ModusToolbox™ Machine Learning Configurator user guide

ModusToolbox™ Tools Package version 3.6

Machine Learning Technology Pack version 3.0.0

Machine Learning Configurator version 3.0.0

About this document

Scope and purpose

The ModusToolbox™ Machine Learning (ML) Configurator is used in ML applications for adapting a pretrained learning model to an Infineon target platform. The tool accepts a pretrained ML model and generates an embedded model as a library. The model can be used along with your application code for a target device. The ModusToolbox™ ML Configurator also allows you to fit the pretrained model of choice to the target device with a set of optimization parameters.

Intended audience

This document helps application developers understand how to use the ML Configurator as part of creating a ModusToolbox™ application.

Document conventions

Convention	Explanation
Bold	Emphasizes heading levels, column headings, menus and sub-menus
Italics	Denotes file names and paths.
`Courier New`	Denotes APIs, functions, interrupt handlers, events, data types, error handlers, file/folder names, directories, command line inputs, code snippets
File > New	Indicates that a cascading sub-menu opens when you select a menu item

Abbreviations and definitions

The following define the abbreviations and terms used in this document:

- ML – machine learning
- NPZ – NumPy array in the zipped format. When unzipped, it provides validation data as NumPy arrays used by the tool.
- MAE – maximum absolute error
- MACC – Machine-learned ASAS Classification Catalog

Reference documents

Refer to the following documents for more information as needed:

- ModusToolbox™ Machine Learning user guide

About this document

- ModusToolbox™ Tensorflow tflite-micro library readme file

Table of contents

1 Overview 4
1.1 Supported library 4

2 Launch the ML Configurator 5
2.1 make command 5
2.2 VS Code and Eclipse 5
2.3 Executable (GUI) 5
2.4 Executable (CLI) 5

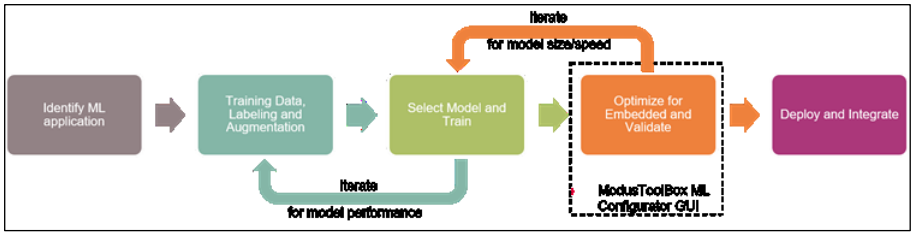
3 Quick start 6

4 GUI description 7
4.1 Menus 7
4.1.1 File 7
4.1.2 Edit 7
4.1.3 View 7
4.1.4 Help 8
4.2 Notice List 8
4.3 Status bar 8
4.4 General Settings tab 8
4.4.1 Output file prefix 8
4.4.2 Output folder 8
4.4.3 Target device 8
4.4.4 Model 8
4.4.5 Data 9
4.4.6 Calibration 10
4.4.7 Vela options 10
4.4.8 Verbose output 11
4.4.9 Generate Source 11
4.4.10 Cancel 11
4.5 Validate tabs 12
4.5.1 Available ports 12
4.5.2 Baud rate 13
4.5.3 Quantization 13
4.5.4 Visual samples 13
4.5.5 Validate 13
4.5.6 Cancel 13
4.5.7 Results table 13
4.5.8 Graph 13
4.6 Log pane 14

5 Version changes 15

1 Overview

The following shows the design flow for a typical application. The ModusToolbox™ ML Configurator GUI forms a vital part in fitting the model to the target platform



The ModusToolbox™ ML Configurator requires the support of the machine-learning tool ecosystem. This tool forms the central asset, which brings together other assets of the machine learning tool ecosystem, including Core tools, Inference engine, etc. as described in the ModusToolbox™ Machine Learning user guide. You can create a new application using the ML code example or you can add the ML library to an existing application using the Library Manager.

1.1 Supported library

Name	Version	Link
ModusToolbox ™ Tensorflow tflite-micro library	3.1.0	https://github.com/Infineon/ml-tflite-micro
ModusToolbox ™ Machine Learning Middleware	3.1.0	https://github.com/Infineon/ml-middleware

2 Launch the ML Configurator

There are numerous ways to launch the ML Configurator, and those ways depend on how you use the various tools in ModusToolbox™ software.

