intel Eclipse IDE with oneAPI Toolkits User Guide

intel Eclipse IDE with oneAPI Toolkits

Contents

Local Development of Eclipse Projects

The Intel® oneAPI Toolkits support these compilers:

- Intel® oneAPI DPC++ Compiler
- Intel® Fortran Compiler
- Intel® C++ Compiler

If you have not installed an Intel oneAPI Toolkit, **install a toolkit** before proceeding.

If you have not configured your system and built and run a sample project, please refer to the appropriate toolkit Get Started guide and complete those steps:

- **Get Started with the Intel® oneAPI Base Toolkit**
- **Get Started with the Intel® oneAPI HPC Toolkit**
- **Get Started with the Intel® oneAPI IoT Toolkit**

When you have completed those steps, Develop your project with Eclipse.
To develop an Intel oneAPI project on FPGA, see **Intel® oneAPI DPC++ FPGA Workflows on Third-Party IDEs**

Docker Development of Eclipse Projects

The Intel® oneAPI Toolkits support these compilers:

- Intel® oneAPI DPC++ Compiler
- Intel® Fortran Compiler
- Intel® C++ Compiler

If you have not installed an Intel oneAPI Toolkit, **install a toolkit** before proceeding.

If you have not configured your system and built and run a sample project using a Docker Container, please refer to the appropriate toolkit Get Started guide and complete those steps:

- **Get Started with the Intel® oneAPI Base Toolkit**
- **Get Started with the Intel® oneAPI HPC Toolkit**
- **Get Started with the Intel® oneAPI IoT Toolkit**

Containers allow you to set up and configure environments for building, running and profiling oneAPI applications and distribute them using images:

- You can install an image containing an environment pre-configured with all the tools you need, then develop within that environment.
- You can save an environment and use the image to move that environment to another machine without additional setup.
- You can prepare containers with different sets of languages and runtimes, analysis tools, or other tools, as needed.

Singularity Containers

Build a Singularity image using a **Singularity file.**

When you have completed those steps, Develop your project with Eclipse

Remote Development of Eclipse Projects

The Intel® oneAPI Toolkits support these compilers:

- Intel® oneAPI DPC++ Compiler
- Intel® Fortran Compiler
- Intel® C++ Compiler

If you have not installed an Intel oneAPI Toolkit, **install a toolkit** before proceeding.

If you have not run an application on an SSH target, please refer to the appropriate toolkit Get Started guide and complete those steps:

Intel® oneAPI Base Toolkit Intel® oneAPI HPC Toolkit Intel® oneAPI IoT Toolkit	**Build and Run a Sample Project Using Eclipse*** **Run an Application on an SSH Target** **IBuild and Run a Sample Project Using Eclipse*** **Run an Application on an SSH Target** **Build and Run a Sample Project Using Eclipse*** **Run an Application on an SSH Target**

When you have completed those steps, Develop your project with Eclipse.

Developing an Eclipse Project 4

Create a Blank Project

If you do not have the Intel Samples plug-in installed, you can create a blank project in Eclipse:

1. Click File > New > Project. The New Project wizard appears.
2. Expand the C++ folder and select C++ Project and click Next.
3. Add a name for your project.
4. If you wish to change the default location, deselect the Use default location checkbox and specify a new location.
5. In the Project Types area, select Executable > Empty Project.
6. In the Toolchain area, select one of the available toolchains.
7. Click Next.
8. Select one or more of the available configurations.
9. Click Finish.

Import an Existing Project

1. Select File>Import.
2. From the pop up window expand the General option, select Existing Projects into Workspace and click the Next

> button.

3. Click Browse.
4. Find the project, select it and click OK.

Debugging with Eclipse

A Data Parallel C++ program can be debugged simply by right clicking on the Java editor class file from Package explorer.

1. Select Debug As → Data Parallel C++ Application
2. To define a breakpoint in your source code, right-click in the left margin in the Java editor and select Toggle Breakpoint
3. The Debug Perspective will appear. You may use the stepping buttons at the top to review the output.

Notices and Disclaimers

Intel technologies may require enabled hardware, software or service activation.
No product or component can be absolutely secure.
Your costs and results may vary.

Product and Performance Information

Performance varies by use, configuration and other factors. Learn more at www.Intel.com/PerformanceIndex.
Notice revision #20201201

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.
The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.
Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Documents / Resources

Using the Eclipse* IDE with Intel® oneAPI Toolkits	intel Eclipse IDE with oneAPI Toolkits [pdf] User Guide Eclipse IDE with oneAPI Toolkits, oneAPI Toolkits, Toolkits

References

- oneapi-containers/images/singularity at master · intel/oneapi-containers · GitHub
- FPGA Workflows on Third-Party IDEs for Intel® oneAPI Toolkits
- Configure Your CPU or GPU System
- Intel | Data Center Solutions, IoT, and PC Innovation
- Build and Run a Sample Project Using Eclipse*
- Using Containers
- Configure Your System
- SSH: Running Applications Built with Eclipse*
- Run a Sample Project with Eclipse*
- Using Containers
- Get Started with the Intel® oneAPI IoT Toolkit for Linux*
- Configure Your System
- Intel | Data Center Solutions, IoT, and PC Innovation
- Using Containers
- Intel® oneAPI Toolkits
- Overview - 1 | Performance Index