

**Vivante DPU VTuner**

**User Guide**

for

Software Release Version

Beta 1.1.0

August 2023Legal Notes

**COPYRIGHT INFORMATION**

This document contains proprietary information of Vivante Corporation and VeriSilicon Holdings Co., Ltd. They reserve the right to make changes to any products herein at any time without notice and do not assume any responsibility or liability arising out of the application or use of any product described herein, except as expressly agreed to in writing by Vivante and/or VeriSilicon; nor does the purchase or use of a product from Vivante or VeriSilicon convey a license under any patent rights, copyrights, trademark rights, or any other of the intellectual property rights of Vivante, VeriSilicon or third parties.

**DISCLOSURE/RE-DISTRIBUTION LIMITATIONS**

The information contained in this Vivante Confidential document may be directly re-distributed or may be adapted for re-distribution by Vivante customer SoC vendors who have licensed the related core IP, with the understanding that re-distribution is limited to those customers of the SoC vendor who have active and valid NDA or licenses applicable for the SoC which contains the related Vivante core IP. Otherwise, the information contained herein is not to be used by or disclosed to the third parties without the express written permission of an officer of Vivante Corporation or VeriSilicon Holdings Co., Ltd.

If you plan to customize this document for adaptive inclusion in documentation intended for your customers, VeriSilicon requests a pre-release copy of the adaptation for review and approval. In the adapted document please identify the technology and features described in this document using the Vivante trademark and include a reference to the copyright owner, for example: "This section describes the Vivante® GPU and contains copyright material disclosed with permission of VeriSilicon®, who has authorized re-distribution of this material restricted to those NDA partners and licensees of [the SoC vendor] who are engineering [SoC vendor's product] which includes the discussed Vivante IP."

**TRADEMARK ACKNOWLEDGMENT**

VeriSilicon® and the VeriSilicon logo design are the trademarks or the registered trademarks of VeriSilicon Holdings Co., Ltd. Vivante® is a registered trademark of Vivante Corporation. All other brand and product names may be trademarks of their respective companies.

For our current distributors, sales offices, design resource centers, and product information, visit our web pages located at http://www.verisilicon.com.

For technical support, please email [vivante-support@verisilicon.com](mailto:vivante-support@verisilicon.com).

© 2016-2023, VeriSilicon Holding Co., Ltd.. Distribution Level B: Confidential — Distribution Restricted. All rights reserved.

Contents

[2 Overview 4](#_Toc136959285)

[2.1 Component Overview 4](#_Toc136959286)

[2.2 Operating System Compatibility 4](#_Toc136959287)

[3 Software Release Package 4](#_Toc136959288)

[4 VTunerServer Usage 5](#_Toc136959289)

[4.1 FPGA Configuration 5](#_Toc136959290)

[4.2 Cmodel Configuration 6](#_Toc136959291)

[5 VTunerClient Usage 6](#_Toc136959292)

[5.1 Install and Uninstall Client 6](#_Toc136959293)

[5.2 Launch Client GUI 9](#_Toc136959294)

[5.3 Go to DPU Tuner Perspective 9](#_Toc136959295)

[5.4 Create a DPU Project 10](#_Toc136959296)

[5.5 Server Configuration 11](#_Toc136959297)

[5.6 Tune Features 12](#_Toc136959298)

[5.6.1 Gamma 12](#_Toc136959299)

[5.6.2 3DLUT 14](#_Toc136959300)

[5.7 View Log 20](#_Toc136959301)

[5.8 Export and Import 20](#_Toc136959302)

[5.9 Refresh 22](#_Toc136959303)

# Overview

This document describes the Vivante VTuner Tools designed to remotely perform picture quality calibration and tuning of the Vivante Display Processing Unit (DPU) output effect running on an FPGA. It sends requests to the host via an HTTP protocol and controls DPU output effect.

The tool allows the user do,

1. Fine tune configurations in DPU modules.
2. Check the results real-time or through write back images.
3. Manage the setting, such as save, export and import, etc.

## Component Overview

|  |  |  |
| --- | --- | --- |
| Componet | File Name | Description |
| Server | VTunerServer | Monitors client requests and controls the DPU to execute corresponding operations. This could work with C-Model and FPGA. |
| Client | VTunerClient | Provides GUI Environment and sends the requests from GUI to the server via an HTTP protocol. |

## Operating System Compatibility

|  |  |
| --- | --- |
| Componet | System |
| Server | Ubuntu 18 x64 or other Unix-like operating systems |
| Client | Windows 10 x64 and Ubuntu 18 x64 |

# Software Release Package

The release packages will have names similar to the following examples:

VTunerClient:

VTunerClient-<version>-Linux-x86\_64-<date>-Install

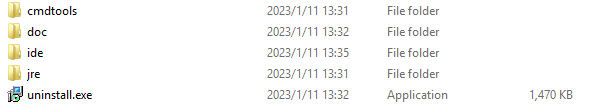
VTunerClient-<version>-Win32-x86\_64-<date>-Setup.exe

VTunerServer:

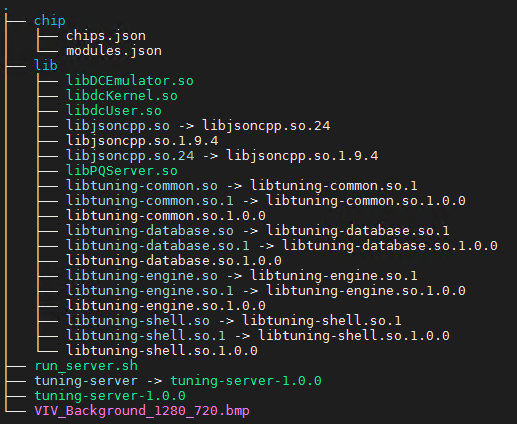
VTunerServer-<version>-cmodel-<version>-Linux-x86\_64-<date>.tgz or

VTunerServer-<version>-hardware-<date>. tgz

VTunerClient installation directory is as below:



VTunerServer package is as below:



# VTunerServer Usage

The Server runs on an FPGA via VTunerServer. The Server could handle requests from the Client via Ethernet and directly control DPU output effect.

**Note:** Make sure the Server is ready before you start up Client.

## FPGA Configuration

* + - 1. Startup FPGA.
      2. Run application to setup pipeline.

example:

*" modetest -M vs-drm -a -s 133@110:1024x600 -P 32@110:1024x600@UYVY"*

Note: For Color Management and Gamma2, make sure properties G2\_EN and G4\_EN are opened.

example:

*vsdb disp*

*vsdb/disp> set\_property 32@42:1*

*vsdb/disp> set\_property 110@112:1*

*vsdb/disp> commit*

* + - 1. Copy the package of VTunerServer to FPGA/board.
      2. tar -xzpvf VTunerServer-\*.tgz
      3. cd tuning-server
      4. ./run\_server.sh -port 6060

The port and IP address will be used to connect the client.

