



Quick Start Guide

# IMPINJ OCTANE EMBEDDED TOOLKIT

QUICK START GUIDE

## TABLE OF CONTENTS

<b>Overview</b> .....	<b>1</b>
<b>Create a Linux Environment</b> .....	<b>1</b>
Ubuntu 20.04 .....	1
<b>Setting Up a Development Environment</b> .....	<b>1</b>
Download the Impinj Octane ETK .....	1
Extract the Impinj Octane ETK .....	1
<b>Configuring and Compiling a Sample Application</b> .....	<b>2</b>
Documentation .....	2
Compile the Application .....	2
Docker Integration .....	2
<b>Building the Installation Package</b> .....	<b>3</b>
Modify the Example to Meet your Request .....	3
Build CAP File .....	3
<b>Impinj Octane CAP File</b> .....	<b>3</b>
Copy All Files into the CAP Directory .....	4
The CAP Description File .....	4
Example Impinj Octane ETK CAP Description File .....	4
Create a Reader Config File .....	5
Create a CAP File .....	5
Speedway Reader or Gateway .upg CAP file .....	5
Update the CAP File Location .....	6
<b>Deploying and Running the Application</b> .....	<b>6</b>
Deploying the CAP .....	6
Enter OSSHELL .....	6
Stop the Running Application on Impinj Octane .....	6
Running the Application .....	7
<b>Notices</b> .....	<b>8</b>

## OVERVIEW

The Impinj Octane embedded toolKit (Impinj Octane ETK) provide the tools required to develop applications for an Impinj RAIN RFID reader (excluding Impinj R700). The Impinj Octane ETK contain cross-compilation toolchains, third-party and Impinj libraries, and header files. Wherever possible both dynamic and static libraries are provided. It also contains tools for the host (development) machine that facilitate the software development process (e.g. gdb) and for packaging an application to be installed on a reader.

This guide explains how to use the Impinj Octane ETK version that accompanies the Impinj reader firmware version 7.5.x.x.

## CREATE A LINUX ENVIRONMENT

Use of the Impinj Octane ETK requires a GNU/Linux Operating System. This guide uses Ubuntu 20.04.

### Ubuntu 20.04

Using your preferred method (Desktop, Virtual Machine, Vagrant, etc.) create a Linux environment. This guide assumes the following commands have been run successfully against that Ubuntu 20.04 Linux system, at a minimum:

```
[user@machine]$ dpkg --add-architecture i386
[user@machine]$ sudo apt-get update --fix-missing
[user@machine]$ sudo apt-get install -y apt-utils
[user@machine]$ sudo DEBIAN_FRONTEND=noninteractive apt-get install -y build-essential
[user@machine]$ sudo DEBIAN_FRONTEND=noninteractive apt-get install -y cmake
[user@machine]$ sudo DEBIAN_FRONTEND=noninteractive apt-get install -y libc6-i386
[user@machine]$ sudo DEBIAN_FRONTEND=noninteractive apt-get install -y libssl-dev
[user@machine]$ sudo DEBIAN_FRONTEND=noninteractive apt-get install -y zlib1g:i386
```

**Note:** In the above commands, vim is installed to perform file editing. Feel free to install any editor of your choosing.

## SETTING UP A DEVELOPMENT ENVIRONMENT

### Download the Impinj Octane ETK

From the [Impinj Support Portal](#), download the Impinj Octane ETK 7.5.0\_Octane\_Embedded\_Development\_Tools.tar.gz file.

### Extract the Impinj Octane ETK

Navigate to the folder where the tarballs are located. Use the following commands to extract the Impinj Octane ETK

```
[user@machine]$ tar xzfv 7.5.0_Octane_Embedded_Development_Tools.tar.gz
[user@machine]$ cd 7.5.0_Octane_Embedded_Development_Tools
```

Now your environment is ready to compile code to run on the Impinj Octane.

**Note:** The commands in this guide assume they begin in the 7.5.0\_Octane\_Embedded\_Development\_Tools folder.

# CONFIGURING AND COMPILING A SAMPLE APPLICATION

## Documentation

Building the example applications that come with the Impinj Octane ETK can be done by either reading the rest of this manual or reading the examples/README.md file

```
[user@machine]$ cat examples/README.md
```

## Compile the Application

Execute the build.sh script in the folder of the c or c++ (aka cxx) language you want to use.

```
[user@machine]$ rm -f examples/output examples/ltk-c/output examples/ltk-cxx/output
[user@machine]$ cd examples/ltk-c/ && ./build.sh
[user@machine]$ cd examples/ltk-cxx/ && ./build.sh
```

## Docker Integration

Impinj Octane ETK version 7.5 introduces support for the development environment to be run via a Docker container instead of natively on the host machine. You are required to have Docker installed on your system to be able to use this approach. Should you decide to use it, please refer to the instructions in the examples/README.md file or just issue the following commands if you have Docker installed:

```
[user@machine]$ rm -f examples/output examples/ltk-c/output examples/ltk-cxx/output
[user@machine]$ cd examples && ./dockerized.sh /bin/bash
[user@docker_machine]$ cd /etk/examples/ltk-c/ && ./build.sh
[user@docker_machine]$ cd /etk/examples/ltk-cxx/ && ./build.sh
[user@docker_machine]$ cd ../../
```

**Note:** The examples/output folder is being removed prior to building the Docker container to prevent prior compilation errors from appearing.

