



Vivante VTuner

User Guide

for use with Vivante DC8200 Series IP

Document Revision 0.81

30 August 2023

Compatible with VTuner Beta Release Version 1.2.0

VERISILICON

LEVEL B: CONFIDENTIAL – DISTRIBUTION RESTRICTED

Legal Notices

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 herein may be directly re-distributed or may be adapted for re-distribution by SoC vendors who have licensed the related Vivante 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.

(VeriSilicon Distribution **LEVEL B: CONFIDENTIAL – DISTRIBUTION RESTRICTED**).

Upon request VeriSilicon provides this document in Word format for adaptive inclusion in documentation intended for the SoC vendor's 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® DPU 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 page located at <http://www.verisilicon.com>.

Vivante and VeriSilicon Proprietary. Copyright © 2023 by Vivante Corporation and VeriSilicon Holdings Co., Ltd. All rights reserved.

Table of Contents

| | |
|---|-----------|
| Legal Notices..... | 2 |
| Table of Contents..... | 3 |
| List of Figures..... | 4 |
| List of Tables..... | 5 |
| 1 Overview | 6 |
| 1.1 VTuner Introduction | 6 |
| 1.1.1 VTunerServer | 6 |
| 1.1.2 VTunerClient | 6 |
| 1.2 OS Compatibility | 7 |
| 1.3 Software Release Packages | 7 |
| 2 Setting Up VTunerServers..... | 8 |
| 2.1 Setting Up a VTunerServer for a DPU on an FPGA Board..... | 8 |
| 2.2 Setting Up a VTunerServer for a DPU on a CModel | 9 |
| 3 Using VTunerClients..... | 10 |
| 3.1 Installing the VTunerClient..... | 10 |
| 3.2 Checking the Environment..... | 14 |
| 3.3 Launching the VTunerClient Workbench | 15 |
| 3.3.1 DPU Tuner Perspective | 17 |
| 3.4 Creating a DPU Project..... | 17 |
| 3.5 Setting Up a Connection to a VTunerServer..... | 19 |
| 3.6 Exporting and Importing Feature Settings | 21 |
| 3.6.1 Exporting Feature Settings from a DPU Project | 21 |
| 3.6.2 Importing Feature Settings to a DPU Project | 22 |
| 3.7 Tuning Features of a DPU..... | 23 |
| 3.7.1 Tuning the Image Feature..... | 23 |
| 3.7.2 Tuning the Display Feature | 25 |
| 3.7.3 Tuning the Writeback Feature..... | 26 |
| 3.7.4 Tuning the Gamma Feature | 27 |
| 3.7.5 Tuning the Degamma Feature..... | 28 |
| 3.7.6 Tuning the 3DLUT Feature | 30 |
| 3.7.7 Tuning the Color Management Feature..... | 39 |
| 3.7.8 Tuning the Gamma 2 Feature..... | 40 |
| 3.8 Refreshing Feature Settings | 42 |
| Document Revision History..... | 43 |

List of Figures

| | |
|---|----|
| Figure 1. VTunerServer Setup Complete..... | 8 |
| Figure 2. Install VTunerClient Dialog | 11 |
| Figure 3. VTunerClient Setup Wizard..... | 11 |
| Figure 4. License Agreement Page of the VTunerClient Setup Wizard | 11 |
| Figure 5. Choose Destination Location Page of the VTunerClient Setup Wizard | 12 |
| Figure 6. Start Copying File Page of the VTunerClient Setup Wizard..... | 12 |
| Figure 7. Installing Page of the VTunerClient Setup Wizard..... | 13 |
| Figure 8. Installation Complete | 13 |
| Figure 9. Environment Check Result Example on Linux..... | 14 |
| Figure 10. Environment Check Result Example on Windows..... | 14 |
| Figure 11. Workspace Launcher | 15 |
| Figure 12. Configuration Dialog..... | 16 |
| Figure 13. DPU Tuner Perspective..... | 16 |
| Figure 14. File > New > DPU Project in Top Menu Bar..... | 18 |
| Figure 15. New DPU Project Window | 18 |
| Figure 16. Project Properties Window..... | 19 |
| Figure 17. Server Configuration Dialog..... | 19 |
| Figure 18. Status Bar..... | 20 |
| Figure 19. Export Configuration Window | 21 |
| Figure 20. Import Configuration Window..... | 23 |
| Figure 21. Details View When the Image Feature is Selected..... | 24 |
| Figure 22. Details View When the Display Feature is Selected | 25 |
| Figure 23. Details View When the Writeback Feature is Selected | 26 |
| Figure 24. Details View for PQ Curve When the Gamma Feature is Selected..... | 27 |
| Figure 25. Details View for User Curve When the Gamma Feature is Selected | 28 |
| Figure 26. Details View for PQ Curve When the Degamma Feature is Selected | 29 |
| Figure 27. Details View for User Curve When the Degamma Feature is Selected..... | 29 |

| | |
|---|----|
| Figure 28. Details View When Precision Mode under 3DLUT Is Selected | 31 |
| Figure 29. Pick Color Checker Window | 32 |
| Figure 30. Input Patch File Page for 24-Patch Color Checker | 33 |
| Figure 31. Target Patch File Page for 24-Patch Color Checker | 33 |
| Figure 32. Example CSV File | 34 |
| Figure 33. Input Patch File Page for Custom Color Checker | 34 |
| Figure 34. Target Patch File Page for Custom Color Checker | 35 |
| Figure 35. Advance Page of the Pick Color Checker Window | 35 |
| Figure 36. Details View with Basic Calibration Settings When Precision Mode under 3DLUT Is Selected | 36 |
| Figure 37. Details View with Fine Tuning Settings When Precision Mode under 3DLUT Is Selected | 37 |
| Figure 38. Details View in Dragging Mode When Direct Mode under 3DLUT Is Selected | 38 |
| Figure 39. Details View in Single Point Precision Mode When Direct Mode under 3DLUT Is Selected | 39 |
| Figure 40. Details View When a Range Pair Under Color Management is Selected | 40 |
| Figure 41. Details View When the Gamma 2 Feature Is Selected | 41 |

List of Tables

| | |
|---|----|
| Table 1. OS Compatibility | 7 |
| Table 2. Views in the DPU Tuner Perspective | 17 |
| Table 3. Toolbar in the DPU Tuner Perspective | 17 |
| Table 4. Properties of the Image Feature | 24 |
| Table 5. Properties of the Display Feature | 25 |
| Table 6. Properties of the Writeback Feature | 26 |

1 Overview

This chapter gives an overview of the Vivante VTuner.

1.1 VTuner Introduction

Vivante VTuner is a tool that adopts an HTTP-based server-client architecture for you to remotely tune DPUs running on FPGA boards and CModels. You can check the results either at the client in real time or by dumping the result images.

1.1.1 VTunerServer

VTunerServer is a VTuner server that works with a DPU running on an FPGA board or CModel.

A VTunerServer monitors requests from the client. Once receiving a request, it instructs the DPU to perform corresponding operations based on the configuration updates in the request.

1.1.2 VTunerClient

VTunerClient is a VTuner client that runs on your local personal computer (PC).

A VTunerClient provides a GUI for you to tune the features of DPU. When you update a feature configuration in the GUI, the VTunerClient sends a tuning request to the server via the HTTP protocol.

1.2 OS Compatibility

The following table lists the operating systems (OSes) recommended for VTunerServers and VTunerClients to run.

Table 1. OS Compatibility

| Component | Operating System |
|--------------|---|
| VTunerServer | Ubuntu 18 x64 and other Unix-like operating systems |
| VTunerClient | Windows 10 x64 and Ubuntu 18 x64 |

1.3 Software Release Packages

The VTunerServer release package is:

- VTunerServer-*<version>*-hardware-*<date>*.tgz for use with DPUs running on FPGA boards
- VTunerServer-*<version>*-cmodel-*<version>*-Linux-x86_64-*<date>*.tgz for use with DPUs running on CModels

The VTunerClient executable file is:

- VTunerClient-*<version>*-Win32-x86_64-*<date>*-Setup.exe for Windows 10 x64
- VTunerClient-*<version>*-Linux-x86_64-*<date>*-Install for Ubuntu 18 x64

2 Setting Up VTunerServers

This chapter provides a guide for you to set up VTunerServers for work with DPUs running on FPGA boards and CModels.

2.1 Setting Up a VTunerServer for a DPU on an FPGA Board

This section describes how to set up a VTunerServer for use with a DPU running on an FPGA board.

Prerequisites

- An FPGA board with DPU is available.
- The target operating system is Ubuntu 18 x64 or another Unix-like operating system.
- A VTunerServer release package VTunerServer-*<version>*-hardware-*<date>*.tgz is available.
- The third-party IP group G2 is enabled if you want to tune the **Color Management** feature.
- The third-party IP group G4 is enabled if you want to tune the **Gamma 2** feature.

Procedure

1. Start up the FPGA board.
2. Run your application to set up a pipeline.

An example command is provided as follows:

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

3. Copy the VTunerServer release package to the FPGA board.
4. Run the following commands to set up a VTunerServer:

```
tar -xvpvf VTunerServer-*.tgz
cd tuning-server
./run_server.sh -port 6060
```

The specified VTunerServer port is used to connect to VTunerClients. For more information, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Result

When the VTunerServer is set up, the following figure is displayed.



Figure 1. VTunerServer Setup Complete

2.2 Setting Up a VTunerServer for a DPU on a CModel

This section describes how to set up the VTunerServer for use with a DPU running on a CModel.

Prerequisites

- The target operating system is Ubuntu 18 x64 or another Unix-like operating system.
- A VTunerServer release package `VTunerServer-<version>-cmodel-<version>-Linux-x86_64-<date>.tgz` is available.

Procedure

1. Copy the VTunerServer release package to the operating system.
2. Run the following commands to set up a VTunerServer:

```
tar -xvpvf VTunerServer-*.tgz
./run_server.sh -port 6060
```

The specified VTunerServer port is used to connect to VTunerClients. For more information, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Result

When the VTunerServer is set up, [Figure 1](#) is displayed.

3 Using VTunerClients

This chapter guides you through the workflow of installing and using the VTunerClient for calibration and fine-tuning of DPUs running on FPGA boards and CModels.

In general, the workflow is as follows:

1. Install the VTunerClient.
For details, see [Section 3.1, Installing the VTunerClient](#).
2. Check whether all the libraries required by the VTunerClient are installed on your environment.
For details, see [Section 3.2, Checking the Environment](#).
3. Launch the VTunerClient workbench.
For details, see [Section 3.3, Launching the VTunerClient Workbench](#).
4. Create a project for the target DPU.
For details, see [Section 3.4, Creating a DPU Project](#).
5. Set up a connection to the VTunerServer.
For details, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).
6. (Optional) Import feature settings to the DPU project.
For details, see [Section 3.6.2, Importing Feature Settings to a DPU Project](#).
7. Tune the DPU by updating feature settings.
For details, see [Section 3.7, Tuning Features of a DPU](#).
8. Check the tuning results by viewing logs in real time or dumping the DPU result image:
For details, see [Section 3.3.1, DPU Tuner Perspective](#) and [Section 3.7.3, Tuning the Writeback Feature](#).