## Cmodel Configuration

1. Copy the package of VTunerServer to FPGA/board.
2. tar -xzpvf VTunerServer-\*.tgz
3. ./run\_server.sh -port 6060

The port and IP address will be used to connect the client.

After the Server is ready, the image “VIV\_Background\_1280\_720.bmp” will be displayed in the X11 window.

# VTunerClient Usage

VTunerClient provides the GUI to tune DPU modules and send the requests to Server via network.

The basic work flow is as below:

1. Launch VTunerClient.
2. Configure Server IP and port in [**Server Configurations**](#_Server_Configuration).
3. Create/Select a DPU project.
4. Select the feature need tuning in the [**Features**](#_Tune_Features) view.
5. Change the options of the selected feature, then data will be sent to the server in real time.

## Install and Uninstall Client

VTunerClient installations support the following platforms:

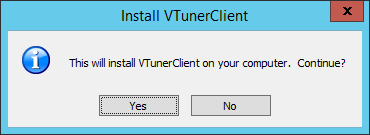
* **Linux**
* **Windows**

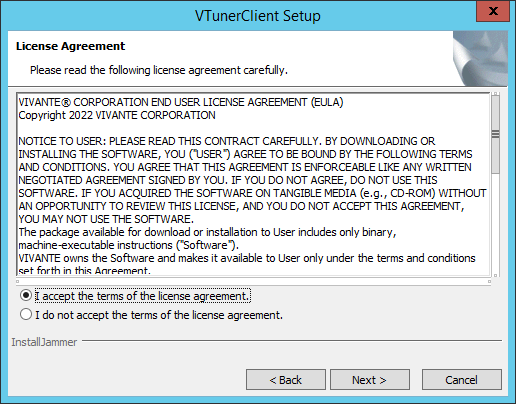
VTunerClient has been verified to work Windows 10 and Ubuntu 18.04.

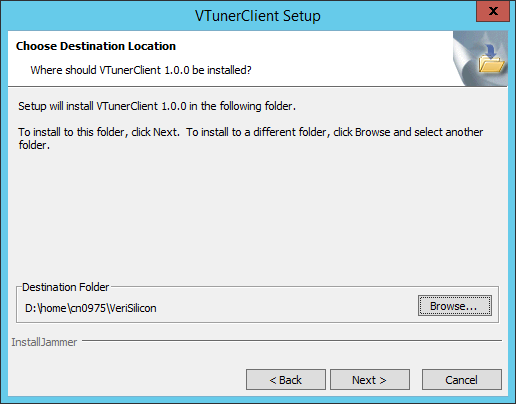
Note: on Linux, 3DLUT module depends on "MKL" dynamic link library.

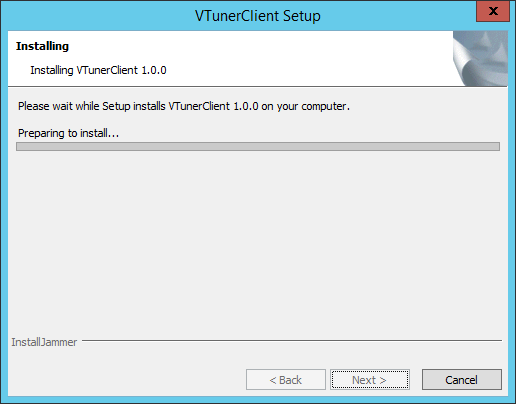
1. **Install Client**

Run VTunerClient-<version>-Linux-x86\_64-<date>-Install or VTunerClient-<version>-Win32-x86\_64-<date>-Setup.exe to launch the installation wizard (x.x.x is the version). Follow the wizard to finish installation.



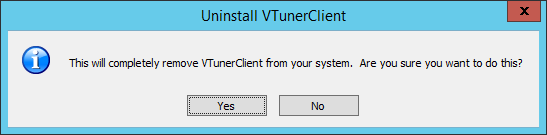






1. **Uninstall Client**

Run installation\_dir/uninstall to launch the uninstallation wizard. Follow the wizard to finish uninstallation.



## Launch Client GUI

1. **Run Client**

Run installation\_dir/ide/vtunerclientx.x.x in a BASH on Linux. Double click the shortcut 'VTunerClient x.x.x' on desktop or run installation\_dir\ide\vtunerclientx.x.x.

1. **Select Workspace**

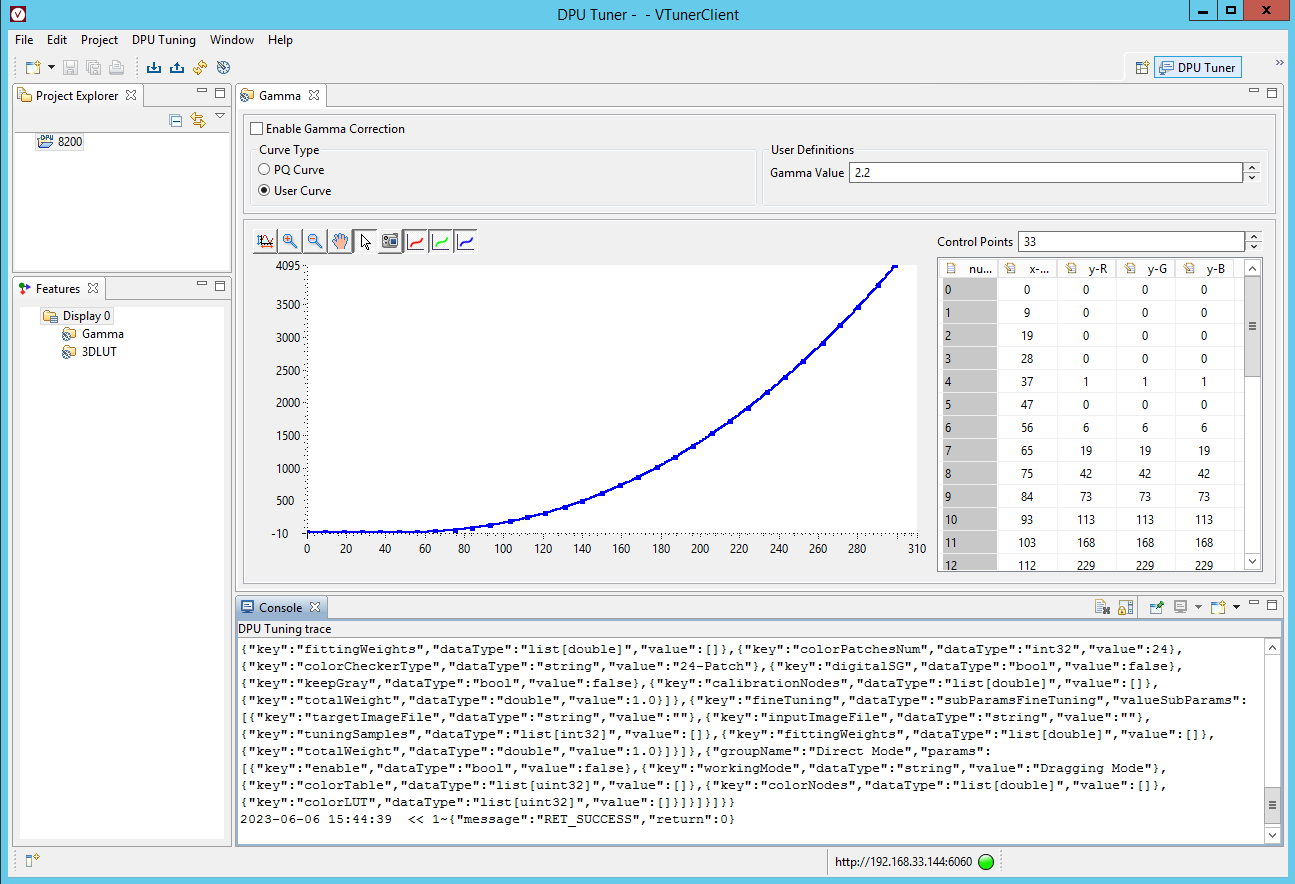
When VTunerClient is opened, the **Workspace Launcher - Select a workspace** dialog will pop up by default. Specify the workspace location and click ok, then will go to the workbench.

