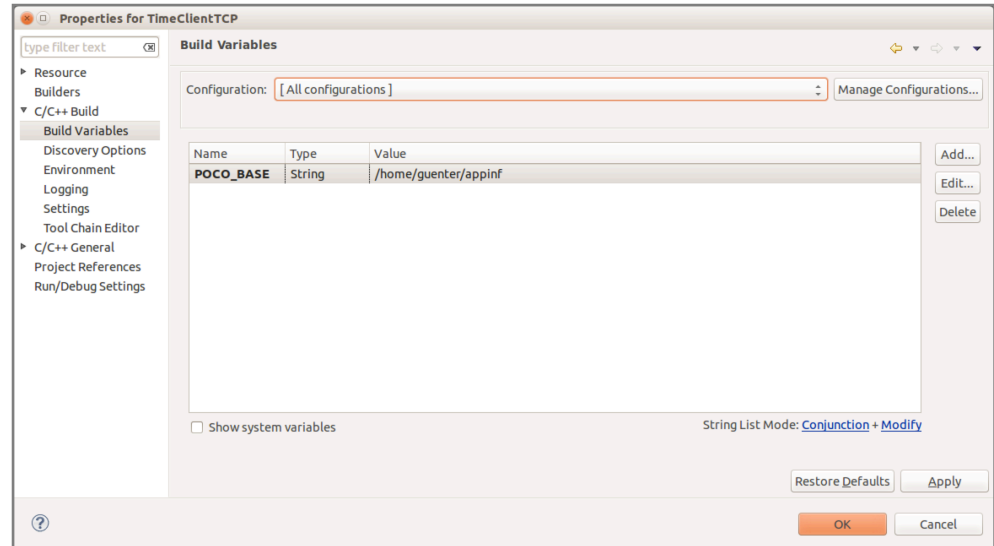
In the following we’ll show the steps to build the Remoting NG `TimeClientTCP` sample using Eclipse CDT, with CDT’s build system.

3.3.1.1 Create New Project

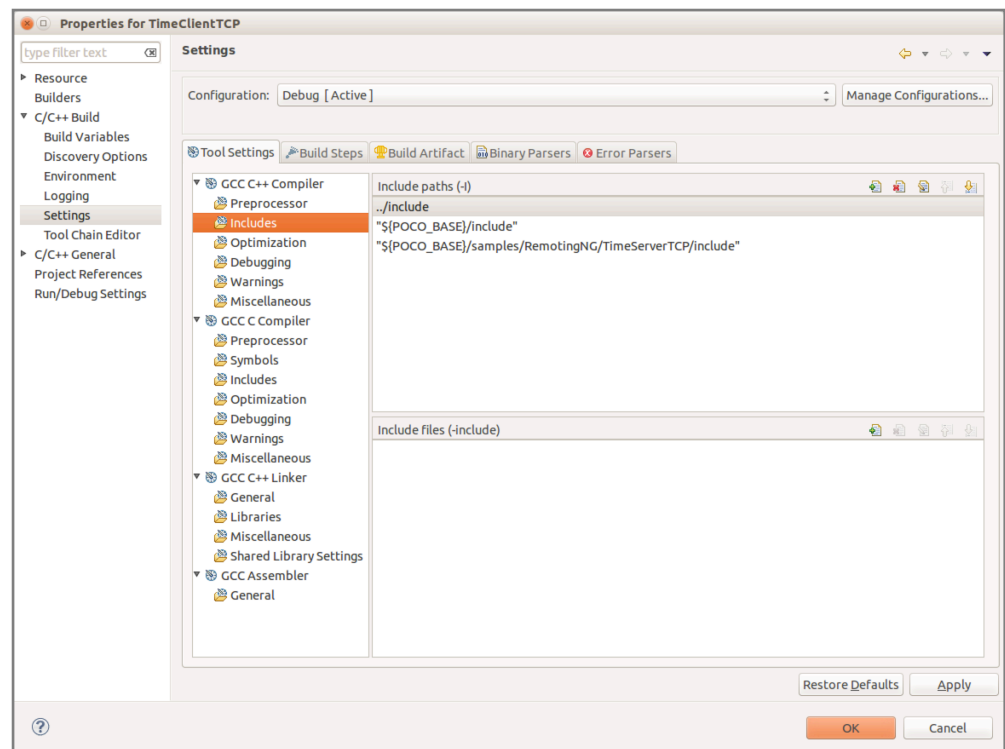
The first step is to create a new empty C++ project. Then, copy into the newly created project directory the `include` and `src` directories and their content from `samples/RemotingNG/TimeClientTCP`.

