Manuals+ — User Manuals Simplified.

Intel oneAPI DL Framework Developers Toolkit for Linux Owner’s Manual

Home » Intel » Intel oneAPI DL Framework Developers Toolkit for Linux Owner’s Manual

Intel oneAPI DL Framework Developers Toolkit for Linux

Contents

Follow These Steps for the Intel® oneAPI DL Framework Developer Toolkit:

The following instructions assume you have installed the Intel® oneAPI software. Please see the Intel oneAPI Toolkits page for installation options.

1. Configure Your System
2. Build and run a sample project using the Command Line.

Introduction

If you wish to use oneDNN and oneCCL samples, you must install the Intel® oneAPI Base Toolkit. The Base Kit contains all Intel® oneAPI DL Framework Developer Toolkit (DLFD Kit) components with all required dependencies.

If you wish to use the DL DevKit libraries without trying the provided samples, you only need to install the DLFD Kit. Otherwise, install the Intel® oneAPI Base Toolkit.

This toolkit is a suite of development libraries that make it fast and easy to build or optimize a deep learning framework that gets every last ounce of performance out of the newest Intel® processors. This toolkit enables Deep Learning Framework with flexible options including optimal performance on a CPU or GPU.

- Intel® oneAPI Deep Neural Network Library
- Intel® oneAPI Collective Communications Library

Intel® oneAPI Deep Neural Network Library

The Intel® oneAPI Deep Neural Network Library is an open-source performance library for deep learning applications. The library includes basic building blocks for neural networks optimized for Intel® Architecture Processors and Intel® Processor Graphics. This library is intended for deep learning applications and framework developers interested in improving application performance on Intel CPUs and GPUs. Many popular Deep Learning frameworks are integrated with this library.

Intel® oneAPI Collective Communications Library

The Intel® oneAPI Collective Communications Library is a library providing an efficient implementation of communication patterns used in deep learning.

- Built on top of Intel® MPI Library, allows for use of other communication libraries.
- Optimized to drive scalability of communication patterns.
- Works across various interconnects: Intel® Omni-Path Architecture, InfiniBand*, and Ethernet
- Common API to support Deep Learning frameworks (Caffe*, Theano*,Torch*, etc.)
- This package comprises the Intel® MLSL Software Development Kit (SDK) and the Intel® MPI Library Runtime components.

Configure Your System

Intel® oneAPI DL Framework Developer Toolkit
To run samples using the Intel® oneAPI DPC++/C++ Compiler and Intel® Threading Building Blocks, you must install the **Intel® oneAPI Base Toolkit** before configuring your system.

For a complete list of system requirements, see the **Intel® oneAPI Deep Neural Network Library Release Notes**.

To configure your system, you need to:

- Set Environment Variables for CPU/GPU or FPGA
- For GPU users, install GPU drivers
- Disable Hangcheck for applications with long-running GPU compute workloads
- For GPU users, add a user to the video group

Set Environment Variables for CLI Development

For working at a Command Line Interface (CLI), the tools in the oneAPI toolkits are configured via environment variables. Set up your CLI environment by sourcing the setvars script:

Option 1: Source setvars.sh once per session

Source setvars.sh every time you open a new terminal window:
You can find the setvars.sh script in the root folder of your oneAPI installation, which is typically /opt/ intel/oneapi/ for sudo or root users and ~/intel/oneapi/ when installed as a normal user.

For root or sudo installations:
. /opt/intel/oneapi/setvars.sh
For normal user installations:
. ~/intel/oneapi/setvars.sh

Option 2: One time setup for setvars.sh

To have the environment automatically set up for your projects, include the command source /setvars.sh in a startup script where it will be invoked automatically (replace with the path to your oneAPI install location). The default installation locations are /opt/ intel/oneapi/ for sudo or root users and ~/intel/oneapi/ when installed as a normal user.

For example, you can add the source /setvars.sh command to your ~/.bashrc or ~/.bashrc_profile or ~/.profile file. To make the settings permanent for all accounts on your system, create a one-line .sh script in your system's /etc/profile.d folder that sources setvars.sh (for more details, see **Ubuntu documentation on Environment Variables**).

NOTE
The setvars.sh script can be managed using a configuration file, which is especially helpful if you need to initialize specific versions of libraries or the compiler, rather than defaulting to the "latest" version.
For more details, see **Using a Configuration File to Manage Setvars.sh.**. If you need to setup the environment in a non-POSIX shell, see **oneAPI Development Environment Setup** for more configuration options.

For GPU Users, Install GPU Drivers

If you followed the instructions in the Installation Guide to install GPU Drivers, you may skip this step. If you have not installed the drivers, follow the directions in the **Installation Guide**.

GPU: Disable Hangcheck

This section applies only to applications with long-running GPU compute workloads in native environments. It is not recommended for virtualizations or other standard usages of GPU, such as gaming.

A workload that takes more than four seconds for GPU hardware to execute is a long running workload. By default, individual threads that qualify as long-running workloads are considered hung and are terminated.
By disabling the hangcheck timeout period, you can avoid this problem.