3.1 Installing the VTunerClient

This section describes how to install the VTunerClient.

Prerequisites

- The target operating system is either Linux or Windows.
Ubuntu 18.04 and Windows 10 are recommended.
- A VTunerClient executable file is available.
For the name of the VTunerClient executable file, see [Section 1.3, Software Release Packages](#).

Procedure

1. Run the VTunerClient executable file.
2. In the **Install VTunerClient** dialog, click **Yes**.

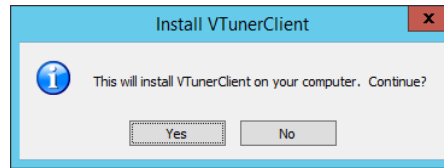


Figure 2. Install VTunerClient Dialog

3. In the **VTunerClient Setup** wizard, follow the instructions to complete the installation.
 - a. On the welcome page, click **Next**.

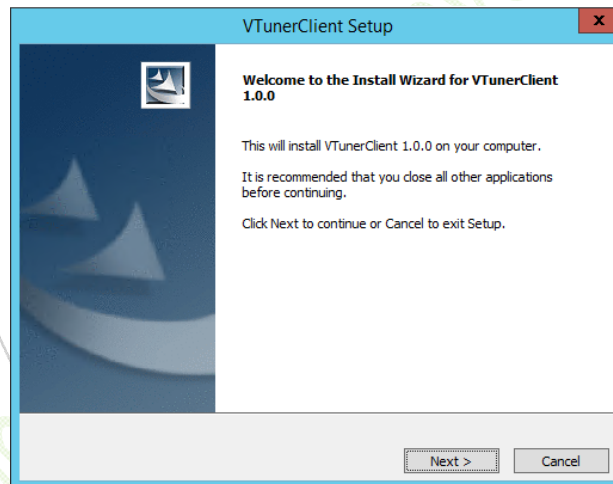


Figure 3. VTunerClient Setup Wizard

- b. On the **License Agreement** page, read the license agreement.
- c. To accept the agreement, select **I accept the terms of the license agreement** and click **Next**.

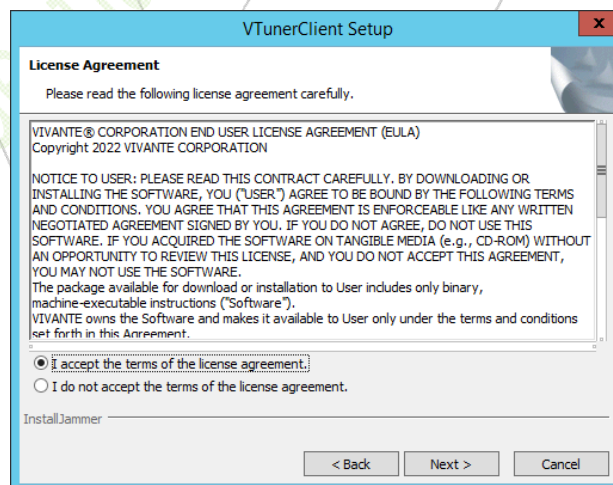


Figure 4. License Agreement Page of the VTunerClient Setup Wizard

- d. On the **Choose Destination Location** page, click **Browse** to select the destination directory where the VTunerClient is to be installed and click **Next**.

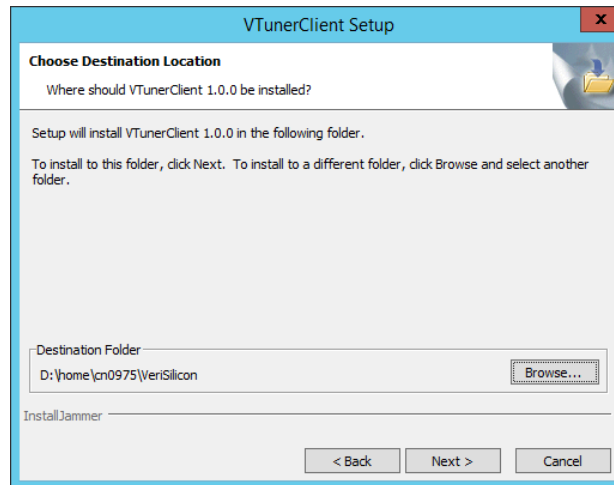


Figure 5. Choose Destination Location Page of the VTunerClient Setup Wizard

- e. On the **Start Copying Files** page, check the installation directory.
- If an unexpected directory is selected for installation, click **Back** and go to Step d.
 - If the installation directory is the same as expected, click **Next**.

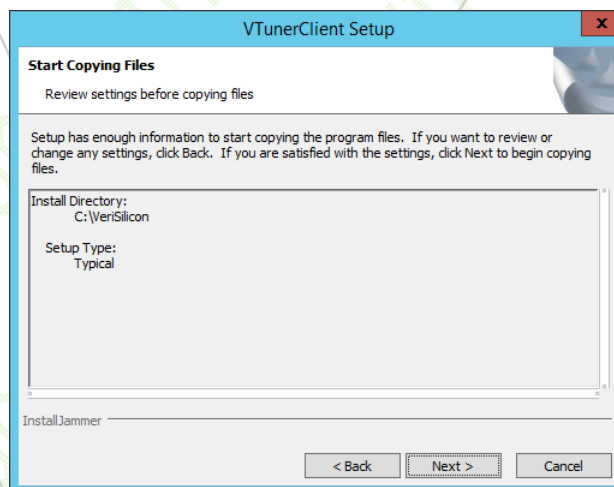


Figure 6. Start Copying File Page of the VTunerClient Setup Wizard

- f. Wait until the installation is completed.

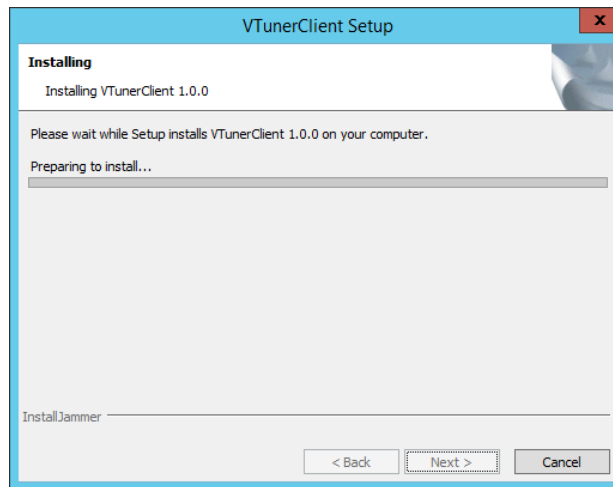


Figure 7. Installing Page of the VTunerClient Setup Wizard

- g. On the installation complete page, click **Finish**.

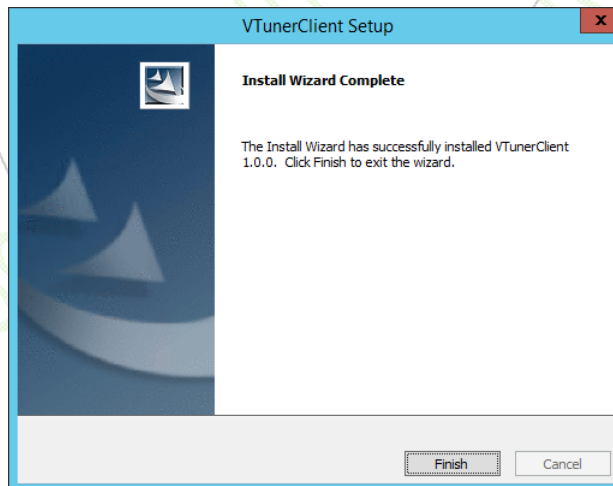


Figure 8. Installation Complete

Result

- The VTunerClient is installed.
- If you are using a Windows system, a **VTunerClient** shortcut is created on desktop.

Subsequent Steps

If you want to uninstall the VTunerClient, you can run `./uninstall` to launch the uninstallation wizard and follow the wizard to complete the uninstallation.

3.2 Checking the Environment

This section describes how to check whether all the libraries required by the VTunerClient are installed on your environment. The VTunerClient works better in an environment where all the required libraries are installed.

Prerequisites

The VTunerClient is installed, as instructed in [Section 3.1, Installing the VTunerClient](#).

Checking the Environment on Linux

If you are using the VTunerClient on Linux, perform the following steps to check the environment:

1. Open a terminal.
2. Go to the directory where the VTunerClient is installed.
3. Run `source ide/setenv-vtunerclient*` to trigger an environment check.
4. Go to the `./cmdtools/bin` directory.
5. Run `./vEcoChecker` to retrieve the environment check result.

```
cn1572@idcbj03:~/VeriSilicon/VTunerClient1.0.0/cmdtools/bin$ ./vEcoChecker
**start checking libraries.**
cannot find ">=>" in /home/cn1572/VeriSilicon/VTunerClient1.0.0/cmdtools/lib/libThreeDLUTExtJNI.so:
/lib/x86_64-linux-gnu/libm.so.6: version 'GLIBC 2.29' not found (required by /home/cn1572/VeriSil
icon/VTunerClient1.0.0/cmdtools/lib/libThreeDLUTExtJNI.so)
The following libraries are not found:
  libopencv_core.so.4.2 (required by libThreeDLUTExtJNI.so)
  libopencv_imgcodecs.so.4.2 (required by libThreeDLUTExtJNI.so)
  libopencv_imgproc.so.4.2 (required by libThreeDLUTExtJNI.so)
  libopencv_highgui.so.4.2 (required by libThreeDLUTExtJNI.so)
  libopencv_ml.so.4.2 (required by libThreeDLUTExtJNI.so)
  libopencv_calib3d.so.4.2 (required by libThreeDLUTExtJNI.so)

Please check the environment.
***end checking libraries.***
```

Figure 9. Environment Check Result Example on Linux

Checking the Environment on Windows

If you are using the VTunerClient on Windows, perform the following steps to check the environment:

1. Open the command prompt of Windows.
2. Run `ide/setenv-vtunerclientx.x.x.bat` to trigger an environment check.
3. Run `cmdtools/bin/vEcoChecker` to retrieve the environment check result.

```
E:\Users\cn1572>E:\Users\cn1572\installed_IDE\VTunerClient1.0.0\cmdtools\bin\vEc
oChecker.exe
**start checking libraries.**
All libraries are OK.
***end checking libraries.***
```

Figure 10. Environment Check Result Example on Windows

Result

- If the environment has all required libraries installed, the message "All libraries are OK" is returned.
- If some required libraries are not found in the environment, a list of the libraries that are not installed is returned.