3.3.1.2 Project Settings

The next step is setting up the project configuration. We create a build variable named `POCO_BASE`, containing the path to the POCO^{PRO} C++ Frameworks installation directory. We’ll use this variable later on to define header and library search paths. Make sure this variable is defined for all build configurations. In the following screenshots, the installation directory is `/home/guenter/pocopro`.

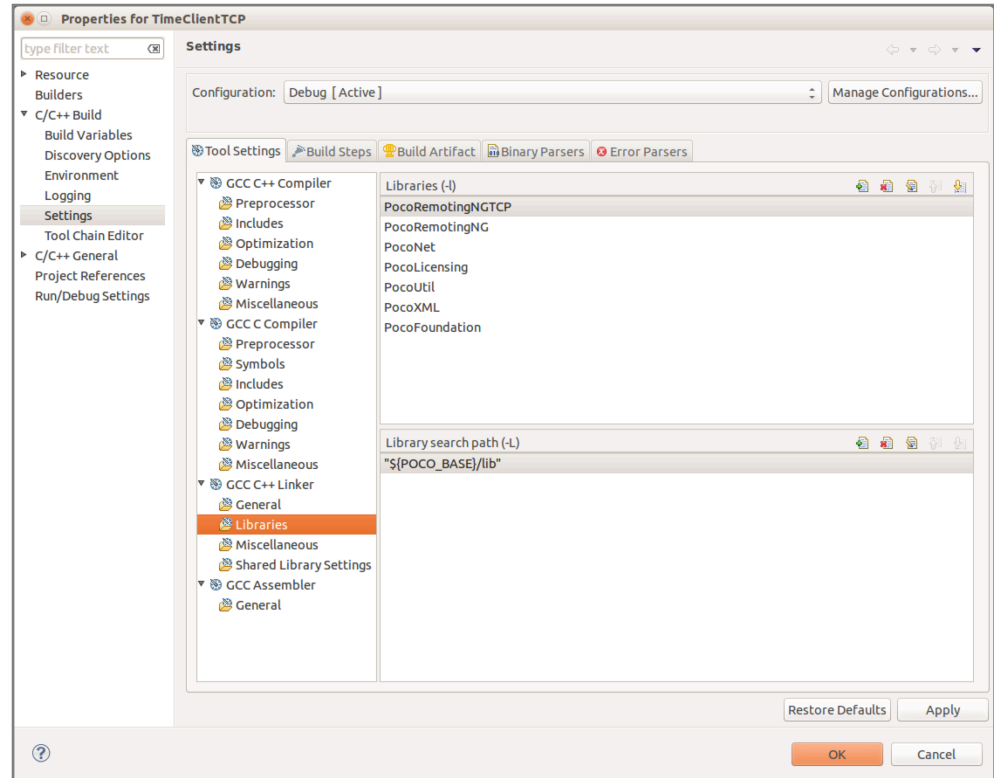


Then we'll set up the header search path.



We have to add the project's *include* directory, the *include* directory of the POCO^{PRO} C++ Frameworks, using the *POCO_BASE* build variable defined previously, as well as the include directory of the *TimeServerTCP* sample, as this contains a header needed by the client. Please make sure to specify the header search paths under *GCC C++ Compiler*.

Finally, the linker settings must be specified.