**Note:** The commands in this guide assume they begin in the /etk folder if Docker is being used.

## BUILDING THE INSTALLATION PACKAGE

Besides using FTP to transfer the files directly to a reader, one of the best ways to copy the cross-compiled application onto Impinj readers is through a package (aka CAP) file. The reader has a “.upg” package file.

### Modify the Example to Meet your Request

Running the sample application may require you to update the NUM\_ANTENNAS value in the main.cpp file. Please edit the main.cpp file and update the NUM\_ANTENNAS variable to the number that the reader you plan to run on supports.

```
[user@machine]$ vi examples/cap/main.cpp
```

### Build CAP File

Impinj Octane ETK now has a folder for creating a sample application and putting it into a cap folder for placement into a cap.upg file. Here is how to create that cap file:

```
[user@machine]$ make -C examples/cap
[user@machine]$ ls -l examples/cap/cap/
total 21652
-rwxr-xr-x 1 root root 22162308 Mar 10 16:15 example
-rwxr-xr-x 1 root root      63 Mar 10 16:15 start
drwxr-xr-x 2 root root   4096 Mar 10 16:15 sys
```

In an upcoming section, there will be instructions for how to build the cap file using one of the example programs.

**Note:** Impinj Octane ETK version 7.5 introduces a start file in the cap folder to automatically start the example program after reboot and continually loop that example if it were to crash or exit:

```
[user@machine]$ cat examples/cap/cap/start
#!/bin/bash

while true; do
    /cust/example
    sleep 1
done
```

Please remove this file prior to creating the cap file if you plan to manually start the application instead.

**Note:** There are instructions below to manually stop the processes on the reader if you leave the start file in-place.

### Impinj Octane CAP File

The Impinj Octane ETK cap file ends in “.upg”

```
[user@machine]$ ls -l examples/cap/cap.upg
-rw-r--r-- 1 root root 4456552 Mar 10 16:37 examples/cap/cap.upg
```

## Copy All Files into the CAP Directory

Copy all the files that you want to show up on the reader (in directory “/cust”) in the “cap” directory. By default, you need to copy the LTK library docsample arm binaries into the “cap” directory. Here is how to copy the first docsample into the cap folder.

```
[user@machine]$ cd examples
[user@machine]$ cp ltk-c/output/target/docsample1/docsample1    cap/cap/docsample1_c
[user@machine]$ cp ltk-cxx/output/target/docsample1/docsample1  cap/cap/docsample1_cxx
[user@machine]$ cd ..
```

## The CAP Description File

With version 7.5 of Impinj Octane ETK, the cap\_description.in file already exists. Using the text editor of your choice, edit the cap\_description.in file to make any appropriate changes for your application.

```
[user@machine]$ vi examples/cap/cap_description.in
```

## Example Impinj Octane ETK CAP Description File

```
# Upgrade Description File
#
# This file contains the settings used by the upgrade generation tool
# to produce an Impinj firmware upgrade file.
[Description]

# Version is a 4 part number in decimal with each part limited to
# 0-255. It is the version of the upgrade file to be generated.
Version = 1.2.3.4

# Valid Reader Hardware is a 3 part number in decimal representing
# the reader model and major/minor revisions on which the upgrade may be
# loaded. Each field may be replaced by the wildcard '*' or '255',
# which matches any.
# Format = aaa.bbb.ccc
#   aaa - Model number
#   bbb - Major revision
#   ccc - Minor revision
Valid Reader Hardware = *.*.*

# File System Layout is an value used by the reader to determine how
# the upgrade partition should be loaded to flash. Currently the only
# supported layout version is 10.
File System Layout = 10
```

```
# Input Directory is the top-level directory of the filesystem to
# create. The files under this directory will be available on the
# reader under /cust after the upgrade is loaded.
Input Directory = ./cap

# Encrypt the upgrade filesystem content. Note this will be
# decrypted when the upgrade file is processed by the reader.
# The default is 'False' (don't encrypt)
#Encrypt = True

# Use the expanded NAND Flash partition sizes available on
# all Revolution based readers and gateways with PCBAs equal or
# greater than v5.00.
# The default is 'False' (build for all PCBAs).
#ExpandedNAND = True
```

## Create a Reader Config File

Optionally, you may create a reader config file to specify the default oshell password to log into the oshell of the reader. With Impinj Octane ETK, the reader.conf file already exists. Using the text editor of your choice, edit the reader.conf file to make any appropriate changes for your application.

```
[user@machine]$ vi examples/cap/reader.conf
```

Using the text editor of your choice, copy the following text into reader.conf:

```
[rshell]
password=developer

[SoftwareFeatures]
StartFTP=yes
```

## Create a CAP File

Impinj Octane ETK already comes with a makefile to build the cap.upg in folder “examples/cap/” as seen previously. The following commands could also be used to generate the cap file using the files beneath folder “examples/cap/cap”.