3.3 Launching the VTunerClient Workbench

This section describes how to launch the VTunerClient workbench.

Prerequisites

The VTunerClient is installed, as instructed in [Section 3.1, Installing the VTunerClient](#).

Procedure

1. Open the VTunerClient.
 - If you are using a Linux system, run the `./ide/vtunerclient<version>` file in a terminal.
 - If you are using a Windows system, double-click the **VTunerClient** shortcut or run the `./ide/vtunerclient<version>` file in a console.
2. In the prompted **Workspace Launcher** dialog, select a workspace and click **OK**.
If the selected workspace does not exist, it will be automatically created.

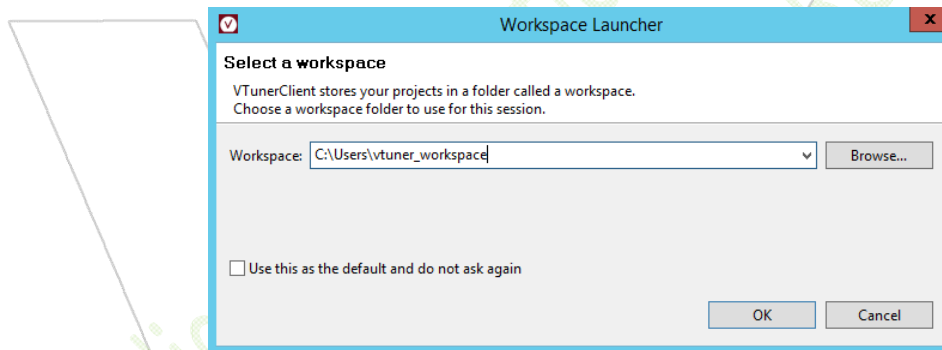


Figure 11. Workspace Launcher

3. In the **Configuration** window, select a project and a VTunerServer host, and click **OK**.
If the target project or VTunerServer host does not exist, you can also create the project or set up a new connection to the VTunerServer in the **Configuration** window.
Alternatively, click **Cancel** to skip this step. You can select a project and a VTunerServer host later.
For more information, see [Section 3.5, Setting Up a Connection to a VTunerServer](#) and [Section 3.4, Creating a DPU Project](#).

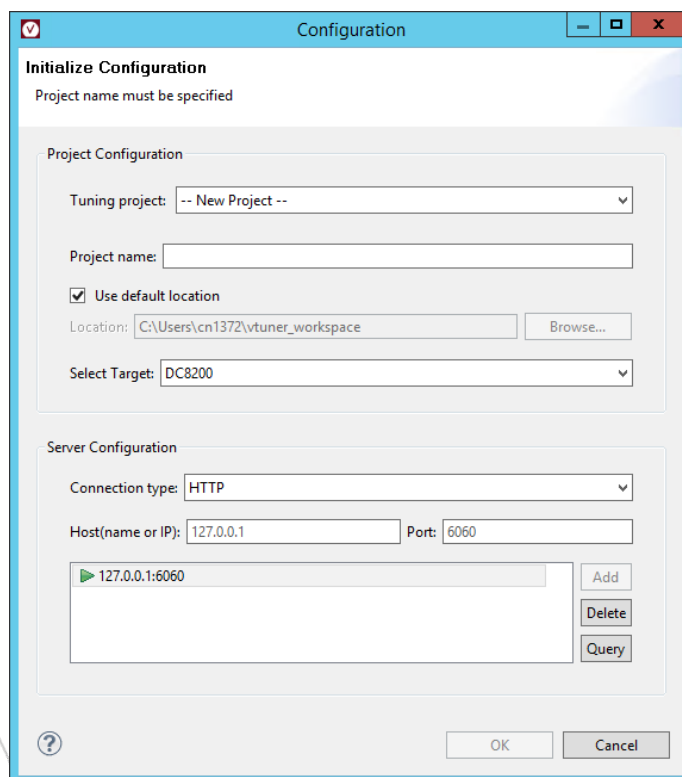


Figure 12. Configuration Dialog

Result

The VTunerClient workspace is created and the workbench is displayed in the **DPU Tuner** perspective.

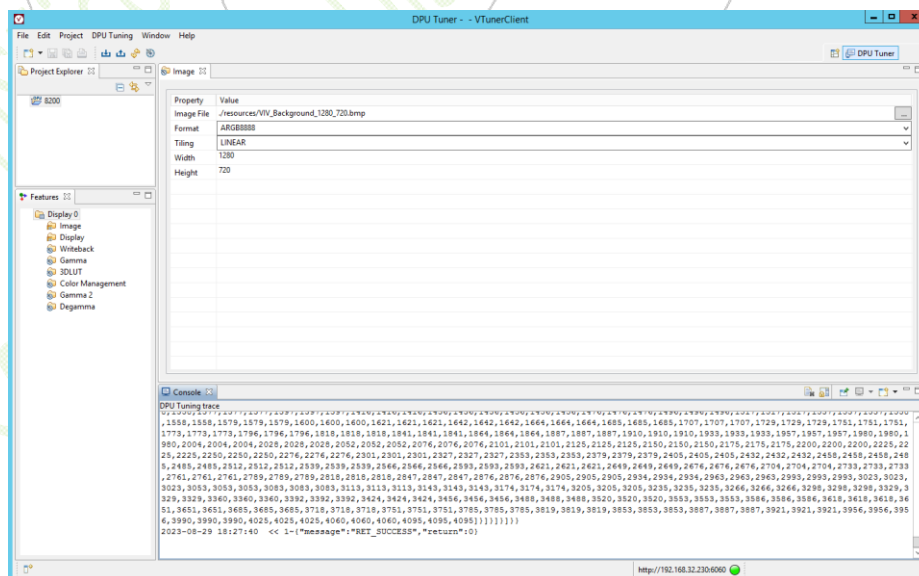


Figure 13. DPU Tuner Perspective

3.3.1 DPU Tuner Perspective





The **DPU Tuner** perspective contains the views and toolbar that you need for tuning DPU features, as shown in [Figure 13](#). The views and the buttons in the toolbar are detailed in [Table 2](#) and [Table 3](#) respectively.

Note: The **DPU Tuner** perspective can be customized. To customize the perspective, choose **Window > Customize Perspective** from the top menu bar.

Table 2. Views in the DPU Tuner Perspective

| View | Description |
|------------------|---|
| Project Explorer | Lists the projects in the current workspace. |
| Features | Lists the features that can be tuned for the project selected in the Project Explorer view. |
| Details | Shows the settings of the feature selected in the Features view. The label of this view varies with the selected feature. |
| Console | Prints logs in real time. |

Table 3. Toolbar in the DPU Tuner Perspective

| Key Button | Description |
|---|--|
|  | Import feature settings to a project. |
|  | Export feature settings from a project. |
|  | Refresh feature settings with data synchronized from a VTunerServer. |
|  | Modify connections to VTunerServers. |

3.4 Creating a DPU Project

This section describes how to create a DPU project. Each DPU project contains a group of feature settings for DPU tuning.

Prerequisites

The VTunerClient workbench is launched, as instructed in [Section 3.3, Launching the VTunerClient Workbench](#).

Procedure

1. In the top menu bar, choose **File > New > DPU Project**.

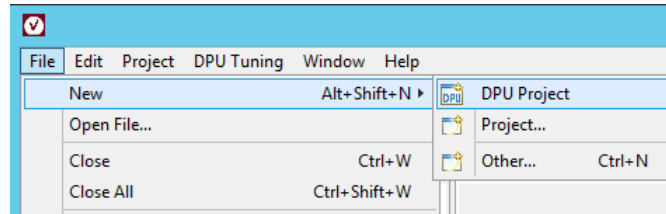


Figure 14. File > New > DPU Project in Top Menu Bar

2. In the prompted **New DPU Project** window, create a DPU project.

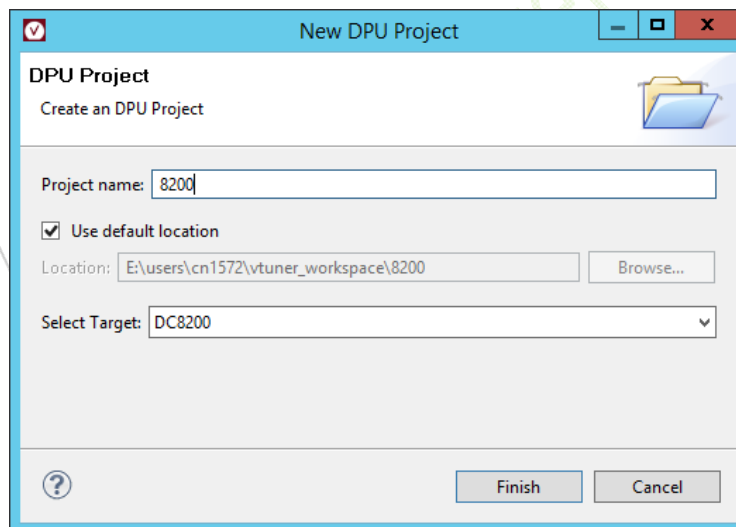


Figure 15. New DPU Project Window

- a. Enter a project name in the **Project name** text box.
The project name must contain only letters, digits, and underscores. Other special characters are not supported.
- b. Select the **Use default location** check box to create the directory of the new DPU project in the default path shown in the **Location** text box.
Alternatively, clear the **Use default location** check box and click **Browse** to select a folder where the new DPU project is to be created.
- c. From the **Select Target** drop-down list, select the DPU variant to be tuned.
- d. Click **Finish** to create the project.

Subsequent Steps

If you want to update the target DPU, you can right-click the DPU project, select **Properties**, and then update the DPU variant on the **General** tab of the properties window.

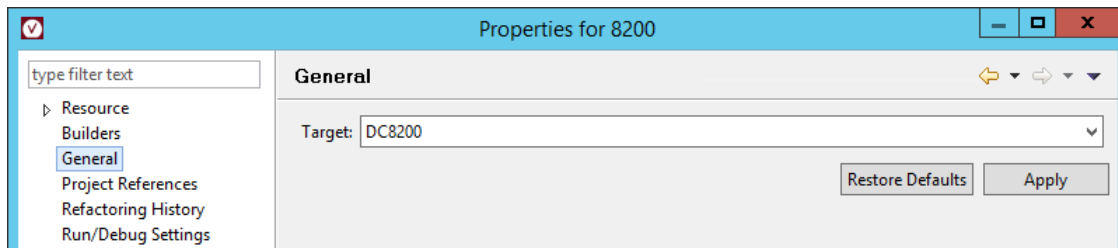


Figure 16. Project Properties Window

3.5 Setting Up a Connection to a VTunerServer

This section describes how to specify the IP address and port of a VTunerServer and accordingly set up the connection to the VTunerServer.

Prerequisites

- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).
- A VTunerServer is accessible.
For information about how to set up a connection to a VTunerServer, as instructed in [Chapter 2, Setting Up VTunerServers](#).