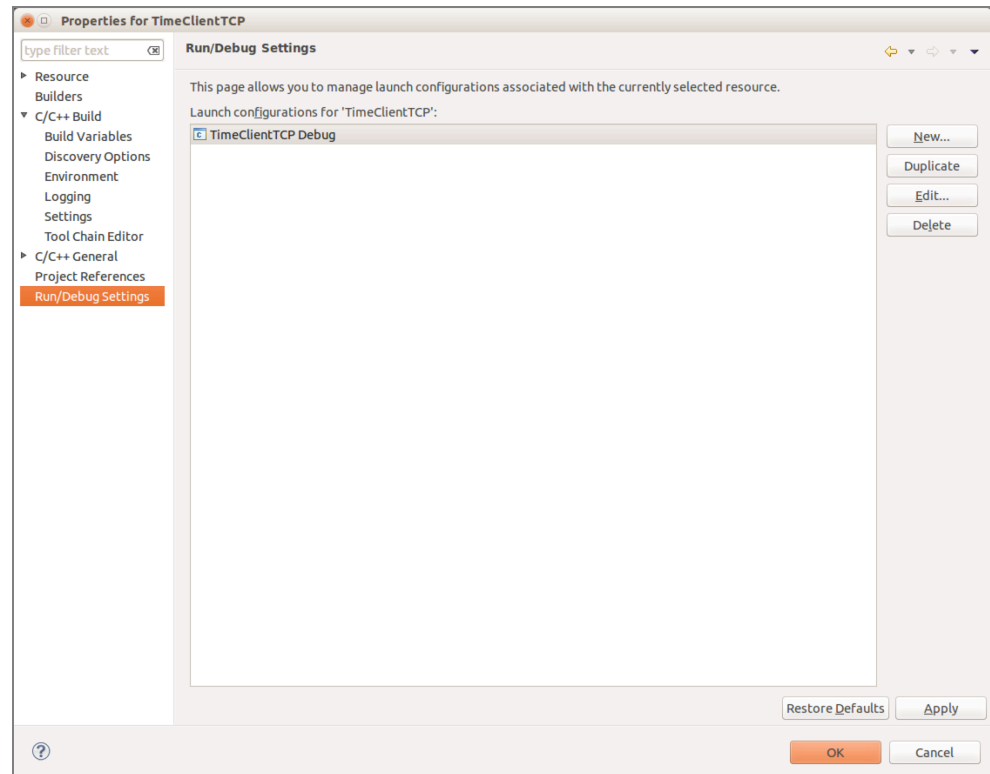


We need to specify the library search path (again, using *POCO_BASE* build variable), as well as which libraries to link.

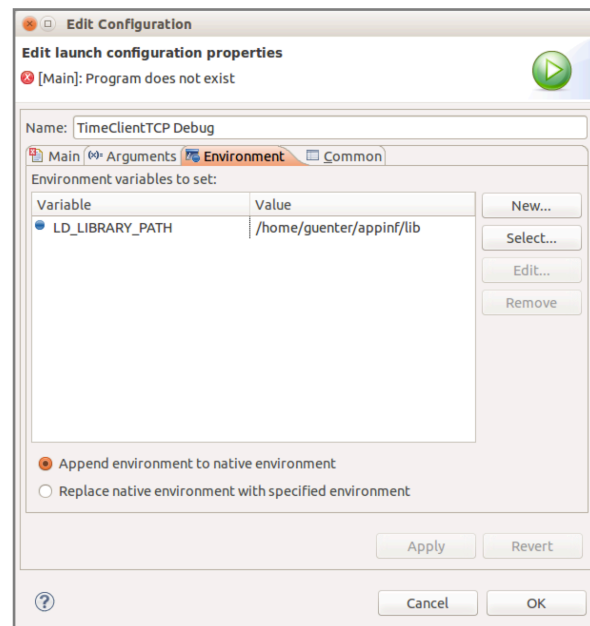
With these settings done, the project can be built.

3.3.1.3 Running the Project

In order to run the project, the shared library search path must be set. This is done under Run/Debug settings, by creating a new launch configuration.



In the launch configuration, specify the shared library search path, using the `LD_LIBRARY_PATH` environment variable.

