

Intel® RealSense™ Product Family

D400 Series Calibration Tools

User Guide

December 2024
Version 2.14.2.0

INTEL CONFIDENTIAL

Copyright (2018 - 2024) Intel Corporation.

This software and the related documents are Intel copyrighted materials, and your use of them is governed by the express license under which they were provided to you ("License"). Unless the License provides otherwise, you may not use, modify, copy, publish, distribute, disclose or transmit this software or the related documents without Intel's prior written permission.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

A "Mission Critical Application" is any application in which failure of the Intel Product could result, directly or indirectly, in personal injury or death. SHOULD YOU PURCHASE OR USE INTEL'S PRODUCTS FOR ANY SUCH MISSION CRITICAL APPLICATION, YOU SHALL INDEMNIFY AND HOLD INTEL AND ITS SUBSIDIARIES, SUBCONTRACTORS AND AFFILIATES, AND THE DIRECTORS, OFFICERS, AND EMPLOYEES OF EACH, HARMLESS AGAINST ALL CLAIMS COSTS, DAMAGES, AND EXPENSES AND REASONABLE ATTORNEYS' FEES ARISING OUT OF, DIRECTLY OR INDIRECTLY, ANY CLAIM OF PRODUCT LIABILITY, PERSONAL INJURY, OR DEATH ARISING IN ANY WAY OUT OF SUCH MISSION CRITICAL APPLICATION, WHETHER OR NOT INTEL OR ITS SUBCONTRACTOR WAS NEGLIGENT IN THE DESIGN, MANUFACTURE, OR WARNING OF THE INTEL PRODUCT OR ANY OF ITS PARTS.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document 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.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

Copies of documents which have an order number and are referenced in this document, or other Intel literature, may be obtained by calling 1-800-548-4725, or go to: <http://www.intel.com/design/literature.htm>.

Code names featured are used internally within Intel to identify products that are in development and not yet publicly announced for release. Customers, licensees and other third parties are not authorized by Intel to use code names in advertising, promotion or marketing of any product or services and any such use of Intel's internal code names is at the sole risk of the user.

Intel and the Intel logo are trademarks of Intel Corporation in the U.S. and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2024, Intel Corporation. All rights reserved.

Contents

1	Introduction	9
1.1	Purpose and Scope of this Document.....	9
1.2	Components	9
1.3	Class 1 Laser and Caution	9
1.4	Hardware Requirements	10
1.5	Software Requirements	11
2	Overview	12
2.1	Calibration Parameters	12
2.2	Types of Dynamic Calibration	12
2.2.1	Target Dynamic Calibration (Depth Scale Calibration)	13
2.2.2	Target-less Dynamic Calibration (Rectification Calibration).....	13
2.3	Depth Quality Tool	13
3	Installation	14
3.1	Installation on Linux (Ubuntu 18.04, Ubuntu 20.04, Ubuntu 22.04, Ubuntu 24.04)	14
3.1.1	Install 3rd-party dependencies	14
3.1.2	Calibration Tools and API Package Installation	15
3.1.3	Checking Package Installation.....	16
3.1.4	Video4Linux backend	16
3.1.5	Test Installation	17
3.1.6	Debian Package Removal	17
3.2	Installation on Windows	17
3.2.1	Install 3rd-party dependencies	18
3.2.2	Package installation	18
4	Calibration with Dynamic Calibrator.....	22
4.1	Dynamic Calibrator Graphical User Interface	22
4.2	Dynamic Calibrator Command Line Interface.....	27
4.3	Connect camera to host system.....	28
4.4	Multiple camera support	28
4.4.1	List all camera devices with Graphical User Interface	29
4.4.2	List all camera devices with Command Line Interface (CLI)	30
4.4.3	Perform calibration on selected device with Command Line Interface (CLI) .	30
4.4.4	Stop Calibration.....	30
4.5	Device USB2 Mode Warning	31
4.6	Targeted Calibration	32
4.6.1	Setup Calibration Target	32
4.6.2	Running Dynamic Calibrator with targeted calibration.....	32
4.6.3	Auto Exposure Adjustment during Targeted Calibration	37
4.6.4	Target calibration phases	39
5	Advanced Calibration Processes	45
5.1	Target-less Calibration.....	45
5.2	Hybrid Calibration	47

	5.2.1	Run Calibration with the Robot	51
6		Handling Calibration Data with CustomRW Tool.....	54
	6.1	Restore Calibration on Device	55
	6.2	Read Calibration from Device.....	55
	6.3	Write Calibration to Device.....	58
	6.4	Read and Write RAW Calibration Tables.....	58
	6.5	Read and Write Custom Fisheye Data	60
	6.6	Read calibration data from device and print extra calibration information.....	61
7		Developing Custom Apps with Calibration API	63
	7.1	API Package Location	63
	7.2	API Package Structure	63
	7.3	Compiling Examples.....	64
	7.4	API Documentation	65
8		Troubleshooting	66
9		Appendix A - Calibration Check	68
	9.1	Quick Check.....	68
	9.2	Accuracy Check	69
10		Appendix B - Calibration Process Highlights.....	70
	10.1	Targeted Calibration	70
11		Appendix C - Intel® RealSense™ Dynamic Calibration Target Setup	77
	11.1	Introduction.....	77
	11.2	Target Image.....	77
	11.3	Printed Target Setup	77
		11.3.1 Print target on letter size pager.....	77
		11.3.2 Check the printed target dimension and accuracy	78
		11.3.3 Setup the printed target on a flat surface	79
	11.4	Perform Dynamic Calibration	80

Tables

Table 1-1. Supported Intel® RealSense™ Depth Modules and Depth Cameras	10
Table 1-2. Software Requirements	11
Table 3-1: Calibration Tools and API Packages	15
Table 4-1: Dynamic Calibrator Command Line Options	27
Table 6-1: CustomRW Command Line Options	54
Table 8-1 Troubleshooting	66

Figures

Figure 1-1. Intel® RealSense™ Depth Camera D435 (Depth Module Integrated)	10
Figure 1-2. Intel® RealSense™ Depth Module D410 with Vision Processor D4 Card (Non-integrated Bare Depth Module)	11
Figure 4-1. Dynamic Calibration Printed Target	32
Figure 4-2 Dynamic Calibrator Initial Start Window	36
Figure 4-3 Dynamic Calibrator app warm up window	37
Figure 4-4 Dynamic Calibrator Targeted Rectification Phase	39
Figure 4-5 Dynamic Calibrator Targeted Rectification Phase (continued)	40
Figure 4-6 Dynamic Calibrator Targeted Rectification Phase (continued)	40
Figure 4-7 Dynamic Calibrator Scale Phase	41
Figure 4-8 Dynamic Calibrator Scale Phase (continued)	41
Figure 5-1 Target-less Dynamic Calibration with Entire FOV	46
Figure 5-2 Target-less Dynamic Calibration with Center FOV	47
Figure 5-3 Hybrid Dynamic Calibration – Target-less Phase	48
Figure 5-4 Hybrid Dynamic Calibration – Targeted Scale Calibration Phase	48
Figure 5-5 Hybrid Dynamic Calibration – Targeted RGB Calibration Phase	49
Figure 5-6 Hybrid Dynamic Calibration – Device and Target Aligned	50
Figure 5-7 Hybrid Dynamic Calibration – Device and Target Aligned	51
Figure 5-8 Dynamic Calibration on Robot	52
Figure 5-9 Dynamic Calibration on Robot	52
Figure 5-10 Dynamic Calibration on Robot	53
Figure 6-1 Restore Calibration on Selected Device	55
Figure 6-2 Read Calibration on Selected Device	56
Figure 6-3 Write Calibration on Selected Device	58
Figure 6-4 Read Calibration RAW Data on Selected Device and Display on Screen	59
Figure 6-5 Read and Write Calibration RAW Data on Selected Device	60
Figure 6-6 Read Fisheye Custom Data on Selected Device and Display on Screen	61
Figure 6-7 Read Calibration Data and Print Extra Calibration Information on Screen	62
Figure 9-1. Calibration Check – Camera in Calibration	68
Figure 9-2. Calibration Check – Camera Out of Calibration	69
Figure 11-1 Target Print Options	78
Figure 11-2 Target Verification	79
Figure 11-3 Printed Target on Flat Surface	80

Revision History

Revision Number	Description	Revision Date
2.5.2.0	First document for Calibration Tool version 2.5.2.0	Jan 2018
2.6.4.0	Support RGB calibration/ Updated kernel patching instruction on Linux	August 2018
2.6.8.0	Support D435i Custom Calibration Data R/W API Support hardware sync in RGB calibration Support calibration raw data read/write Support Windows 10 RS4	December 2018
2.8.0.0	Improved calibration for robotics UI dynamic adjust to image aspect ratios	March 2019
2.11.0.0	Support Intel® RealSense™ Depth Camera D455 New DC algo compatible with production calibration algo v5.1 Support Linux (Ubuntu 16.04 and Ubuntu 18.04) on ARM platforms (Nvidia Jetson TX2, Xavier, and Firefly RK3399) Fix libpng dependency issue on Ubuntu 18.04 Supports Ubuntu 18.04 kernel 5.x Calibration API is combined into Calibration Tool package	July 2020
2.11.0.0	Update installation instruction – change Linux package download server from http://realsense-hw-public.s3.amazonaws.com to https://librealsense.intel.com	April 2021
2.11.1.0	Support D455 Thermal Compensation	April 2021
2.12.0.0	Support Intel® RealSense™ Depth Camera D405 Support Ubuntu 20.04 Detect device USB types and add warnings on USB2 devices Enhancements in custom calibration API and sample app	May 2022
2.13.1.0	Support Intel® RealSense™ Depth Camera D457 Support Ubuntu 22.04	January 2023
2.14.2.0	Support Intel® RealSense™ Depth Camera D421	December 2024

1 Introduction

1.1 Purpose and Scope of this Document

This document is a comprehensive user guide for calibration tools released as part of Intel® RealSense™ Product Family D400 Series Calibration Tools and API software release.

The usage of calibration Tool API is separately covered in document "Intel® RealSense™ Product Family D400 Series Calibration Tools API programmer's guide"

1.2 Components

The Intel® RealSense™ Product Family D400 Series Calibration Tools and API software release package includes:

- Calibration Tools including Dynamic Calibrator
- Calibration API libraries, headers, and sample application
- Intel® RealSense™ Dynamic Target phone application

1.3 Class 1 Laser and Caution

The Intel® RealSense™ Product Family D400 Series is classified as a Class 1 Laser Product under the EN/IEC 60825-1, Edition 3 (2014) internationally and IEC60825-1, Edition 2 (2007) in the US.

This product complies with US FDA performance standards under 21 CFR 1040.10 for laser products except for deviations pursuant to Laser Notice No. 50 dated June 24, 2007.



Caution - Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.



Do not power on the product if any external damage was observed.

There are no service/maintenance, modification, or disassembly procedures for the stereo module and infrared projector. The system integrator must either notify Intel or return modules before any failure analysis is performed.

- Do not attempt to open any portion of this laser product.
- Invisible laser radiation when opened. Avoid direct exposure to beam.
- There are no user serviceable parts with this laser product.
- Modification or service of the stereo module, specifically the infrared projector, may cause the emissions to exceed Class 1.
- No magnifying optical elements, such as eye loupes and magnifiers, are allowed.

Introduction

- Do not try to update camera firmware that is not officially released for specific camera module SKU and revision.

1.4 Hardware Requirements

Table 1-1. Supported Intel® RealSense™ Depth Modules and Depth Cameras

No.	Depth Module/Depth Cameras
1	Intel® RealSense™ Depth Module D400
2	Intel® RealSense™ Depth Module D410
3	Intel® RealSense™ Depth Module D415
4	Intel® RealSense™ Depth Camera D415
5	Intel® RealSense™ Depth Module D420
6	Intel® RealSense™ Depth Module D430
7	Intel® RealSense™ Depth Camera D435
8	Intel® RealSense™ Depth Camera D435i
9	Intel® RealSense™ Depth Camera D455
10	Intel® RealSense™ Depth Camera D405
11	Intel® RealSense™ Depth Camera D435f
12	Intel® RealSense™ Depth Camera D457
13	Intel® RealSense™ Depth Camera D421

The host processor connection to the camera is through USB 3.1 Gen 1.

Figure 1-1. Intel® RealSense™ Depth Camera D435 (Depth Module Integrated)



Introduction

Figure 1-2. Intel® RealSense™ Depth Module D410 with Vision Processor D4 Card (Non-integrated Bare Depth Module)



1.5 Software Requirements

Table 1-2. Software Requirements

Software	Version
Intel® RealSense™ camera firmware	On Linux/Windows platforms 5.10.15.0 or higher on D435i 5.10.6.0 or higher on other devices
Linux*	64-bit host system supporting Ubuntu* 16.04, Ubuntu 18.04, Ubuntu 20.04, Ubuntu 22.04, Ubuntu 24.04 Supports both Intel and ARM platforms. ARM platforms testing is limited to Nvidia Jetson Xavier and Orin.
Linux* Kernel	4.4.0 and later
Windows*	Windows 10 and Windows 11 64-bit

§§

2 Overview

This chapter provides an overview of the calibration parameters and dynamic calibration process.

2.1 Calibration Parameters

Dynamic Calibration is optimizing **extrinsic** parameters, i.e., they refer to calibration done in the field at the user environment with minimal or no user intervention. They are ONLY extrinsic parameters (translation and rotation) of the camera image with regard to the main axis system (the axis between the left and right). Intrinsic parameters, such as distortion, field of view, principal point, are not dynamically calibrated.

Dynamic calibration is run under the assumption that it is the re-calibration of the depth modules/cameras after factory calibration, or at least that the nominal parameters are known.

The left camera is the reference camera and is located at world origin. RGB parameters only apply to depth modules/cameras with RGB sensor for color, e.g., Intel® RealSense™ Depth Camera D415, D435, and D455.

Intrinsic includes

- Focal length - specified as [fx; fy] in pixels for left, right, and RGB cameras
- Principal point - specified as [px; py] in pixels for left, right, and RGB cameras
- Distortion - specified as Brown's distortion model [k1; k2; p1; p2; k3] for left, right, and RGB cameras

Extrinsic includes

- RotationLeftRight - rotation from right camera coordinate system to left camera coordinate system, specified as a 3x3 rotation matrix
- TranslationLeftRight - translation from right camera coordinate system to left camera coordinate system, specified as a 3x1 vector in millimeters
- RotationLeftRGB - rotation from RGB camera coordinate system to left camera coordinate system, specified as a 3x3 rotation matrix
- TranslationLeftRGB - translation from RGB camera coordinate system to left camera coordinate system, specified as a 3x1 vector in millimeters

2.2 Types of Dynamic Calibration

Different types of dynamic calibrations are supported for Intel® RealSense™ Product Family D400 Series

1. **Rectification calibration:** aligning the epipolar line to enable the depth pipeline to work correctly and reduce the holes in the depth image
2. **Depth scale calibration:** aligning the depth frame due to changes in position of the optical elements

Dynamic Calibration Tool API supports these algorithms in two distinctive operating modes: targeted and target-less. The Dynamic Calibrator supports only targeted calibration.

Targeted calibration is the recommended approach because it supports both rectification and depth scale calibrations and will give more accurate results than rectification only calibration done in target-less calibration.

2.2.1 Target Dynamic Calibration (Depth Scale Calibration)

In targeted mode, Dynamic Calibration API supports depth scale calibration and a target is required. The target is predefined and can be displayed on a smartphone through a phone app. A simplified flow is summarized as following:

1. Take the images from L and R camera, including the depth stream (in real-time)
2. Detect the target on the smartphone in both images
3. Similar to target-less calibration, user moves the phone so that it covered most of the image, repeating steps 1-2
4. Once done, user just keeps taking images by positioning the phone anywhere in the image but must move the phone every time
5. After taking 15 images in step 4, the process is complete
6. The process checks for rectification error (absolute Y difference) but also compares the measured pattern size with ground truth

NOTE: The advantage of targeted calibration is that it's accurate and consistent. It calibrates left/right depth as well as RGB (on devices with RGB). The disadvantage is that it requires calibration target which means it cannot be used in cases where calibrating with a target is not feasible.

2.2.2 Target-less Dynamic Calibration (Rectification Calibration)

In target-less mode, Dynamic Calibration API supports rectification calibration without the need of any target. Its basic flow is summarized as following:

1. Take the images from L and R camera (in real-time)
2. Extract features from the images
3. Match the features between L and R camera
4. The image is binned into 6x8 grid. Check whether each bin contains enough corresponding points.
5. The user sees the panel status and moves the device around (bins without features are blue).
6. Steps 1-6 are iterated until all bins have enough feature points
7. Check rectification error (absolute Y difference). If too large, run a solver to optimize extrinsic parameters to minimize it.

NOTE: Note: Targeted calibration is likely to give more accurate results and is therefore the recommended approach.

NOTE: The advantage of target-less calibration is simplicity, no calibration target is needed. It fits with user cases where calibrating with a target may not be feasible. The disadvantage is that it calibrates left/right depth but not RGB and generally less accurate and less consistent than targeted calibration.

2.3 Depth Quality Tool

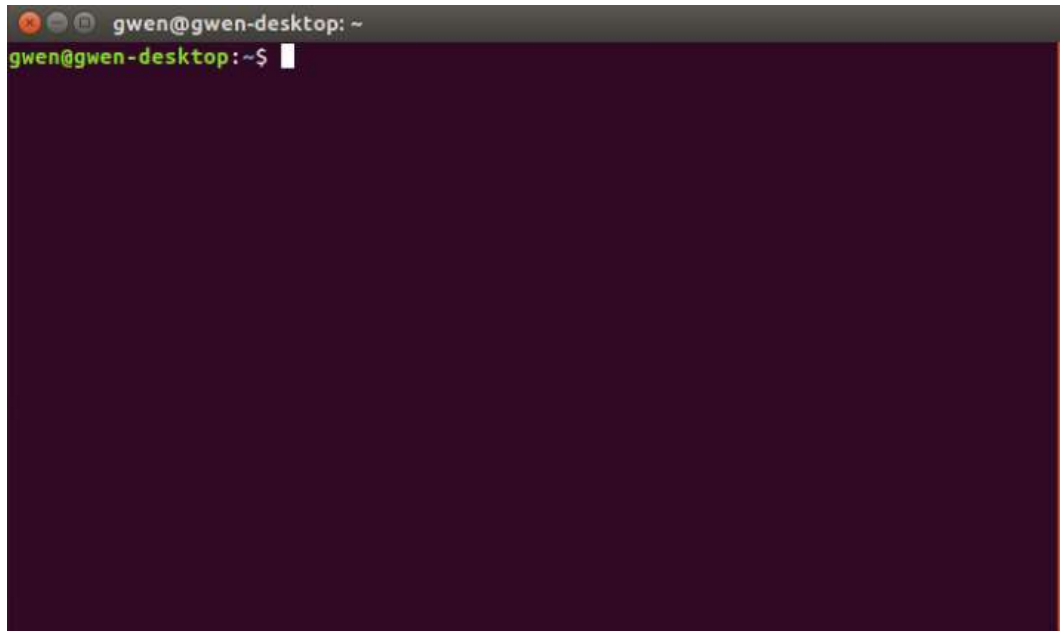
NOTE: The Intel® RealSense™ SDK includes a Depth Quality Tool that can be used to test the quality of the Intel® RealSense™ Product Family D400 Series cameras if it is suspected that the module has gone out of calibration. Refer to the SDK release page at <https://github.com/IntelRealSense/librealsense/releases> to download Depth Quality Tool

3 Installation

The Intel® RealSense™ Dynamic Calibrator is supported on Linux (Ubuntu 18.04, and Ubuntu 20.04), Windows 10 64-bit, and Windows 11 64-bit.

3.1 Installation on Linux (Ubuntu 18.04, Ubuntu 20.04, Ubuntu 22.04, Ubuntu 24.04)

To open a terminal window, hit (Ctrl + Alt + T).



A working Internet connection is required for installing the 3rd party dependent libraries and patching the kernel. For systems behind a firewall or proxy server, proxy will also need to be configured. Please consult with your network administrator for details.

3.1.1 Install 3rd-party dependencies

1. Ensure apt-get is up to date:

```
sudo apt-get update
```

2. Install libusb-1.0:

```
sudo apt-get install libusb-dev libusb-1.0-0-dev
```

3. Install libglfw:

Installation

```
sudo apt-get install libglfw3 libglfw3-dev
```

4. Install freeglut:

```
sudo apt-get install freeglut3 freeglut3-dev
```

3.1.2 Calibration Tools and API Package Installation

Calibration Tool API provides installation package in dpkg format for Debian OS and its derivatives. The package and its respective content are listed below:

Table 3-1: Calibration Tools and API Packages

Name	Content
librscalibrationtool	Intel® RealSense™ Dynamic Calibrator, Calibration API and example sources and other related tools

The package is generally available for installation through Intel server. If access to Intel server is not permitted in specific use case, the Debian package can be obtained and installed locally.

3.1.2.1 Debian Package Installation through Intel Server

The Debian packages are available on Intel server and can be installed with the following steps.

- Add Intel server to the list of repositories:

```
sudo add-apt-repository "deb https://librealsense.intel.com/Debian/apt-repo $(lsb_release -cs) main" -u
```

- Register the server's public key:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-key  
F6E65AC044F831AC80A06380C8B3A55A6F3EFCDE || sudo apt-key adv --keyserver  
hkp://keyserver.ubuntu.com:80 --recv-key F6E65AC044F831AC80A06380C8B3A55A6F3EFCDE
```

- Refresh the list of repositories and packages available:

```
sudo apt-get update
```

- Install the librscalibrationtool package which includes Intel® RealSense™ Dynamic Calibrator:

```
sudo apt-get install librscalibrationtool
```

3.1.2.2 Debian Package Installation through Local Files

The version 2.14.2.0 package is also available in local .deb Debian package file and can be installed with the following steps.

Installation

- Install the librscalibrationtool package which includes Intel® RealSense™ Dynamic Calibrator:

sudo dpkg -i librscalibrationtool_2.14.2.0_<platform>.deb, for example,

```
sudo dpkg -i librscalibrationtool_2.14.2.0_amd64.deb
```

3.1.3 Checking Package Installation

After the debian package install finishes, check for files installed. Under debian convention, the executables are installed under /usr/bin, library files under /usr/lib, and other files including sample code under /usr/share/doc/librscalibrationapi and /opt/intel/librscalibrationapi/.

For example, for Calibration Tool, librscalibrationtool, the files are installed as below:

```
sudo dpkg -L librscalibrationtool
```

```
/usr
|.....bin
| |..... Intel.Realsense.DynamicCalibrator
| |..... Intel.Realsense.CustomRW
|
|----- share
| |----- doc
| | |..... librscalibrationtool
| | | |----- attributions.txt
| | | |----- readme.txt
| | | |----- changelog
| | | |----- copyright
| | | |----- License.txt
| | | |----- README.md
| | | |
| | | |----- target
| | | | |----- print-target-fixed-width.pdf
| | | |
| | | |----- api
| | | | |----- DynamicCalibrationAPI-Linux-2.14.2.0.tar.gz
```

Notes:

- Intel.Realsense.DynamicCalibrator is the main calibrator app
- Intel.Realsense.CustomRW is the custom calibration data read/write tool
- DynamicCalibrationAPI-Linux-2.14.2.0.tar.gz is the Calibration API and examples package, advanced users can develop custom calibration apps with this API, see chapter **Developing Custom Apps with Calibration API** and **Dynamic Calibration Programmer's Guide** for details.

3.1.4 Video4Linux backend

1. Ensure no cameras are presently plugged into the system.
2. Patch and insert modified kernel drivers for Intel® RealSense™ Camera.

Intel® RealSense™ SDK 2.0 provides dpkg package librealsense2-dkms for the kernel rules and kernel drivers. Please follow instructions in the following link for Ubuntu OS version supported and package installation details:

Installation

https://github.com/IntelRealSense/librealsense/blob/development/doc/distribution_linux.md

Please note generally on supported platforms, only the Realsense DKMS kernel driver package librealsense2-dkms is required for Dynamic Calibration. The rules are included in librealsense2-udev-rules which librealsense2-dkms depends on and will be installed automatically.