## **Go to DPU Tuner** Perspective

Each Workbench window contains one or more perspectives. A perspective defines the initial set and layout of views in the Workbench window. Within the window, each perspective shares the same set of editors and provides a set of functionality aimed at accomplishing a specific type of task or works with specific types of resources.

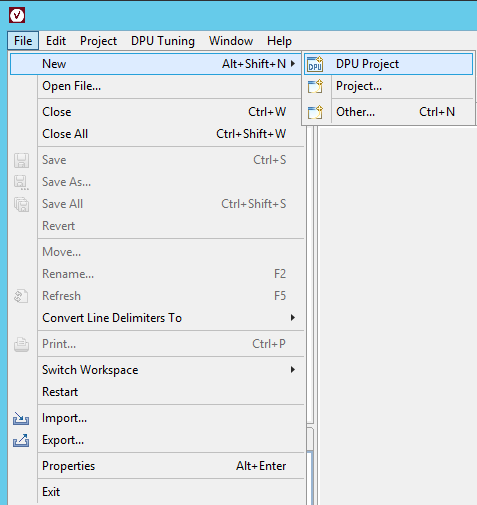
Perspective control what appears in certain menus and toolbars. They define visible action sets, which you can change to customize a perspective. You can save a perspective that you build in this manner, making your own custom perspective that you can open again later.

DPU Tuner perspective combines all the views and toolbars that you would while tuning DPU features. Views contains Project Explorer, Features, Details and Console. Toolbar provides shortcut for Import, Export, Refresh and Server Configuration.



## Create a DPU Project

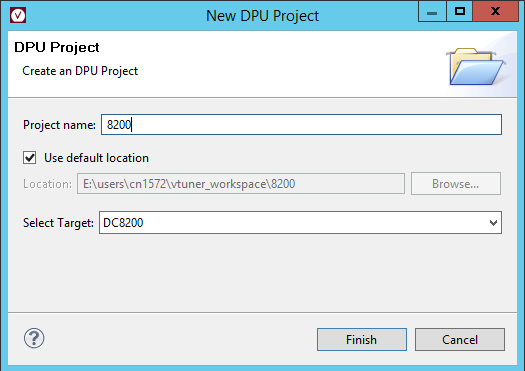
DPU Project could help manage the DPU tuning settings. From the main menu, choose **File-->New-->DPU Project**.



In the **DPU Project** dialog enter the Project name. **Note:** The name of the project should only consist of letters, numbers and underscores, special characters such as "@, &, /, \, :, \*, ?, ", <, >, |" should not be used. For the purposes of this tutorial we will use 8200.

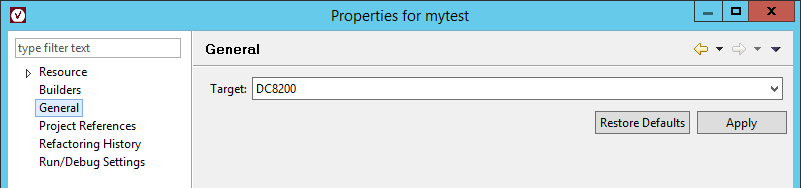
Check the **Use default location** checkbox. This will cause our new directory to be created in our workspace. Note, the directory path is displayed.

In the **Select Target:** combo box, select the target you want to tune.



After the project name and target are selected, click the **Finish** button. The new project is now created. Then select a DPU project, all supported features of the current target will be shown in the Features view.

After project is created, could modify the target by right click a DPU project, select **Properties -> General** to check the current target or change to another target.



## Server Configuration

This is used to connect server by server IP address and port.

**Connection Type:** Only HTTP is supported now.

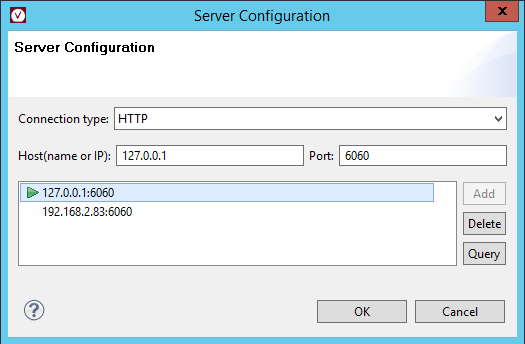
**Host(name or IP):** Specify the host name/IP and port. Please make sure the host is accessible.

Click “**Add**” to add the host into the list.

Click “**Delete**” to delete all selected host items from the list.

Click “**Query**” to test if the current host is accessible.

Click “**OK**” to apply the current selected host as the server.



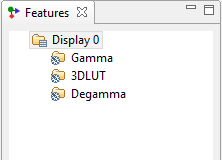
There is a status line showing connection status at the bottom of the window. Green light means server is accessible. Red light means server is inaccessible.