Procedure

- In the toolbar, click .
- In the prompted **Server Configuration** dialog, add the VTunerServer to the host list.

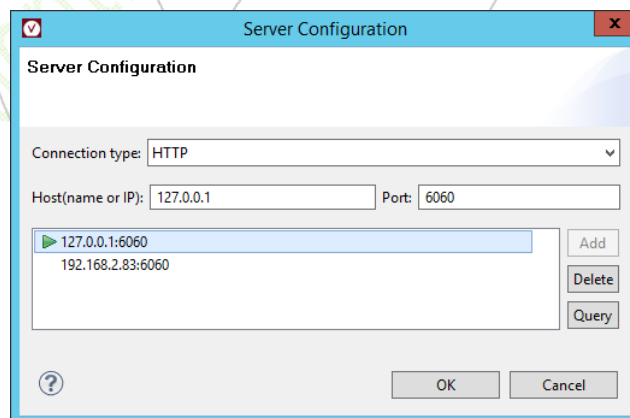


Figure 17. Server Configuration Dialog

- a. Set **Connection type** to the protocol to be used between the VTunerClient and the VTunerServer.
Currently, only HTTP is supported.
- b. Set **Host(name or IP)** to the IP address of the VTunerServer.
- c. Set **Port** to the port specified during VTunerServer setup.
For details, see [Chapter 2, Setting Up VTunerServers](#).
- d. Click **Add** to add the VTunerServer to the host list.
 - If an exclamation mark (!) appears to the right of the host information, accessing the VTunerServer fails. In this case, troubleshoot the connection error, and repeat Step d.
 - If a green tick mark (v) appears to the right of the host information, the VTunerServer is accessible.
- e. Click **OK** to complete the configuration.

Result


- The host address of the VTunerServer in format <IP address>:<port> is added to the host list of the **Server Configuration** dialog.
The  flag before the host address means that the VTunerServer is active. Requests will be sent to the active VTunerServer once feature settings are modified on the VTunerClient.
- The host address of the VTunerServer is displayed in the status bar of the VTunerClient workbench.
The indicator on the right of the host address shows whether the host is accessible.
 - A green indicator means the VTunerServer is accessible.
 - A red indicator means the VTunerServer is inaccessible.



Figure 18. Status Bar

Subsequent Steps

- To check the status of the connection to a VTunerServer, select the corresponding host address and click **Query** in the **Server Configuration** dialog.
- To delete the connection to a VTunerServer, select the corresponding host address and click **Delete** in the **Server Configuration** dialog.
- To switch the active VTunerServer to another, click the host address of the target VTunerServer and then click **OK** in the **Server Configuration** dialog.

3.6 Exporting and Importing Feature Settings

This section describes how to export and import feature settings of a DPU project.

3.6.1 Exporting Feature Settings from a DPU Project

This section describes how to export feature settings from a DPU project to a .json or .bin file.

Prerequisites

- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).
- A DPU project is available in the workspace.
For information about how to create a DPU project, see [Section 3.4, Creating a DPU Project](#).

Procedure

- In the tool bar, click .
- In the prompted **Export Configuration** window, specify the export configurations.

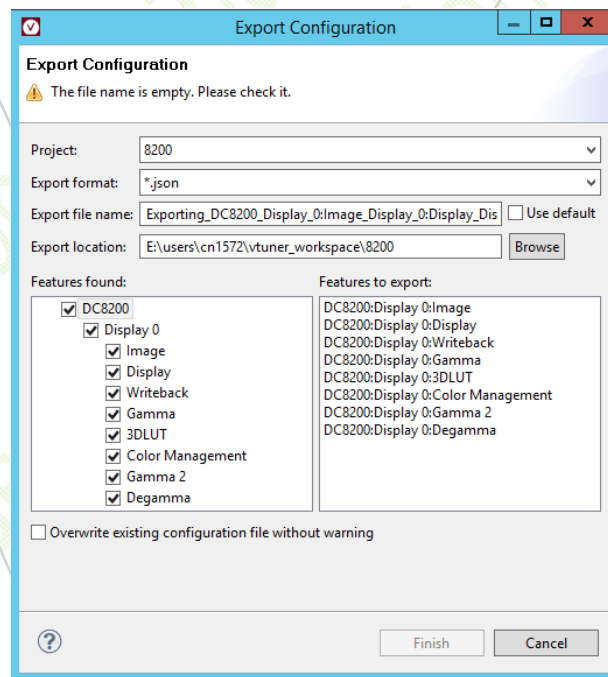


Figure 19. Export Configuration Window

- From the **Project** drop-down list, select the DPU project to be exported.
- From the **Export format** drop-down list, select the format of the exported file.
 - If ***.json** is selected, all feature settings can be exported.
 - If ***.bin** is selected, only settings of the **Gamma** and **3DLUT** features can be exported.

- c. To the right of **Export location**, click **Browse** and select the directory of the file to which you want to export feature settings.
If the specified file does not exist, it is automatically created during the export.
- d. In the **Features found** list, select the features whose settings you want to export.
The **Features to export** list is accordingly updated in real time.
All the settings of the selected features will be exported to the specified file.
- e. Select or clear the **Overwrite existing configuration file without warning** check box.
If the check box is selected and the specified file already exists, the original data in the file will be lost after the export is completed.
- f. Click **Finish** to export feature settings based on the configurations.

Subsequent Steps

You can import feature settings from the exported file to a DPU project by following the instructions in [Section 3.6.2, Importing Feature Settings to a DPU Project](#).


3.6.2 Importing Feature Settings to a DPU Project

This section describes how to import feature settings to a DPU project from a .json or .bin file.

Prerequisites

- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).
- A .json or .bin file exported from the VTunerClient is available.
For information about how to export a .json or .bin file, see [Section 3.6.1, Exporting Feature Settings from a DPU Project](#).

Procedure

1. In the tool bar, click .
2. In the prompted **Import Configuration** window, specify the import configurations.
 - a. From the **Project** drop-down list, select the DPU project to which you want to import feature settings.
 - b. To the right of **Import location**, click **Browse** and select the directory where the files to be imported are stored.
 - c. In the **Setting Files Found** list, select the files that you want to import.
 - d. Click **Finish** to import feature settings based on the configurations.

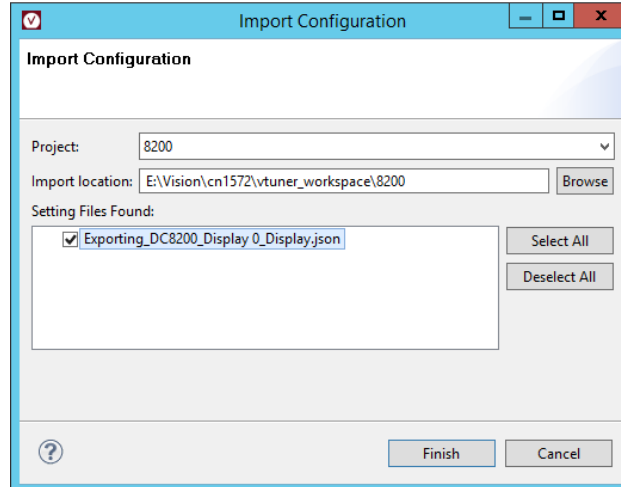


Figure 20. Import Configuration Window

3.7 Tuning Features of a DPU

This section describes how to tune each feature of a DPU by updating feature settings.

3.7.1 Tuning the Image Feature

This section describes how to tune the **Image** feature to control DPU input.

Prerequisites

- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).
- A DPU project is available in the workspace.
For information about how to create a DPU project, see [Section 3.4, Creating a DPU Project](#).
- The VTunerClient is connected to a VTunerServer that works with a DPU running on a CModel.
For information about how to set up a connection to a VTunerServer, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Procedure

1. In the **Project Explorer** view, select the DPU project.
Then, the features supported for tuning are listed in the **Features** view.
2. In the **Features** view, click **Image**.
Then, the settings of the **Image** feature are displayed in the details view.

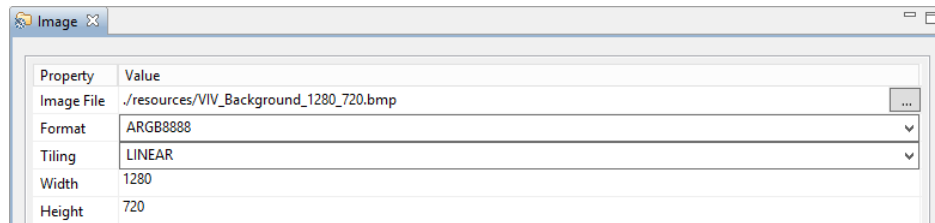


Figure 21. Details View When the Image Feature is Selected

3. In the details view of the **Image** feature, modify the settings of one or more properties.
Once a property is modified, the VTunerClient sends a request to the VTunerServer.

Table 4 describes the properties shown in the details view.

Table 4. Properties of the Image Feature

| Property | Description |
|------------|--------------------------------|
| Image File | The input image file. |
| Format | The input color format. |
| Tiling | The input tile mode. |
| Width | The width of the input image. |
| Height | The height of the input image. |

Result

The tuning results are printed in the **Console** view in real time.

3.7.2 Tuning the Display Feature

This section describes how to tune the **Display** feature to control DPU output.

Prerequisites

- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).
- A DPU project is available in the workspace.
For information about how to create a DPU project, see [Section 3.4, Creating a DPU Project](#).
- The VTunerClient is connected to a VTunerServer that works with a DPU running on a CModel.
For information about how to set up a connection to a VTunerServer, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Procedure

1. In the **Project Explorer** view, select the DPU project.
Then, the features supported for tuning are listed in the **Features** view.
2. In the **Features** view, click **Display**.
Then, the settings of the **Display** feature are displayed in the details view.

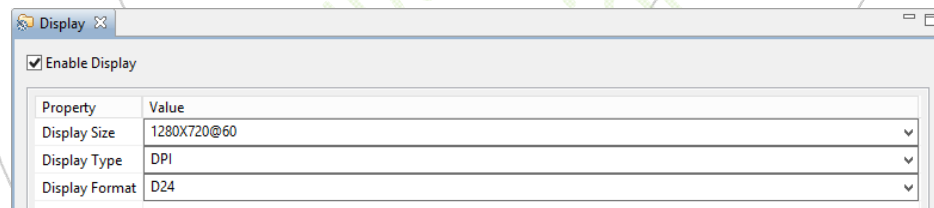


Figure 22. Details View When the Display Feature is Selected

3. In the details view of the **Display** feature, set the settings of one or more properties.
Once a property is modified, the VTunerClient sends a request to the VTunerServer.
[Table 5](#) describes the properties shown in the details view.

Table 5. Properties of the Display Feature

| Property | Description |
|----------------|----------------------------|
| Display Size | The output resolution. |
| Display Type | The output interface type. |
| Display Format | The output color format. |

Result

The tuning results are printed in the **Console** view in real time.