```
sudo apt-get install librealsense2-dkms
```

3.1.5 Test Installation

Now check dynamic calibration tool version by running the following command:

The calibration tool executables are located under `/usr/bin`.

For example,

To print the tool version to screen:

```
/usr/bin/Intel.Realsense.DynamicCalibrator -v
```

To print device information to screen, connect a camera device to the system and execute the following command:

```
/usr/bin/Intel.Realsense.DynamicCalibrator -list
```

To print device calibration data to screen, execute the following command:

```
/usr/bin/Intel.Realsense.CustomRW -r
```

To try out calibrator, execute the following command:

```
/usr/bin/Intel.Realsense.DynamicCalibrator
```

If you see the correct version information, device information and calibration data printed on screen, Dynamic Calibrator launched with correct device listed, congratulations you have the right setup and ready to use the calibration tools.

3.1.6 Debian Package Removal

For any reason, if you would like to uninstall the Calibration Tools and API, the Debian packages can be removed by running dpkg commands.

For example, Calibration Tool, librscaleibrationtool, can be removed as below:

```
sudo dpkg -r librscaleibrationtool
```

3.2 Installation on Windows

The Intel® RealSense™ Dynamic Calibration Tool runs on Windows 10 or Windows 11.

Installation

3.2.1 Install 3rd-party dependencies

Windows dynamic calibration package requires the following installed on the host system:
Microsoft Visual C++ 2015 Redistributable.

The install package installs this library automatically. So normally user will not need to install this separately.

3.2.2 Package installation

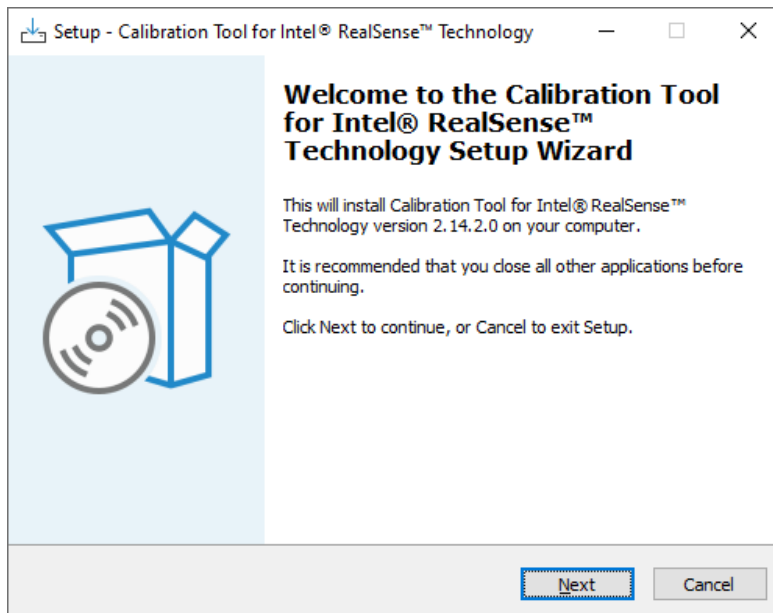
The release includes the following install package for Windows:

- Calibration Tool and API
CalibrationTool-2.14.2.0-Setup.exe

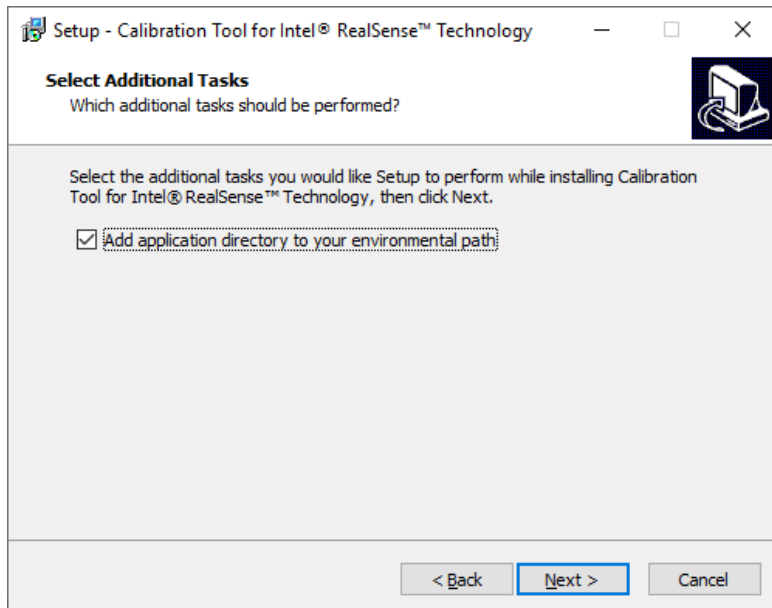
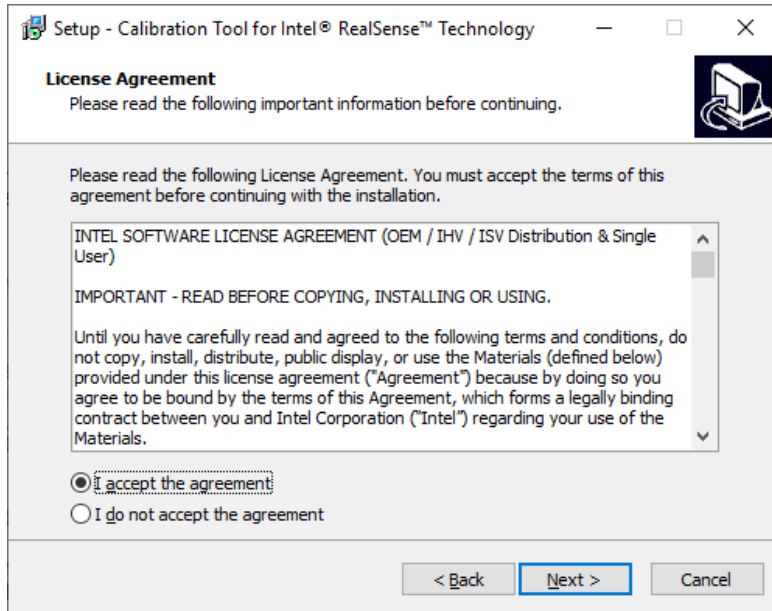
3.2.2.1 Calibration Tool Windows Installer Package

The installer package compresses all contents into a single installer file, for example, CalibrationTool-2.14.2.0-Setup.exe.

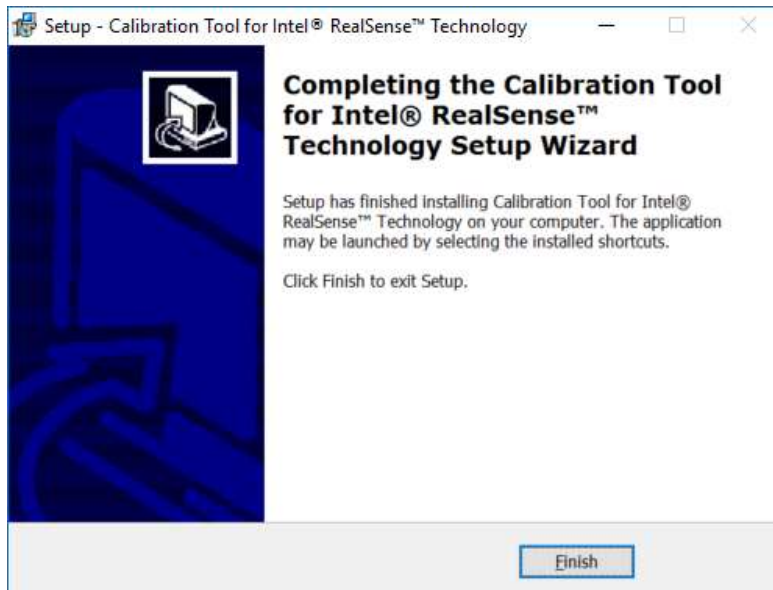
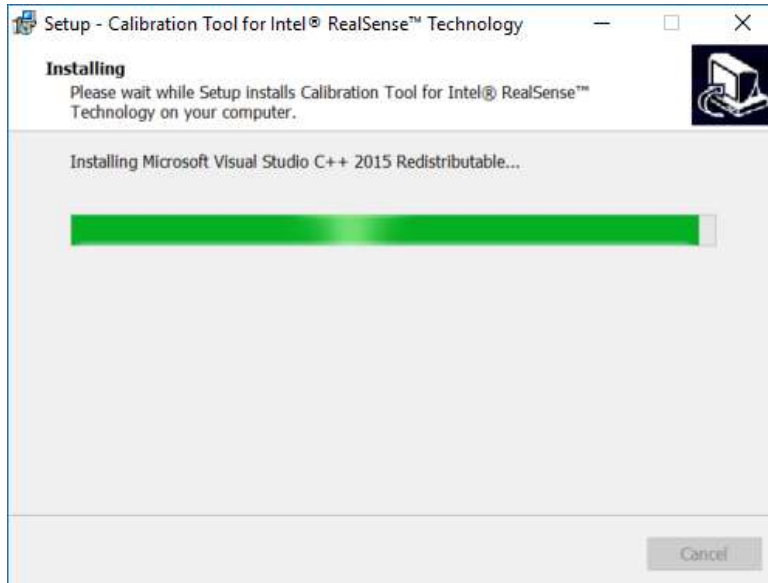
- Right click on CalibrationTool-2.14.2.0-Setup.exe and choose "Run as administrator"
- User Account Control Window pops up, click the "Yes" button
- The install process should start. By default, it installs under C:\Program Files\Intel\CalibrationToolAPI\2.14.2.0.



Installation



Installation



The tools are installed under C:\Program Files\Intel with following directory structure:
CalibrationToolAPI

```

|----- 2.14.2.0
|       |----- bin
|       |       |----- Intel.Realsense.DynamicCalibrator.exe
|       |       |----- Intel.Realsense.DynamicCalibratorCLI.exe
|       |       |----- Intel.Realsense.CustomRW.exe
|       |
|       |----- api
|       |----- DynamicCalibrationAPI.Zip
  
```

Installation

```

|      |
|      |----- attributions.txt
|      |----- license.txt
|      |----- readme.txt
|      |----- target

```

The installed directory contains two versions of the Dynamic Calibrator tool:

- **Intel.Realsense.DynamicCalibrator.exe** with an interactive graphical user interface. It supports targeted calibration and is the primary tool regular users should use to calibrate the devices. This is the tool linked to the Dynamic Calibrator icon on the desktop.
- **Intel.Realsense.DynamicCalibratorCLI.exe** with a command line interface with advanced command line options. It supports the targeted calibration process similarly to the GUI version, in addition, it also supports many other features that advanced users may prefer to use to calibrate their devices or experiment new calibration methods.
- **Intel.Realsense.CustomRW.exe** allows user to dump the calibration data from device, write calibration parameters to device, and reset the device calibration to default gold settings.
- **DynamicCalibrationAPI.Zip** is the Calibration API and examples package, advanced users can develop custom calibration apps with this API, see chapter *Developing Custom Apps with Calibration API* and *Dynamic Calibration Programmer's Guide* for details.

The installer will create Dynamic Calibrator icon link to Intel.Realsense.DynamicCalibrator.exe at the system desktop.



Double click the icon to start using the tool with its graphical user interface.

Of course, if command line interface preferred, users can execute the programs anywhere since installation location is added to system PATH during installation. User can also always navigate to the install directory using a windows terminal.

For example,