## Tune Features

**Features** View shows all the supported features of the target specified in the current selected project.

Click the feature item in Features View, the corresponding module will be shown in the **Details** view.

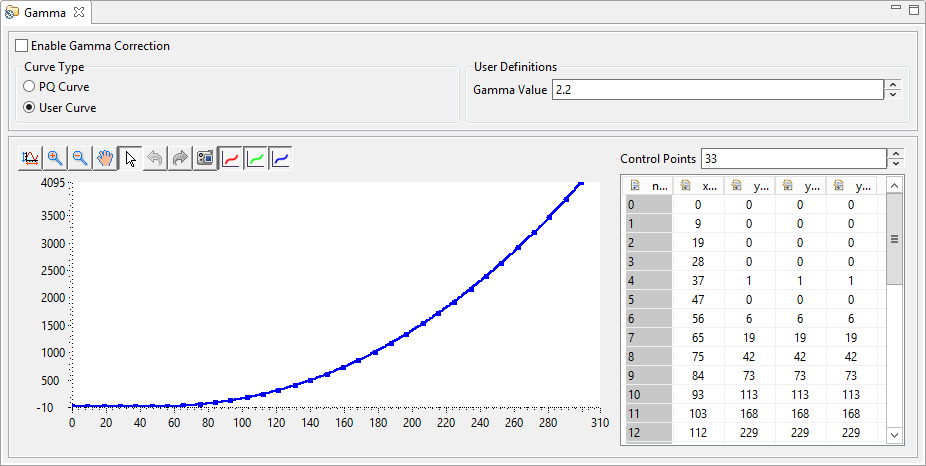


### Gamma

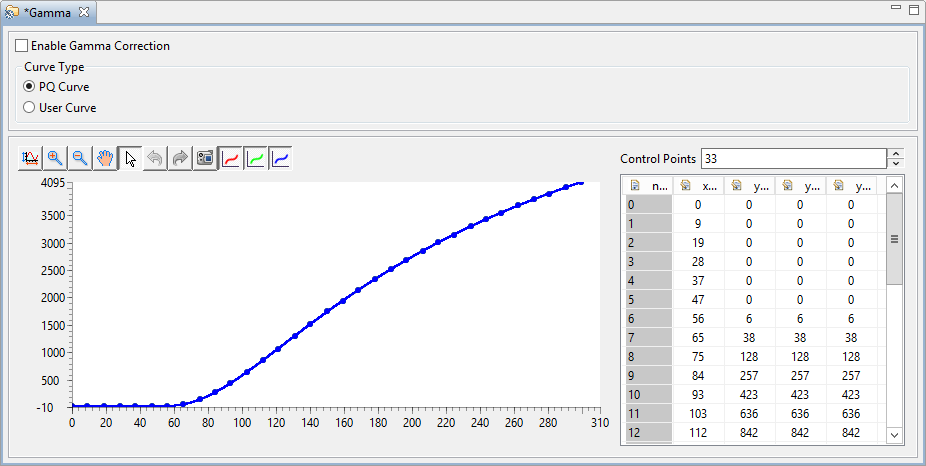
Gamma provides two types of curves to generate the R, G, B table.

**User Curve:**

Change the “**Gamma** **value”** to control the gamma curves.



**PQ Curve:**



**Control Points:** Specify the number of control points in the curve.

In the left plot, each point is draggable. Drag these points to tune the gamma mapping values.

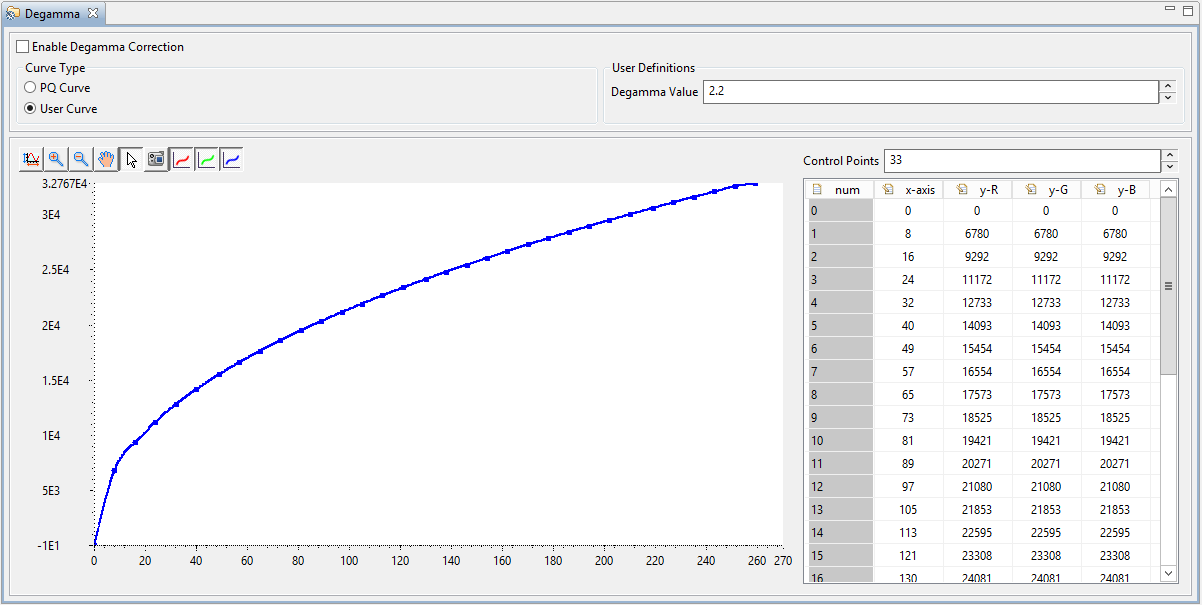
In the right table, double click the cell to modify the value of each point.

