

## Abstract

This application note provides an overview of the build environment for the syn1588® PTP Stack source code and is also applicable for the syn1588 API and shared memory API sources.

With release v1.13 the custom Makefile/Visual Studio Project based build environment was replaced by a general CMake build environment. The structure of the source code has not changed in comparison to the setup for v1.12 and older Releases. The compile-time configuration has changed slightly and has been integrated into a standard CMake flow.

This document shall provide help when switching from the old setup to the new CMake setup.

## Directory Structure

The syn1588® PTP Stack repository has the following top level directories and basic configuration files:

`cmake/config/default.cmake:`

Base compile-time configuration of the syn1588® PTP Stack, can be overruled as described below.

`cmake/config/customer.cmake:`

For Linux environments: When this compile-time configuration file is present it provides an additional layer of configurations and will be applied automatically on-top of the default.cmake file.

For Windows environments: Select this as additional compile-time configuration file as described in the next chapter.

`doc:` Documentation of the syn1588® PTP Stack

`resources:`

Example configuration files for run-time operation of the syn1588® PTP Stack. These files were part of the `bin/` directory (which has been removed for the current setup)

`src:`

The source code of the syn1588® PTP Stack itself

## Building the syn1588<sup>®</sup> PTP Stack

The syn1588<sup>®</sup> PTP Stack can be built via a standard CMake buildflow with the following prerequisites:

- CMake, minimum Version: v3.5
- C++ Compiler supporting C++ 2011

You can overrule compile-time settings by providing your own configuration file which can be loaded on-top of the default.cmake and the customer.cmake file.

Note that the same set of compile-time configuration has to be applied also to projects that utilize the shared memory API and syn1588 API.

For example:

Add and use cmake/config/project.cmake similar to the following example for the Linux build flow:

```
> cmake -DPTP_COMPILE_CONFIG=project ..
```

## Building for Windows

### Visual Studio

The CMake build environment supports Visual Studio 2017 and 2019, different versions can be added on-demand and availability.

Please use “File→Open→CMake” in Visual Studio to import the PTP Stack Sources to Visual Studio.

Take a look at the main directory file `CMakeSettings.json`, here you can setup the different build targets and specify these accordingly, e.g., the output directory. You can also add your own compile-time configurations, e.g., by adding them to the `cmake/config/windows.cmake` file or replacing this file with your own configuration file. In this case you will have to modify the entry in the `CMakeSettings.json` file accordingly.

Generated files will be stored in the `out/build/<target-name>` directory by default.

### Other

Generated files will be stored in the `build` directory by default. For example the ptp binary can be found in `build\src\apps\ptp\Debug`, when the build has been run for the Debug target or `build\src\apps\ptp\Release`, when the build has been run for the Release target.

## Building for Linux

Create a “build” sub-directory in the PTP stack main directory.


```
> cd build  
> cmake ..  
> make
```

This will build the PTP Stack for the default compilation options (set via `cmake/config/default.cmake`).

Generated files will be stored in the build directory. For example the ptp binary can be found in `build\src\apps\ptp\` for all build targets, e.g., Debug, Release).

## Further Information

Oregano Systems is always happy to support its customers. In case you have any other questions, please contact us: [support@oregano.at](mailto:support@oregano.at).

 <p><b>Oregano Systems</b> A Meinberg Company</p> <p>Franzosengraben 8 A-1030 Vienna Austria <a href="http://oregano.at">http://oregano.at</a> <a href="mailto:contact@oregano.at">contact@oregano.at</a></p>	<p>Copyright © 2024 Oregano Systems – Design &amp; Consulting GmbH ALL RIGHTS RESERVED.</p> <p>Oregano Systems does not assume any liability arising out of the application or use of any product described or shown herein nor does it convey any license under its patents, copyrights, or any rights of others.</p> <p>Licenses or any other rights such as, but not limited to, patents, utility models, trademarks or tradenames, are neither granted nor conveyed by this document, nor does this document constitute any obligation of the disclosing party to grant or convey such rights to the receiving party.</p> <p>Oregano Systems reserves the right to make changes, at any time without notice, in order to improve reliability, function or design. Oregano Systems will not assume responsibility for the use of any circuitry described herein.</p> <p>All trademarks used in this document are the property of their respective owners.</p>
--	--