### Speedway Reader or Gateway .upg CAP file

```
[user@machine]$ cd examples/cap
[user@machine]$ PATH=$PATH:../../arm-toolchain/bin/
[user@machine]$ ../../cap_gen.sh -d cap_description.in -o cap.upg
[user@machine]$ ls -l .
total 14452
```

```

-rw-r--r-- 1 1001 1001    1400 Mar 10 03:18 Makefile
drwxr-xr-x 3 root root    4096 Mar 10 16:40 cap
-rw-r--r-- 1 root root 6553704 Mar 10 16:43 cap.upg
-rw-r--r-- 1 1001 1001    1471 Mar 10 03:18 cap_description.in
-rwxr-xr-x 1 root root 8088420 Mar 10 16:37 example
-rw-r--r-- 1 1001 1001    58837 Mar 10 03:18 main.cpp
-rw-r--r-- 1 root root 328192 Mar 10 16:37 main.o
-rw-r--r-- 1 1001 1001     61 Mar 10 03:18 reader.conf
-rwxr-xr-x 1 1001 1001     63 Mar 10 03:18 start
[user@machine]$ cd ../..

```

## Update the CAP File Location

Copy or move the CAP (.upg) file to a location that can be accessible to the Impinj Octane reader using the firmware upgrade mechanism of your choosing.

## DEPLOYING AND RUNNING THE APPLICATION

### Deploying the CAP

Using a web browser, navigate to your reader's WebUI, <http://impinj-AA-BB-CC> where AA-BB-CC are the last three octets of your reader's MAC address. Select the Browse button and navigate to the location of the CAP (.upg) file. Select the Upgrade button. When the reader says reader for reboot, reboot the reader. After reboot take note of the reader's IP address.

### Enter OSSHELL

Using putty SSH into the reader. When connected to RShell, use the following command to get access to the Linux shell.

```
> osshell developer
```

### Stop the Running Application on Impinj Octane

Before you test manually, it is important to stop the application if the start script is present beneath /cust/start. The following instructions will stop the example application running on the Impinj readers and gateways:

```

root@[reader]:~# cd /cust
root@[reader]:/cust# ps aux | egrep "(start|example)"
root      194  0.0  2.5  3188  1516 ?        S    17:20   0:00 /bin/bash
/cust/start
root      293  0.0  3.8  6464  2268 ?        S    17:21   0:00 /cust/example
root      599  0.0  2.0  3472  1212 pts/0    S+   17:21   0:00 egrep
(start|example)
root@[reader]:/cust# kill 194 293

```

Make sure to identify the correct process numbers to kill both the start and example processes.



## NOTICES

Copyright © 2021, Impinj, Inc. All rights reserved.

Impinj gives no representation or warranty, express or implied, for accuracy or reliability of information in this document. Impinj reserves the right to change its products and services and this information at any time without notice.

EXCEPT AS PROVIDED IN IMPINJ'S TERMS AND CONDITIONS OF SALE (OR AS OTHERWISE AGREED IN A VALID WRITTEN INDIVIDUAL AGREEMENT WITH IMPINJ), IMPINJ ASSUMES NO LIABILITY WHATSOEVER AND IMPINJ DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATED TO SALE AND/OR USE OF IMPINJ PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT.

NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY PATENT, COPYRIGHT, MASKWORK RIGHT, OR OTHER INTELLECTUAL PROPERTY RIGHT IS GRANTED BY THIS DOCUMENT.

Impinj assumes no liability for applications assistance or customer product design. Customers should provide adequate design and operating safeguards to minimize risks.

Impinj products are not designed, warranted or authorized for use in any product or application where a malfunction may reasonably be expected to cause personal injury or death, or property or environmental damage ("hazardous uses"), including but not limited to military applications; life-support systems; aircraft control, navigation or communication; air-traffic management; or in the design, construction, operation, or maintenance of a nuclear facility. Customers must indemnify Impinj against any damages arising out of the use of Impinj products in any hazardous uses.

Statements concerning Impinj product performance are based on Impinj's internal modeling and test data, actual results may vary.

Impinj, and Impinj products and features are trademarks or registered trademarks of Impinj, Inc. For a complete list of Impinj Trademarks, visit [www.impinj.com/trademarks](http://www.impinj.com/trademarks). All other product or service names may be trademarks of their respective companies.

The products referenced in this document may be covered by one or more U.S. patents. See [www.impinj.com/patents](http://www.impinj.com/patents) for details.