### Degamma

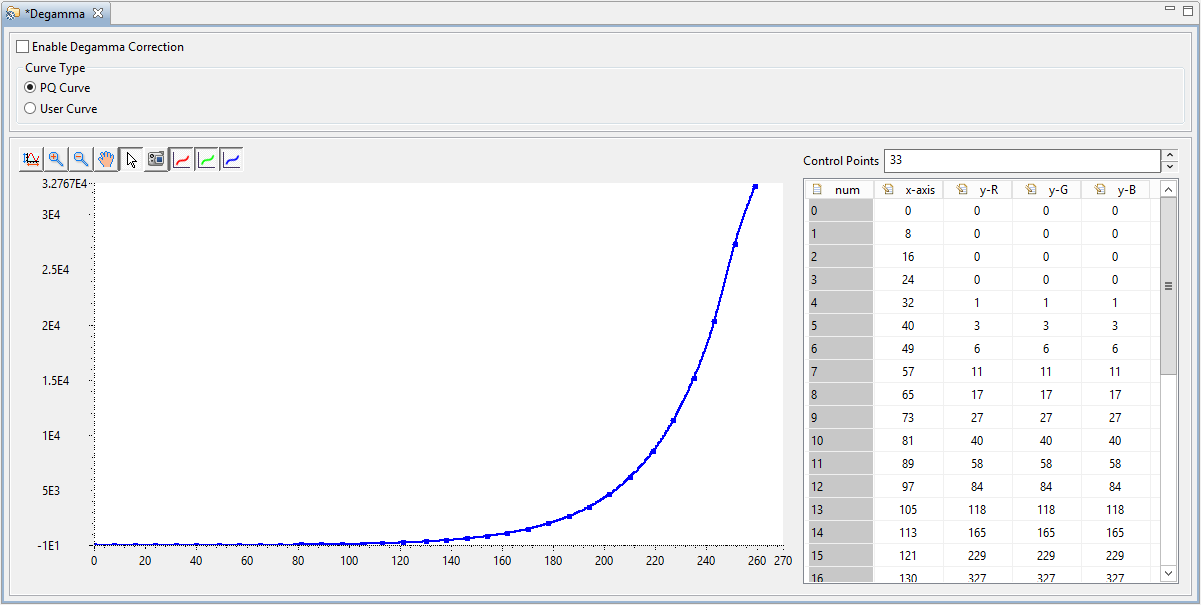
Degamma provides two types of curves to generate the R, G, B table.

**User Curve:**

Change the “**Degamma** **value”** to control the gamma curves.



**PQ Curve:**



**Control Points:** Specify the number of control points in the curve.

In the left plot, each point is draggable. Drag these points to tune the degamma mapping values.

In the right table, double click the cell to modify the value of each point.

### 3DLUT

3DLUT provides two modes to control the color mapping look up table. The size of look up table is 17x17x17.

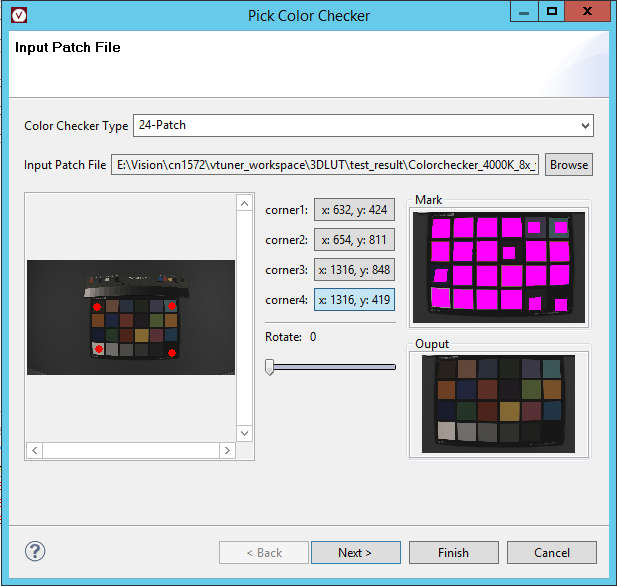
1. **Precision Mode**

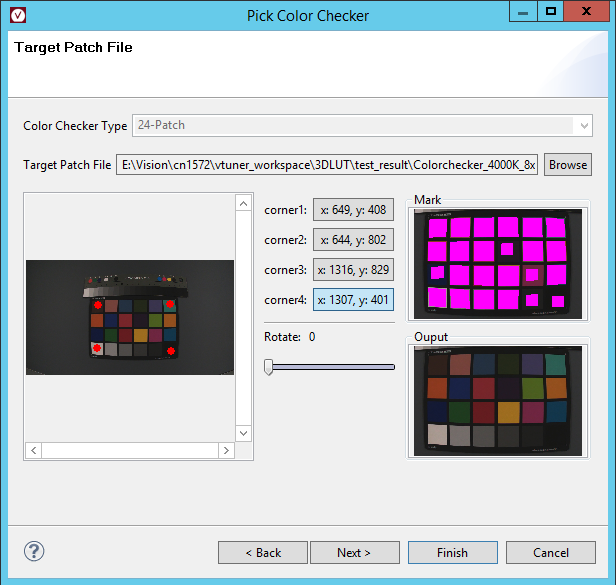
Precision mode provides two target-oriented ways to generate 3d look up table. It has two modes:Basic Calibration and Fine Tuning.

**Basic Calibration:** Generate 3d look up table by using standard color checkers. The **Color Checker Type** supports 24 patches color checkers, 140 patches color checker and csv file.

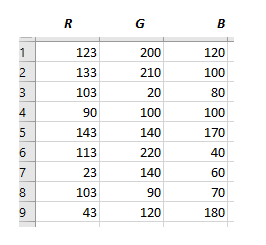
Click **Open...** button to select the location of the input/target file and select **Color Checker Type** in the popup dialog**.**

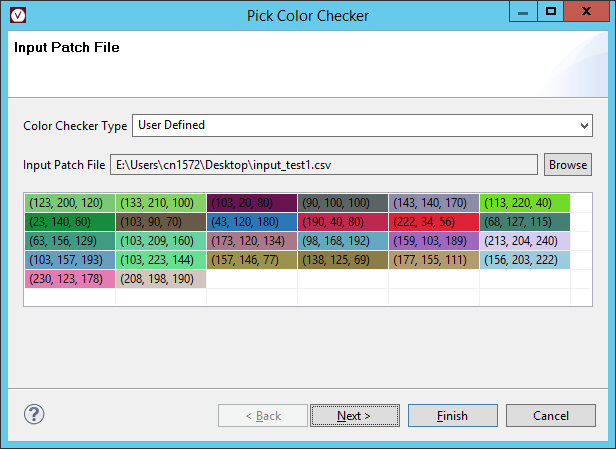
For **Color Checker Type** is 24-patch or 140-patch, select the location of the input/target color checker image. Then select the center of the color patch at the four corners with a mouse, and the colors will be extracted from the image as average of a patch. Click the button of each corner can re-choose this corner. Drag the slider under **Rotate** to rotate the image.