```

cd C:\Program Files\Intel\CalibrationToolAPI\2.14.2.0
cd bin

```

4 Calibration with Dynamic Calibrator

Your camera might be out of calibration if you see the following symptoms:

- Reduced depth density on objects in the operating range (might still get depth on vertical lines).
- Flat surfaces look "wobbly", i.e., there is more deviation from the flatness than usual.
- Measuring the physical distances to objects are not within about 3% of what they should be.

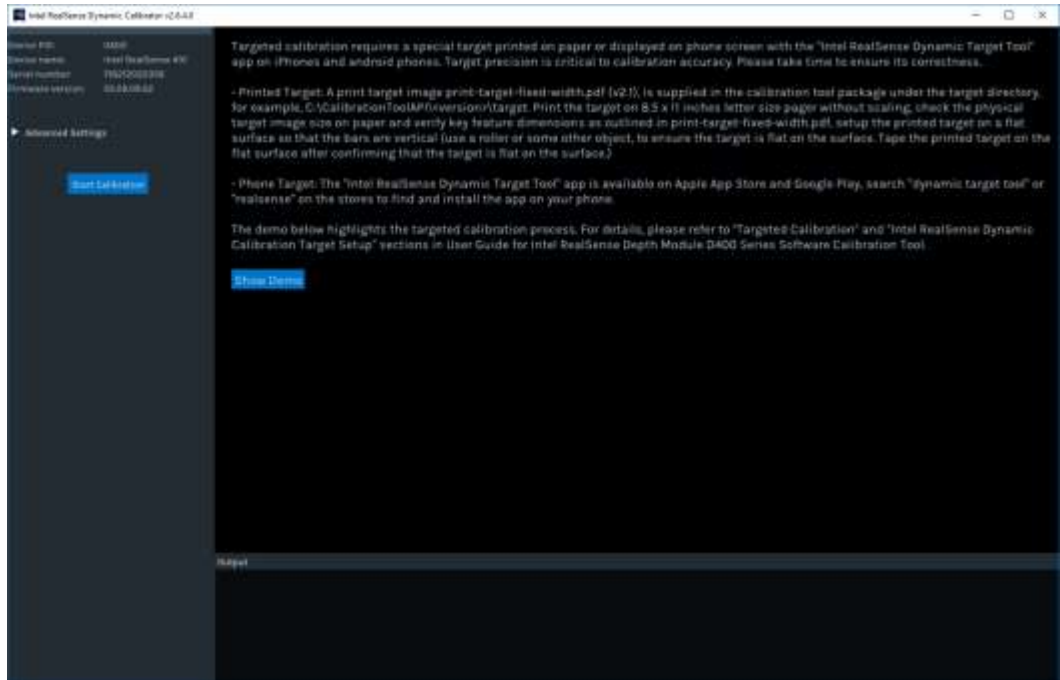
Once you determined the camera is out of calibration, the camera needs to be re-calibrated to correct the changes that might have occurred between the left and right depth sensors.

Dynamic Calibrator is under the bin directory. To support both regular users who prefer graphical user interface (GUI) and OEM/ODMs who prefer command line interface (CLI) for ease of automation, the tool supports both GUI and CLI interfaces.

- On **Linux**, a single executable Intel.Realsense.DynamicCalibrator supports both GUI and CLI interfaces.
- On **Windows**, two separate executables, both share same functionality but different user interfaces, are provided:
 - ✓ Intel.Realsense.DynamicCalibrator.exe with a graphical user interface
 - ✓ Intel.Realsense.DynamicCalibratorCLI.exe with a CLI with command line options

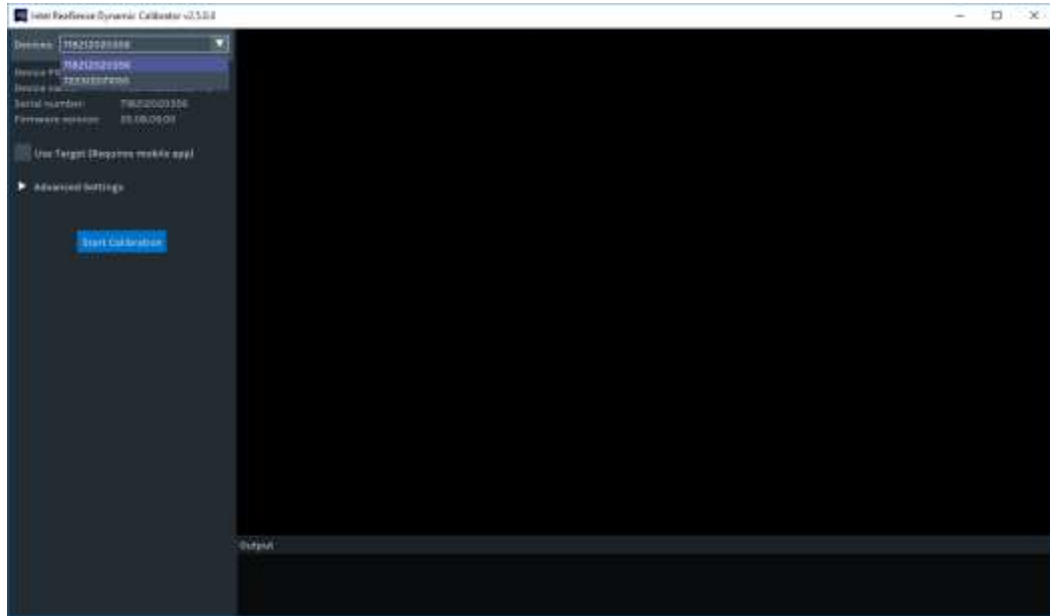
4.1 Dynamic Calibrator Graphical User Interface

The Graphical User Interface (GUI) is provided for easier use. The app detects all Intel® RealSense™ Product Family D400 Series cameras on the system and lists them in a dropdown box on the top left corner.

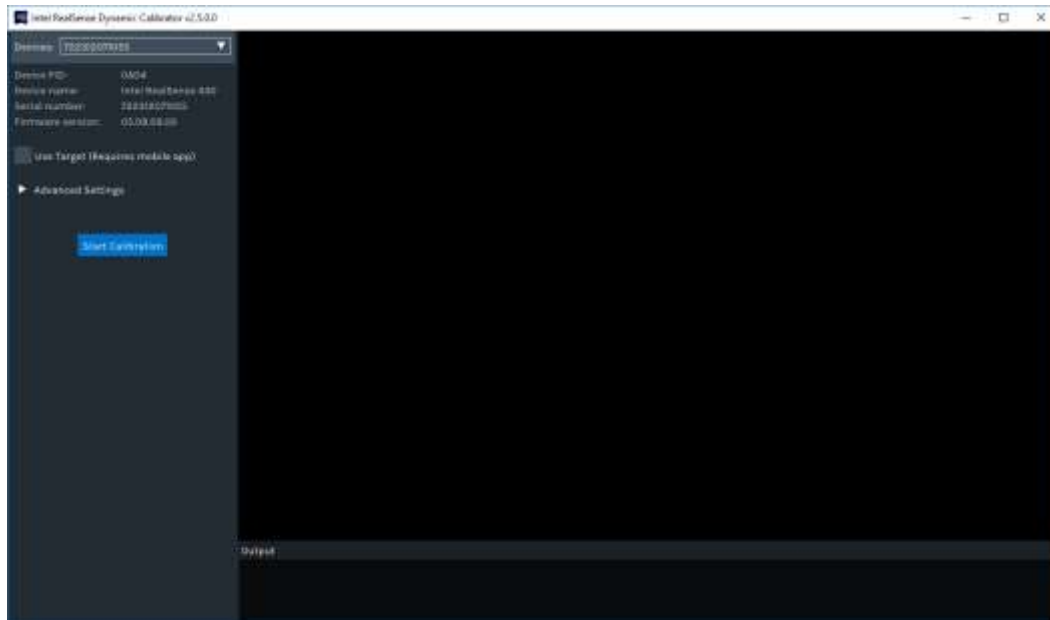


Calibration with Dynamic Calibrator

User selects one of the devices from the list to calibrate:



Details on the device including product id, device name, serial number, and FW version displayed:



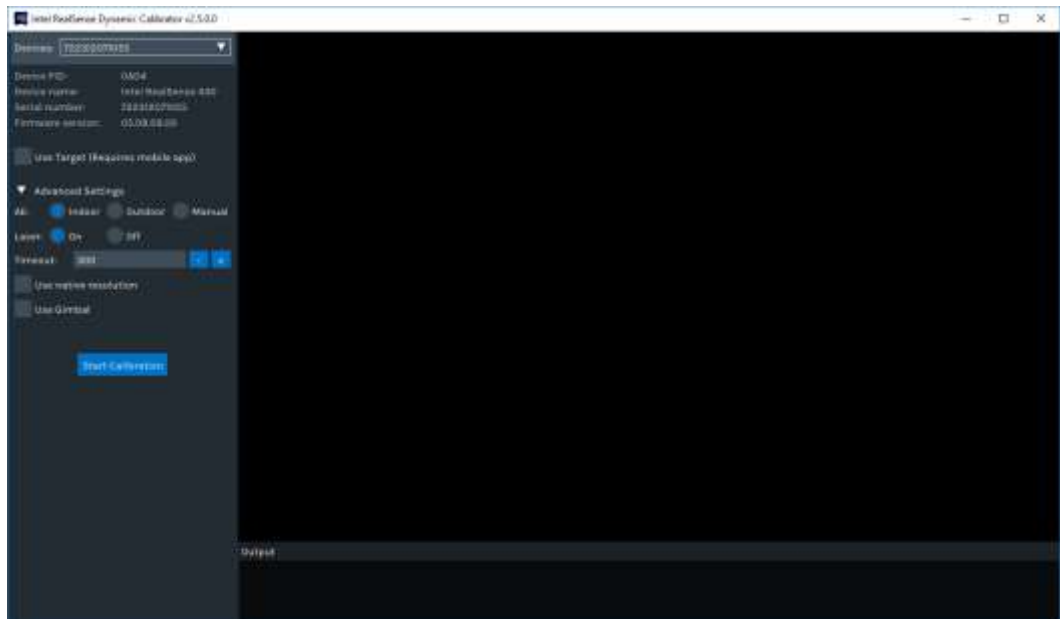
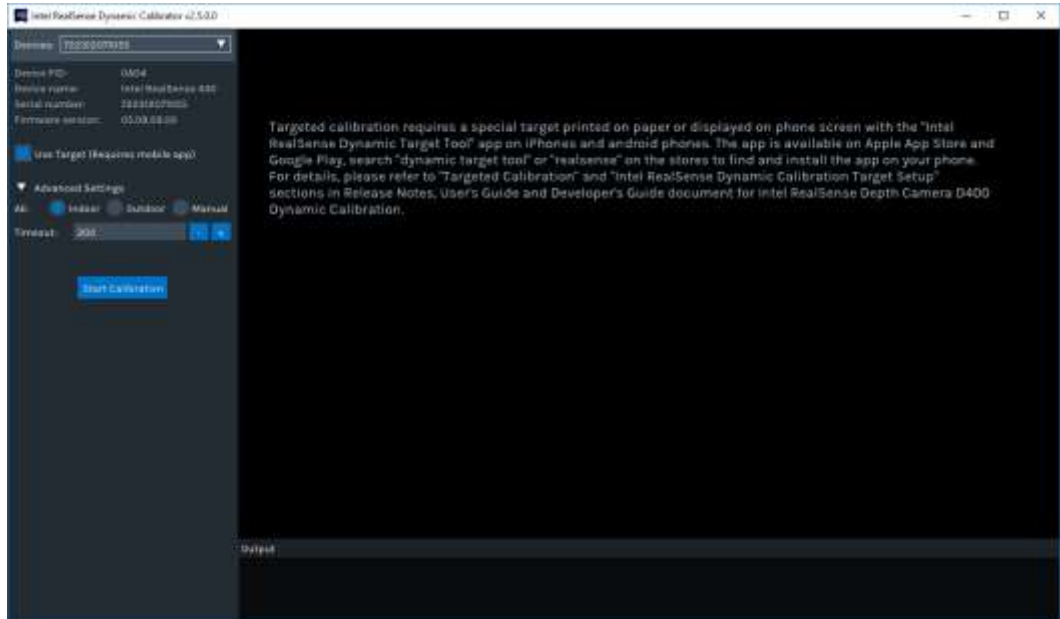
The calibration process requires a special target, either printed or phone target. Please refer to Appendix for target setup details.

By default, the app handles everything automatically. However, a few advanced settings are provided for experienced users who prefer to make custom choices.

For targeted calibration, user can customize AE (auto exposure) setpoint. **Indoor** and **Outdoor** are auto options that let the app to handle it automatically. The app will adjust automatically to the environment light conditions so that calibration target can be found without over exposure. The Manual option allow user to set

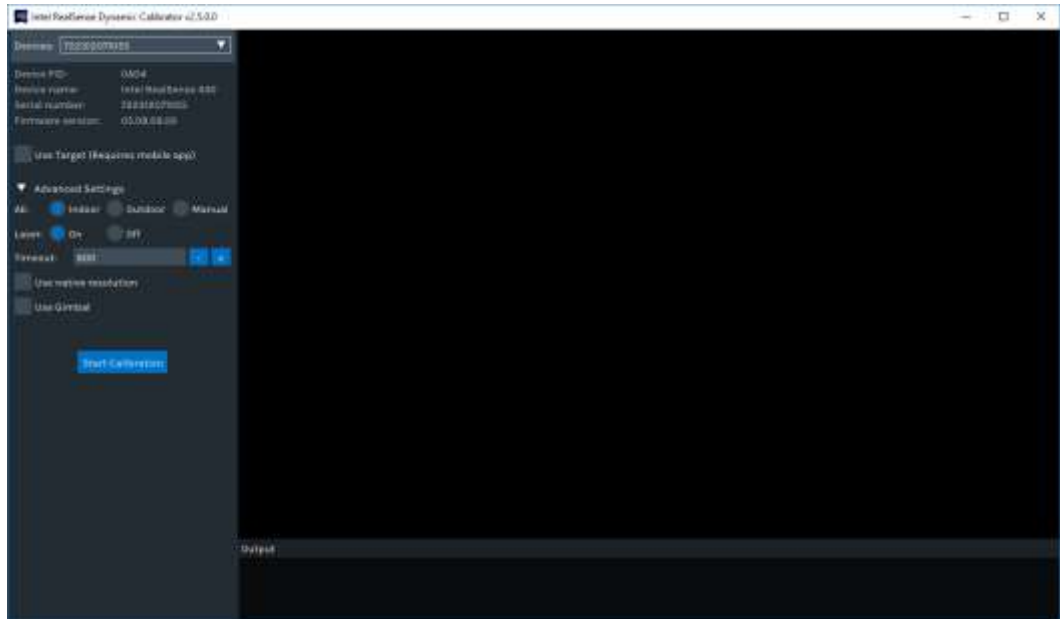
Calibration with Dynamic Calibrator

a fixed setpoint value. This is useful in very complex environment where the app may experience difficulty to sweep through the setpoints automatically. The value set in such cases is completely dependent on the environment lighting, for example, 1000 - 2000 should be sufficient for indoor usage and 3000 – 4000 might work better outdoor. The exact value in user case can be found through trial and error.

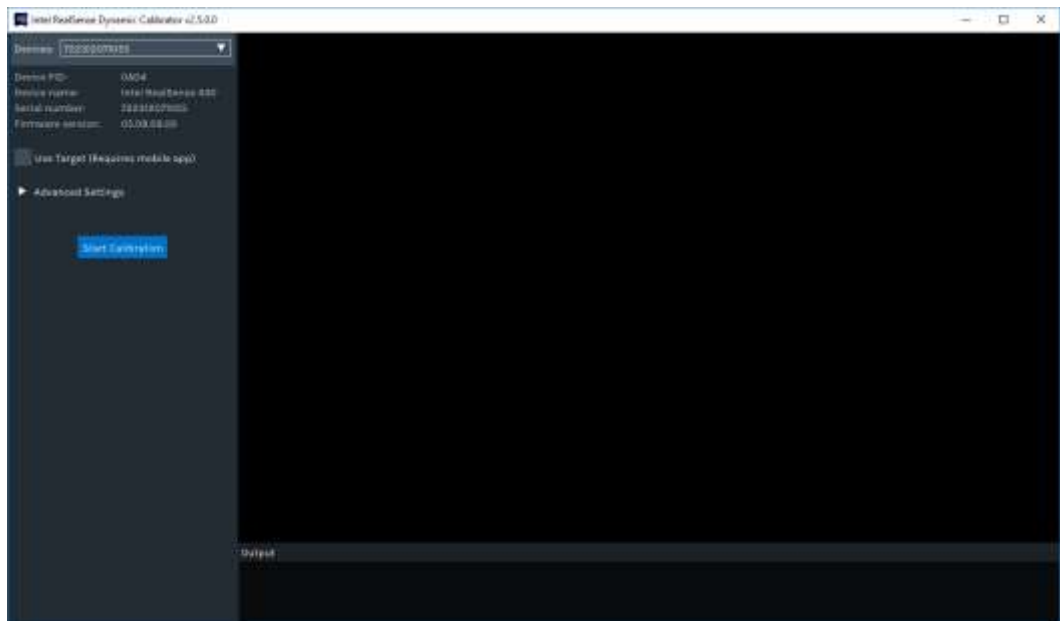


User can also adjust the timeout period. User can adjust it in case an extended period is required in difficult situations:

Calibration with Dynamic Calibrator

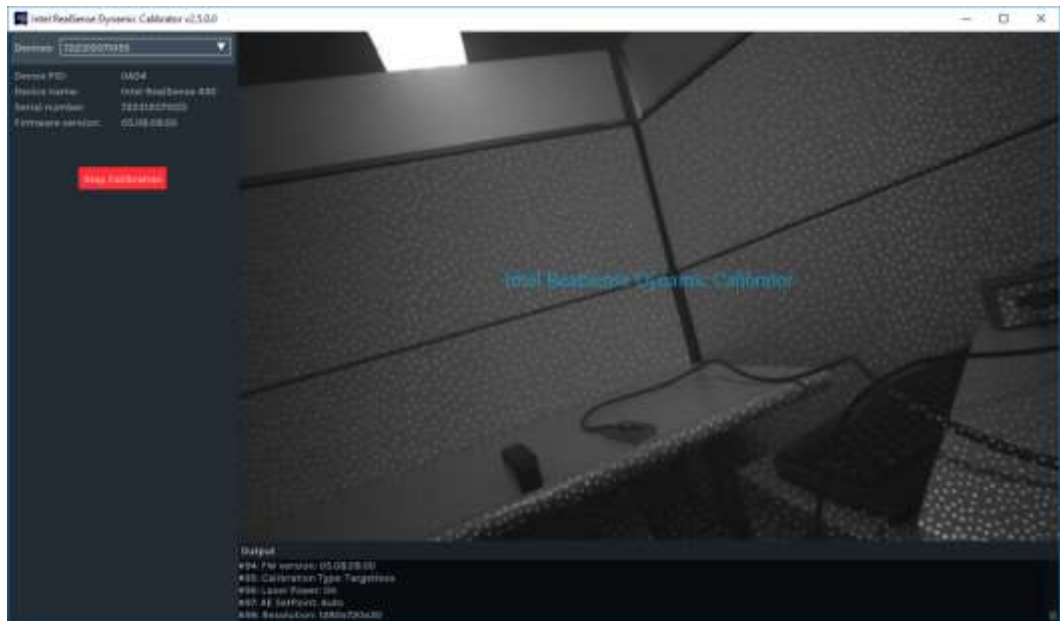


The last step is to start calibration by clicking the “Start Calibration” button:



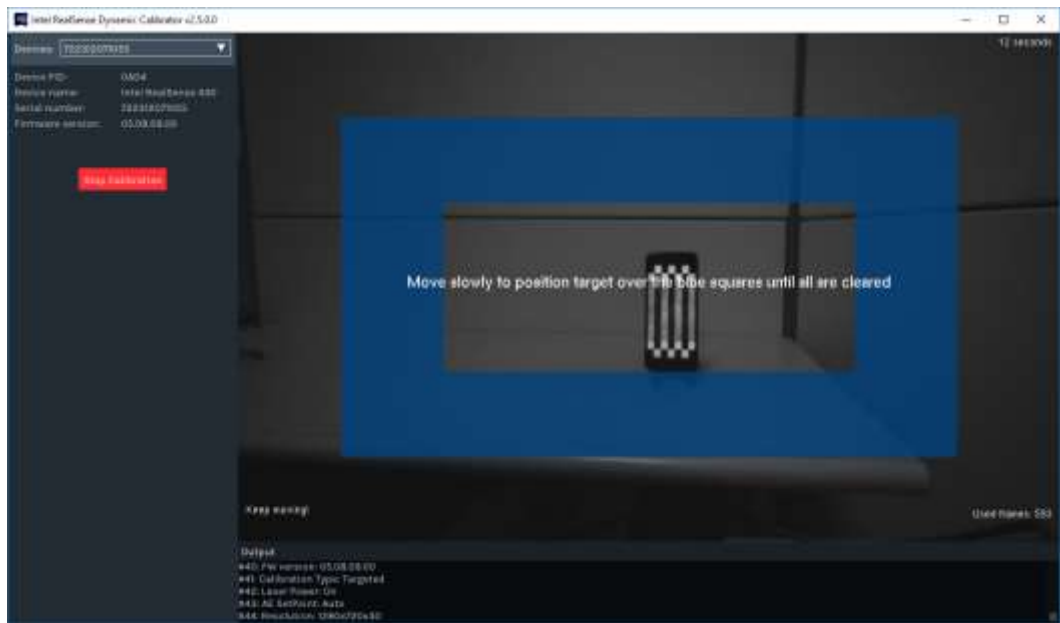
User can always stop/cancel calibration and return to main interface at any time by clicking the “Stop Calibration” button”:

Calibration with Dynamic Calibrator



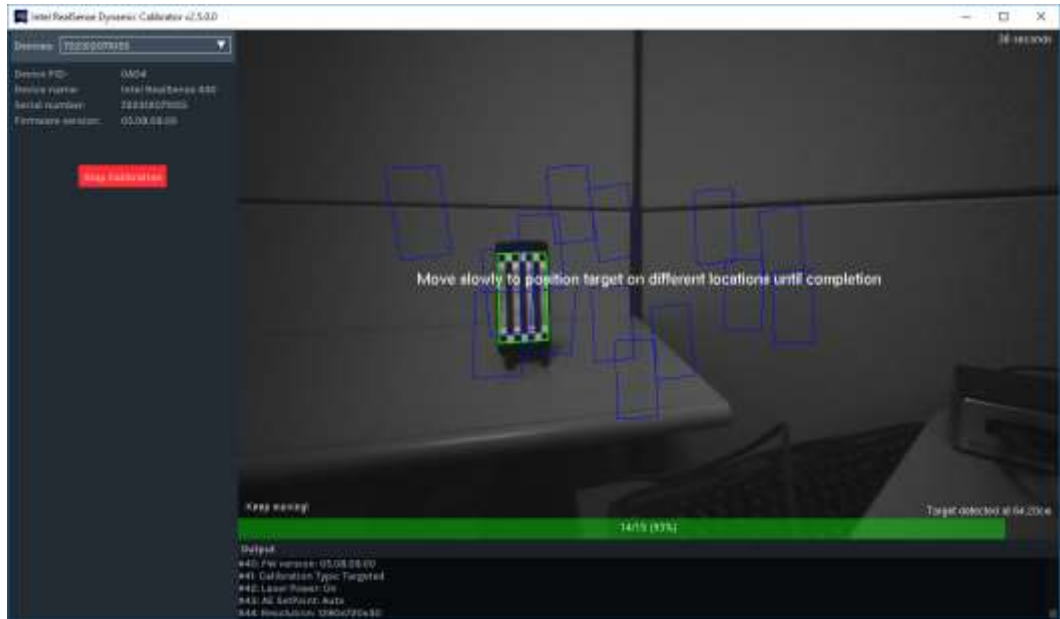
During calibration, the UI displays the elapsed time in the top right corner, and status feedback on the lower left corner, and either the number of frames or target distance on the lower right corner:

Rectification Phase:



Scale Phase:

Calibration with Dynamic Calibrator



4.2 Dynamic Calibrator Command Line Interface

A number of command line options are supported in Dynamic Calibrator Command Line Interface (CLI):

- `Intel.Realsense.DynamicCalibratorCLI.exe` on Windows
- `Intel.Realsense.DynamicCalibrator -cli` on Linux

Table 4-1: Dynamic Calibrator Command Line Options

Option	Description
-help or -?	Display list of command line options.
-version -v	Show Calibration Tool API version info
-cli	Run app in CLI mode
-show or -list	Display list of connected cameras with device name, serial number and firmware version information.
-sn <camera serial number> -serial < camera serial number>	If multiple cameras connected to the current system, choose one of the camera devices to calibrate by specifying its serial number.
-mode <0, 1, 2, 3, or 4> -m <0, 1, 2, 3, or 4>	select calibration mode: 0 for target-less calibration 1 for targeted calibration 2 for hybrid calibration 3 for scale calibration only (for debugging) 4 for rgb calibration only (for debugging)

Calibration with Dynamic Calibrator

-dump <0 or 1> -d <0 or 1>	For debugging use. Dump calibration images and other data. 0 to dump only frames used in calibration and results to files. 1 to dump every frame received.
-setpoint <0 to 4095> -s <0 to 4095>	Force AE setpoint (reference point). For dark environment, choose a low setpoint, 400, for example. For normal office lighting, a setpoint around 800 is appropriate. For outdoor bright sunlight environment, a higher setpoint is required, for example, 1200 or higher.
-timeout -t <Time out duration in seconds>	Customize time out duration. By default, calibration session will time out in 180 seconds for target-less and 300 seconds for targeted calibration.
-error -e	Report rectification (vertical alignment) error in pixels periodically until timeout
-a -aligned	device and target aligned in orientation, for example, device is mounted vertically, and target is positioned vertically
-skiprgb	Skip RGB calibration. This is useful if device like D435 and D415 equipped with a RGB camera but user does not use it. Skip RGB phase during calibration process will not impact depth calibration and it makes the process much faster.
-ignore-borders	ignore the border blocks during target-less rectification
-max-images <more than 6>	Specify number of images to capture in scale calibration and rgb calibration phases. This is used when user usage is limited by time or space and would like to experiment best settings for their usage. <ul style="list-style-type: none"> • In targeted calibration, the default is 16 images. • In hybrid calibration, the default is either 6 or 8 images depends on device/target orientation. • For optimal results, recommend at least 6 images.
-force	Force the app to accept specified parameters, even if it's outside of the recommended range. Currently this only applies to -max-images.
-verbose	Display more detailed messages to assist support and debug

4.3 Connect camera to host system

The Intel® RealSense™ camera to be calibrated should be connected to the system where dynamic calibrator software will run. This is usually done through a USB cable (for peripherals). On systems with camera integrated, it's already connected, so no extra step is required.

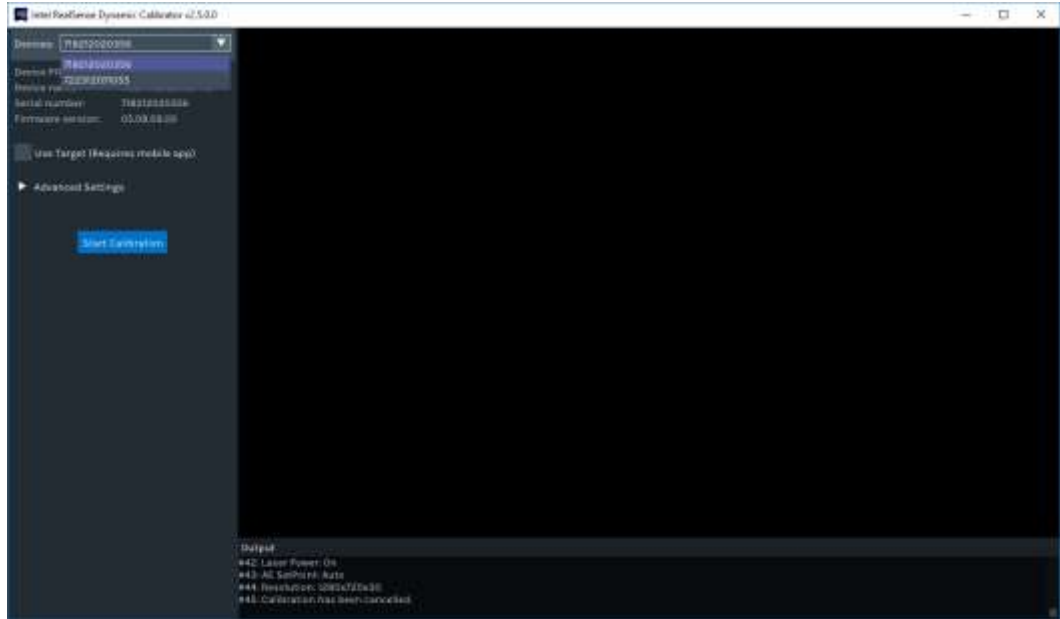
4.4 Multiple camera support

If multiple cameras connected to the current system, dynamic calibrator allows user to list all camera devices on the system and choose one of the camera devices to calibrate. Only a single camera device can be calibrated at a time.

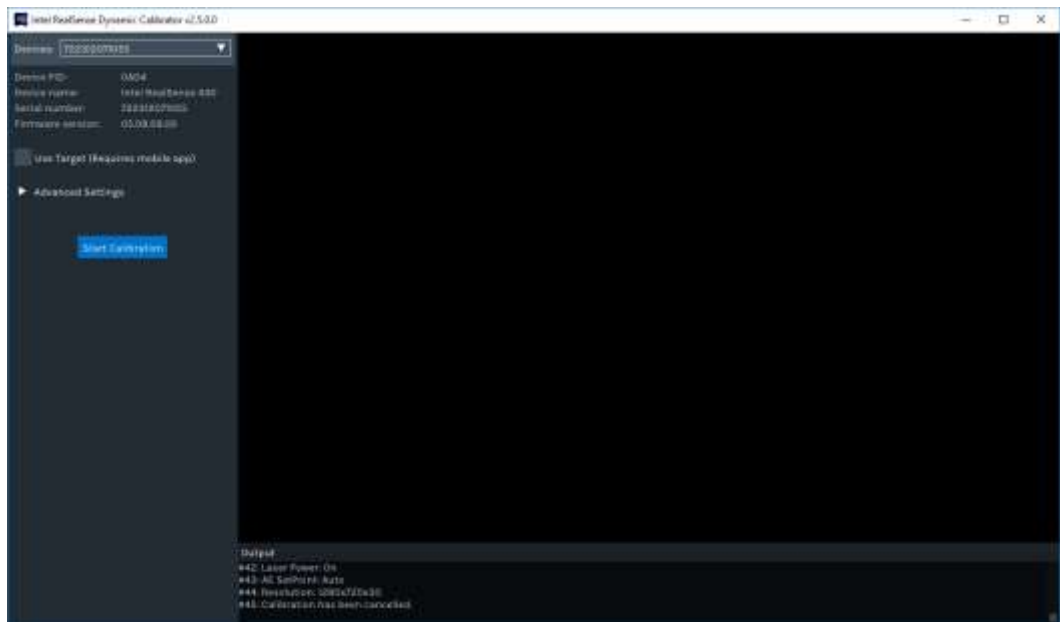
Calibration with Dynamic Calibrator

4.4.1 List all camera devices with Graphical User Interface

The app detects all Intel® RealSense™ Product Family D400 Series devices on the system and lists them in a dropdown box on the top left corner.



User selects one of the devices from the list to calibrate and details on the device including product ID, device name, serial number, and Firmware version is displayed:



Calibration with Dynamic Calibrator

4.4.2 List all camera devices with Command Line Interface (CLI)

To list all devices, use the `–show` or `–list` option in DynamicCalibrator. For example:

On Linux:

```
/usr/bin/Intel.Realsense.DynamicCalibrator –list
```

Device Name	Serial Number	Firmware Version	USB Type
Intel RealSense 410	718212020356	05.10.06.00	3.2
Intel RealSense 430	722312071055	05.10.06.00	3.2

On Windows:

```
cd bin
Intel.Realsense.DynamicCalibrator.exe –list
```

Device Name	Serial Number	Firmware Version	USB Type
Intel RealSense 410	718212020356	05.10.06.00	3.2
Intel RealSense 430	722312071055	05.10.06.00	3.2

4.4.3 Perform calibration on selected device with Command Line Interface (CLI)

Then choose one of the camera devices to calibrate by specifying its serial number though the `–sn` option in DynamicCalibrator app. For example,

On Linux:

```
/usr/bin/Intel.Realsense.DynamicCalibrator –sn 722312071055
```

On Windows:

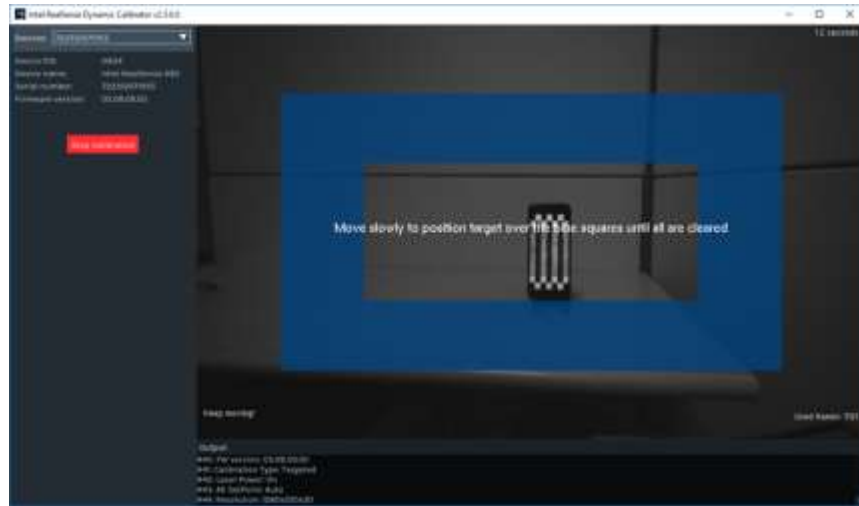
```
Intel.Realsense.DynamicCalibratorCLI.exe –sn 722312071055
```

4.4.4 Stop Calibration

After calibration is started, user can stop the process at any time by doing any **one** of the following:

- a) ESC key – press **ESC** key anytime in both GUI and CLI mode will stop the calibration process.
- b) Or, in GUI mode, press the **“Stop Calibration”** button at top left corner of the calibration window.

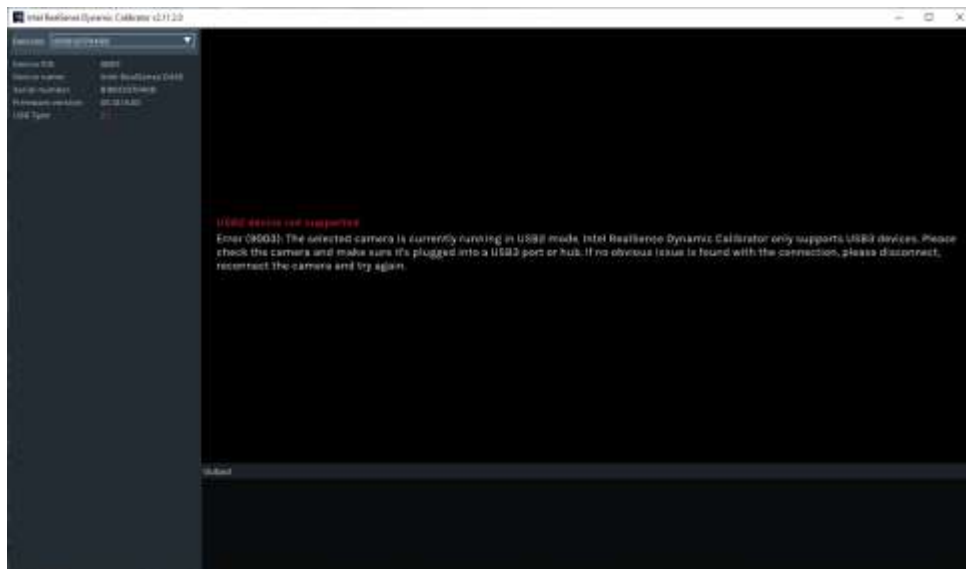
Calibration with Dynamic Calibrator



- c) Or, in CLI mode, press the “Q” or “q” key will quit the process
- d) Or, simply close the calibration window

4.5 Device USB2 Mode Warning

Intel® RealSense™ Dynamic Calibrator requires cameras running in USB3 mode. If a camera runs in USB2 mode, it's currently not supported. If such device is selected for calibration, the tool will display an error message at the command line or in the GUI to inform the user.

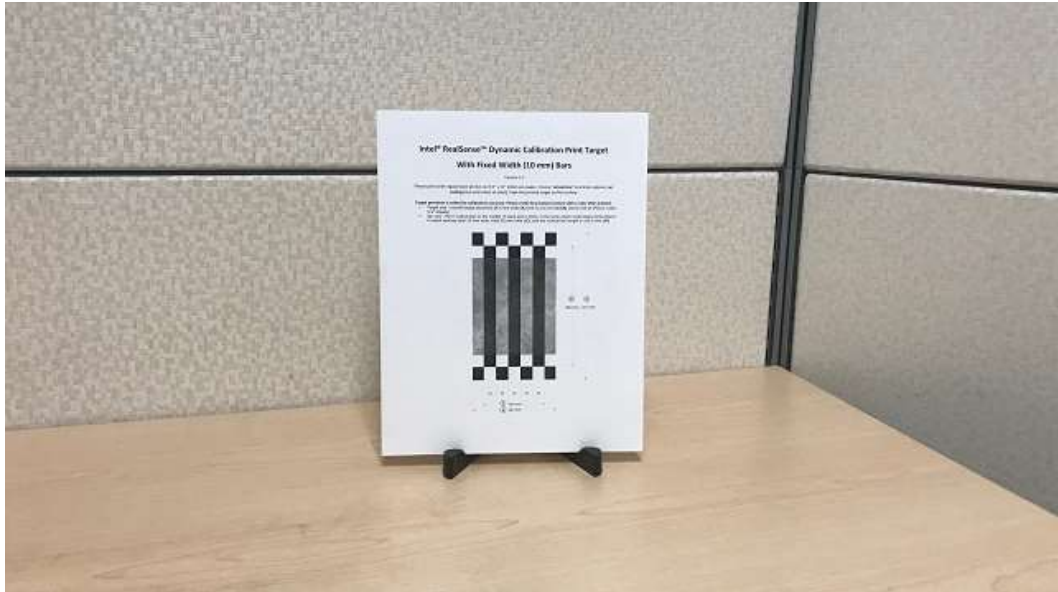


4.6 Targeted Calibration

4.6.1 Setup Calibration Target

Calibration Tools API supports printed target on letter sized paper. Please refer to details in Appendix *INTEL® REALSENSE™ DYNAMIC CALIBRATION TARGET SETUP*.

Figure 4-1. Dynamic Calibration Printed Target



The camera should be pointing to the target at a distance of approximately 65 cm to 100 cm. It should not be too close or too far. Due to various phone sizes, camera device models, and lighting conditions, exact distance cannot be specified. User will need to move the camera in the specified range to find a distance that works best. In most cases, a distance of around 70 cm is sufficient.

4.6.2 Running Dynamic Calibrator with targeted calibration

On Linux:

```
/usr/bin/Intel.Realsense.DynamicCalibrator
```

By default, the application displays Graphical User Interface (GUI). It detects all Intel® RealSense™ Product Family D400 Series devices on the system and lists them in a dropdown box on the top left corner. Selects one of the device from the list to calibrate. Details of the device is displayed as well:

Calibration with Dynamic Calibrator

If command line interface is preferred, user can display DynamicCalibrator command line options by running:

```
/usr/bin/ Intel.Realsense.DynamicCalibrator -help
```

And specify appropriate options on the command line. The GUI control panel is not displayed in command line mode. Calibration process starts immediately by default.

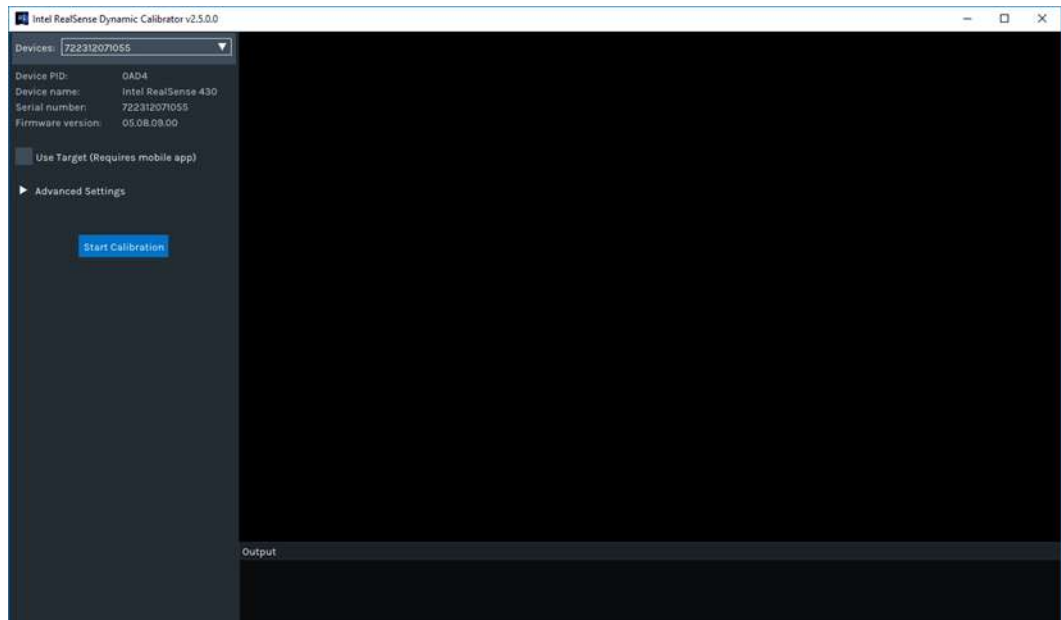
On **Windows**:

If the application is installed through an installer, an app icon “Dynamic Calibrator” should be already created on your desktop.



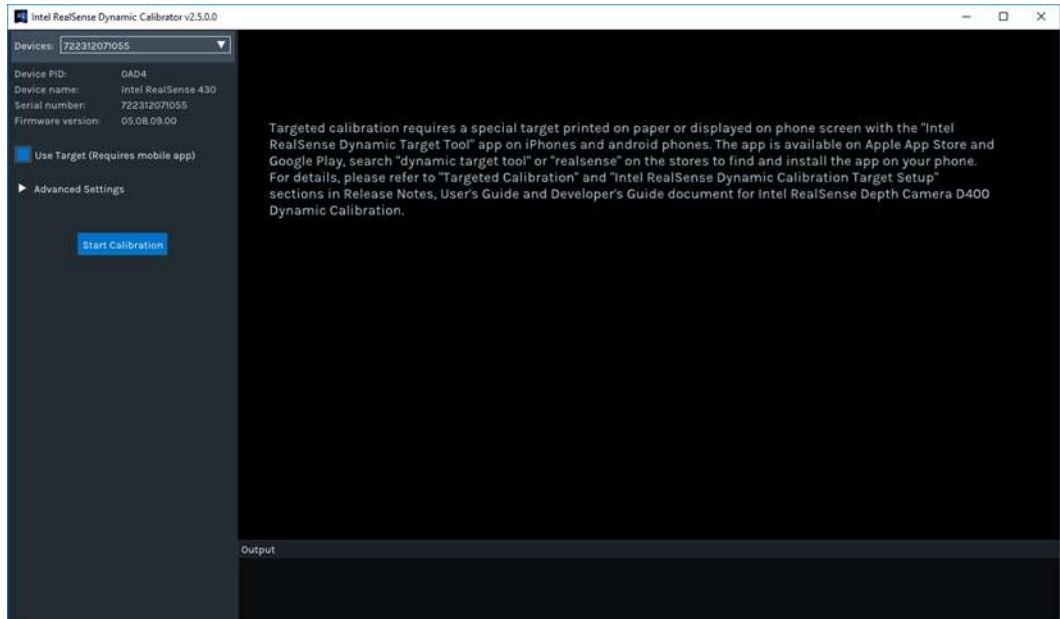
Double click the icon to start the app.

By default, the app starts with Graphical User Interface (GUI). It detects all Intel RealSense D400 devices on the system and lists them in a drop down box on the top left corner. Selects one of the device from the list to calibrate. Details on the device displayed as well:

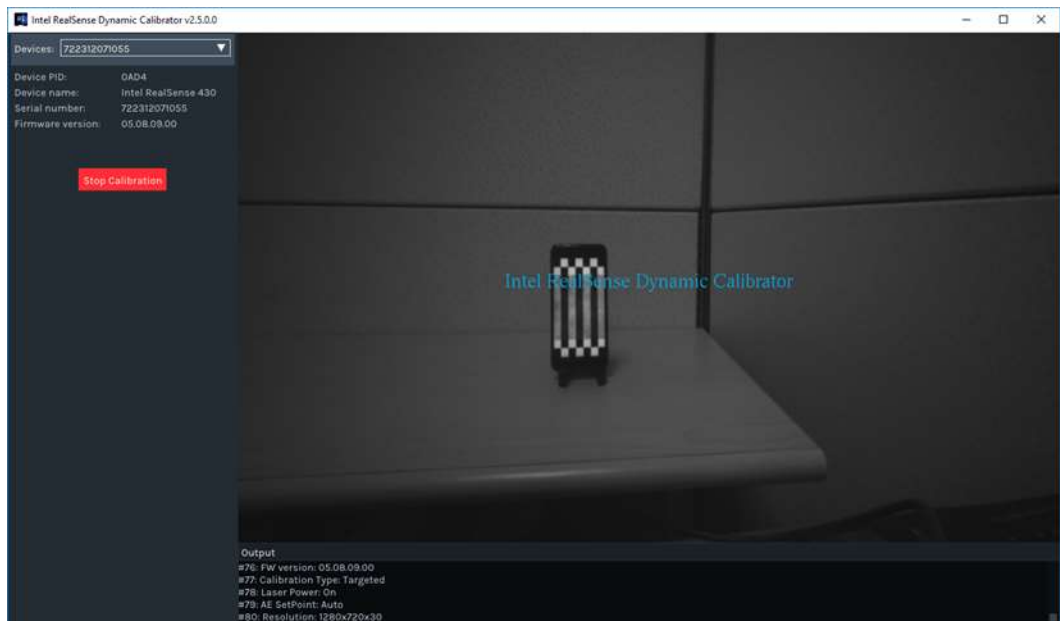


This calibration process requires a special target setup described in previous section “Setup Calibration Target”. If you haven't done that, please follow instructions and complete the target set up either a printed target or phone target.

Calibration with Dynamic Calibrator



Next to start calibration by clicking the “Start Calibration” button:



If command line interface is preferred, user can display DynamicCalibrator command line options by running:

```
Cd bin
Intel.Realsense.DynamicCalibratorCLI.exe -help
```

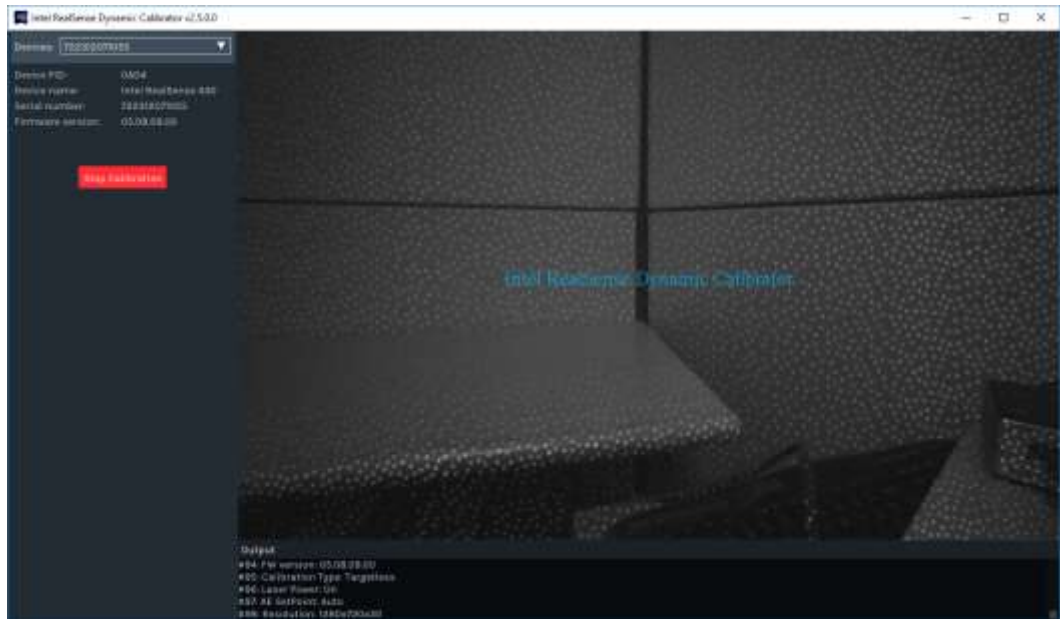
Calibration with Dynamic Calibrator

And specify appropriate options on the command line. The GUI control panel is not displayed in command line mode. Calibration process starts immediately by default.