3.7.3 Tuning the Writeback Feature

This section describes how to tune the **Writeback** feature to dump a DPU result image.

Prerequisites

- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).
- A DPU project is available in the workspace.
For information about how to create a DPU project, see [Section 3.4, Creating a DPU Project](#).
- The VTunerClient is connected to a VTunerServer.
For information about how to set up a connection to a VTunerServer, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Procedure

1. In the **Project Explorer** view, select the DPU project.
Then, the features supported for tuning are listed in the **Features** view.
2. In the **Features** view, click **Writeback**.
Then, the settings of the **Writeback** feature are displayed in the details view.

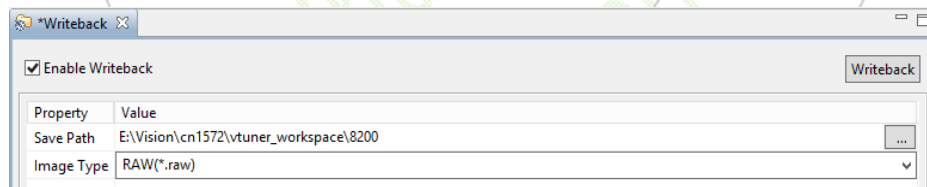


Figure 23. Details View When the Writeback Feature is Selected

3. In the details view of the **Writeback** feature, modify the settings.
Once a setting is modified, the VTunerClient sends a request to the VTunerServer.
 - a. Select the **Enable Writeback** check box to enable write-back at DPU output.
 - b. Set the properties to configure the dumped file of the result image.

[Table 6](#) describes the properties shown in the details view.

Table 6. Properties of the Writeback Feature

| Property | Description |
|------------|--|
| Save Path | The folder where you want to save the result image file. |
| Image Type | The file format of the result image. |

- c. Click the **Writeback** button to dump a result image based on the property settings.

Result

- The tuning results are printed in the **Console** view in real time.
- The DPU result image is dumped to a file in the folder specified by **Save Path**.

3.7.4 Tuning the Gamma Feature

This section describes how to tune the **Gamma** feature to control the 12-bit R, G, and B gamma lookup tables used by the gamma correction module after alpha blending.

Prerequisites

- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).
- A DPU project is available in the workspace.
For information about how to create a DPU project, see [Section 3.4, Creating a DPU Project](#).
- The VTunerClient is connected to a VTunerServer.
For information about how to set up a connection to a VTunerServer, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Procedure

- In the **Project Explorer** view, select the DPU project.
Then, the features supported for tuning are listed in the **Features** view.
- In the **Features** view, click **Gamma**.
Then, the settings of the **Gamma** feature are displayed in the details view.

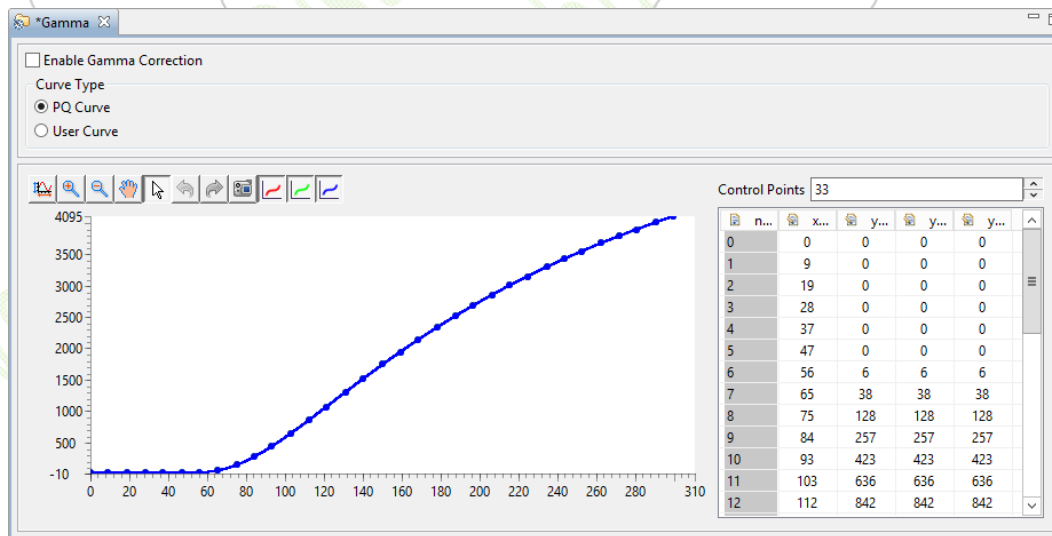


Figure 24. Details View for PQ Curve When the Gamma Feature is Selected

3. In the details view of the **Gamma** feature, modify the settings as follows:

Once a setting is modified, the VTunerClient sends a request to the VTunerServer.

- a. Select the **Enable Gamma Correction** check box to enable the gamma correction module.
- b. Under **Curve Type**, select **PQ Curve** or **User Curve** for gamma curve configuration.
- c. If **User Curve** under **Curve Type** is selected, specify the gamma value in the **Gamma Value** spin box.

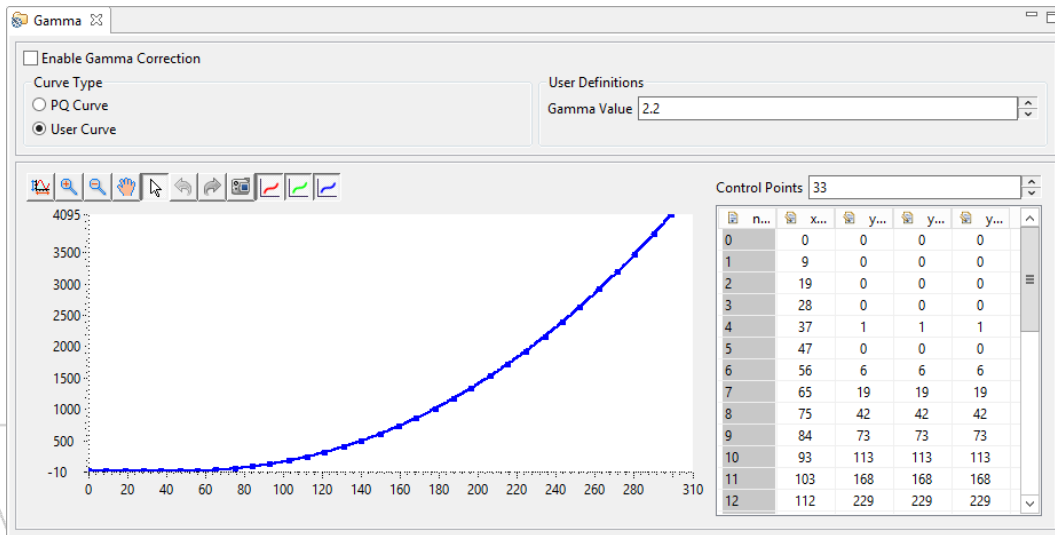


Figure 25. Details View for User Curve When the Gamma Feature is Selected

- d. In the **Control Points** text box, enter the number of control points in the gamma curve.
- e. To further adjust the gamma curve, drag each point in the plot to tune the gamma mapping value.
Alternatively, double-click each cell in the control point table and modify the value of the point.

Result

The tuning results are printed in the **Console** view in real time.

3.7.5 Tuning the Degamma Feature

This section describes how to tune the **Degamma** feature to control the 12-bit R, G, and B de-gamma lookup tables.

Prerequisites

- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).
- A DPU project is available in the workspace.
For information about how to create a DPU project, see [Section 3.4, Creating a DPU Project](#).
- The VTunerClient is connected to a VTunerServer.
For information about how to set up a connection to a VTunerServer, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Procedure

1. In the **Project Explorer** view, select the DPU project.
Then, the features supported for tuning are listed in the **Features** view.
2. In the **Features** view, click **Degamma**.
Then, the settings of the **Degamma** feature are displayed in the details view.

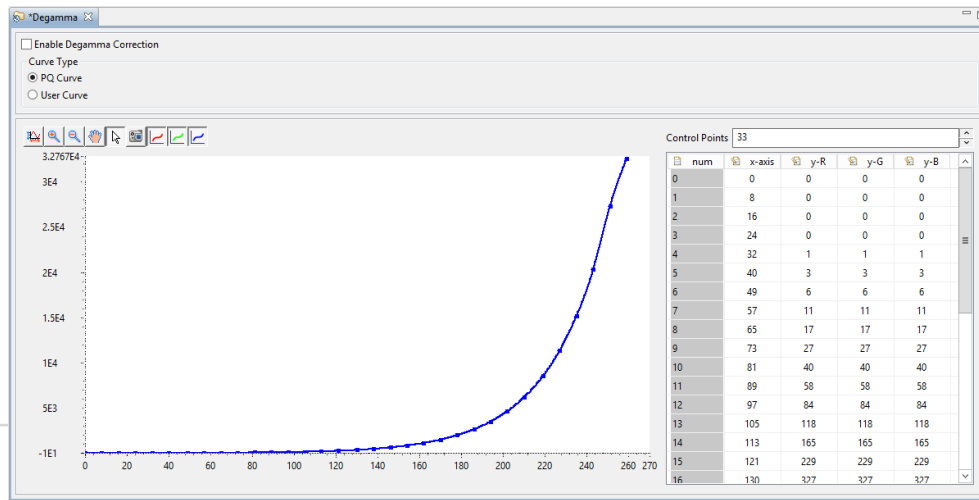


Figure 26. Details View for PQ Curve When the Degamma Feature is Selected

3. In the details view of the **Degamma** feature, modify the settings as follows:
Once a setting is modified, the VTunerClient sends a request to the VTunerServer.
 - a. Select the **Enable Degamma Correction** check box to enable the de-gamma correction module.
 - b. Under **Curve Type**, select **PQ Curve** or **User Curve** for de-gamma curve configuration.
 - c. If **User Curve** under **Curve Type** is selected, specify the de-gamma value in the **Degamma Value** spin box.