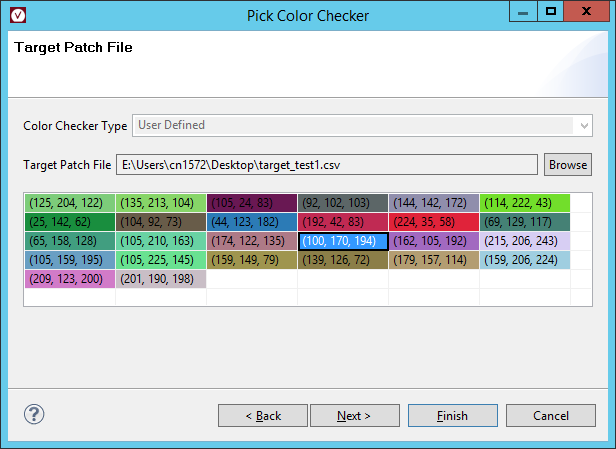




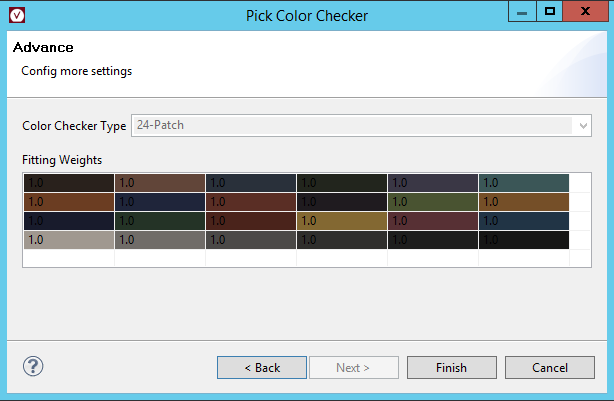
For **Color Checker Type** is User Defined, select the location of the input/target csv file. The csv file describes the R/G/B value of each patch.







Click the **Next** button, a weight table for each color patch will be displayed, user could modify each weight of the color to generate the 3d look up table.



After click the **Finish** button, the global maximum and average Delta C and Delta E will be shown in the **Summary** table. The attributes of each color patch are shown in the **Attributes** table, including color patch index, weight and color of input, target and optimized, Delta C, Delta E before and after optimized.

Click **Create 3D LUT** button, a new 3d look up table will be generated again after modify the weights of each color**.**

**Coordinates** is the chromaticity coordinates of the current calibration node are being loaded.

**Enlarge** is enlarge mode of color channel values.

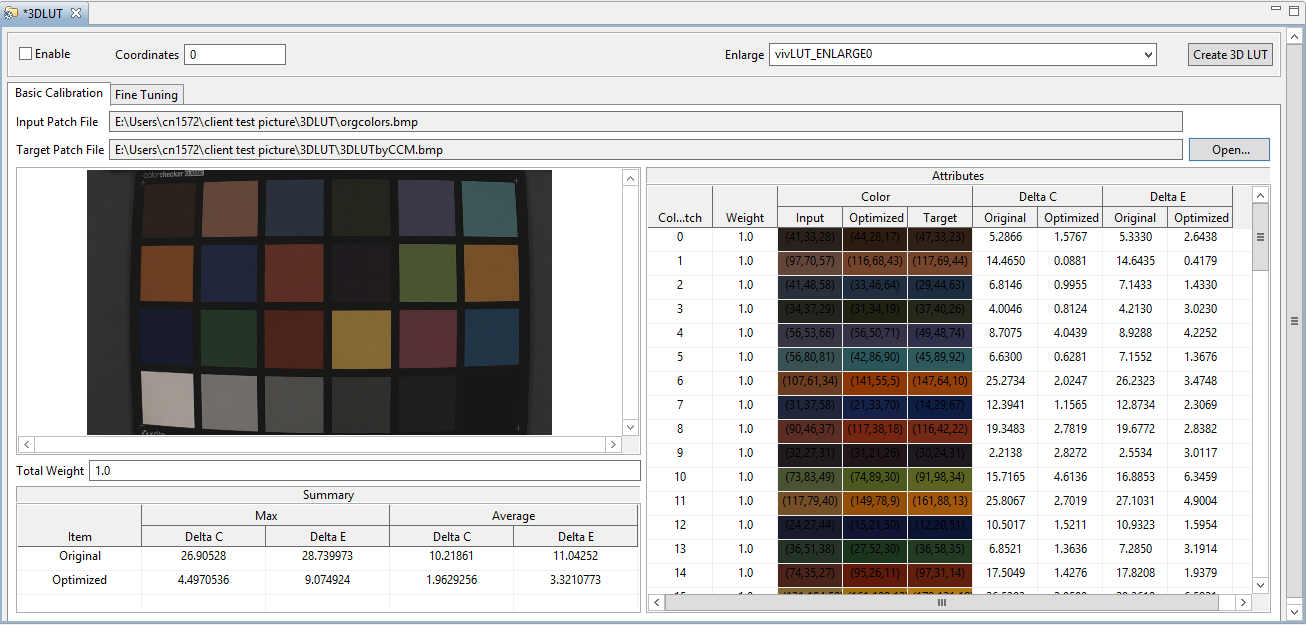
**vivLUT\_ENLARGE0**: Dose not enlarge the color channel values.

**vivLUT\_ENLATGE9**: Enlarges each color channel value by 2n, where

n = 9 - Bitwidth of the greatest value among the red, green and blue channels.

**vivLUT\_ENLATGE10**: Enlarges each color channel value by 2n, where

n = 10 - Bitwidth of the greatest value among the red, green and blue channels.