```
cd bin
Intel.Realsense.DynamicCalibratorCLI.exe
```

When DynamicCalibrator calibration process starts, it briefly displays a logo screen:

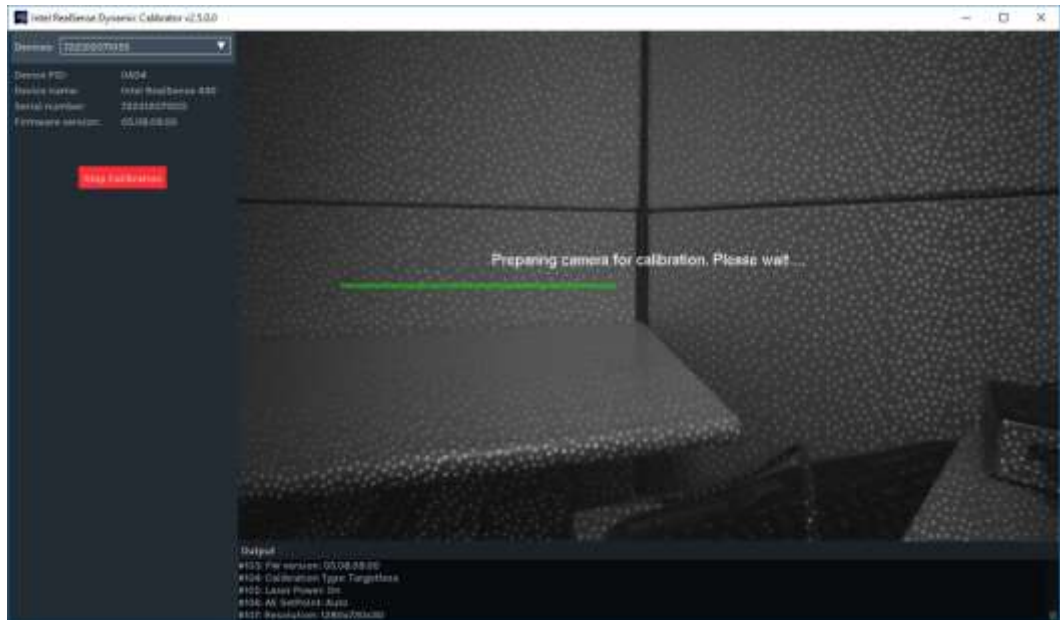
Figure 4-2 Dynamic Calibrator Initial Start Window



Followed by a warm up screen that indicates progress in getting the camera ready for calibration.

Calibration with Dynamic Calibrator

Figure 4-3 Dynamic Calibrator app warm up window

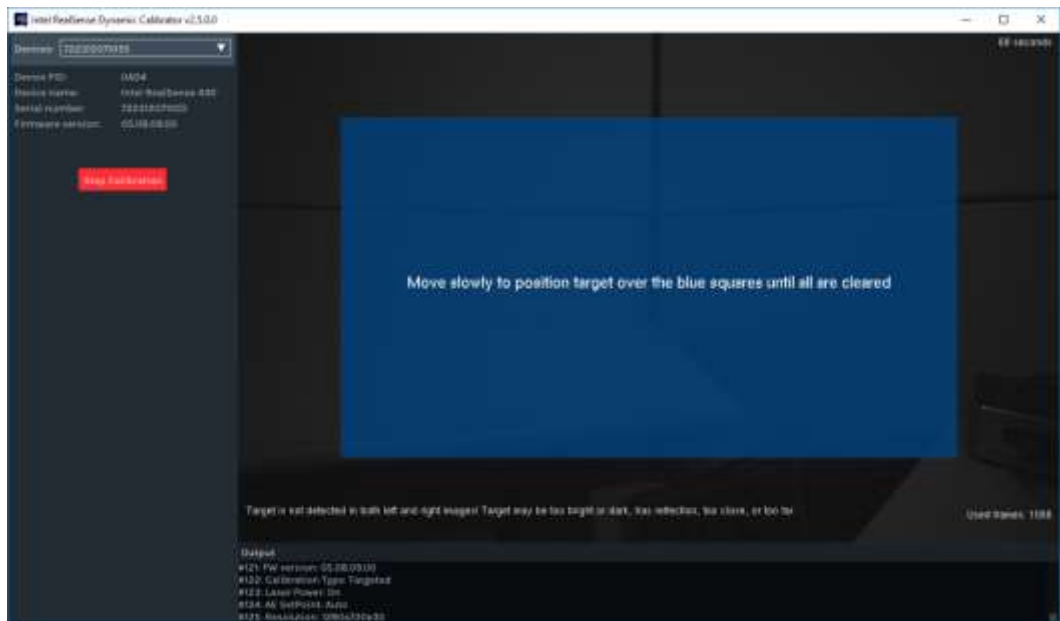
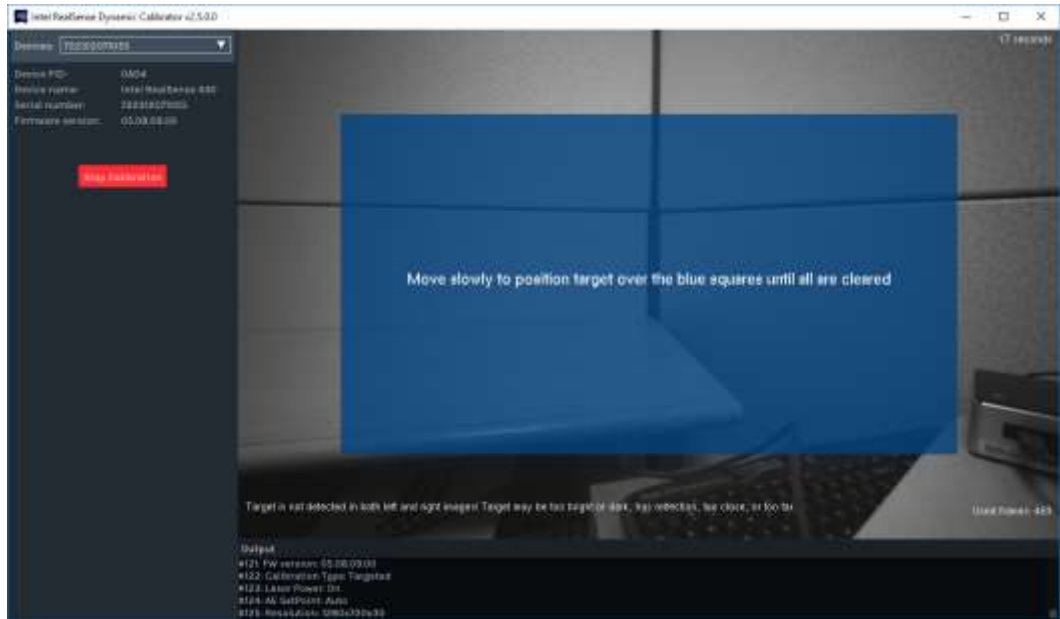


4.6.3 Auto Exposure Adjustment during Targeted Calibration

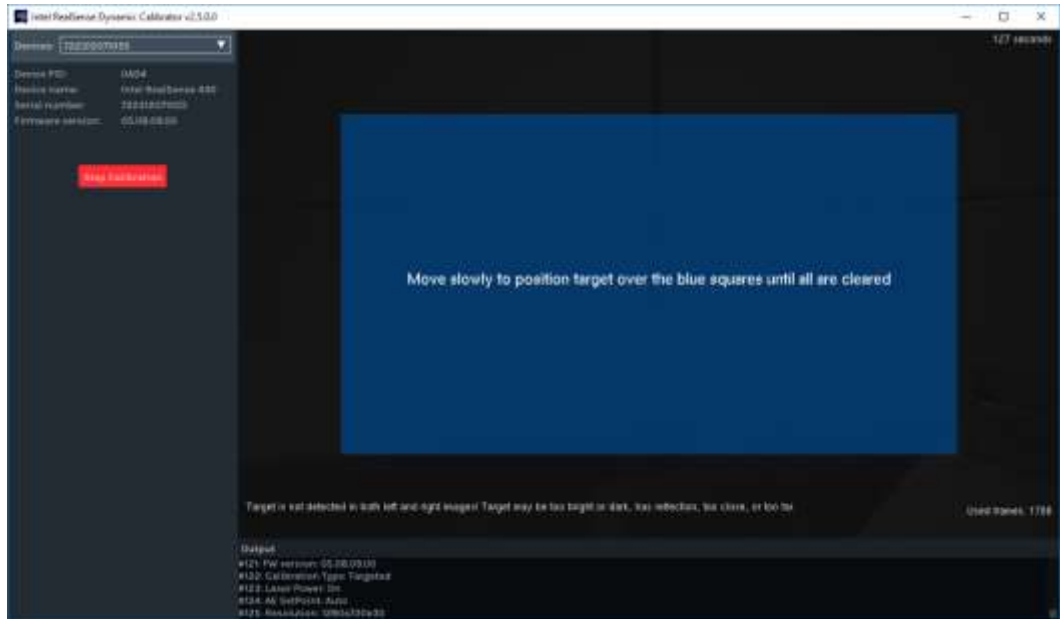
Throughout the targeted calibration process, exposure is adjusted automatically. The tool tries to detect and recognize the calibration target in all lighting conditions. When a target is not found, sometimes because the surrounding environment has a reflection on the phone screen or the target is too dark/bright to detect, it will adjust exposure automatically so that a target is detected. So from time to time, you may see the exposure changes from dark to bright and gradually reduces brightness. This is expected behavior.

In some cases, if the tool adjusted too bright or too dark, and the tool can't make progress, you can point the camera away from the target and wait a couple seconds until the brightness comes to a normal level and then point back to the target.

Calibration with Dynamic Calibrator



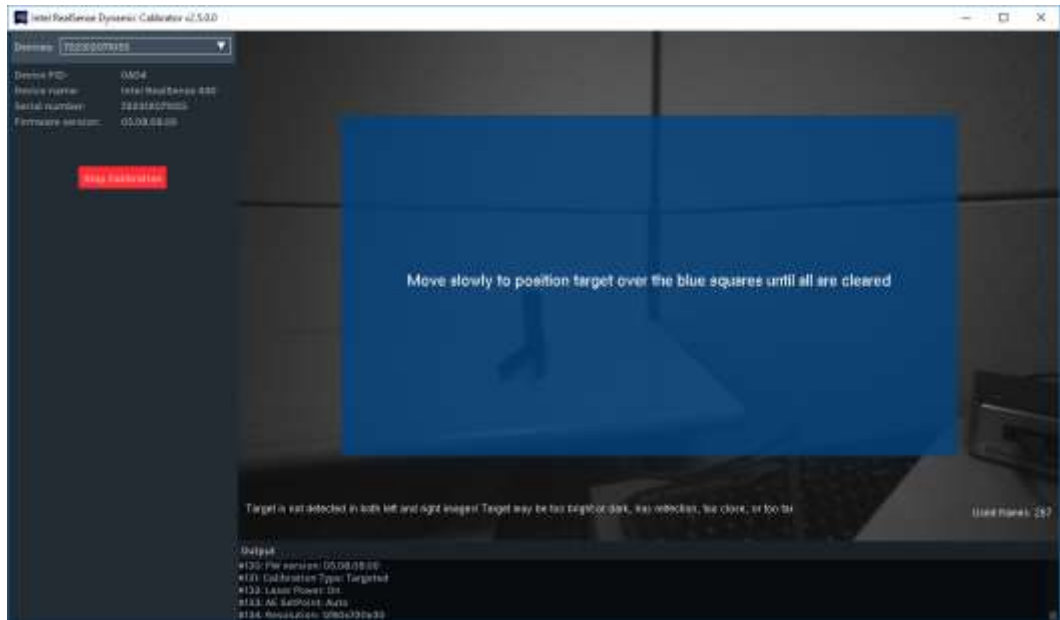
Calibration with Dynamic Calibrator



4.6.4 Target calibration phases

Target calibration process consists of two phases – **rectification** and **scale** calibration in sequential order. For the first phase, rectification, a block of be squares are present in middle part of the window:

Figure 4-4 Dynamic Calibrator Targeted Rectification Phase



Calibration with Dynamic Calibrator

Point the camera to the calibration target and move camera slowly to position target over the blue squares until all are cleared:

Figure 4-5 Dynamic Calibrator Targeted Rectification Phase (continued)

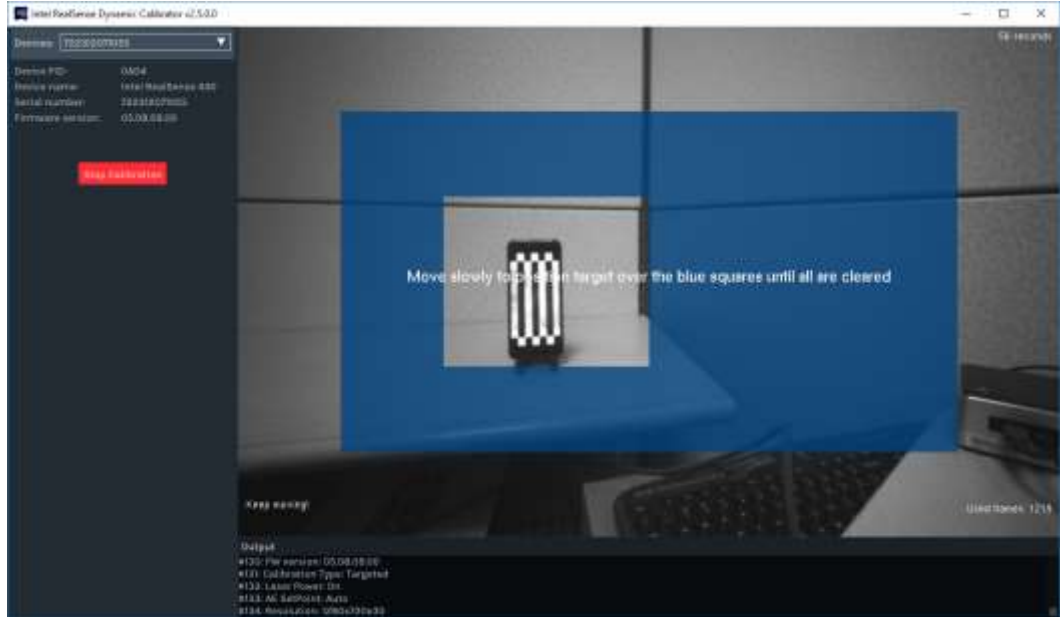
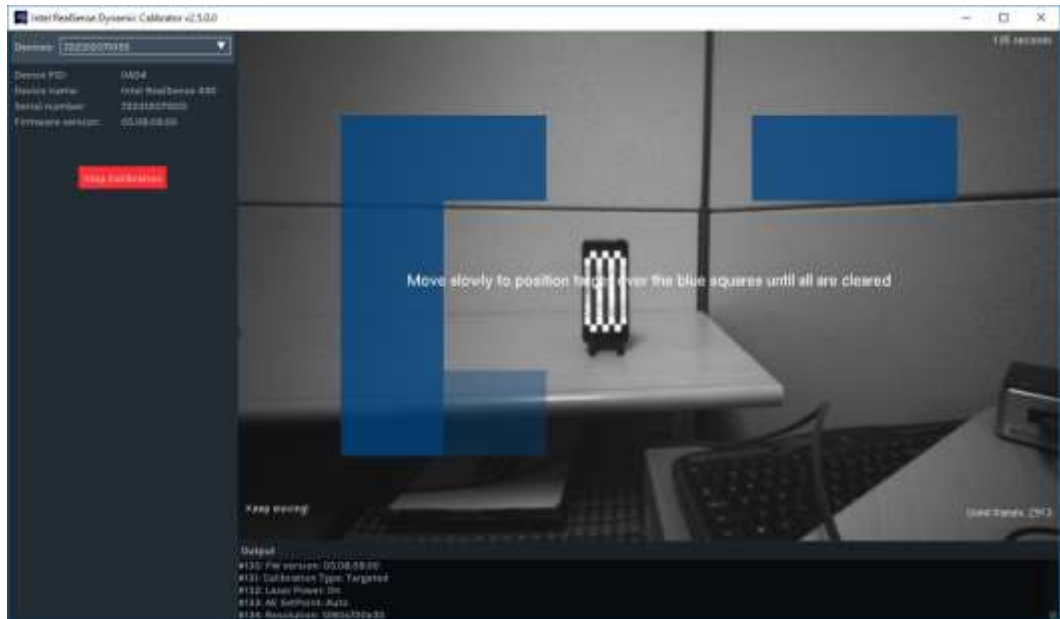


Figure 4-6 Dynamic Calibrator Targeted Rectification Phase (continued)



Once all blue squares are cleared, rectification phase is completed. Intermediate result is applied to the current stream.

Calibration with Dynamic Calibrator

Scale calibration phase will start immediately afterwards. Point the camera to the target and move slowly. A green progress bar will appear as it makes progress in scale calibration. Target images will be captured and analyzed and accepted target positions highlighted on screen. Total 15 successful target images will be accepted.

Figure 4-7 Dynamic Calibrator Scale Phase

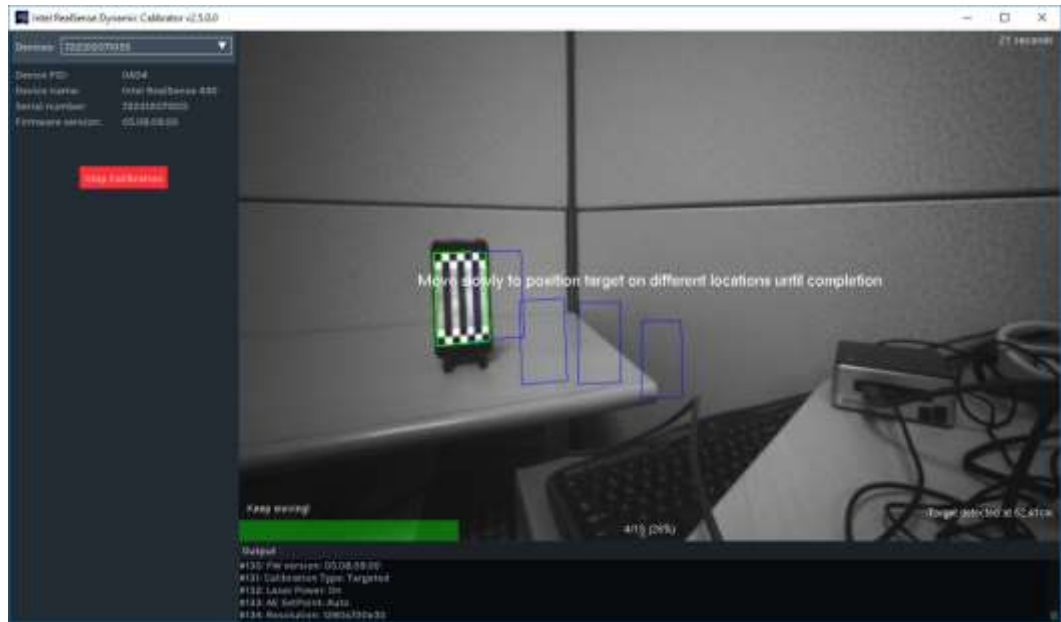
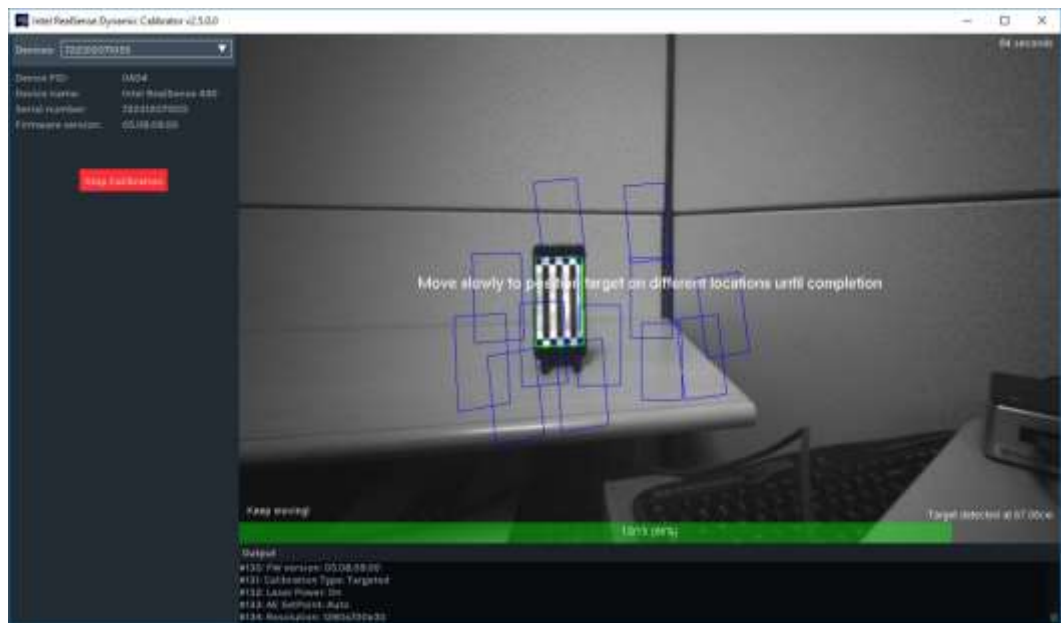


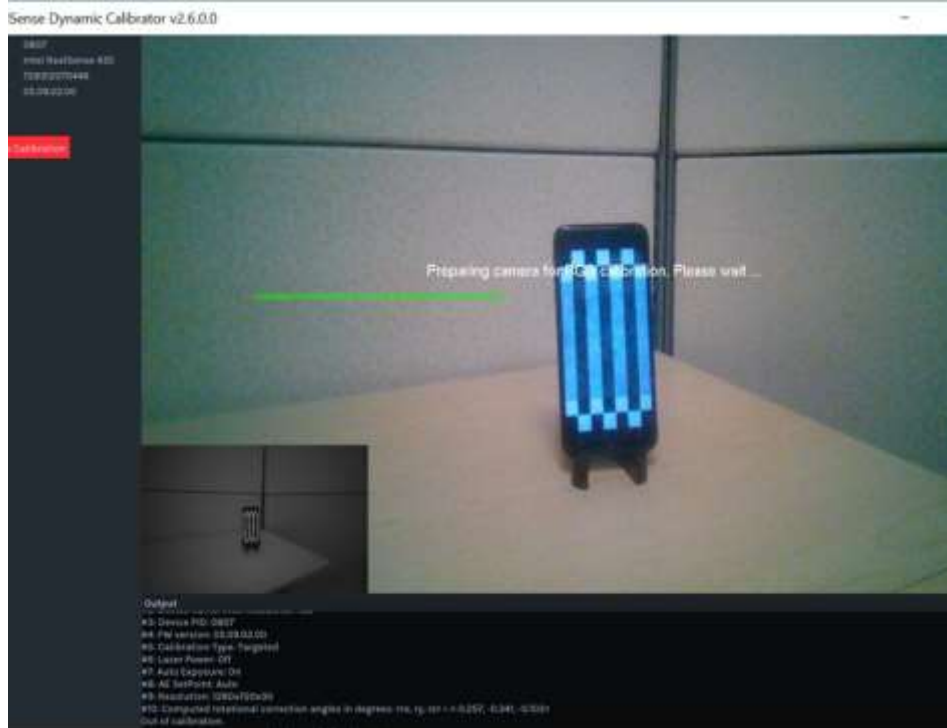
Figure 4-8 Dynamic Calibrator Scale Phase (continued)



Calibration with Dynamic Calibrator

After the green bar extends to full, scale calibration completes.

On D435 and D415 devices, an extra step will be run to calibrate the RGB camera. The usage is similar as the depth scale phase.

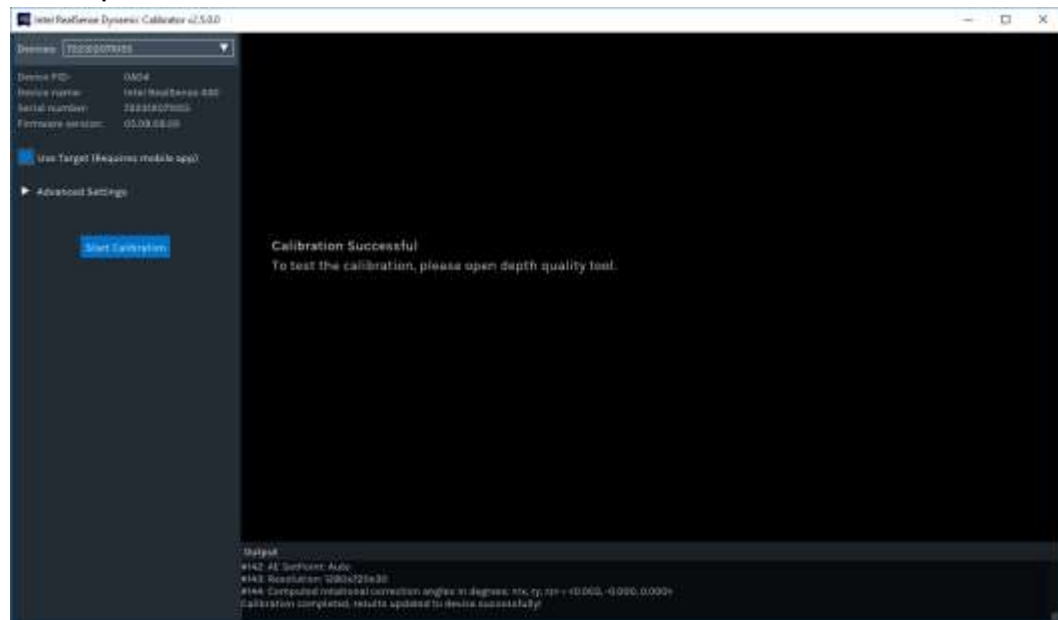


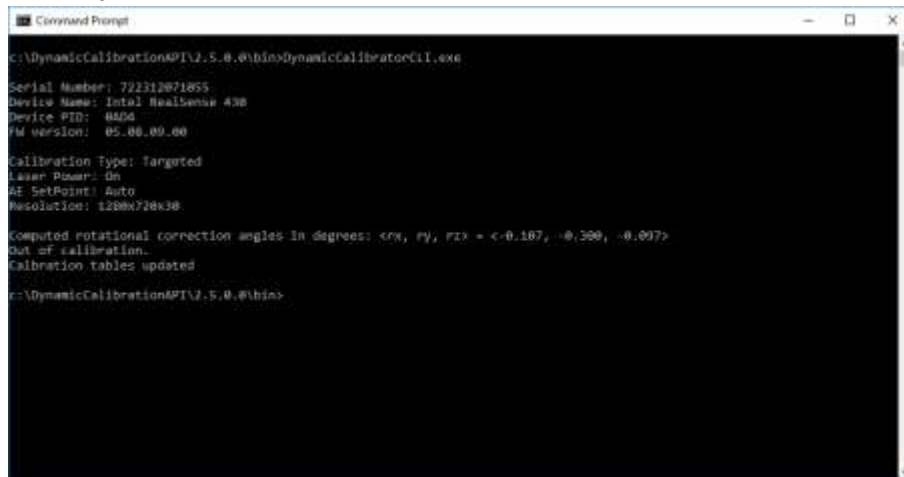
Calibration with Dynamic Calibrator



Depends on what user interface (GUI or CLI) used, result will be either on the GUI status box or displayed in the command line window. Results will be updated to device automatically. Success or fail status will also be displayed along with the results.

Results update in GUI mode:



Calibration with Dynamic Calibrator**Results update in CLI mode:**

```
Command Prompt
c:\DynamicCalibrationAPI\1.5.0.\bin\dynamicCalibratorCLI.exe
Serial Number: 722312871855
Device Name: Intel Realsense 438
Device PID: 8404
FW version: 05.06.00.00

Calibration Type: Targeted
Laser Power: On
AI SetPoint: Auto
Resolution: 1280x720x30

Computed rotational correction angles in degrees: <rx, ry, rz> = <-0.187, -0.309, -0.007>
Out =F calibration.
Calibration tables updated

c:\DynamicCalibrationAPI\1.5.0.\bin>
```

This completes the targeted calibration process.

5 Advanced Calibration Processes

As described in previous chapter, Dynamic Calibrator Command Line Interface (CLI) supports advanced options for user to explore.

- **Intel.Realsense.DynamicCalibratorCLI.exe** on Windows
- **Intel.Realsense.DynamicCalibrator -cli** on Linux

In addition to Targeted Calibration, it also supports a few advanced calibration processes for automation like calibrating devices in robotics.

The sample app source code is included as part of the Calibration API package, so users can experiment and develop own calibration app for their usage.

5.1 Target-less Calibration

This calibration mode does not require a target. User simply points the device to a scene with textures and move slowly until the calibration process finishes. The advantage is no target is required, so it's convenient in integration and automation where a target is not available. The disadvantage is that it's less accurate as Targeted Calibration. It rectifies the left and right cameras (improves fill rate) but does not improve scaling error as well as Targeted Calibration (less accurate and consistent Z-accuracy).

(Windows)

```
Intel.Realsense.DynamicCalibratorCLI.exe -m 0
```

(Linux)

```
Intel.Realsense.DynamicCalibrator -cli -m 0
```

Advanced Calibration Processes**Figure 5-1 Target-less Dynamic Calibration with Entire FOV**

By default, Target-less considers features in the entire FOV. In some cases, this can be troublesome, for example, device mounted on a robot and very close to the ground floor, or cases where features not available in the edges or corners of the FOV. Since the target-less calibration algorithm uses features in the entire FOV, it will take much longer than usual to complete the calibration process or in worst cases may not complete at all.

The `-ignore-borders` option is provided to handle such difficult situations. Accuracy may be slightly impacted.

(Windows)

```
Intel.Realsense.DynamicCalibratorCLI.exe -m 0 -ignore-borders
```

(Linux)

```
Intel.Realsense.DynamicCalibrator -cli -m 0 -ignore-borders
```

Figure 5-2 Target-less Dynamic Calibration with Center FOV

5.2 Hybrid Calibration

Hybrid Calibration combines best of Target-less Calibration (convenience) and Targeted Calibration (accuracy) into a single process. It starts with a target-less calibration to rectifies the left/right images and then followed with a targeted scale calibration. It provides flexibility as well as accuracy.

(Windows)

```
Intel.Realsense.DynamicCalibratorCLI.exe -m 2
```

(Linux)

```
Intel.Realsense.DynamicCalibrator -cli -m 2
```

Hybrid Calibration starts the initial phase in target-less calibration:

Advanced Calibration Processes

Figure 5-3 Hybrid Dynamic Calibration – Target-less Phase



Once the target-less rectification is completed, it continues to a scale calibration with the same target used in Targeted Calibration.

Figure 5-4 Hybrid Dynamic Calibration – Targeted Scale Calibration Phase



On devices with RGB, like D435 and D415, the calibration process will continue to a third phase to calibrate the RGB. However, this phase can be skipped with the `-skiprgb` option if user does not use the RGB in their usage.

Figure 5-5 Hybrid Dynamic Calibration – Targeted RGB Calibration Phase



A few additional command line options are provided to fit into user usage:

Option	Description
-a -aligned	device and target aligned in orientation, for example, device is mounted vertically, and target is positioned vertically
-skiprgb	Skip RGB calibration. This is useful if device like D435 and D415 equipped with a RGB camera but user does not use it. Skip RGB phase during calibration process will not impact depth calibration and it makes the process much faster.
-ignore-borders	ignore the border blocks during target-less rectification
-max-images <more than 6>	Specify number of images to capture in scale calibration and rgb calibration phases. This is used when user usage limited by time or space and would like to experiment best settings for their usage. <ul style="list-style-type: none"> • In targeted calibration, the default is 16 images. • In hybrid calibration, the default is either 6 or 8 images depends on device/target orientation. • For optimal results, recommend at least 6 images.
-force	Force the app to accept specified parameters, even if it's outside of the recommended range. Currently this only applies to -max-images.

When the device is mounted vertically on a robot and the robot is limited in any vertical movement. A very limited number of target images can be fit into the FOV due to the movement limitation.

Advanced Calibration Processes**Figure 5-6 Hybrid Dynamic Calibration – Device and Target Aligned**

In this case place the target with the stripes vertically as well and use -a option to align the target and the device. More target images can be captured in the scale calibration and RGB calibration phases and therefore better accuracy in the calibration result.

(Windows)

```
Intel.Realsense.DynamicCalibratorCLI.exe -m 2 -a
```

(Linux)

