



---

# Getting Started

## Intel® Linux RealSense™ SDK



---

Quick Start.....	3
The Intel® Linux RealSense™ SDK Architecture .....	4
Hardware and Software Requirements and Tools.....	5
Install the SDK .....	6



---

## QUICK START

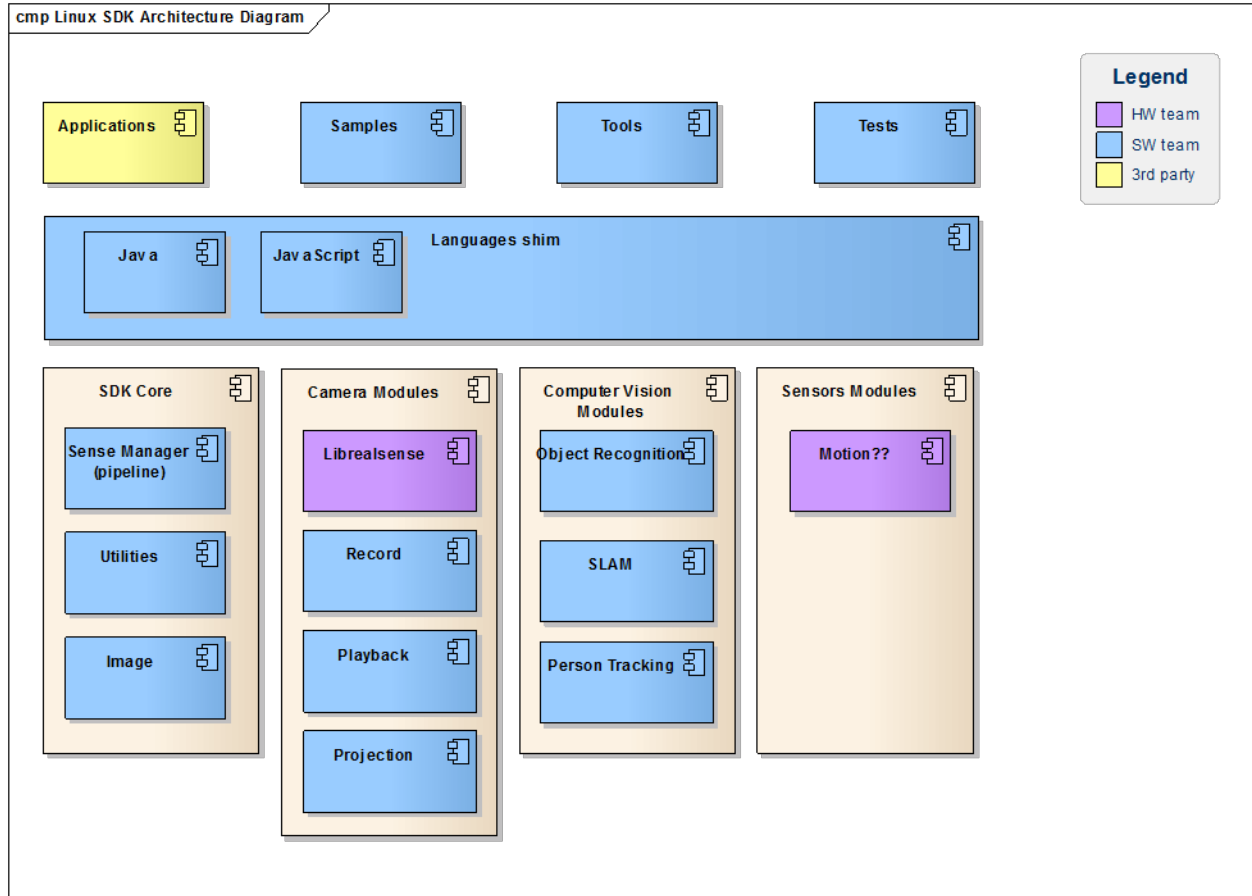
To get started quickly, follow these steps:

Download and build the SDK

Set up your development environment and verify setup by running Hello World Intel RealSense SDK



## THE INTEL® LINUX REALSENSE™ SDK ARCHITECTURE





---

## HARDWARE AND SOFTWARE REQUIREMENTS AND TOOLS

Compatible Devices	Intel® RealSense™ ZR300
Compatible Platforms	<ol style="list-style-type: none"><li>1. Ubuntu 16.04 x64 (GCC 4.9 toolchain)</li><li>2. Ostro Build 102 - Intel® RealSense™ layers for Yocto build can be found at <a href="https://github.com/IntelRealSense/meta-intel-realsense">https://github.com/IntelRealSense/meta-intel-realsense</a></li></ol>
Supported Languages	C++



---

## INSTALL THE SDK

Pre installation - dependency list:

In order to successfully compile and use the SDK, you should install the following list of dependencies:

- librealsense v1.9.3
- log4cxx
- opencv3.1
- cmake
- gtest (googletest)

Installation:

```
git clone https://github.com/IntelRealSense/realsense_sdk
```

```
cd realsense_sdk
```

```
mkdir build
```

```
cd build
```

```
cmake ../
```

```
make
```

```
sudo make install
```

**\*Note:** Running the samples from this installed local folder, will work with the local SDK. Running Samples from other folder will work with the SDK installed on the machine.

## How to enable logging in your application

Prerequisites:

Install log4cxx-

```
sudo apt-get install liblog4cxx10 liblog4cxx10-dev
```

1. log is not compiling by default . In order to compile it, uncomment line 17 in `sdk/src/utilities/logger/CMakeLists.txt`
2. Copy "RSLogs" directory to your home directory.
3. In `~/RSLogs/`, edit `rslog.properties` file to the output logs files you want to create. Root logger is the logger that always exists, but you may add your own logger. Pay attention to the log level hierarchy.
4. Include file "log\_utils.h" to your source/header files.
5. Add "log\_utils" to your link libraries (liblog\_utils.so)
6. Use defines from "log\_utils.h" to log, (File name and line number will be logged automatically) in example: >  
`LOG_DEBUG("This is my demo DEBUG message number %d with string %s\n", 1, "HELLO WORLD");`