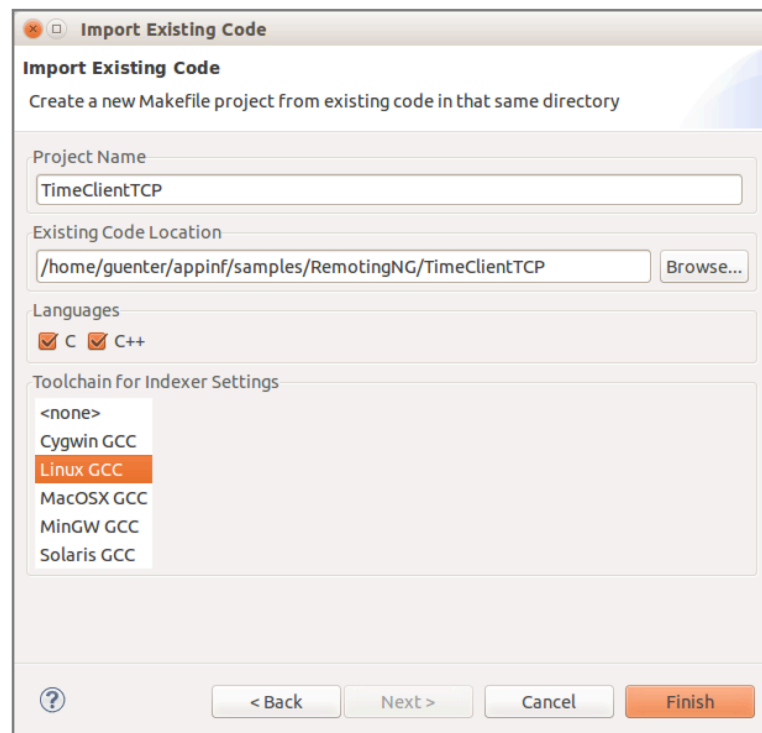
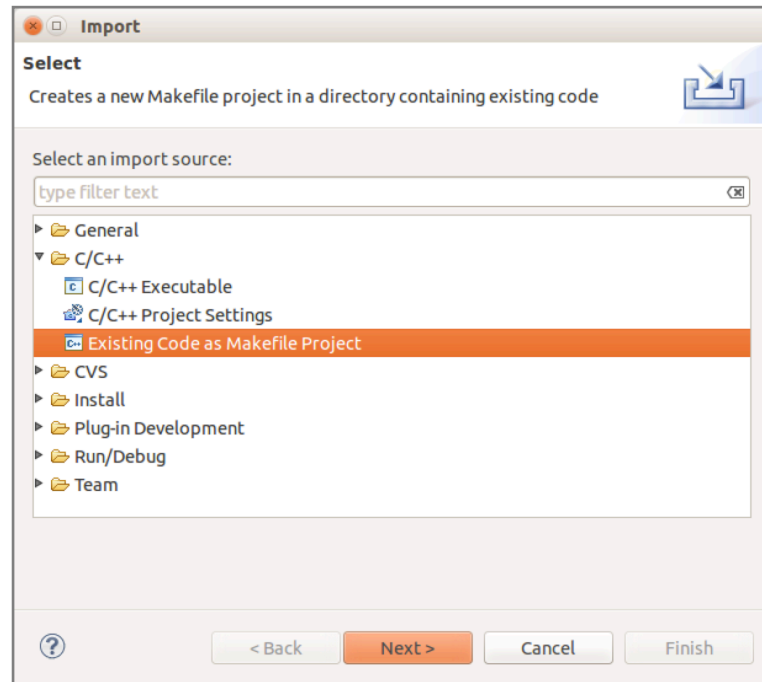


With these settings, you should be able to launch (and debug, if necessary) the executable directly from Eclipse.