```
Intel.Realsense.DynamicCalibrator -cli -m 2 -a
```

Figure 5-7 Hybrid Dynamic Calibration – Device and Target Aligned



5.2.1 Run Calibration with the Robot

Target-less Calibration with a robot:

(Windows)

Intel.Realsense.DynamicCalibratorCLI.exe -m 0 -gimbal <COM port on your system>

(Linux)

Intel.Realsense.DynamicCalibrator -cli -m 0 -gimbal /dev/ttyUSB0

Hybrid Calibration with a robot:

(Windows)

Intel.Realsense.DynamicCalibratorCLI.exe -m 2 -gimbal <COM port on your system>

(Linux)

Intel.Realsense.DynamicCalibrator -cli -m 2 -gimbal /dev/ttyUSB0

The calibration process only rotates the device through the simple robot used in this sample. When user develops their own calibration app on their robot, it can take advantage of horizontal or other movement if available to improve the process.

Advanced Calibration Processes

Figure 5-8 Dynamic Calibration on Robot



Figure 5-9 Dynamic Calibration on Robot



Advanced Calibration Processes

Figure 5-10 Dynamic Calibration on Robot



6 Handling Calibration Data with CustomRW Tool

While users calibrate devices with dynamic calibrator or through their own developed calibration tools based on Calibration API, there is a need to read, write, and restore calibration data on the device. CustomRW Tool provides this ability to users. CustomRW Tool is included in the Calibration Tool package and is installed by default under C:\Program Files\Intel\CalibrationToolAPI\2.13.1.0\bin on Windows and /usr/bin on Linux.

The tool supports options to restore and read/write calibration data:

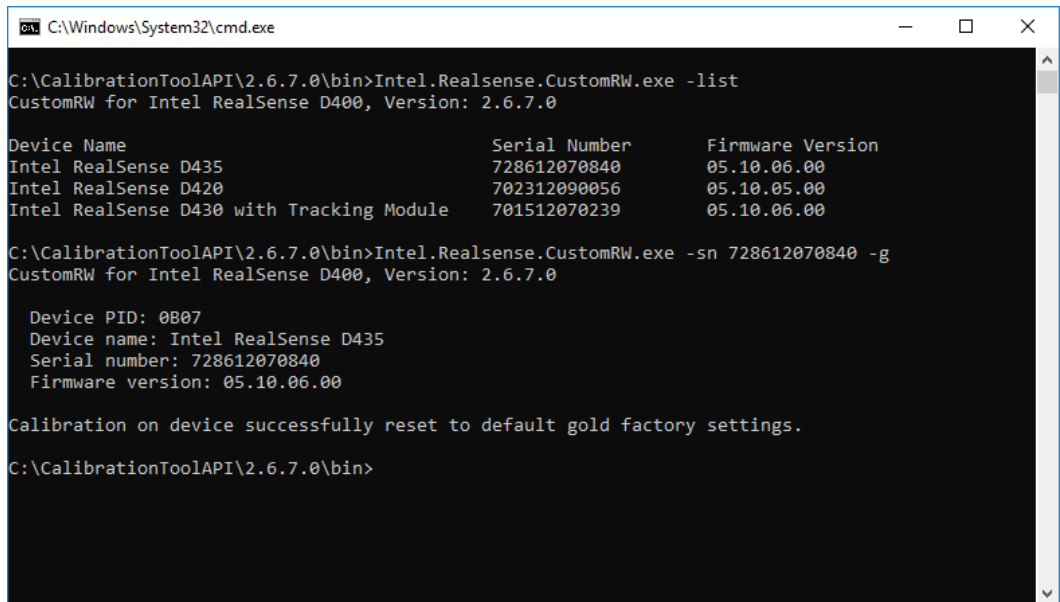
Table 6-1: CustomRW Command Line Options

Option	Description
-help or -?	Display list of command line options.
-version -v	Show version info
-list	Display list of connected devices with device name, serial number and firmware version information.
-sn <camera serial number> -serial < camera serial number>	If multiple cameras connected to the current system, choose one of the camera devices to calibrate by specifying its serial number. When a serial number is not specified, the tool will detect devices on the system and choose the first one to operate.
-g	Reset calibration on selected device to default gold factory settings
-r	Read calibration data
-w	Write calibration data
-raw <19, 1f, 20, or ff>	Specify calibration table raw binary content to read/write. Valid table id 19, 1f, 20, and ff
-fe	Specify custom fisheye data to read/write. Available ONLY on selected device SKU.
-file <file name> -f <file name>	Specify file source/destination. When read, by default, without specifying a file destination, the data will be display on screen in the command console.
--force-rgb -rgb	force read/write rgb calibration data even if device does not have rgb sensor (for 3rd party rgb custom calibration)
--extra -ex	print extra calibration information to console

6.1 Restore Calibration on Device

The `-g` option restores the calibration on device to default gold settings. Depending up on the type of calibration done on the device, the gold settings represent the original factory setting or the calibration setting from OEM Calibration if such a calibration is done on the device after factory calibration.

Figure 6-1 Restore Calibration on Selected Device



```

C:\Windows\System32\cmd.exe
C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -list
CustomRW for Intel RealSense D400, Version: 2.6.7.0

Device Name                Serial Number      Firmware Version
Intel RealSense D435       728612070840      05.10.06.00
Intel RealSense D420       702312090056      05.10.05.00
Intel RealSense D430 with Tracking Module 701512070239      05.10.06.00

C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 728612070840 -g
CustomRW for Intel RealSense D400, Version: 2.6.7.0

Device PID: 0B07
Device name: Intel RealSense D435
Serial number: 728612070840
Firmware version: 05.10.06.00

Calibration on device successfully reset to default gold factory settings.

C:\CalibrationToolAPI\2.6.7.0\bin>

```

6.2 Read Calibration from Device

The tool supports `-r` option to read the calibration parameters from the device. By default, the parameters are printed on screen. If a file name is provided, the parameters can also be dumped into a file in XML format.

Handling Calibration Data with CustomRW Tool

Figure 6-2 Read Calibration on Selected Device

```

C:\Windows\System32\cmd.exe
C:\CalibrationToolAPI\2.6.7.0\bin>Intel.RealSense.CustomRW.exe -sn 728012878848 -r -f mycalib.xml
CustomRW for Intel RealSense D488, Version: 2.6.7.0

Device PID: 0807
Device name: Intel RealSense D435
Serial number: 728012878848
Firmware version: 05.10.06.00

Calibration parameters from the device:
resolutionLeftRight: 1280 800

FocalLengthLeft: 641.271118 641.170941
PrincipalPointLeft: 641.274963 403.309265
DistortionLeft: -0.055031 0.064661 0.000622 -0.000774 -0.020866

FocalLengthRight: 641.384277 641.169739
PrincipalPointRight: 644.808350 405.958710
DistortionRight: -0.055748 0.065381 0.001820 -0.000605 -0.021290

RotationLeftRight: 0.999997 -0.001372 0.001770
                   0.001370 0.999991 -0.003506
                   -0.001764 0.003908 0.999991

TranslationLeftRight: -49.960100 -0.023940 0.159185

HasRGB: 1

resolutionRGB: 1920 1080

FocalLengthColor: 1389.219849 1389.348877
PrincipalPointColor: 963.673706 548.193542
DistortionColor: 0.000000 0.000000 0.000000 0.000000 0.000000
RotationLeftColor: 0.999835 -0.014481 0.018044
                   0.014530 0.999885 -0.004413
                   -0.018078 0.004591 0.999930
TranslationLeftColor: 15.371313 -0.011960 0.853173

C:\CalibrationToolAPI\2.6.7.0\bin>

```

The calibration parameters are defined in section 2.1 “**Calibration Parameters**”. The XML file format is as below:

```

<?xml version="1.0"?>
<Config>
  <param name = "ResolutionLeftRight">
    <value>1280</value>
    <value>800</value>
  </param>
  <param name = "FocalLengthLeft">
    <value>641.271</value>
    <value>641.177</value>
  </param>
  <param name = "PrincipalPointLeft">
    <value>641.275</value>
    <value>403.309</value>
  </param>
  <param name = "DistortionLeft">
    <value>-0.0550314</value>
    <value>0.0646809</value>
    <value>0.00062232</value>
    <value>-0.000774438</value>
    <value>-0.0208662</value>
  </param>
  <param name = "FocalLengthRight">
    <value>641.384</value>
    <value>641.17</value>
  </param>
  <param name = "PrincipalPointRight">
    <value>644.808</value>

```

Handling Calibration Data with CustomRW Tool

```

    <value>405.959</value>
</param>
<param name = "DistortionRight">
  <value>-0.055748</value>
  <value>0.0653811</value>
  <value>0.00102014</value>
  <value>-0.000605032</value>
  <value>-0.0212901</value>
</param>
<param name = "RotationLeftRight">
  <value>0.999997</value>
  <value>-0.00137212</value>
  <value>0.00176977</value>
  <value>0.00137902</value>
  <value>0.999991</value>
  <value>-0.00390553</value>
  <value>-0.0017644</value>
  <value>0.00390796</value>
  <value>0.999991</value>
</param>
<param name = "TranslationLeftRight">
  <value>-49.9601</value>
  <value>-0.0239399</value>
  <value>0.159105</value>
</param>
<param name = "HasRGB">
  <value>1</value>
</param>
<param name = "ResolutionRGB">
  <value>1920</value>
  <value>1080</value>
</param>
<param name = "FocalLengthRGB">
  <value>1389.22</value>
  <value>1389.35</value>
</param>
<param name = "PrincipalPointRGB">
  <value>983.674</value>
  <value>548.194</value>
</param>
<param name = "DistortionRGB">
  <value>0</value>
  <value>0</value>
  <value>0</value>
  <value>0</value>
  <value>0</value>
</param>
<param name = "RotationLeftRGB">
  <value>0.999835</value>
  <value>-0.0144807</value>
  <value>0.0109438</value>
  <value>0.0145299</value>
  <value>0.999885</value>
  <value>-0.00443285</value>
  <value>-0.0108784</value>
  <value>0.00459113</value>
  <value>0.99993</value>

```

Handling Calibration Data with CustomRW Tool

```

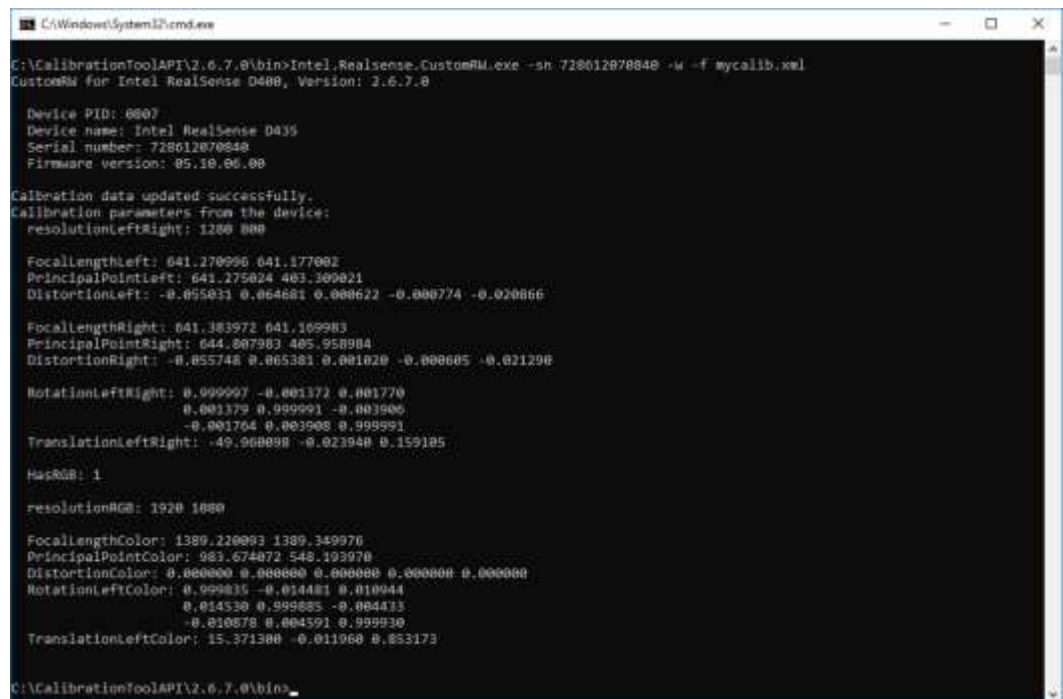
</param>
<param name = "TranslationLeftRGB">
  <value>15.3713</value>
  <value>-0.0119596</value>
  <value>0.853173</value>
</param>
</Config>

```

6.3 Write Calibration to Device

User can come up with the calibration parameters from their own calibration algorithm or from previous backups of the calibration data in XML format and write it to the device through the `-w` option.

Figure 6-3 Write Calibration on Selected Device



6.4 Read and Write RAW Calibration Tables

The device keeps the calibration data in its internal raw tables. Intel maintains the tables in its proprietary formats. The read/write options above convert the internal formats to XML so users can understand the actual calibration parameters. There are cases where read/write the raw tables is necessary. For example, these raw tables can be generated by other Intel tools like OEM Calibration Tool with offline pre-captured images, user can then write the raw tables directly to the device through the `-w` option. Similarly, user can also choose to dump the raw tables for records and restore the device to this calibration data later.

Handling Calibration Data with CustomRW Tool

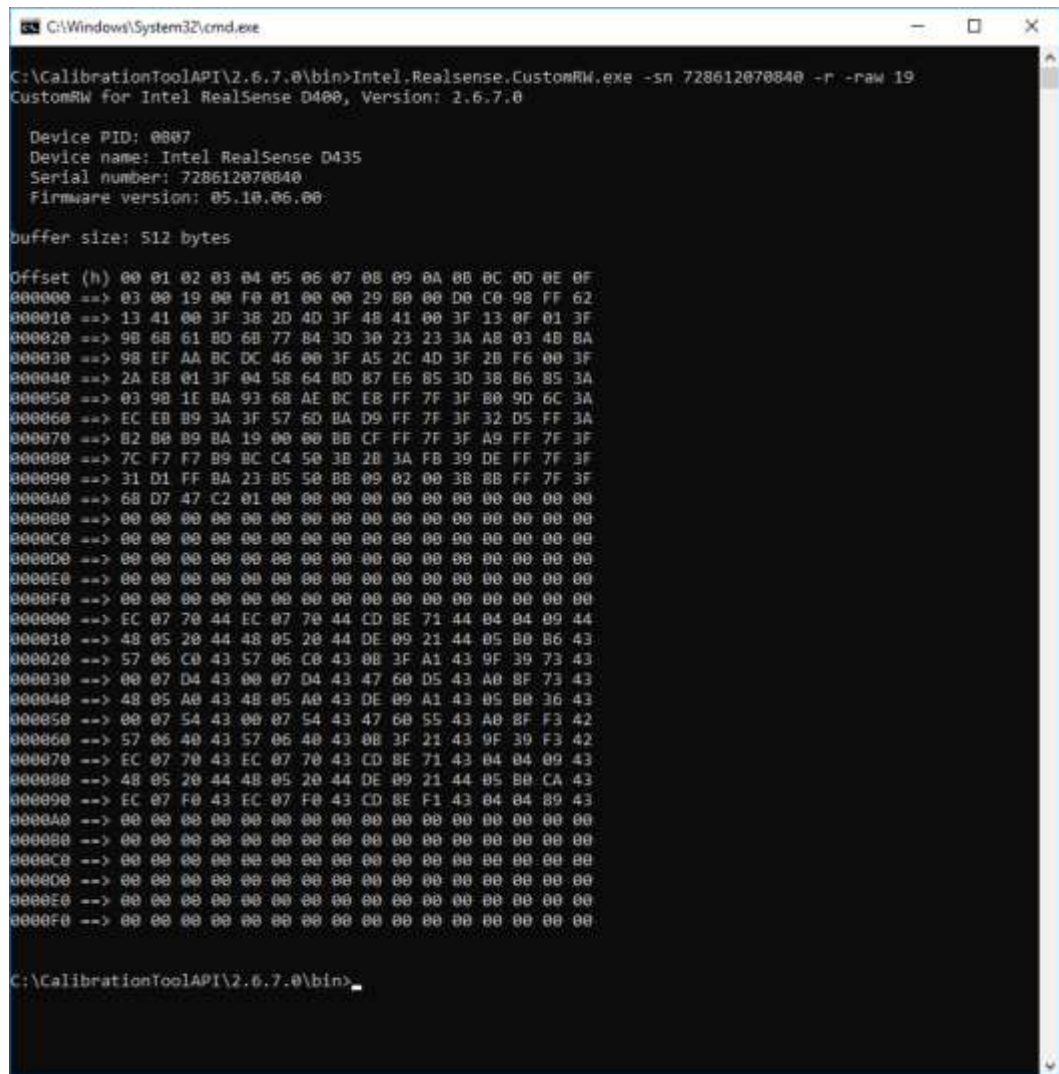
The `-raw <table id>` option supports raw tables 19 (coefficient table), 1f (depth table) and 20 (RGB table). Coefficient table and depth table are available on all devices. RGB table only exists on devices with RGB, like D415 and D435.

For example,

Example: read raw calibration table from device and display to the terminal, supported table id 19, 1F, and 20, for example,

```
Intel.Realsense.CustomRW.exe -r -raw 19
```

Figure 6-4 Read Calibration RAW Data on Selected Device and Display on Screen



Example: read raw calibration table from device and save into a binary file, supported table id 19, 1F, and 20, for example,

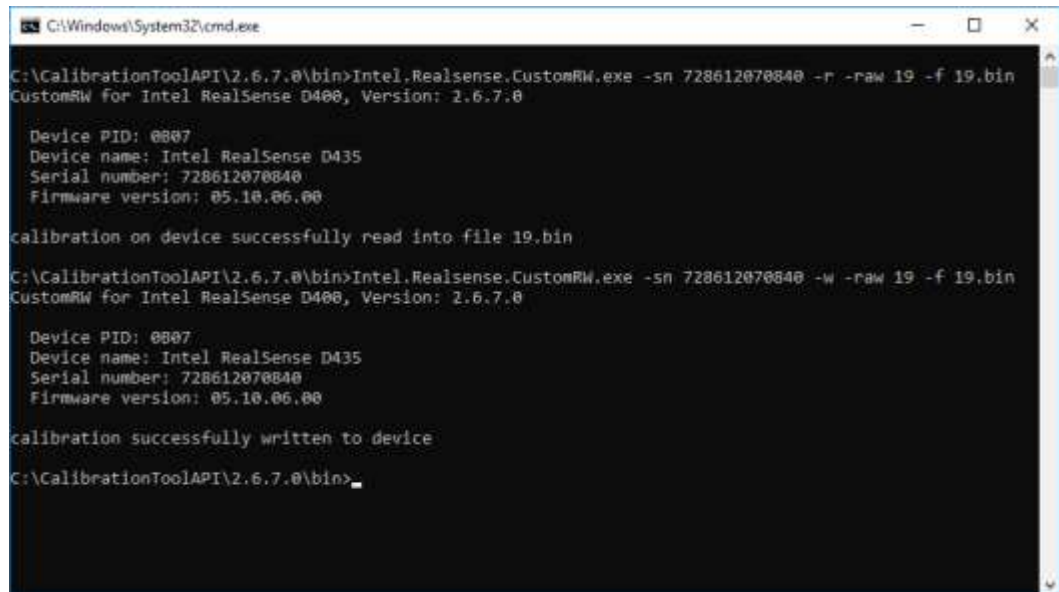
Handling Calibration Data with CustomRW Tool

Intel.Realsense.CustomRW.exe -r -raw 19 -f mytable-19.bin

Example: write raw calibration table from binary file into the device, supported table id 19, 1F, and 20, for example,

Intel.Realsense.CustomRW.exe -w -raw 19 -f mytable-19.bin

Figure 6-5 Read and Write Calibration RAW Data on Selected Device



```

C:\Windows\System32\cmd.exe
C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 728612070840 -r -raw 19 -f 19.bin
CustomRW for Intel RealSense D400, Version: 2.6.7.0

Device PID: 0807
Device name: Intel RealSense D435
Serial number: 728612070840
Firmware version: 05.10.06.00

calibration on device successfully read into file 19.bin

C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 728612070840 -w -raw 19 -f 19.bin
CustomRW for Intel RealSense D400, Version: 2.6.7.0

Device PID: 0807
Device name: Intel RealSense D435
Serial number: 728612070840
Firmware version: 05.10.06.00

calibration successfully written to device

C:\CalibrationToolAPI\2.6.7.0\bin>

```

6.5 Read and Write Custom Fisheye Data

The -fe option supports read/write the custom fisheye data on selected device SKU. It only works on very limited range of devices. Use this option ONLY when the device is supported.

Example #8: read custom fisheye data from device and display to the terminal
 Intel.Realsense.CustomRW.exe -r -fe

Handling Calibration Data with CustomRW Tool

Figure 6-6 Read Fisheye Custom Data on Selected Device and Display on Screen

```

C:\Windows\System32\cmd.exe

C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -list
CustomRW for Intel RealSense D400, Version: 2.6.7.0

Device Name                Serial Number              Firmware Version
Intel RealSense D435       728612070840              05.10.06.00
Intel RealSense D420       702312090056              05.10.05.00
Intel RealSense D430 with Tracking Module  701512070239              05.10.06.00

C:\CalibrationToolAPI\2.6.7.0\bin>Intel.Realsense.CustomRW.exe -sn 701512070239 -r -fe
CustomRW for Intel RealSense D400, Version: 2.6.7.0

Device PID: 0AD5
Device name: Intel RealSense D430 with Tracking Module
Serial number: 701512070239
Firmware version: 05.10.06.00

Buffer size: 136 bytes

Offset (h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
000000 --> 00 00 00 40 BB 80 7F 43 00 00 00 00 96 0A A2 43
000010 --> 00 00 00 00 68 5B 80 43 40 58 7E 43 00 00 00 00
000020 --> 00 00 00 00 00 00 00 3F 04 21 68 3F 00 00 C0 FF
000030 --> 00 00 C0 FF 00 00 C0 FF 00 00 C0 FF 94 F5 7F BF
000040 --> 95 0F 81 8C C1 CC 08 8C 21 59 81 3C 44 F7 7F BF
000050 --> 9A 04 88 BB CC 85 07 8C FE 50 8C BB 20 FD 7F 3F
000060 --> 89 A2 20 BC ED C0 08 3C 99 0B 46 B8 FF FF FF FF
000070 --> FF FF FF FF 12 34 56 7F FF FF FF FF FF FF FF FF
000080 --> FF FF AB CD EF FF FF FF

C:\CalibrationToolAPI\2.6.7.0\bin>

```

Example: read custom fisheye data from device and save into a binary file
 Intel.Realsense.CustomRW.exe -r -fe -f myfe.bin

Example: write custom fisheye data from binary file into the device
 Intel.Realsense.CustomRW.exe -w -fe -f myfe.bin

6.6 Read calibration data from device and print extra calibration information

With -ex option, extra calibration information can be printed, for example, world to left/right rotation matrix, RGB extrinsics in rectified coordinate system. This information can be useful when calibrating custom 3rd party rgb sensor along with Intel® RealSense™ Depth Cameras.

Example: read calibration data from device from device and print extra calibration information to the terminal
 Intel.Realsense.CustomRW.exe -r -ex

Handling Calibration Data with CustomRW Tool

Figure 6-7 Read Calibration Data and Print Extra Calibration Information on Screen

```

C:\Windows\System32\cmd.exe
C:\Intel>Intel.Realsense.CustomRW.exe -sn 013322250242 -r -ex
CustomRW for Intel RealSense D400, Version: 2.12.0.0