2.1 make command

As described in the ModusToolbox™ tools package user guide build system chapter, you can run numerous make commands in the application directory, such as launching the ML Configurator. After you have created a ModusToolbox™ application, navigate to the application directory and type the following command in the appropriate bash terminal window:

```
make ml-configurator
```

This command opens the ML Configurator GUI for the specific application in which you are working.

2.2 VS Code and Eclipse

VS Code and Eclipse have tools to launch the ML Configurator from within an open application. Refer to the applicable user guide for more details:

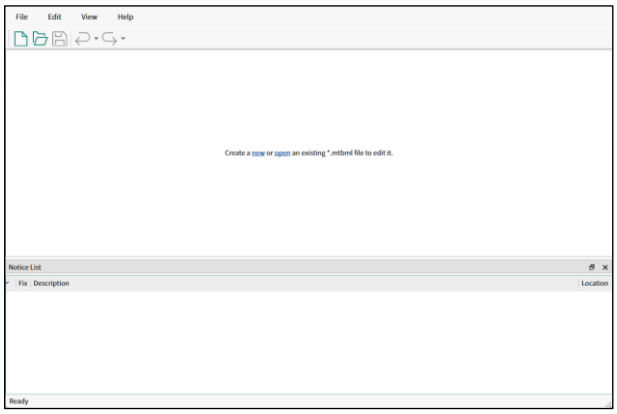
- VS Code for ModusToolbox™ user guide
- Eclipse IDE for ModusToolbox™ user guide

2.3 Executable (GUI)

If you don't have an application or if you just want to see what the configurator looks like, you can launch the ML Configurator GUI by running its executable as applicable for your operating system (for example, double-click it or select it using the Windows Start menu). By default, it is installed here:

<install_dir>/ModusToolbox/packs/ModusToolbox-Machine-Learning-Pack/tools/ml-configurator/

When launched this way, the ML Configurator opens without any settings configured. You can either open a specific configuration file or create a new one. See Menus for more information.



2.4 Executable (CLI)

The ML Configurator executable can be run from the command line, and it also has a command-line version of the executable as well. Running configurator executables from the command line can be useful as part of batch files or shell scripts to re-generate the source code based on the latest configuration settings. The exit code for the executable is zero if the operation is successful, or non-zero if the operation encounters an error. For more information about the command-line options, run the executable using the `-h` option.

3 Quick start

This section provides a simple workflow for how to use the ML Configurator.

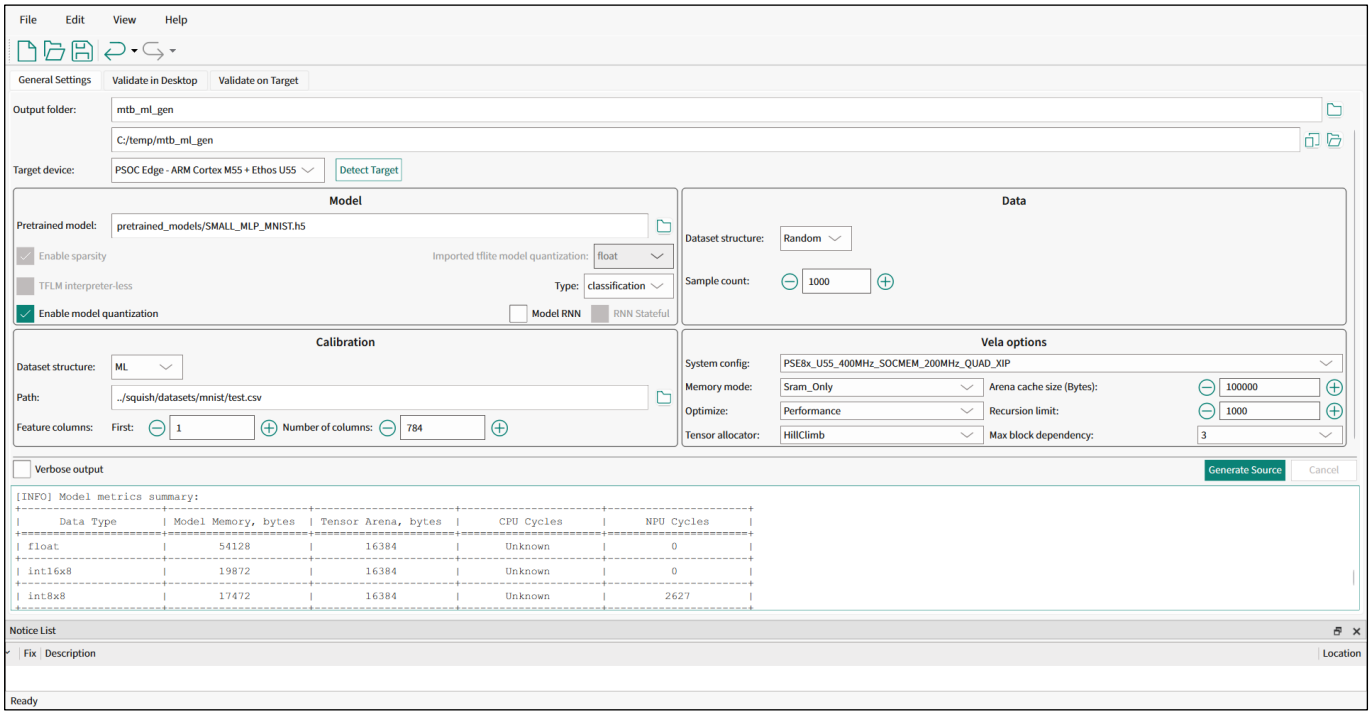
1. Create a new application based on a code example. For example:

   https://github.com/Infineon/mtb-example-ml-profiler
   https://github.com/Infineon/mtb-example-psoc-edge-ml-profiler

2. Launch the ML Configurator GUI.
3. Under the **General Settings** tab, enter/select project elements and model elements.
4. Click the **Generate Source** button and specify the configuration **.mtbml* file, as needed.
5. Under the **Validate in Desktop** tab, use the various setting to visualize output data.
6. If you have a device connected to your computer, under the **Validate on Target** tab, use the various setting to visualize output data.

4 GUI description

The ML Configurator GUI contains menus, tabs, and various fields to generate data for the selected project.



4.1 Menus

4.1.1 File

- **New** – Creates a new file with new configuration.
- **Open…** – Opens the configuration file.
- **Save** – Saves the existing file.
- **Save As…** – Saves the existing file under a different name.
- **Open in System Explorer** – This opens your computer's file explorer tool to the folder that contains the **.mtbml* file.
- **Settings**
  - **Automatically generate code after save** – Select this check box (not selected by default) to automatically generate code upon saving the configuration file.
- **Exit** – Closes the Configurator.

4.1.2 Edit

- **Undo** – Undoes the last action or sequence of actions.
- **Redo** – Redoes the last undone action or sequence of undone actions.

4.1.3 View

- **Notice List** – Displays or hides the Notice List pane. Displayed by default.
- **Toolbar** – Displays or hides the Toolbar.
- **Chart** – Displays or hides the Chart pane. Displayed by default.

- **Log** – Displays or hides the Log pane. Displayed by default.

4.1.4 Help

- **View Help** – Opens this document.
- **About ML Configurator** – Opens the About box for version information with links to open https://www.infineon.com and the current session log file.

4.2 Notice List

The **Notice List** pane combines notices (errors, warnings, tasks, and infos) from many places in the configuration into a centralized list. You can double-click the notice location to show the parameter causing the error or warning. For more information about the **Notice List**, refer to Infineon Device Configurator.

4.3 Status bar

The status bar displays various information about the status of the tool.

4.4 General Settings tab

The **General Settings** tab is the main tab. It opens when the ML Configurator opens.

4.4.1 Output file prefix

Provides the name for the output file. This name is used as the prefix for the generated files.

4.4.2 Output folder

This is the name and location where the generated files are placed relative to the location of the saved configuration file.

4.4.3 Target device

Allows you to select the target device to use. The supported devices:

- PSOC™ 6 – ARM Cortex M4
- PSOC™ Edge – ARM Cortex M55 + Ethos U55
- PSOC™ Edge – ARM Cortex M33 + NNLite2

4.4.4 Model

Use this section to import a pretrained model and set the parameters.

4.4.4.1 Pretrained model

This parameter allows you to select a pretrained model file from 3rd party ML frameworks.

Note: *For this version, the ML Configurator supports the h5 and Tflite file formats.*

4.4.4.2 Enable sparsity

This check box enables the use of a memory-efficient packed format for any sparse weights present in the model. Unchecked by default.

4.4.4.3 TFLM interpreter-less

If this check box is selected, only float quantization is supported and only for **Validate on Target**. Unchecked by default.

4.4.4.4 Enable model quantization

Select this check box to enable the Calibration section to generate an 8-bit quantized model, otherwise only a floating-point model is generated.