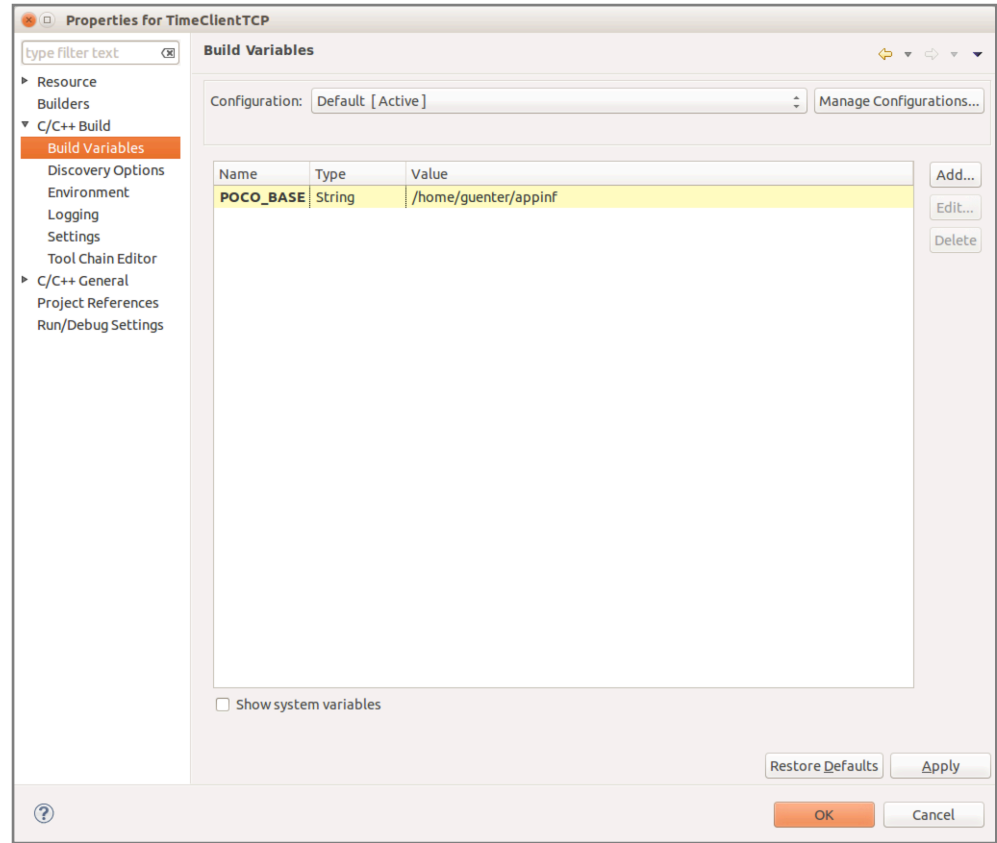
3.3.2 External Build System

You can import the samples directly into Eclipse and use the GNU Make-based build system used by the POCO^{PRO} C++ Frameworks. To do so,

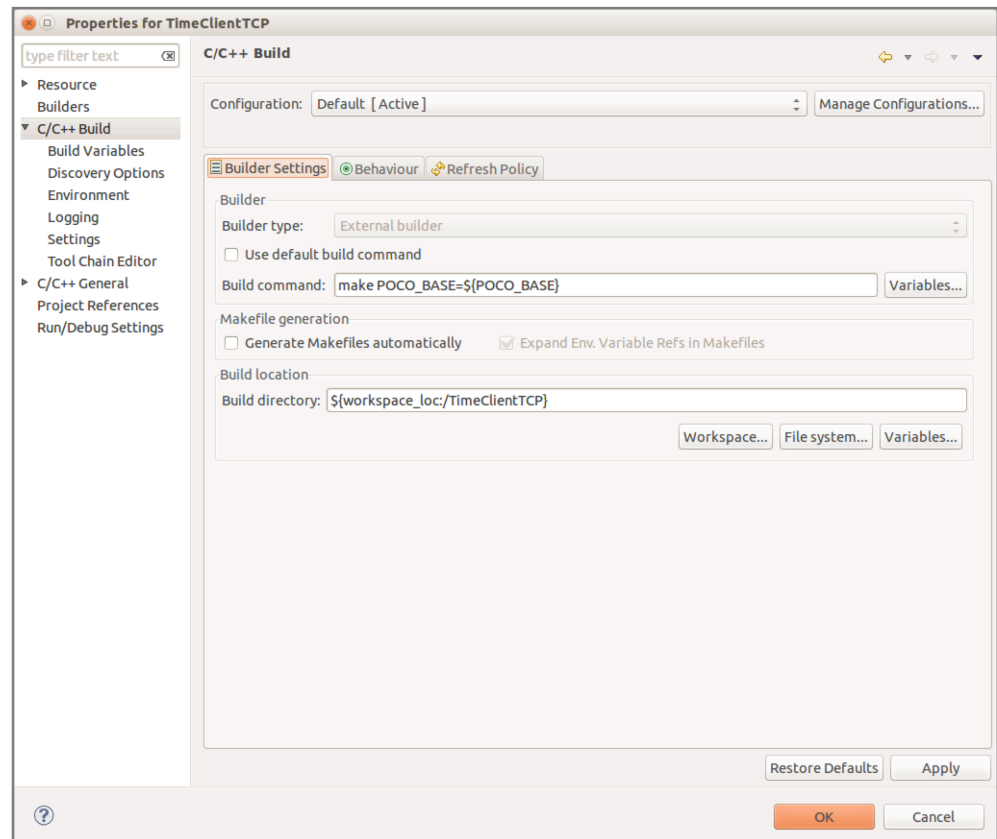
import a project (*Existing Code as Makefile Project*) and specify the sample directory (e.g., *samples/RemotingNG/TimeClientTCP*) as project directory.