Device PID: 085C
Device name: Intel RealSense D455
Serial number: 013322250242
Firmware version: 05.12.14.107
USB Type: 3.2

Calibration parameters from the device:
resolutionLeftRight: 1280 800

FocallengthLeft: 637.578369 637.653992
PrincipalPointLeft: 639.241699 395.139008
DistortionLeft: -0.055249 0.061241 -0.002801 -0.000915 -0.018776

FocallengthRight: 642.286011 642.163025
PrincipalPointRight: 629.955444 408.075806
DistortionRight: -0.052595 0.056032 -0.000915 -0.000780 -0.017061

RotationLeftRight: 0.999964 0.000954 -0.008480
                  -0.001014 0.999975 -0.006994
                  0.008473 0.007002 0.999940
TranslationLeftRight: -95.212959 -0.268179 -0.369349

HasRGB: 1

resolutionRGB: 1280 800

FocallengthColor: 638.267151 638.466492
PrincipalPointColor: 644.835938 408.008484
DistortionColor: -0.052774 0.054780 -0.001483 -0.000992 -0.016103
RotationLeftColor: 0.999988 0.001019 -0.004814
                  -0.001041 0.999989 -0.004523
                  0.004809 0.004528 0.999978
TranslationLeftColor: -58.986320 -0.240117 -0.081141

Extra calibration information:

World to left rotation matrix (inverse rotation of the left camera in rectified coordinate system):
0.999982 -0.003814 0.004607
0.003798 0.999987 0.003501
-0.004621 -0.003483 0.999983

World to right rotation matrix (inverse rotation of the right camera in rectified coordinate system):
0.999988 -0.002830 -0.003869
0.002817 0.999990 -0.003498
0.003879 0.003487 0.999986

RGB camera intrinsics:
width: 1280
height: 800
principal point ppx: 644.835938
principal point ppy: 408.008484
focal length fx: 638.267151
focal length fy: 638.466492
distortion coefficients: -0.0527738594 0.0547797643 -0.0014833275 -0.0009916656 -0.0161027312
distortion model: BROWN CONRADY

RGB extrinsics in rectified coordinate system:
Rotation Matrix:
0.9999961257 -0.0027781273 -0.0002028379
0.0027779175 0.9999955893 -0.0010272462
0.0002056911 0.0010266788 0.9999994636

Translation Vector:
-58.9867782593 -0.0763279945 -0.0689293966

C:\Intel>

```

7 Developing Custom Apps with Calibration API

A programmable interface is provided for developing custom calibration apps. A compressed package of the interface and examples is included in the tool installation.

7.1 API Package Location

On **Linux**:

```

/usr
|
|---- share
|     |---- doc
|         |.... libscalibrationtool
|             |
|             |---- api
|                 |---- DynamicCalibrationAPI-Linux-2.14.2.0.tar.gz

```

On **Windows**:

```

C:\Program Files\Intel\CalibrationToolAPI
|----- 2.14.2.0
|     |
|     |----- api
|         |----- DynamicCalibrationAPI.Zip

```

7.2 API Package Structure

Copy DynamicCalibrationAPI-Linux-2.14.2.0.tar.gz or DynamicCalibrationAPI.Zip to your working folder and unzip it into similar directory structure as below:

```

DynamicCalibrationAPI
|
|---- 2.14.2.0
|     |---- CMakeLists.txt
|     |---- lib
|         |---- libDSDynamicCalibrationAPI.so or DSDynamicCalibrationAPI.dll
|     |.... examples
|         |---- DynamicCalibrator
|         |---- CustomRW
|         |---- CustomCalibration
|         |---- rs-calibration-converter
|         |---- simple-rw
|         |---- simple-fe
|     |---- Include
|         |---- DSDynamicCalibration.h

```

Developing Custom Apps with Calibration API

```

|      | |---- DSOSUtils.h
|      | |---- DSShared.h
|      | |---- DSCalData.h
|      |
|      | |..... attributions.txt
|      | |..... License.txt
|      | |..... README.md
|      | |..... target
|      | |---- librealsense
|      | |---- 3rdparty

```

Notes:

- API - libDSDynamicCalibrationAPI.so or DSDynamicCalibrationAPI.DLL is the calibration API library and DSDynamicCalibration.h is its main header file
- Under examples folder
 - ✓ DynamicCalibrator contains the source code for the Dynamic Calibrator
 - ✓ CustomRW contains the source code for the CustomRW tool
 - ✓ CustomCalibration contains sources for an example custom calibration of depth and RGB sensors with non-Intel algorithms, the example is described in detail in the *Intel® RealSense™ Product Family D400 Series Custom Calibration white paper*

7.3 Compiling Examples

The API package support cmake.

On Linux:

```

cd DynamicCalibrationAPI/2.14.2.0
mkdir build
cd build
cmake ..
-- The C compiler identification is GNU 5.4.0
-- The CXX compiler identification is GNU 5.4.0
-- Check for working C compiler: /usr/bin/cc
-- Check for working C compiler: /usr/bin/cc -- works
-- Detecting C compiler ABI info
-- Detecting C compiler ABI info - done
-- Detecting C compile features
-- Detecting C compile features - done
-- Check for working CXX compiler: /usr/bin/c++
-- Check for working CXX compiler: /usr/bin/c++ -- works
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
-- Detecting CXX compile features
-- Detecting CXX compile features - done
-- CMAKE_SYSTEM_PROCESSOR=x86_64
-- BUILD_SHARED_LIBS=ON
--
LIBRS_INCLUDE_DIR=/home/gwen/dc/test/DynamicCalibrationAPI/2.13.1.0/librealsense/i
nclude

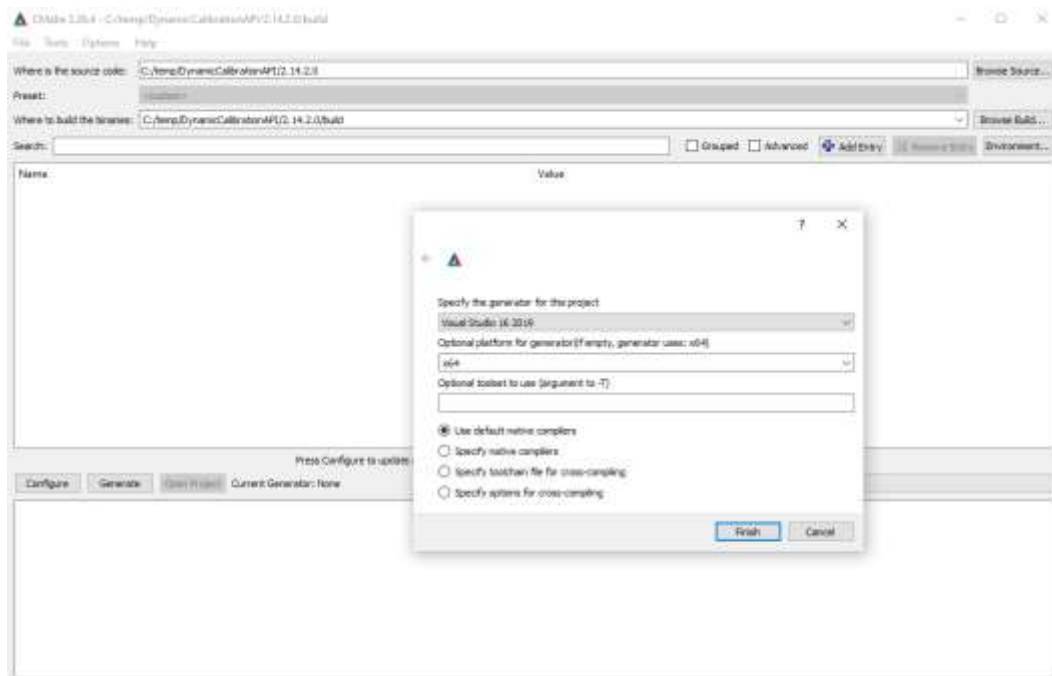
```

Developing Custom Apps with Calibration API