NOTE If the system is rebooted, hangcheck is automatically enabled. You must disable hangcheck again after every reboot or follow the directions to disable hangcheck persistently (across multiple reboots).

To disable hangcheck until the next reboot:
sudo sh -c "echo N> /sys/module/i915/parameters/enable_hangcheck"

To disable hangcheck across multiple reboots:

NOTE If the kernel is updated, hangcheck is automatically enabled. Run the procedure below after every kernel update to ensure hangcheck is disabled.

1. Open a terminal.
2. Open the grub file in /etc/default.
3. In the grub file, find the line GRUB_CMDLINE_LINUX_DEFAULT="".

   Enter this text between the quotes (""):

   i915.enable_hangcheck=0
4. Run this command:

   sudo update-grub
5. Reboot the system. Hangcheck remains disabled.

GPU: Add User to Video Group

For GPU compute workloads, non-root (normal) users do not typically have access to the GPU device. Make sure to add your normal user(s) to the video group; otherwise, binaries compiled for the GPU device will fail when executed by a normal user. To fix this problem, add the non-root user to the video group: sudo usermod -a -G video

For the most up-to-date requirements list, see the Intel® oneAPI Collective Communications Library Release Notes.

Run a Sample Project
Run a sample project using the Command Line.

Run a Sample Project Using the Command Line

Intel® oneAPI DL Framework Developer Toolkit

If you wish to use oneDNN and oneCCL samples, you must install the Intel® oneAPI Base Toolkit (BaseKit) . The BaseKit contains all Intel® oneAPI DL Framework Developer Toolkit components with all required dependencies.

After the BaseKit is installed, you can run a sample using the instructions in Build and Run a Intel® oneAPI DL Framework Developer Toolkit Sample Using the Command Line.

Using Containers

Intel® oneAPI DL Framework Developer 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.

Download Docker* Image

You can download a Docker* image from the Containers Repository.

NOTE The Docker image is ~5 GB and can take ~15 minutes to download. It will require 25 GB of disk space.
image=intel/oneapi-dlfdkit
docker pull “$image”

Using Containers with the Command Line

Intel® oneAPI DL Framework Developer Toolkit
Compile and run the containers directly.

The below enables the GPU, if available, using –device=/dev/dri (may not be available in Linux* VM or Windows*).

The command will leave you at a command prompt, inside the container, in interactive mode.

```
image=intel/oneapi-dlfdkit
# –device=/dev/dri enables the gpu (if available). May not be available in Linux VM or Windows docker run –
device=/dev/dri -it “$image”
```

Once in the container, you can interact with it using Run a Sample Project Using the Command Line.

NOTE You may need to include proxy settings before -it “$image”if you are behind a proxy:

```
docker run -e http_proxy=”$http_proxy” -e https_proxy=”$https_proxy” -it “$image”
```

Using Intel® Advisor, Intel® Inspector or VTune™ with Containers

When using these tools, extra capabilities have to be provided to the container:

```
–cap-add=SYS_ADMIN –cap-add=SYS_PTRACE
docker run –cap-add=SYS_ADMIN –cap-add=SYS_PTRACE \
–device=/dev/dri -it “$image”
```

Next Steps

Intel® oneAPI DL Framework Developer Toolkit

After you have built your own project, review **Intel® oneAPI DL Framework Toolkit Code Samples** to understand the capabilities of this toolkit.

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.

Optimization Notice

Intel’s compilers may or may not optimize to the same degree for non-Intel microprocessors for optimizations that are not unique to Intel microprocessors. These optimizations include SSE2, SSE3, and SSSE3 instruction sets and other optimizations. Intel does not guarantee the availability, functionality, or effectiveness of any optimization on microprocessors not manufactured by Intel. Microprocessor dependent optimizations in this product are intended for use with Intel microprocessors. Certain optimizations not specific to Intel microarchitecture are reserved for Intel microprocessors. Please refer to the applicable product User and Reference Guides for more information regarding the specific instruction sets covered by this notice.
Notice revision #20110804

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

Get Started with the Intel® oneAPI DL Framework Developers Toolkit for Linux*	**Intel oneAPI DL Framework Developers Toolkit for Linux** [pdf] Owner's Manual oneAPI DL Framework Developers Toolkit for Linux, Framework Developers Toolkit for Linux, Developers Toolkit for Linux, Toolkit for Linux

References

- **Installation Guides — Intel® software for general purpose GPU capabilities documentation**
- **Intel® oneAPI Base Toolkit: Essential oneAPI Tools & Libraries**
- **GitHub - oneapi-src/oneAPI-samples: Samples for Intel® oneAPI Toolkits**
- **EnvironmentVariables - Community Help Wiki**
- **oneAPI Collective Communications Library Release Notes**
- **Intel® oneAPI Deep Neural Network Library Release Notes**
- **Build and Run a Intel® oneAPI DL Framework Developer Toolkit Sample...**
- **oneAPI Development Environment Setup**
- **Use a Config file for setvars.sh on Linux or macOS**
- **oneAPI: A New Era of Heterogeneous Computing**
- **Intel® oneAPI Base Toolkit: Essential oneAPI Tools & Libraries**