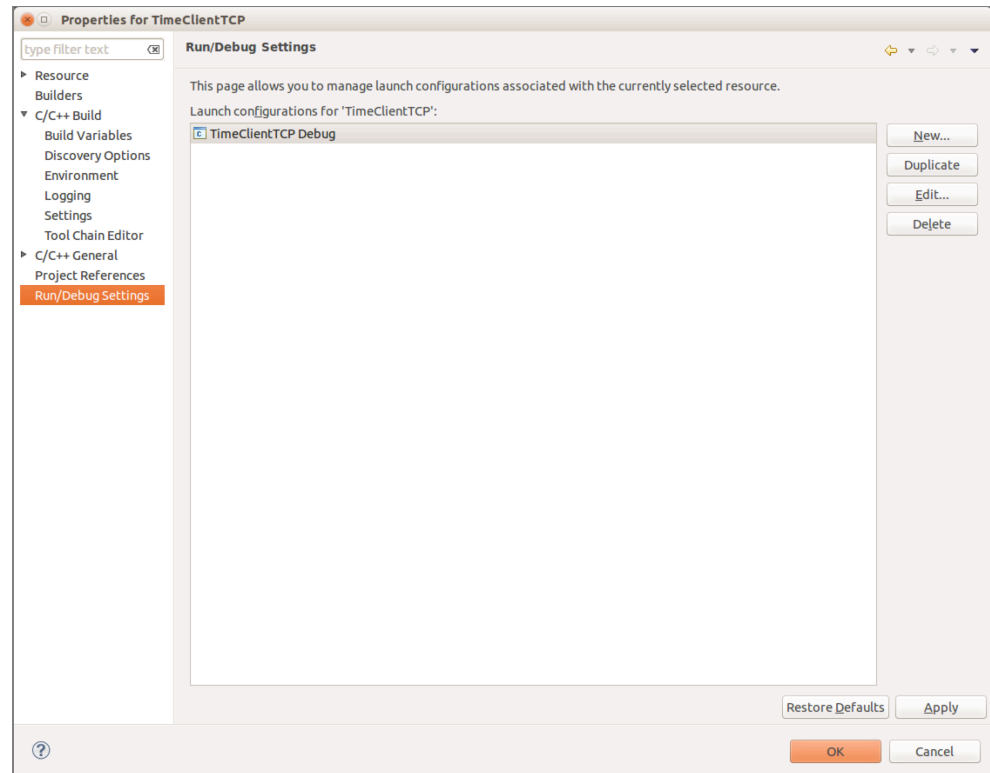
After importing the project, the project settings must be adjusted. First, create the `POCO_BASE` build variable, which we'll use later on.



Then, modify the build command used to build the project. We'll have to pass the *POCO_BASE* environment variable to GNU Make. This can be done in C/C++ Build > Builder Settings.