```
--
LIBRS_LIBRARY_DIR=/home/gwen/dc/test/DynamicCalibrationAPI/2.14.2.0/librealsense/lib/linux/x86_64
-- DCAPI_INCLUDE_DIR=/home/gwen/dc/test/DynamicCalibrationAPI/2.14.2.0/Include
-- DCAPI_LIBRARY_DIR=/home/gwen/dc/test/DynamicCalibrationAPI/2.14.2.0/lib
-- Configuring done
-- Generating done
-- Build files have been written to:
/home/gwen/dc/test/DynamicCalibrationAPI/2.14.2.0/build
make -j8
```

On **Windows**:

Config cmake for compiling CalibrationToolAPI\2.14.2.0 and generate project files for Visual Studio 2019 and x64 platform.



Open rs-dynamic-calibration-api-samples.sln under build directory to compile.

7.4 API Documentation

For details of the calibration API and examples, please refer to the **Intel® RealSense™ Product Family D400 Series Software Calibration Tool Programmer Guide**.

8 Troubleshooting

Table 8-1 Troubleshooting

Issue	Resolution
<p>With targeted calibration on Linux, after rectification phase is successfully completed, sample app windows may become black</p>	<p>The most like cause is the kernel not improperly patched or os booted into the non-patched kernel. Please follow instructions to redo the patching in installation section Video4Linux backend.</p> <p>By default, Ubuntu auto kernel update feature is enabled. So you may experience the same screen blacking issue again if your Ubuntu kernel is automatically updated.</p> <p>To avoid this, you can choose to do one of the following:</p> <p>Method 1 - Pin/hold your kernel Examples: Discover the current latest kernel currently installed. It's the one with the biggest version number.</p> <pre>\$ dpkg -l grep linux-image \$ uname -r 4.4.0-79-generic \$ sudo apt-mark hold linux-image-4.4.0-79-generic linux-image-4.4.0-79-generic set on hold. \$ sudo apt-mark hold linux-headers-4.4.0-79-generic \$ sudo apt-mark hold linux-image-extra-4.4.0-79-generic linux-image-extra-4.4.0-79-generic set on hold. \$ dpkg --get-selections grep linux-image-4.4.0-79-generic linux-image-4.4.0-79-generic hold</pre> <p>Method 2 - Disable automatic updates on Ubuntu:</p> <p>Open the Unity Dash Search for 'Software & Updates' Select the 'Updates' tab. Change 'Automatically check for updates' from 'Daily' to 'Never'. This setting will stop the system from checking for ANY updates</p>
<p>In certain cases, when the device is badly out of calibration or negative effect of lighting conditions, dynamic calibration may not be able to find the right correction and fail to complete the process.</p>	<p>Try to find a different scene or environment to run dynamic calibration again. If multiple tries do not resolve the issue, the device may shifted too much for Dynamic Calibration Tool to recover and require to perform a complete recalibration with OEM Calibration Tool.</p>

Troubleshooting

<p>Devices calibrated through targeted dynamic calibration shows greater depth error than expected (i.e., more than 2%)</p>	<p>Follow target setup instruction to check the target physical sizes. Target size is critical to calibration accuracy.</p>
<p>With modules with old calibration tables, target calibration does not change calibration on device</p>	<p>A simple workaround is to perform a target-less dynamic calibration on the device first and then targeted calibration would have no issue. Additional calibration would have no such limitation.</p>

9 Appendix A - Calibration Check

This section describes two methods to check the camera calibration. The first method is a quick and easy way to check the alignment of left and right cameras. The second method is to test the accuracy of depth measurement.

The two methods are explained below:

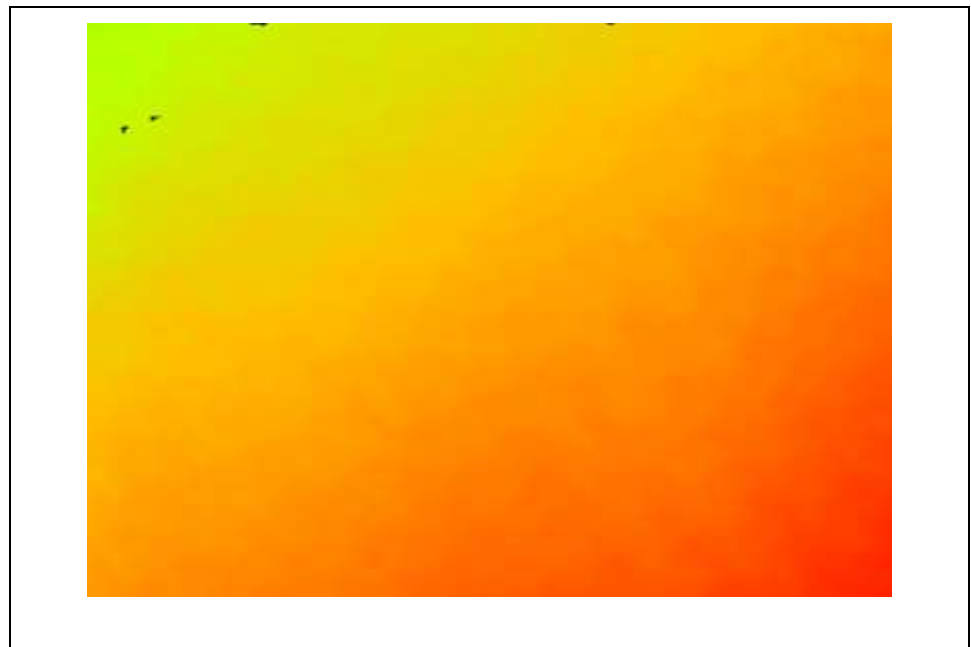
9.1 Quick Check

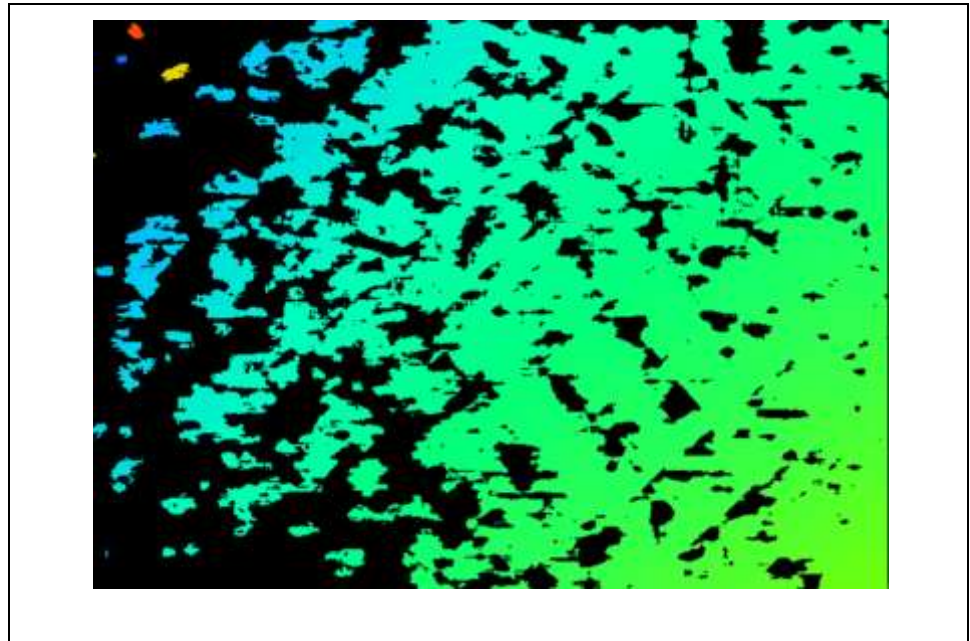
With the quick check, the user should view the depth image from the camera with a tool that can display depth, Intel® RealSense™ Viewer or Depth Quality Tool in Intel® RealSense™ SDK 2.0.
<https://github.com/IntelRealSense/librealsense/releases/latest>

Point the camera to a flat surface such as a wall about 1 to 2 meters away (3 to 6 feet). Avoid black surfaces.

Visually inspect the depth image display of the wall. A lot of black dots or holes on the image is an indication of the camera being out of calibration.

Figure 9-1. Calibration Check – Camera in Calibration



Appendix A - Calibration Check**Figure 9-2. Calibration Check – Camera Out of Calibration**

9.2 Accuracy Check

This procedure should be used to check the accuracy of the camera. Place the camera in parallel to a flat wall and exactly two meter (2000 mm) away.

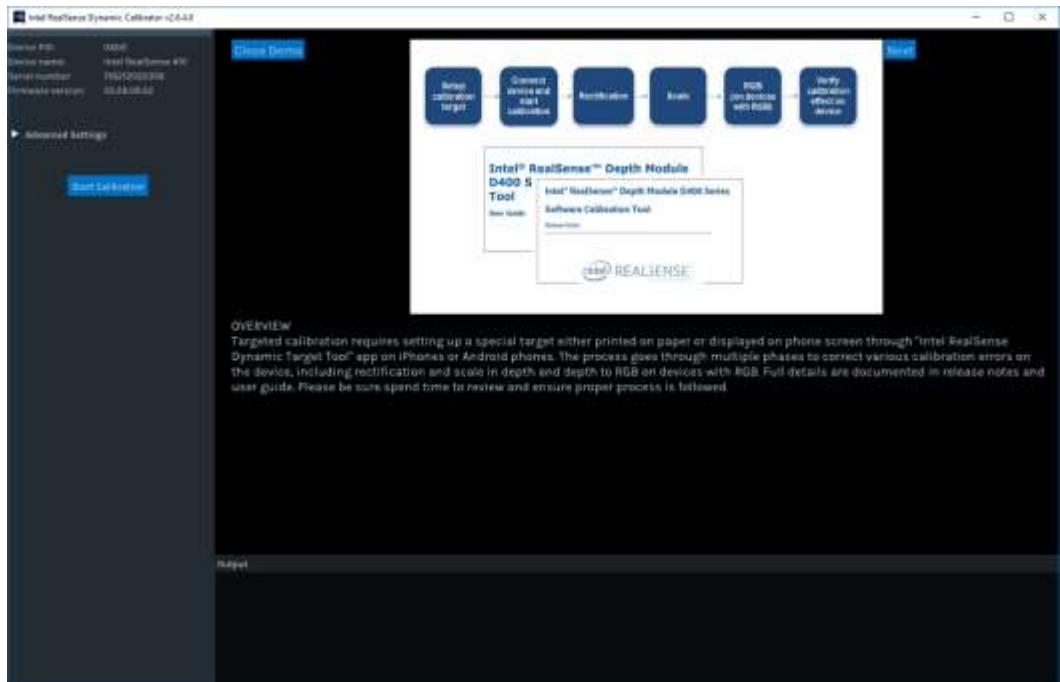
Once the camera is placed in its position, Use Intel® RealSense™ Viewer or Depth Quality Tool to measure the absolute distance. For a flat surface at a distance of 2 meter the absolute distance should be within 2% or better at 2 meter (2000mm).

If the distance is not within the defined range, then the camera needs to be calibrated.

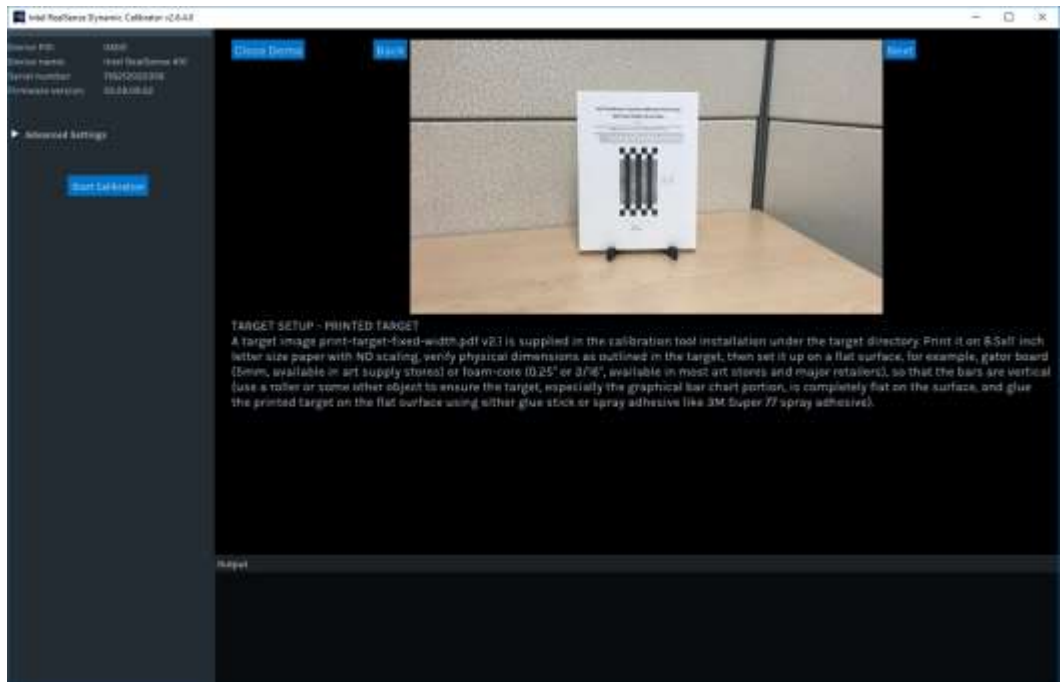
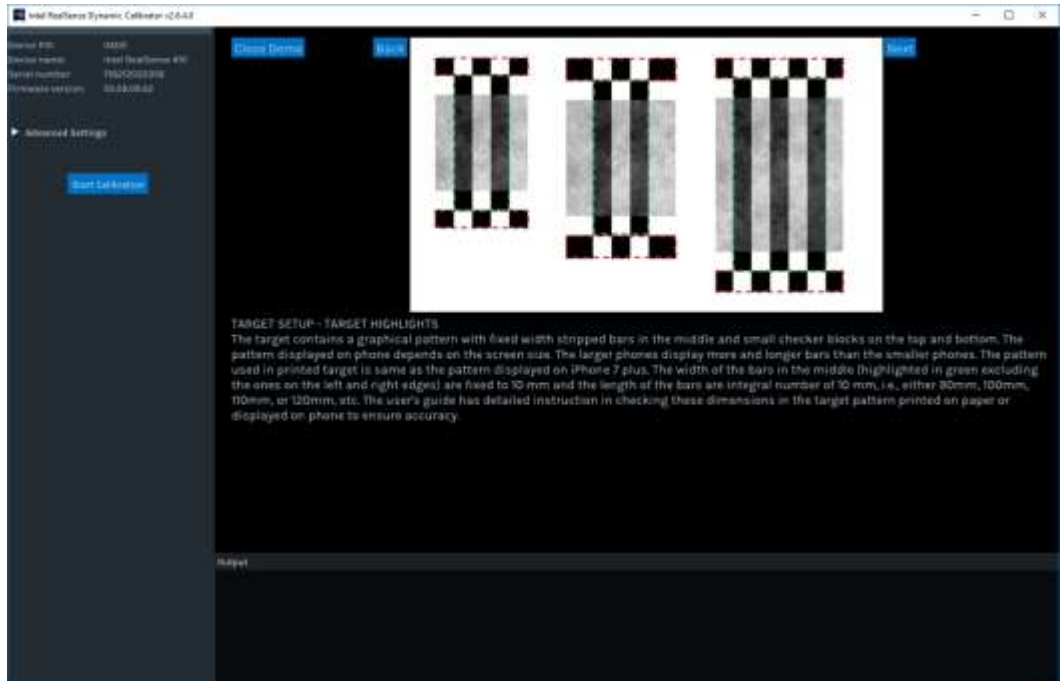
10 Appendix B - Calibration Process Highlights

10.1 Targeted Calibration

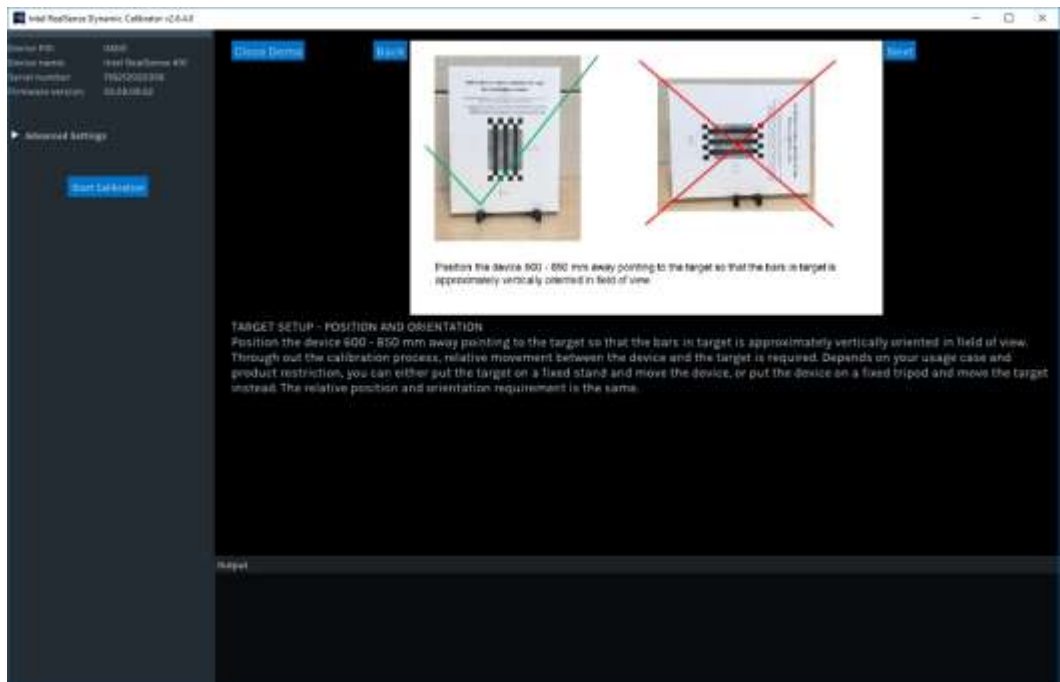
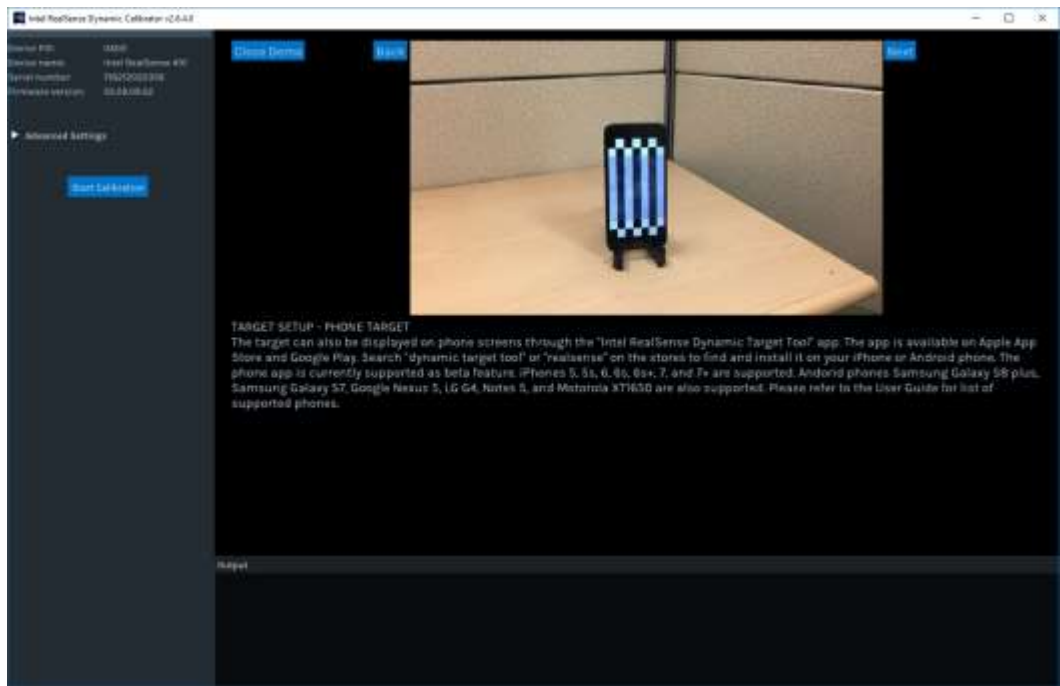
Targeted calibration requires setting up a special target either printed on paper or displayed on phone screen through "Intel RealSense Dynamic Target Tool" app on iPhones or Android phones. The process goes through multiple phases to correct various calibration errors on the device, including rectification and scale in depth and depth to RGB on devices supporting RGB. Targeted calibration should be used to calibrate all D400 Series devices if a target setup is permitted.



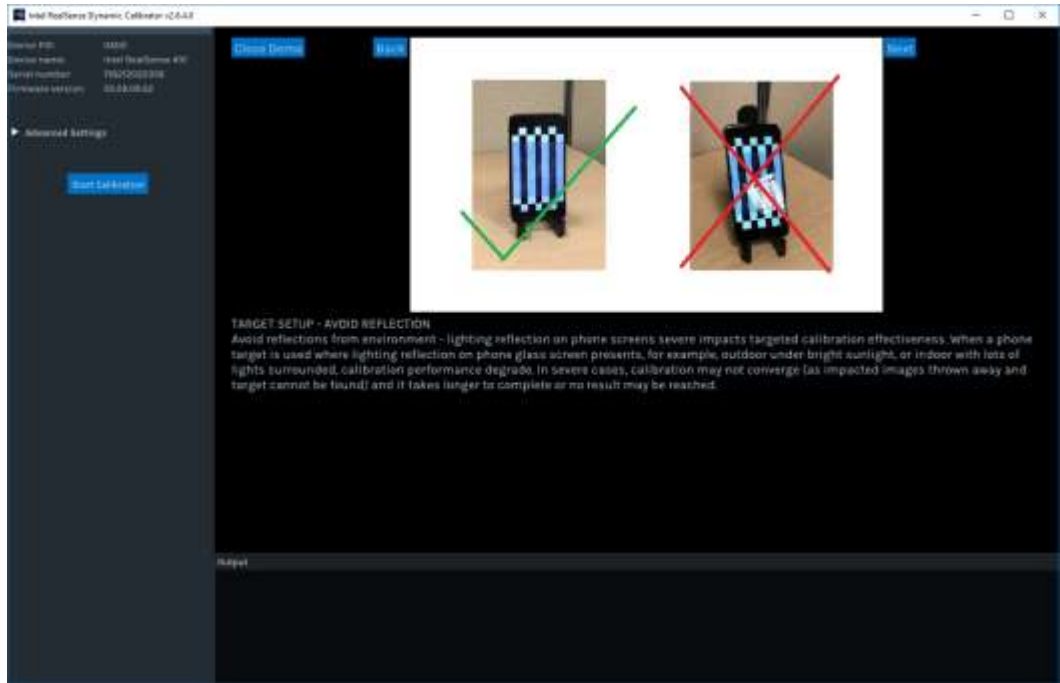
Appendix B - Calibration Process Highlights



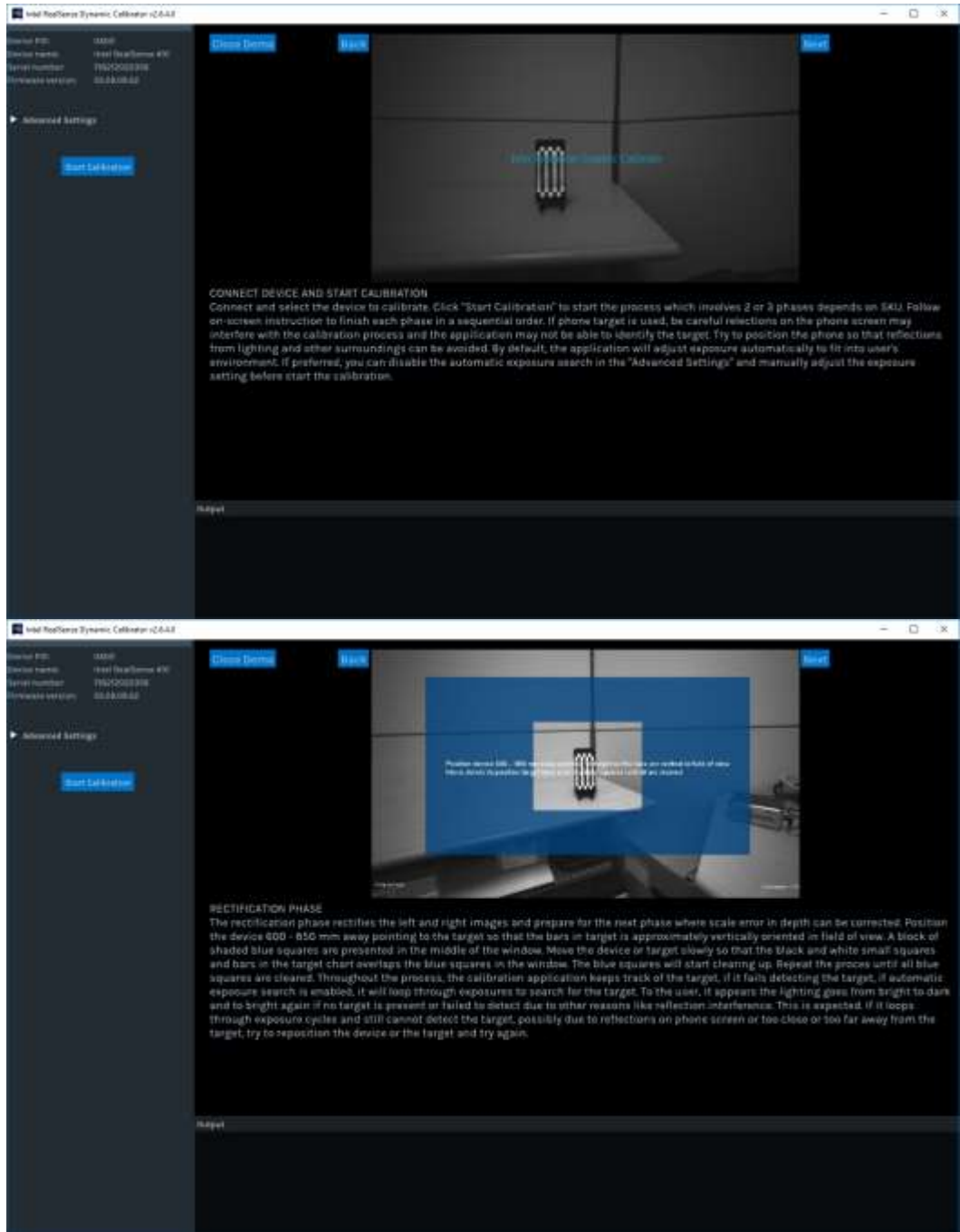
Appendix B - Calibration Process Highlights



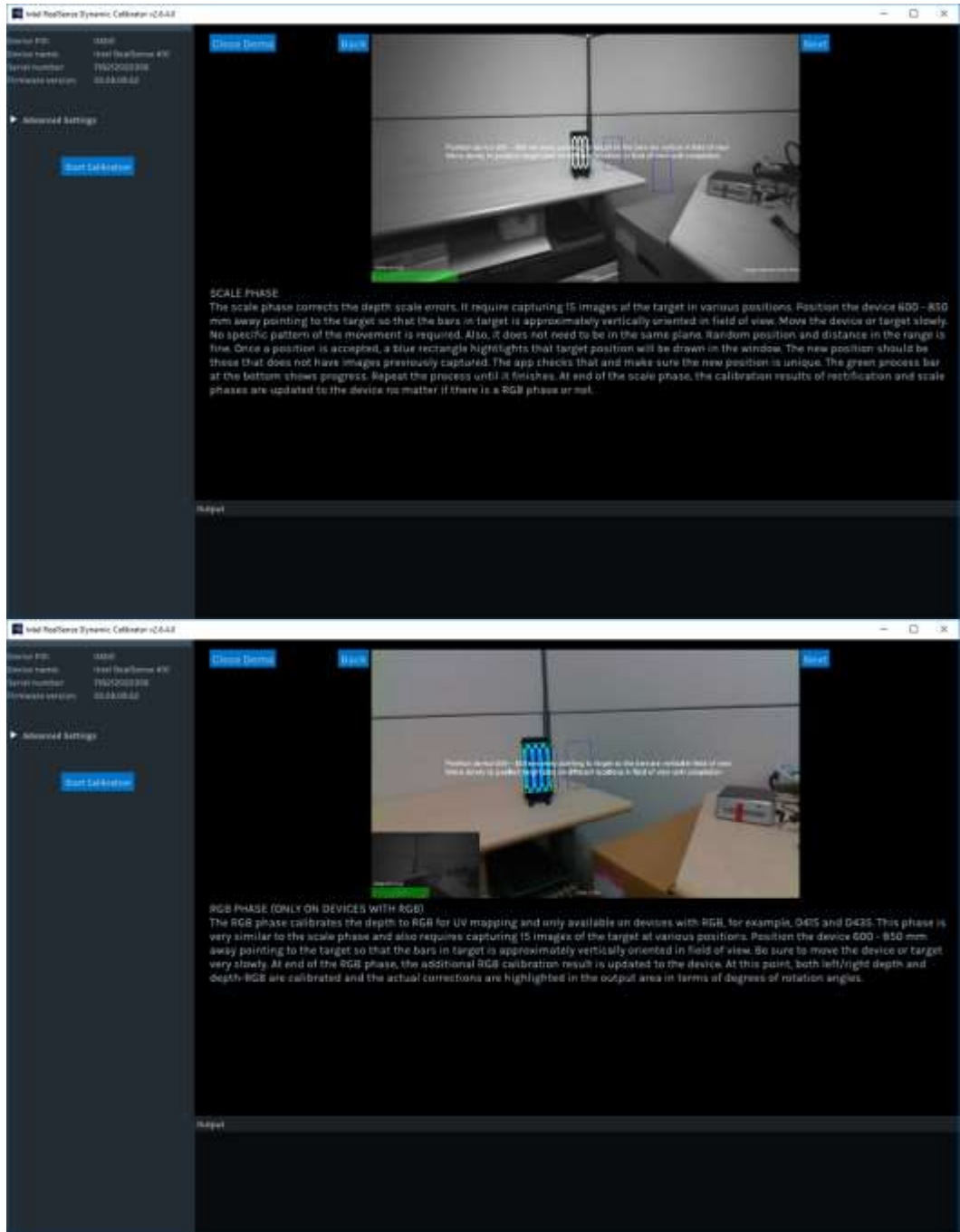
Appendix B - Calibration Process Highlights



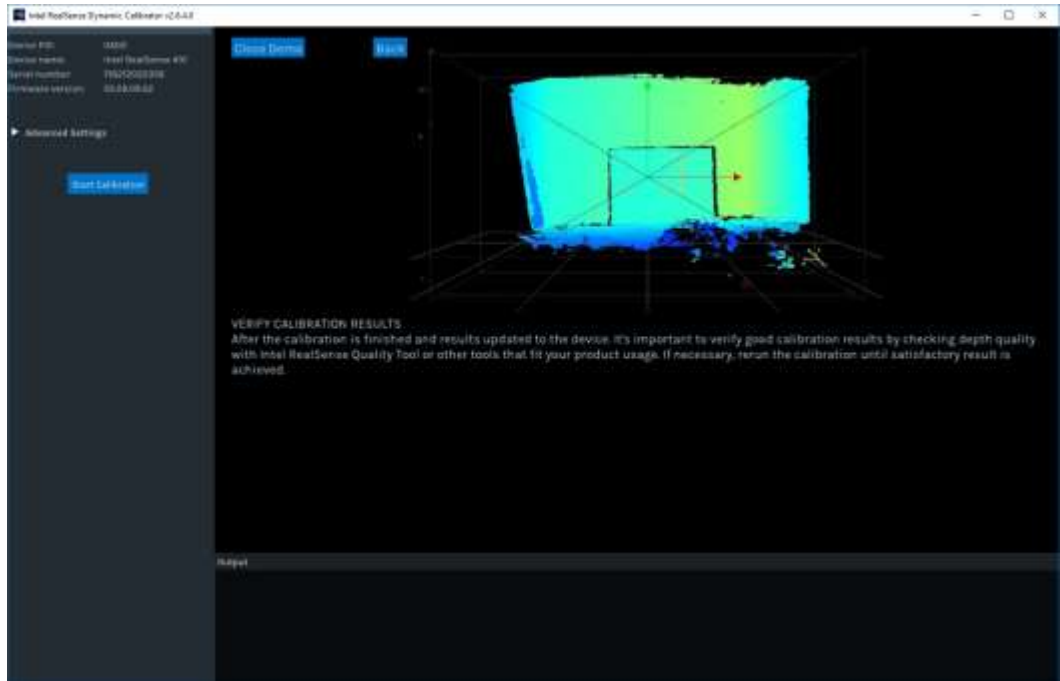
Appendix B - Calibration Process Highlights



Appendix B - Calibration Process Highlights



Appendix B - Calibration Process Highlights



11 Appendix C - Intel® RealSense™ Dynamic Calibration Target Setup

11.1 Introduction

Dynamic Calibration is used to re-calibrate the Intel® RealSense™ Product Family D400 Series cameras. The process requires a special calibration target that can be printed from a supplied target pdf image.

This section provides instructions to setup the target for dynamic calibration.

11.2 Target Image

The dynamic calibration target image `print-target-fixed-width.pdf` can be found in the Dynamic Calibration installation directory or downloaded from the RealSense website. The latest target image version is v2.1.

- Under the target subdirectory in Dynamic Calibration installation directory. For example, `C:\Program Files\Intel\CalibrationToolAPI\2.14.2.0\target\print-target-fixed-width.pdf`
- Download from RealSense website:
<https://dev.intelrealsense.com/docs/dynamic-calibration-print-target>

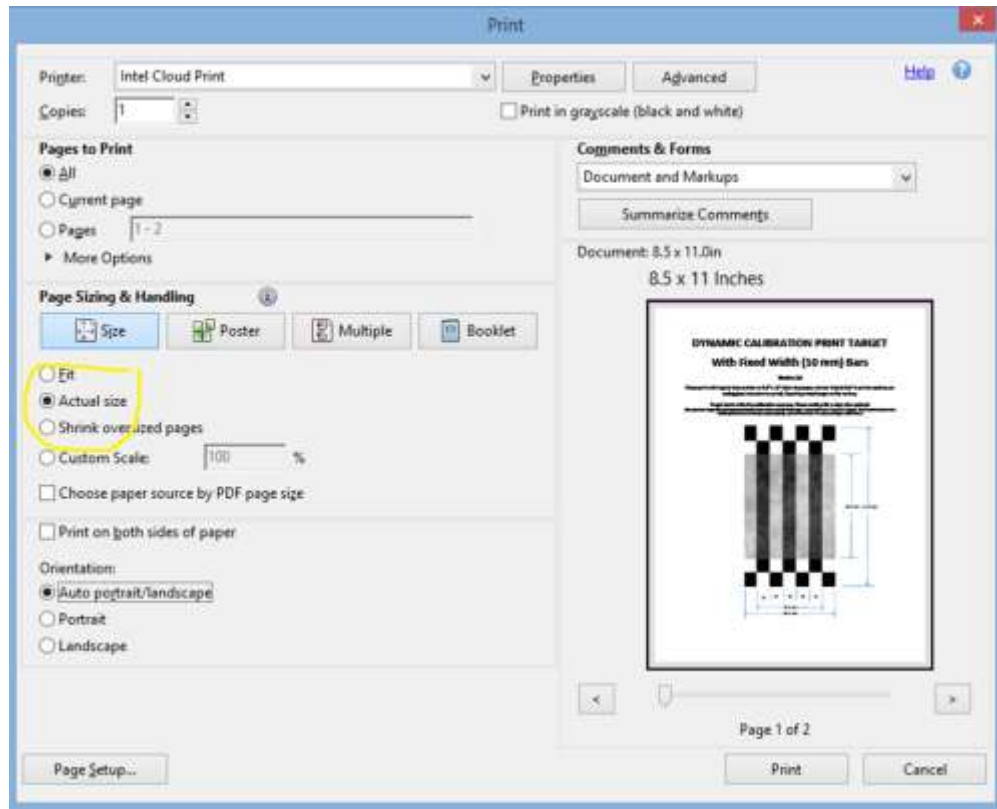
11.3 Printed Target Setup

Please follow the steps below to print and setup the target. The instruction is also included in the ***print-target-fixed-width.pdf***.

11.3.1 Print target on letter size pager

Please print ***print-target-fixed-width.pdf*** with regular laser printer on 8.5" x 11" letter size paper, choose "actual size" in printer options, **no scaling**.

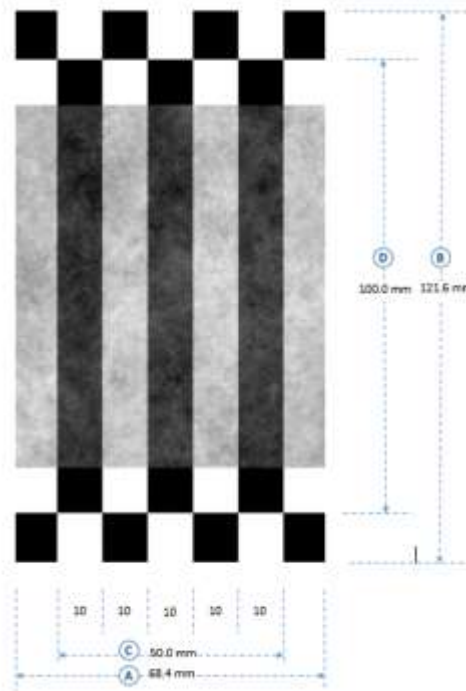
Figure 11-1 Target Print Options



11.3.2 Check the printed target dimension and accuracy

After the target is printed, please check the physical target image on the paper. The specific dimensions are marked around the image.

Figure 11-2 Target Verification



Target precision is critical to calibration accuracy. Please verify key features below with a ruler after printed.

- **Target size** - overall image should be 68.4 mm wide (A) and 121.6 mm tall (B)
- **Bar size** - the 5 vertical bars in the middle (3 black and 2 white, in the order, black-white-black-white-black) in equal spacing each 10 mm wide, total 50 mm wide (C), and the vertical bar length is 100.0 mm (D)

11.3.3 Setup the printed target on a flat surface

Setup the printed target on a flat surface, for example, a flat foam board or flat wall, so that the **bars are vertical** (use a roller or some other object, to ensure the target is flat on the surface).

Tape the printed target on the flat surface after confirming that **the target is flat**.

Figure 11-3 Printed Target on Flat Surface



11.4 Perform Dynamic Calibration

For detailed calibration process, please refer to section **4.6 Targeted Calibration** in *Calibration with Dynamic Calibrator App* Chapter in this document.

In a summary,

- Place the camera about 60 – 85 cm away and point to the target. It should not be too close or too far. User will need to move the camera in the specified range to find a distance that works best. In most cases, distance of around 70 cm is sufficient.
- On the host, start Intel RealSense Dynamic Calibrator application.

On Linux

```
/usr/bin/Intel.Realsense.DynamicCalibrator
```

On Windows

Double click the “Dynamic Calibrator” icon on Windows desktop to start the calibrator app.