Note: *If the user does not import calibration data, the Notice List will display an error message.*

4.4.4.5 Imported *.tflite* model quantization

This option is only available if the Pretrained model is in the *.tflm* file format. There are two possible values: "float" (by default) and "int8x8". Select the value, which corresponds to the quantization of the selected model.

Note: *Quantization must correspond to the model quantization, otherwise, it will call an error from ml-coretools.*

4.4.4.6 Type

Selects the type of model to be imported. The supported values are:

- classification
- regression

The default value is "classification".

4.4.4.7 RNN model definition

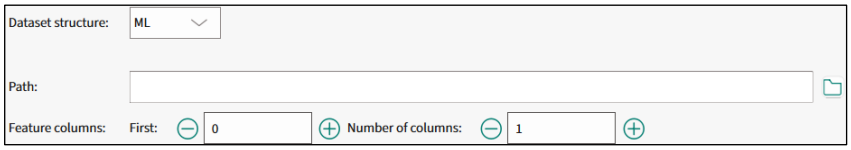
Select this check box to specify the model with the RNN operations.

4.4.4.8 RNN Stateful model definition

First, check the **Model RNN** flag. Then, select this check box to specify the model with the Stateful RNN operations.

4.4.5 Data

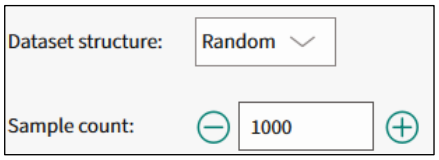
This section allows you to select the type of data to use as the validation input. This can be random data generated by the ML Configurator or an input file typically with data from an actual application source.



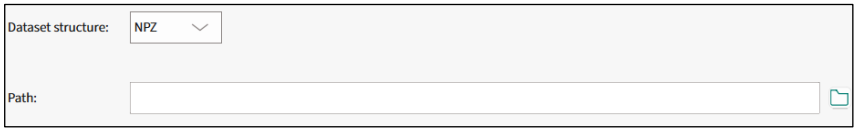
4.4.5.1 Dataset structure

The Dataset structure pull-down menu displays:

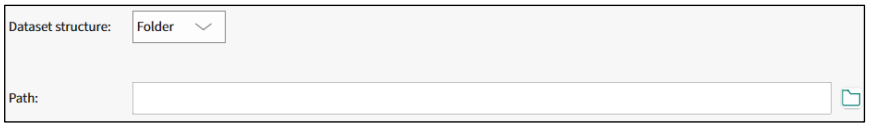
- **Random** – Random data generated by the ML Configurator. This option displays only in the **Data** section.



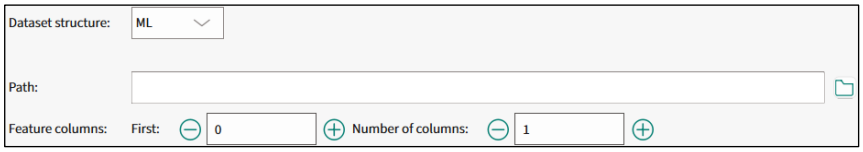
- **NPZ** – The **.npz* file (NumPy arrays in the zipped format).



- **Folder** – The **.txt* file used by ml-coretools. The file contains the paths to the data files (**.csv* or **.jpeg*).



- **ML** – The **.csv* file used by ml-coretools. The file contains only numeric data with no header or sample ID columns.



4.4.5.2 Path

The directory to a file or folder (depending on the option selected).

4.4.5.3 Sample count

The range is 1-1000.

4.4.6 Calibration

This section allows you to import model calibration data. This section is only enabled if you select the Enable model quantization check box.

4.4.6.1 Dataset structure

The same as Dataset structure in the **Data** section but without the Random option.

4.4.6.2 Path

The directory to a file or folder (depending on the option selected).

4.4.7 Vela options

This section only displays if the PSOC Edge - ARM Cortex M55 + Ethos U55 device is selected as the target device.

4.4.7.1 System config

- PSE8x_U55_400MHz_SOCMEM_200MHz_QUAD_XIP
- PSE8x_U55_400MHz_SOCMEM_300MHz_QUAD_XIP

4.4.7.2 Memory mode

- Dedicated_Sram
- Shared_Sram
- Sram_Only

4.4.7.3 Optimize

- Size
- Performance

4.4.7.4 Tensor allocator

- LinearAlloc
- Greedy
- HillClimb

4.4.7.5 Arena cache size

Any integer greater than or equal to 0.

4.4.7.6 Recursion limit

Any integer greater than or equal to 200.