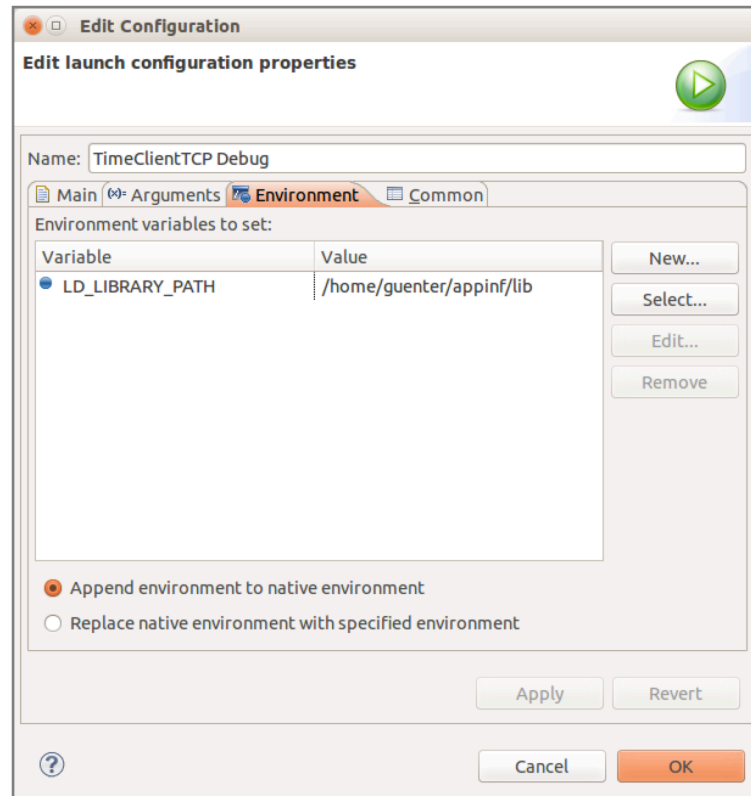
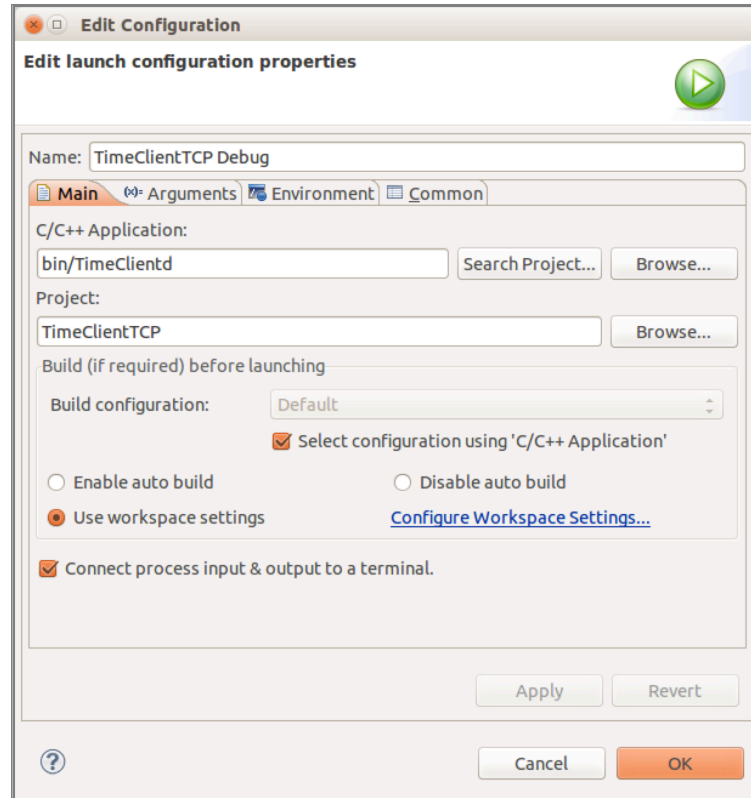


In order to run the project, a new launch configuration must be created. This can be done under Run/Debug settings.



The launch configuration must specify the path to the executable, as well as the shared library search path (`LD_LIBRARY_PATH`).

Note that the executable is located in the *bin* directory within the project directory.



3.4 Sample Notes

3.4.1 XML Samples

A simple XML file for use with the XML samples is located in the `samples/XML/data` directory.

3.4.2 Remoting NG Samples

The samples for Remoting NG come in pairs – there is always a server application and the client application. Be sure to start the server application before the client application. Example (Windows command line):

```
c:\>cd POCopro\samples\RemotingNG
c:\POCopro\samples\RemotingNG>start TimeServer\bin\TimeServer
c:\POCopro\samples\RemotingNG>TimeClient\bin\TimeClient
2012-11-17 10:23:14
```

The **Pizzeria** sample is the server application for the **PizzeriaClient** and **PizzeriaClientWS** client applications.

There is a known issue with the RemoteGenNG code generator and GCC 4.6.3 on Ubuntu 12.04 that will cause the GCC preprocessor to fail when invoked by RemoteGenNG. As a workaround, install GCC 4.4 (apt-get install gcc-4.4) and change the name of the compiler executable in the RemoteGen.xml configuration file from gcc to gcc-4.4 (or create a new compiler definition for GCC 4.4).