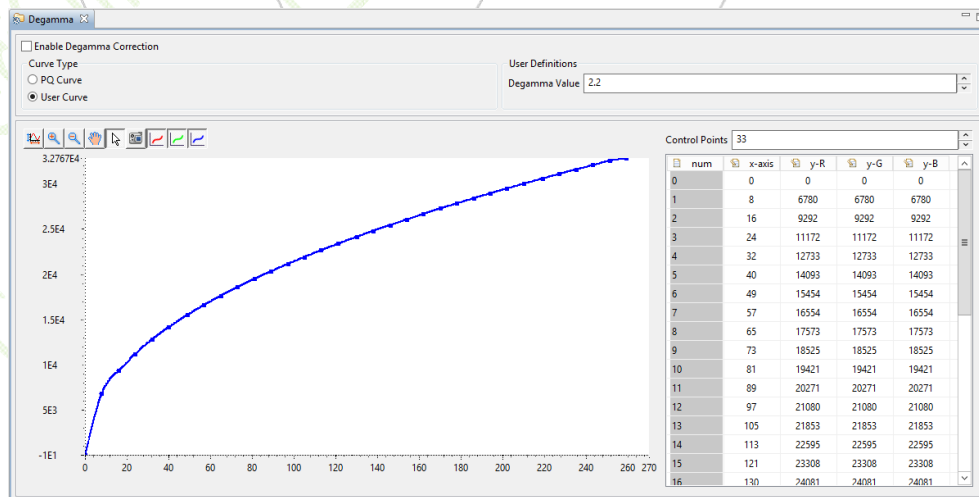


Figure 27. Details View for User Curve When the Degamma Feature is Selected

- d. In the **Control Points** text box, enter the number of control points in the de-gamma curve.
- e. To further adjust the de-gamma curve, drag each point in the plot to tune the de-gamma mapping value. Alternatively, double-click each cell in the control point table and modify the value of the point.

Result

The tuning results are printed in the **Console** view in real time.

3.7.6 Tuning the 3DLUT Feature

This section describes how to tune the **3DLUT** feature to control the three-dimensional lookup table (3D LUT). You can tune the **3DLUT** feature in either precision or direct mode.

Prerequisites

- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).
- A DPU project is available in the workspace.
For information about how to create a DPU project, see [Section 3.4, Creating a DPU Project](#).
- The VTunerClient is connected to a VTunerServer.
For information about how to set up a connection to a VTunerServer, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Procedure

1. In the **Project Explorer** view, select the DPU project.
Then, the features supported for tuning are listed in the **Features** view.
2. In the **Features** view, click **Precision Mode** or **Direct Mode** under **3DLUT**.
Then, the corresponding settings of the **3DLUT** feature are displayed in the details view.
3. In the details view, modify the settings as instructed in [Section 3.7.6.1, Tuning the 3DLUT Feature in Precision Mode](#) and [Section 3.7.6.2, Tuning the 3DLUT Feature in Direct Mode](#).
Once a setting is modified, the VTunerClient sends a request to the VTunerServer.

Result

The tuning results are printed in the **Console** view in real time.

This section describes how to tune the **3DLUT** feature in precision mode.

- The details view shows the settings of the **3DLUT** feature in precision mode.
For information about how to show the settings in the details view, see [Section 3.7.6, Tuning the 3DLUT Feature](#).
- A VTunerServer is accessible.
For information about how to set up a connection to a VTunerServer, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

In the details view of the **3DLUT** feature in precision mode, modify the settings as follows.



1. Select the **Enable** check box to enable the 3D LUT module.
2. In the **Coordinates** text box, enter the chromaticity coordinates of the current calibration node being loaded.
3. In the **Enlarge** text box, enter one of the following values to specify the enlarge mode for color channel values:
 - **vivLUT_ENLARGE0**: does not enlarge color channel values.
 - **vivLUT_ENLATGE9**: enlarges each color channel value by 2^n , where:
 $n = 9$ — Bit width of the greatest value among the red, green, and blue channels
 - **vivLUT_ENLATGE10**: enlarges each color channel value by 2^n , where:
 $n = 10$ — Bit width of the greatest value among the red, green, and blue channels
4. Select the **Basic Calibration** or **Fine Tuning** tab.
 - If the **Basic Calibration** tab is selected, the 3D LUT is generated using standard checkers.
 - If **Fine Tuning** is selected, the 3D LUT is generated by mapping specified input colors to target colors.

5. If the **Basic Calibration** tab is selected, perform the following steps:
 - a. To the right of **Target Patch File**, click the **Open** button.
 - b. In the prompted **Pick Color Checker** window, select **24-patch**, **140-patch**, or **User Defined** from the **Color Checker Type** drop-down list.

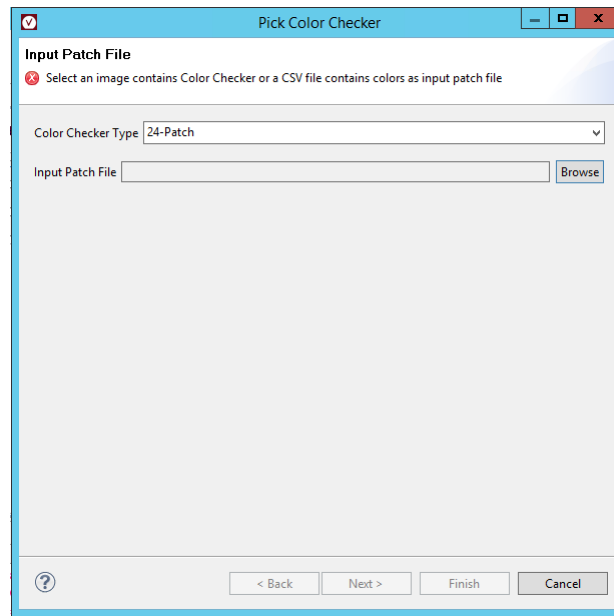


Figure 29. Pick Color Checker Window

- c. If **24-patch** or **140-patch** is selected for **Color Checker Type**, perform the following steps:
 - i. On the **Input Patch File** page, click the **Browse** button and select the input color checker image.
 - ii. In the displayed image, click the center of the color patch at each corner of the color checker to explicitly specify the color checker.
The average color of each patch in the color checker is calculated and used as the patch color.
 - iii. (Optional) To change a color checker corner, click the corresponding mark to remove it and then click the center of a color patch to mark it as the new color checker corner.
 - iv. Drag the slider under **Rotate** to rotate the image.
 - v. Click **Next**.

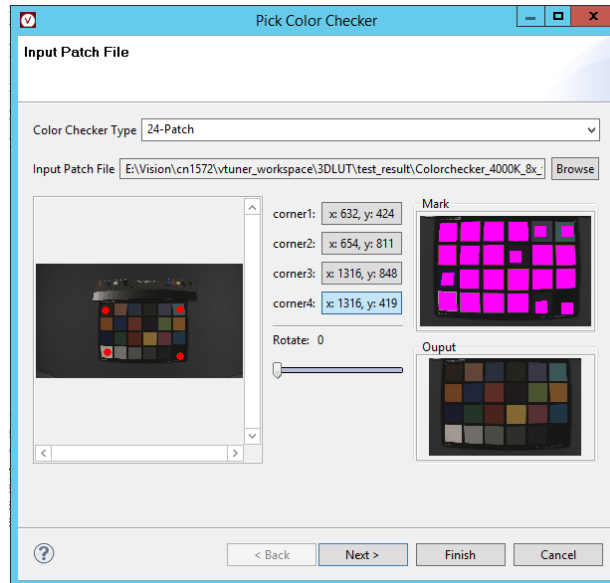


Figure 30. Input Patch File Page for 24-Patch Color Checker

- vi. On the **Target Patch File** page, click the **Browse** button and select the target color checker image.
- vii. In the displayed image, click the center of the color patch at each corner of the color checker by referring to Step ii and Step iii.
- viii. Drag the slider under **Rotate** to rotate the image.
- ix. Click **Next**.

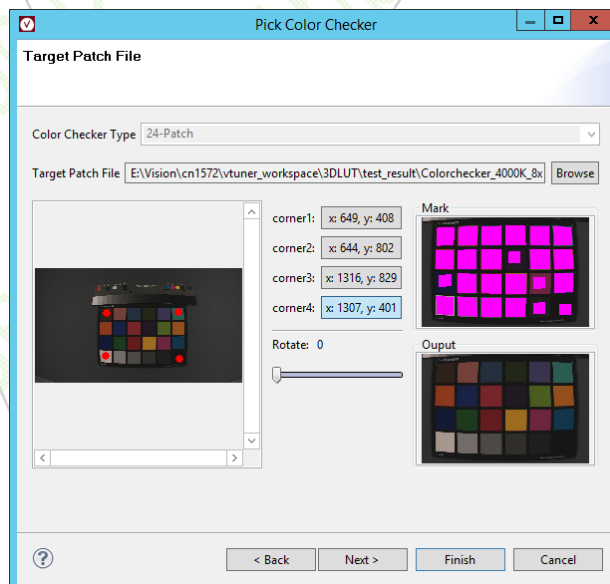


Figure 31. Target Patch File Page for 24-Patch Color Checker

- d. If **User Defined** is selected for **Color Checker Type**, perform the following steps:
 - i. On the **Input Patch File** page, click the **Browse** button and select the .csv file that describes the R, G, and B values of each patch in the input color checker.
 - ii. Click **Next**.

| | <i>R</i> | <i>G</i> | <i>B</i> |
|---|----------|----------|----------|
| 1 | 123 | 200 | 120 |
| 2 | 133 | 210 | 100 |
| 3 | 103 | 20 | 80 |
| 4 | 90 | 100 | 100 |
| 5 | 143 | 140 | 170 |
| 6 | 113 | 220 | 40 |
| 7 | 23 | 140 | 60 |
| 8 | 103 | 90 | 70 |
| 9 | 43 | 120 | 180 |

Figure 32. Example CSV File

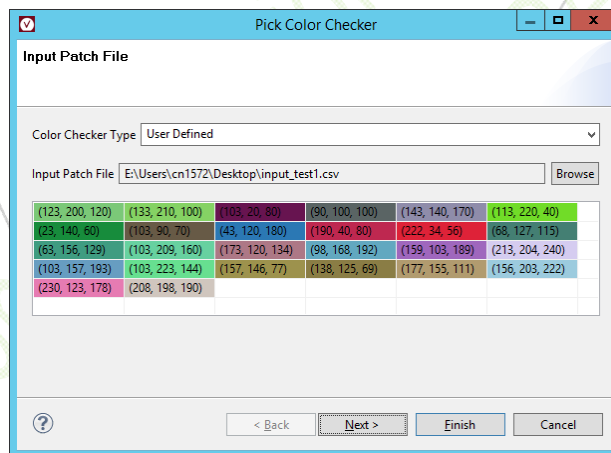


Figure 33. Input Patch File Page for Custom Color Checker

- iii. On the **Target Patch File** page, click the **Browse** button and select the .csv file that describes the R, G, and B values of each patch in the target color checker.
- iv. Click **Next**.

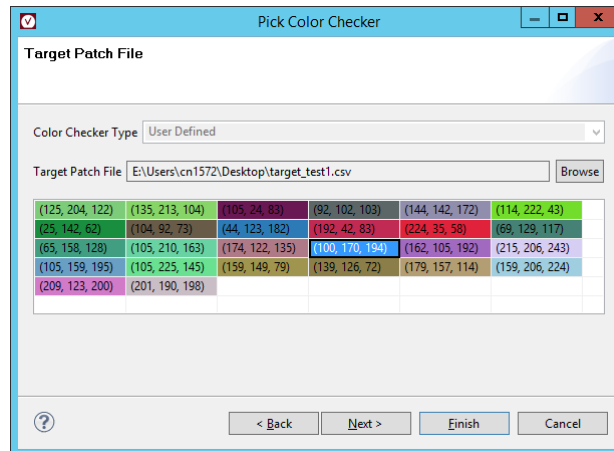


Figure 34. Target Patch File Page for Custom Color Checker

- v. On the **Advance** page, modify the weight of each color for generating the 3D LUT.
- vi. Click **Finish**.