4.4.7.7 Max block dependency

The options: 0; 1; 2; 3.

4.4.8 Verbose output

Select this checkbox to use the option of ml-coretools to print more detailed information at runtime. Unchecked by default.

4.4.9 Generate Source

The **Generate Source** button is used to analyze and validate pretrained models, as well as generate source data. The output of this analysis displays on the Log pane. The ML Configurator also generates an output file in the specified Output folder.

Specify the name and location of the existing **.mtbml* file before clicking this button for the first time. Otherwise, the ML Configurator will ask you to do that.

4.4.10 Cancel

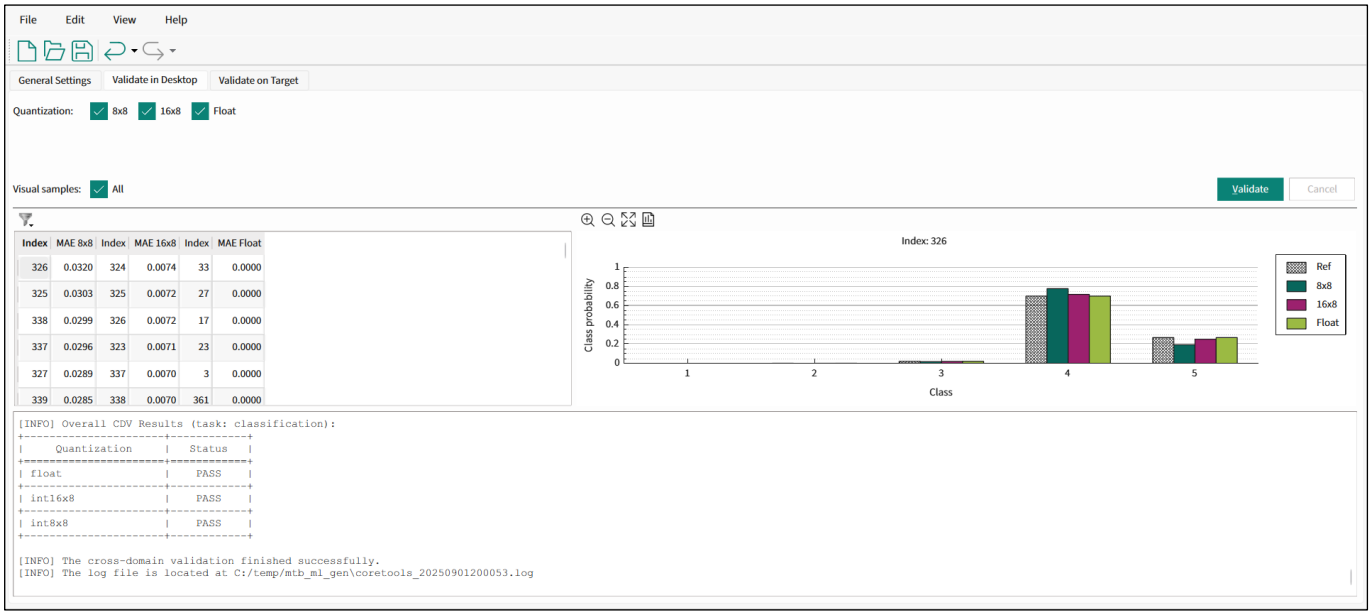
This button stops the generation process early by sending Ctrl+C to the subprocess and removing subsequent subprocesses from the queue.

4.5 Validate tabs

The ML Configurator contains two validate tabs: **Validate in Desktop** and **Validate on Target**. Both tabs provide controls to evaluate and visualize the output data.