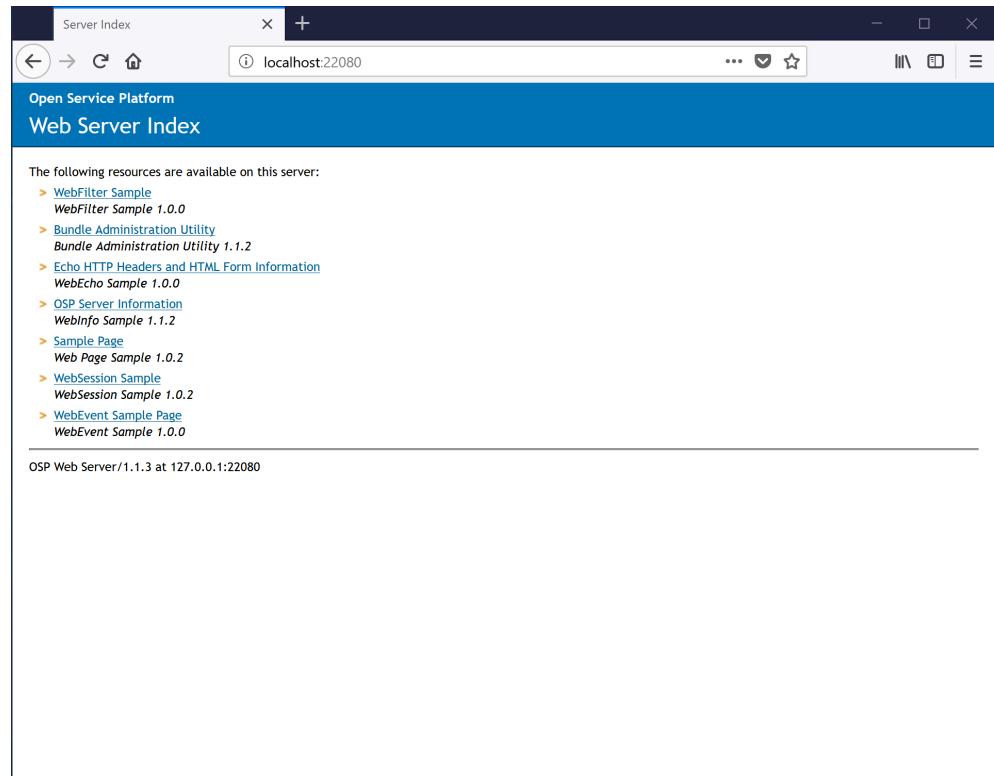
3.4.3 Open Service Platform Samples

The best way to try out the Open Service Platform is to run the **BundleServer** sample. This server application includes a Web server running on port 22080.

Start the **BundleServer** application:

```
c:\>cd POCopro\samples\OSP
c:\POCopro\samples\OSP>BundleServer\bin\BundleServer
osp.core.ServiceRegistry: [Information] Service registered: osp.core.xp
...
Application: [Information] Startup complete.
```

Wait until the “Startup complete” message appears, then direct your favorite Web browser to `http://localhost:22080`.



Click through the various Web pages to try out the different sample bundles. The user name and password for the Bundle Administration Utility is *admin* and *admin*, respectively.

You can also start the OSP Shell client to try out the command line administration facilities of OSP. To do so, start the *ospsh* executable. The user name and password for the shell is *admin* and *admin*, respectively. After starting the shell, enter *help* for a list of commands, or enter the *ls* command to see all installed bundles.

Please be sure to read the Open Service Platform introduction, tutorial and reference pages that are part of the online documentation.

4 Support, Licensing and Contact

If you need any support during the evaluation period, please contact us via email at support@appinf.com.

The Foundation, XML, JSON, Net, Util, Data and Zip libraries are available under the Boost open source license, which makes them free for both open source and commercial use. The complete source code for these libraries can be downloaded for free from <https://pocoproject.org>.

For a complete description of the license regulations and support options for the other libraries, please visit <https://pocoproject.org/pro>.

Applied Informatics can be reached at one of the following addresses:

Applied Informatics Software Engineering GmbH

Maria Elend 143
9182 St. Jakob im Rosental
Austria

Phone: +43 4253 32596

E-Mail: info@appinf.com

Web: <https://pocoproject.org/pro>