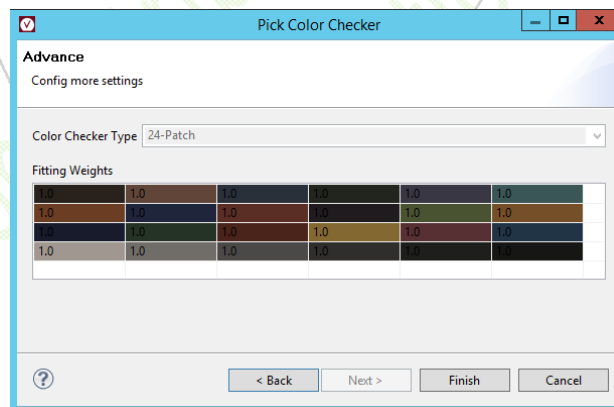


Figure 35. Advance Page of the Pick Color Checker Window

Then, on the **Basic Calibration** tab of the details view:

- The **Summary** table shows the global maximum and average Delta C and Delta E.
- The **Attributes** table shows the following attributes of each color patch: index, weight, input, target, and optimized colors, and Delta C and Delta E before and after optimization.

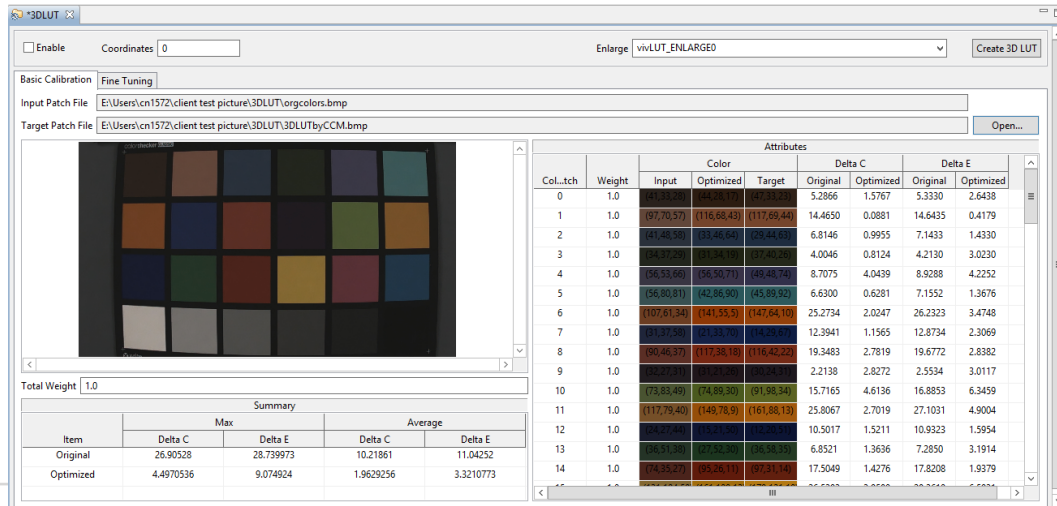


Figure 36. Details View with Basic Calibration Settings When Precision Mode under 3DLUT Is Selected

6. If the **Fine Tuning** tab is selected, perform the following steps:

- To the right of **Input Image File**, click the **Open** button and select the input image.
- To the right of **Target Image File**, click the **Open** button and select the target image.
- Add each sample patch as follows:
 - In the **Input Sample Size** spin box of the **Sample Setting** section, specify the sample patch size of the input image.
 - Select the sample input color by double-clicking a proper position in the input image.
Then, the R, G, and B values of the sample input color is automatically filled in the **Input** section under **Sample Detail**.
 - In the **Target Sample Size** spin box of the **Sample Setting** section, specify the sample patch size of the target image.
 - Select the sample target color by double-clicking a proper position in the target image.
Then, the R, G, and B values of the sample target color is automatically filled in the **Target** section under **Sample Detail**.
 - In the **Weight** text box of the **Sample Detail** section, enter the weight of the sample color.
 - Click **Add Sample**.

Then, the attributes of the sample patch are added to the **Attributes** table. The attributes include index, weight, input color, target color, optimized color, and Delta C and Delta E before and after optimization.

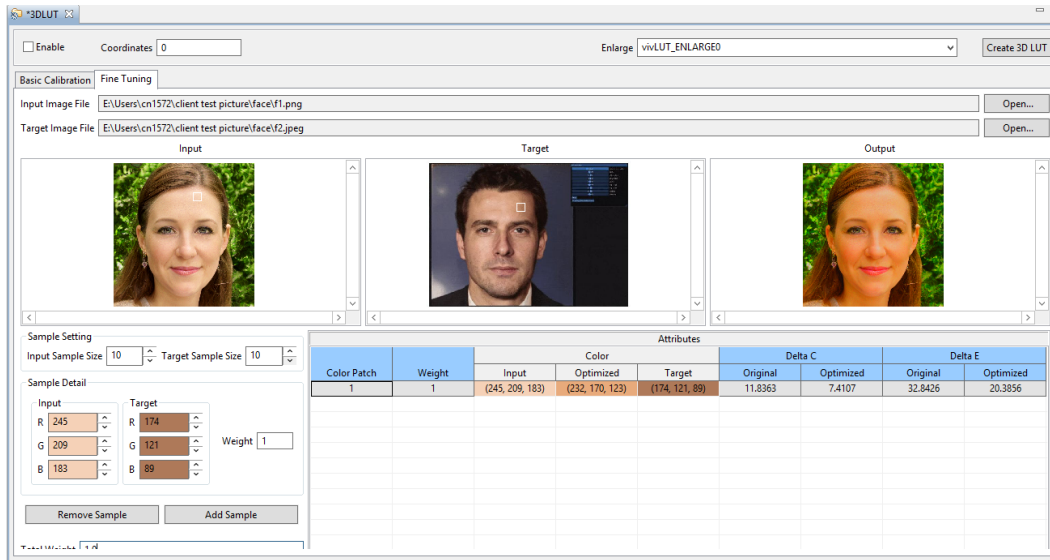


Figure 37. Details View with Fine Tuning Settings When Precision Mode under 3DLUT Is Selected

- d. (Optional) To update the weights of all sample patches to the same value, enter the value in the **Total Weight** text box.
7. In the upper-right corner, click the **Create 3D LUT** button to re-generate a 3D LUT with the modified color weights.

Result

The tuning results are printed in the **Console** view in real time.

3.7.6.2 Tuning the 3DLUT Feature in Direct Mode

This section describes how to tune the **3DLUT** feature in precision mode.

Prerequisites

- The details view shows the settings of the **3DLUT** feature in direct mode.
For information about how to show the settings in the details view, see [Section 3.7.6, Tuning the 3DLUT Feature](#).
- A VTunerServer is accessible.
For information about how to set up a connection to a VTunerServer, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Procedure

In the details view of the **3DLUT** feature in direct mode, modify the settings as follows:

- Select the **Enable** check box to enable the 3D LUT module.

2. In the **Enlarge** text box, enter one of the following values to specify the enlarge mode for color channel values:
 - **vivLUT_ENLARGE0**: does not enlarge color channel values.
 - **vivLUT_ENLARGE9**: enlarges each color channel value by 2^n , where:

$$n = 9 - \text{Bit width of the greatest value among the red, green, and blue channels}$$
 - **vivLUT_ENLARGE10**: enlarges each color channel value by 2^n , where:

$$n = 10 - \text{Bit width of the greatest value among the red, green, and blue channels}$$
3. Select **Single Point Precision Mode** or **Dragging Mode**.
4. If **Dragging Mode** is selected, drag mesh grid nodes to tune color mappings.
 You can use the color searcher on the right to easily locate a color zone. To use the color searcher, enter a set of RGB indexes in the text box and click **Search**. Each index must be in the range from 0 to 16, inclusive, and separated with a comma (,).
Note: HSV-based tuning is not supported. Intersection between grid nodes is not supported.

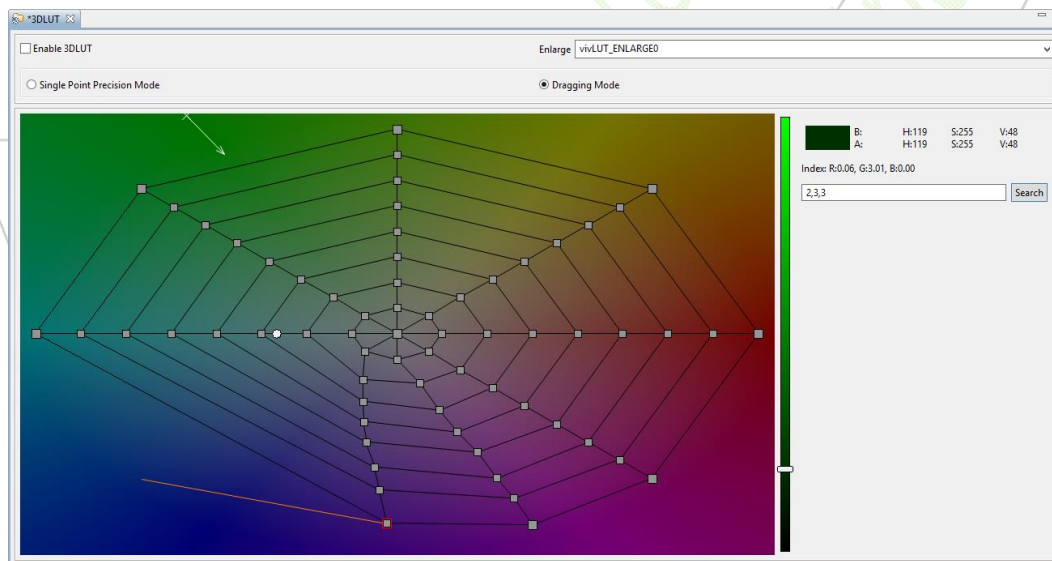


Figure 38. Details View in Dragging Mode When Direct Mode under 3DLUT Is Selected

5. If **Single Point Precision Mode** is selected, perform the following steps:
 - a. In the **Index** text box, enter a set of RGB indexes to choose a point from the table.
 Each index must be in the range from 0 to 16, inclusive, and separated with a comma (,).
 You can use the index filter in the bottom-left corner to query the RGB indexes of a color. To use the index filter, enter a RGB color based condition and click **Search** to list the indexes of all RGB colors that meet the condition.
 - b. Click the **Color Picker** button and select a color as the mapped color.
 Then, the color mapping is added to the color table on the right.
 - c. Repeat Step a and Step b to add all color mappings.
 - d. (Optional) To delete a color mapping from the color table, right-click the corresponding row and choose **Delete** from the shortcut menu.