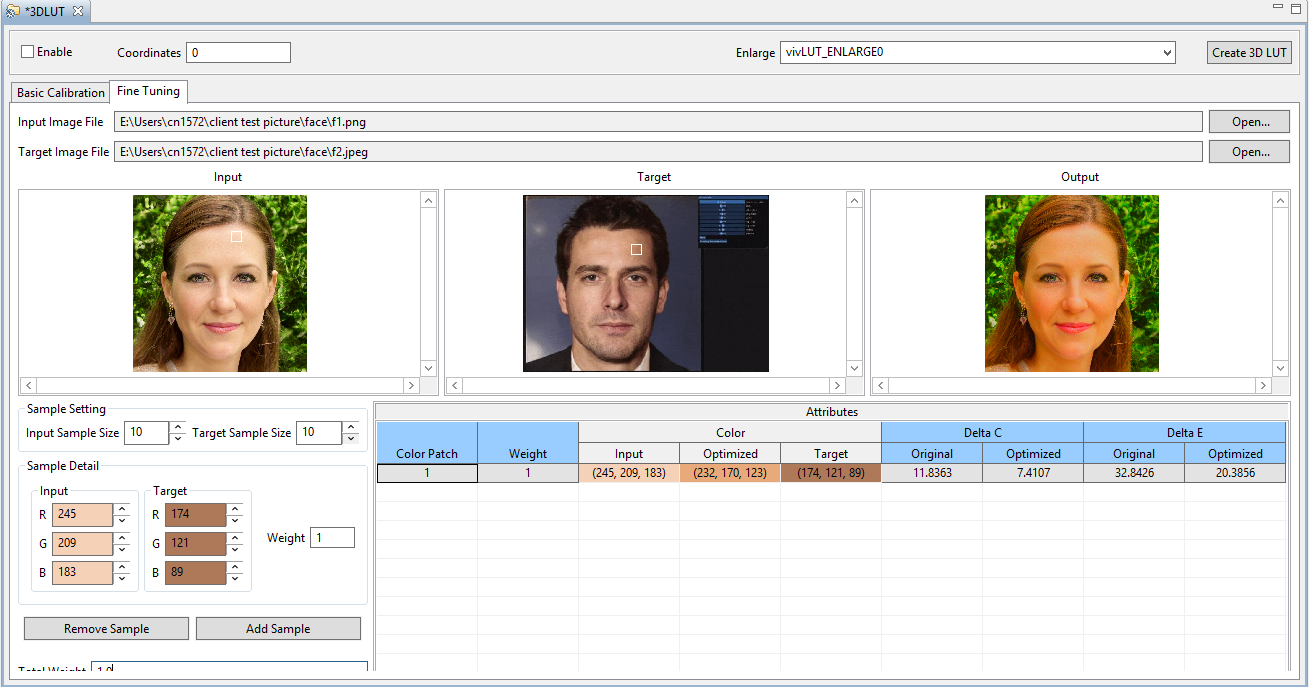


**Fine Tuning**: Map specified input color to target color and generate 3d look up table.

Click **Open…** button of **Input Image File** to select the location of the input image. Click **Open…** button of **Target Image File** to select the location of the target image. In **Sample Setting**, **Input Sample Size** specifies the sample patch size of the input image. Select the input color by double clicking on the input image and then the sample color’s R/G/B will be displayed in **Sample Detail**. Operation of target image is the same as the input, then input the **Weight** for sampled color. After clicking button **Add Sample**, the attributes of each sample patch will be shown in the **Attributes** table, including color patch index, weight and color of input, target and optimized, Delta C, Delta E before and after optimized.

Change the number of **Total Weight**, weight of each sample will be changed uniformly.

Click **Create 3D LUT** button, a new 3d look up table will be generated again after modify the weights of each color**.**



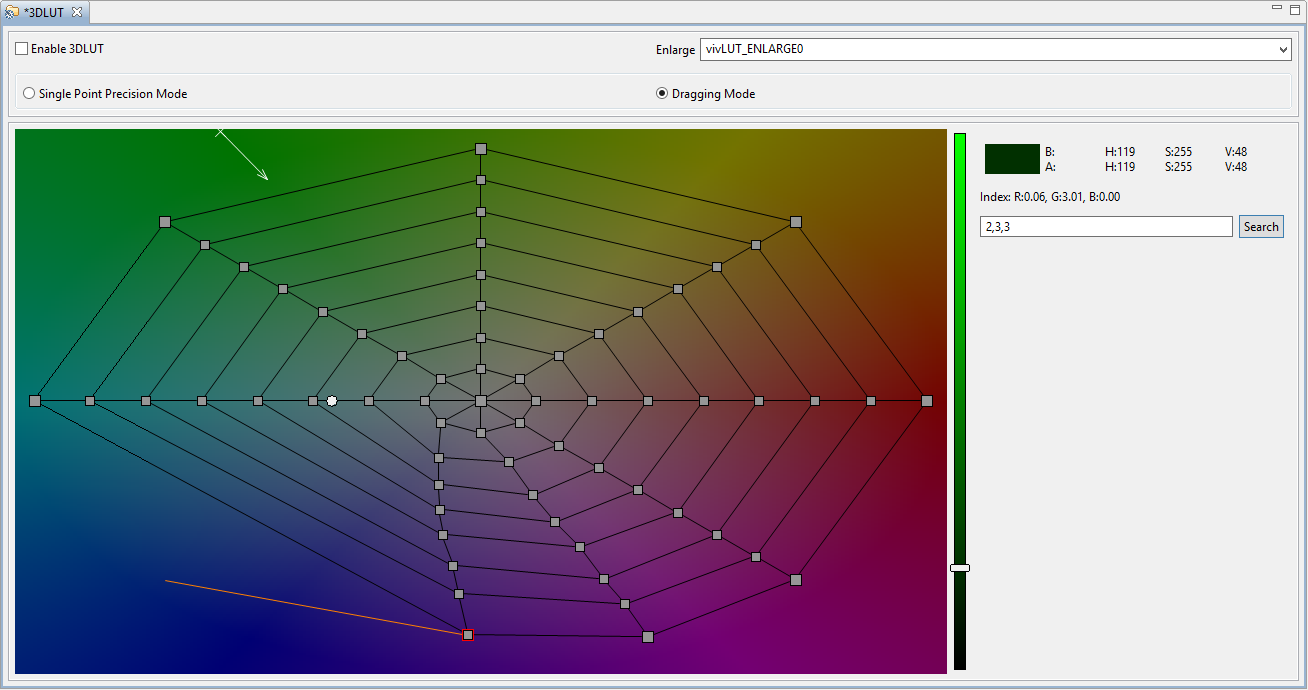
1. **Direct Mode**

Direct mode provides two direct ways to generate 3d look up table. It has two modes:Single Point Precision Mode and dragging mode.

**Dragging Mode:** Drag the mesh grid nodes to tune the color mapping.

Color searcher: enter the indexes of RGB in the search text box, the corresponding color will be marked in the right color zone.

Note: Value tuning of HSV Model is not supported, and intersection between grid nodes is not supported