- e. (Optional) To modify a mapped color, double-click the cell in the **New(R,G,B)** column and enter the new RGB color.

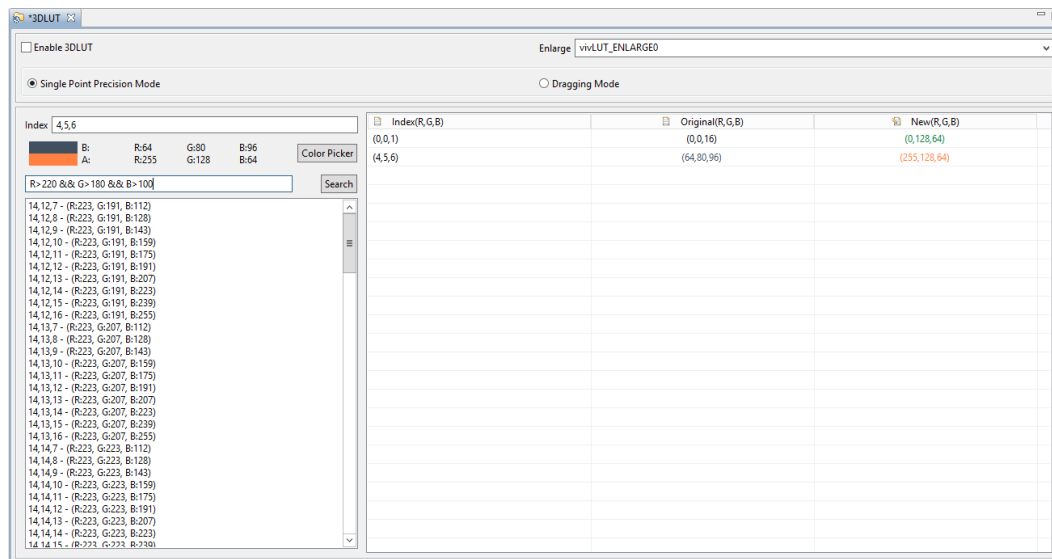


Figure 39. Details View in Single Point Precision Mode When Direct Mode under 3DLUT Is Selected

Result

The tuning results are printed in the **Console** view in real time.

3.7.7 Tuning the Color Management Feature

This section describes how to tune the **Color Management** feature to control the color management module in the third party IP group G2. This feature supports tuning six range pairs and two ranges in each pair.

Prerequisites

- The third party IP group G2 is supported and enabled.
- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).
- A DPU project is available in the workspace.
For information about how to create a DPU project, see [Section 3.4, Creating a DPU Project](#).
- The VTunerClient is connected to a VTunerServer that works with a DPU running on an FPGA board.
For information about how to set up a connection to a VTunerServer, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Procedure

1. In the **Project Explorer** view, select the DPU project.
Then, the features supported for tuning are listed in the **Features** view.
2. In the **Features** view, click a color range under **Color Management**.

Then, the settings of the color range are displayed in the details view.

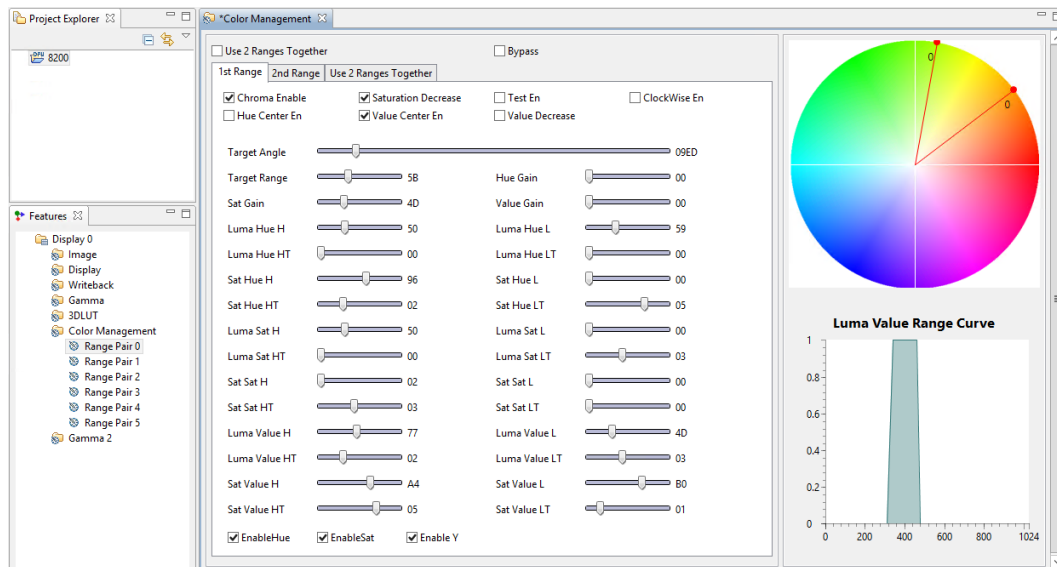


Figure 40. Details View When a Range Pair Under Color Management is Selected

3. In the details view of the **Color Management** feature, modify the settings.
Once a setting is modified, the VTunerClient sends a request to the VTunerServer.
 - a. Clear the **Bypass** check box to enable the color management module.
 - b. Select or clear the **Use 2 Range Together** check box.
 - If the check box is selected, specify color configurations on the **Use 2 Ranges Together** tab.
 - If the check box is cleared, specify color configurations on the **1st Range** and **2nd Range** tabs.

Result

The tuning results are printed in the **Console** view in real time.

3.7.8 Tuning the Gamma 2 Feature

This section describes how to tune the **Gamma 2** feature to control the 10-bit R, G, and B gamma lookup table used by the gamma module in the third party IP group G4.

Prerequisites

- The third party IP group G4 is supported and enabled.
- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).

- A DPU project is available in the workspace.
For information about how to create a DPU project, see [Section 3.4, Creating a DPU Project](#).
- The VTunerClient is connected to a VTunerServer that works with a DPU running on an FPGA board.
For information about how to set up a connection to a VTunerServer, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Procedure

1. In the **Project Explorer** view, select the DPU project.
Then, the features supported for tuning are listed in the **Features** view.
2. In the **Features** view, click **Gamma 2**.
Then, the settings of the **Gamma 2** feature are displayed in the details view.

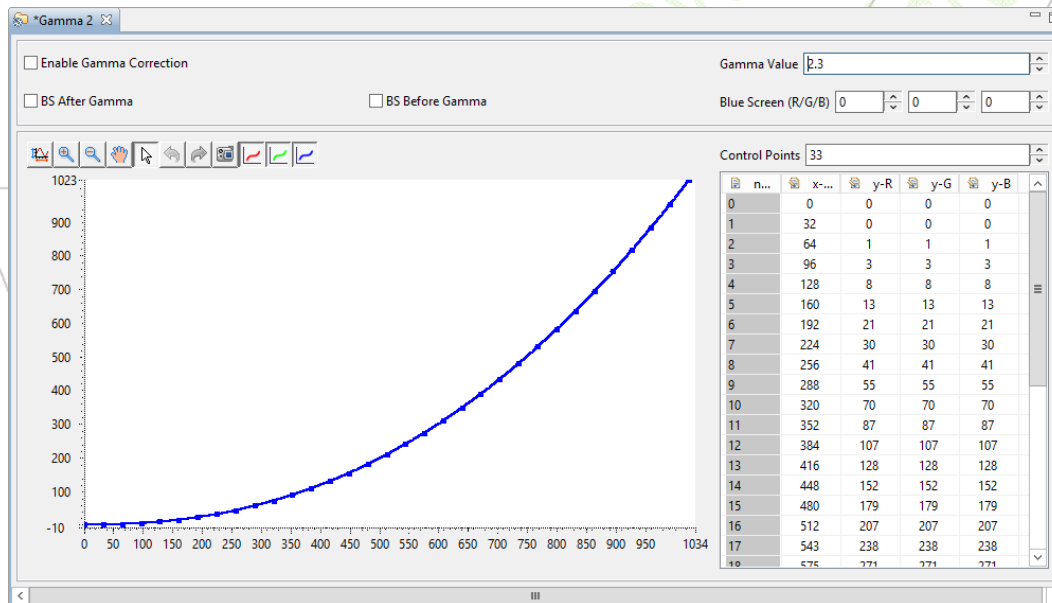


Figure 41. Details View When the Gamma 2 Feature Is Selected

3. In the details view for the **Gamma 2** feature, modify the settings.
Once a setting is modified, the VTunerClient sends a request to the VTunerServer.
 - a. Select the **Enable Gamma Correction** check box to enable the gamma module.
 - b. Select or clear the **BS After Gamma** and **BS Before Gamma** to control whether to enable BS before and after the gamma module in G4.
 - c. In the **Gamma Value** spin box, specify the gamma value.
 - d. In the **Blue Screen (R/G/B)** spin boxes, specify the R, G, and B values of the blue screen color.
 - e. In the **Control Points** text box, specify the number of control points in the curve.
 - f. To further adjust the gamma curve, drag each point in the plot to tune the gamma mapping value.
Alternatively, double-click each cell in the control point table and modify the value of the point.

Result

The tuning results are printed in the **Console** view in real time.


3.8 Refreshing Feature Settings

This section describes how to refresh the feature settings of a DPU project with data synchronized from a VTunerServer.

Prerequisites

- The VTunerClient workbench is launched and displayed in the **DPU Tuner** perspective.
For information about how to launch the workbench, see [Section 3.3, Launching the VTunerClient Workbench](#).
- A DPU project is available in the workspace.
For information about how to create a DPU project, see [Section 3.4, Creating a DPU Project](#).
- A VTunerServer is accessible.
For information about how to set up a connection to a VTunerServer, see [Section 3.5, Setting Up a Connection to a VTunerServer](#).

Procedure

To refresh the feature settings of a DPU project with data synchronized from a VTunerServer, click  in the tool bar.

Document Revision History

This chapter describes the differences between different revisions of this document.

Note: This document is not necessarily updated for each patch or minor revision.

| Document Revision | Date | Compatibility | Change History |
|-------------------|------------|--|--|
| 0.81 | 2023-08-30 | <ul style="list-style-type: none"> Vivante VTuner Beta V1.2.0 Vivante DC8200 series IP | <ul style="list-style-type: none"> Added Section 3.7.5, Tuning the Degamma Feature. Updated Section 3.7.6, Tuning the 3DLUT Feature by adding the setting of enlarge mode for color channel values. Made miscellaneous refinements. |
| 0.80 | 2023-07-21 | <ul style="list-style-type: none"> Vivante VTuner Beta V1.1.0 Vivante DC8200 series IP | Initial release. |