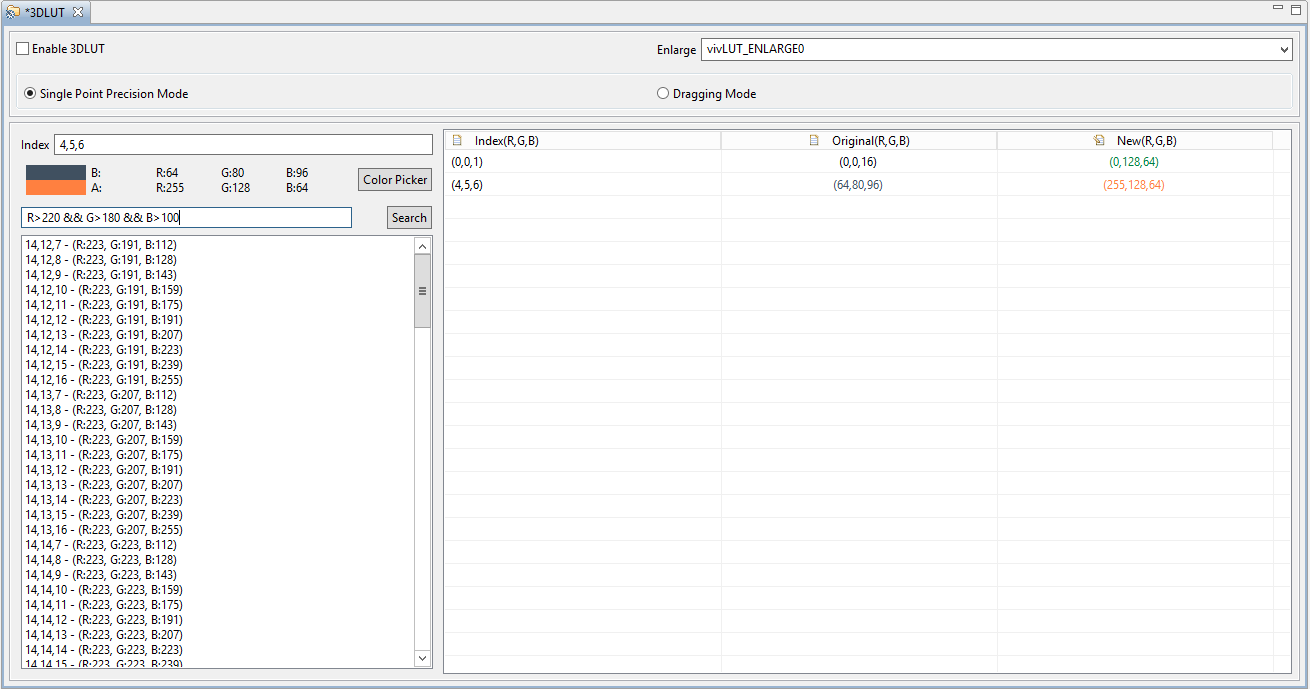


**Single Point Precision Mode:** Precisely adjust the mapping value for each point in the table.

**Index:** choose the point in the table through the combination of the index of R, G, B separated by commas, the range of each index is [0, 16).

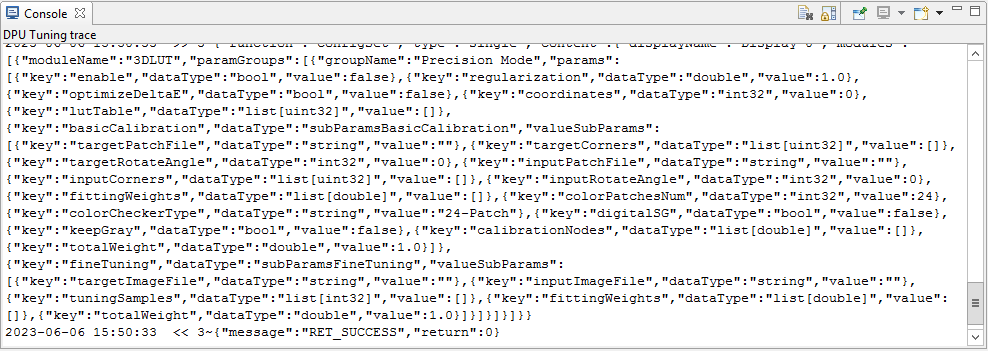
**Color Picker:** click to open the color dialog to choose a color that you want to mapping. The mapped colors will be displayed in the right table.

**Color Table:** in the right color table, right click to delete the selected items. Double click in a cell in “**New(R,G,B)**” column to edit the color value manually.

**Index Filter:** In the text box in the lower left corner, enter the conditions of RGB what you want to find, and the indexes that meet the conditions will be listed.

## View Log

Console view will show all interaction messages between Client and Server.



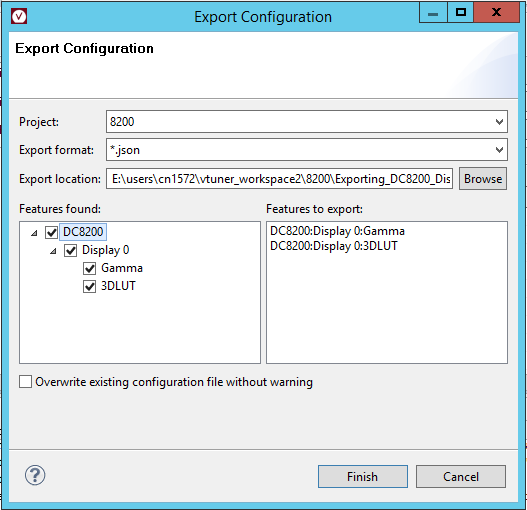
## Export and Import

Client support export and import the features configuration.

1. **Export**

This feature supports exporting the selected settings of the selected project into the specified file.

Open export dialog by click the icon on main toolbar. In the pop-up dialog, select the project which features need to be exported, the export type and export file location and need exported features. Click **Finish**, these settings of the selected features will be saved to the specified location.

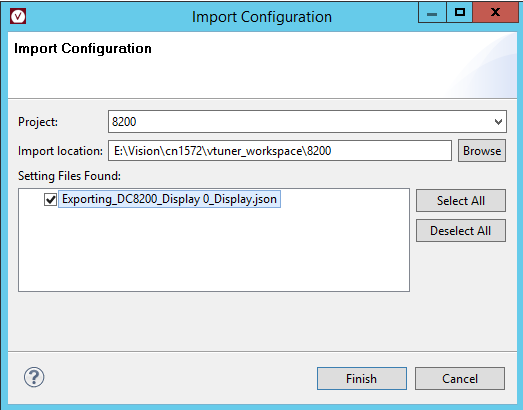


|  |  |
| --- | --- |
| **Element** | **Description** |
| Project | Specify the project which features need to be exported. |
| Export format | Specify the format of the exported settings, "**\*.json**" supports all settings of the modules, "**\*.bin**" only supports the data of Gamma and 3DLUT. |
| Export location | Specify the file location to save the exported settings. |
| Features Found | Show all the features of selected project and could select the features want to be exported. |
| Features to Export | Show all the selected features which will be exported. |

1. **Import**

This feature supports importing the settings from exported features file.

Open import dialog by click the icon on main toolbar. In the pop-up dialog, select the project which features settings need import, the import files directory and select the exported file which need import. Click **Finish**, these settings of the selected file will be import to the project configuration.



|  |  |
| --- | --- |
| **Element** | **Description** |
| Project | Specify the project which need import feature configuration. |
| Import location | Specify the directory where export files locate. |
| Setting Files Found | Show the file in import location. |

## Refresh

This function support synchronizing data from the Server to Client.