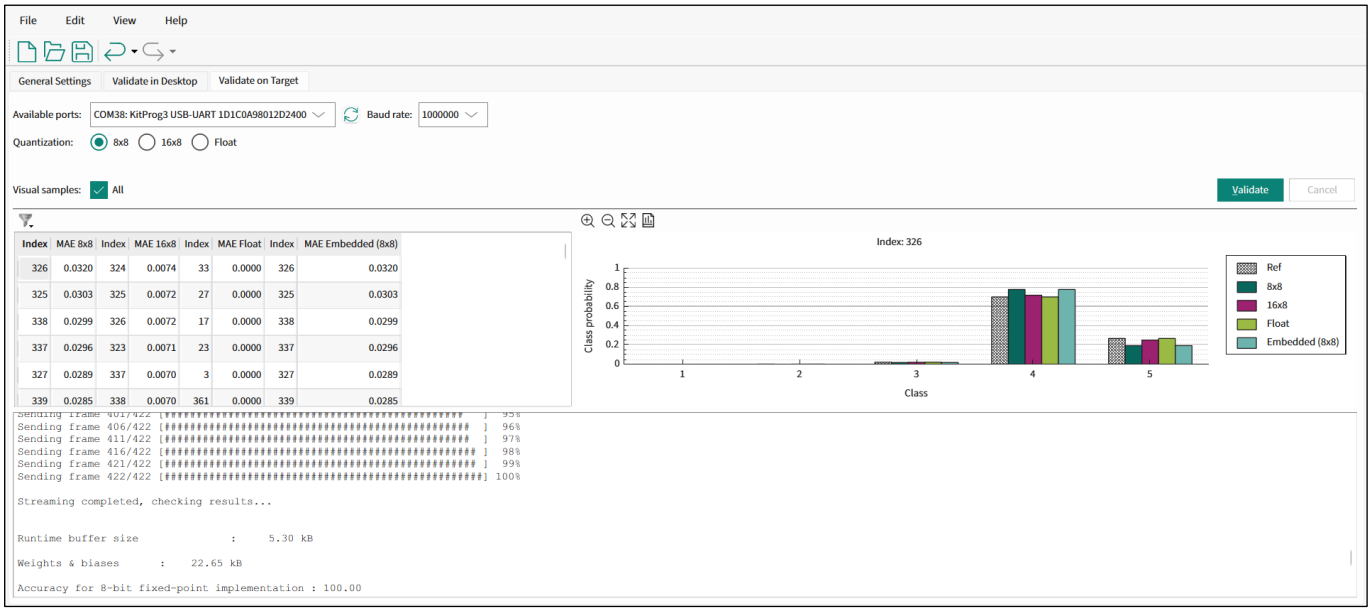
Validate in Desktop

Allows you to run validation without being connected to a device.



Validate on Target

Requires a device be connected to the computer. When using **Validate on Target**, ensure that you build streaming validation firmware on the same host OS, which the ML Configurator is run on.



4.5.1 Available ports

The **Available ports** pull-down menu allows you to specify the COM port, which connects to the target device when using the **Validate on Target** tab.

4.5.2 Baud rate

The **Baud rate** pull-down menu allows you to specify the baud rate from the values 9600, 19200, 38400, 57600, 115200, 250000, 500000, 1000000, and 2000000 when using the **Validate on Target** tab. The default value is 1000000.

4.5.3 Quantization

The **Quantization** check box allows you to select the 8-bit and/or float quantization options for the validation results to display in the table and graph as well as for one or more output files to be generated. For the **Validate in Desktop** tab, you can select multiple options; for the **Validate on Target** tab, you can select only one.

4.5.4 Visual samples

The **Visual samples** group contains the **All** check box and the input field, which allows you to select a number of samples to display in the table window. If **All** is checked, the input field will be hidden and the Notice List will display the warning "Displaying all samples may take a long time to complete the check". The default value is 100.

4.5.5 Validate

The **Validate** button generates validation result based on the quantization selected and the model data input.

4.5.6 Cancel

The **Cancel** button stops the validation process. Available only in running process.

4.5.7 Results table

The **Validate** tabs contain the table, which displays the validation result data with the input Index and its corresponding MAE value. The table may have up to three columns, depending on the selected quantization and filter options.

- Each of the table column headings provides the sorting option to view data in the ascending/descending order.
- The **Filter** button above the table allows you to select and deselect the quantization options, and thus display or hide the corresponding table.
- When you select each of the Indexes and their corresponding MAE value, the data displays in the graph form on the right side of the dialog.

4.5.8 Graph

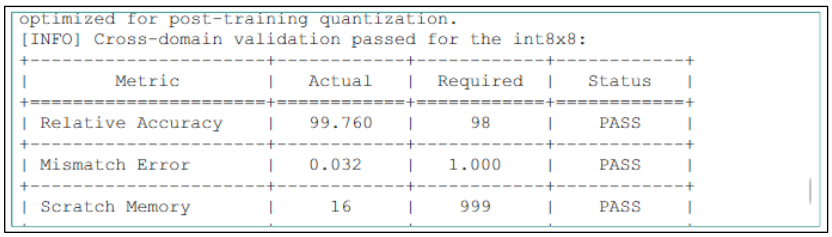
The **Graph** is a simple representation of the data to allow for easy viewing. The graph includes several buttons as follows:

- **Zoom in** – Increases the graph.
- **Zoom out** – Decreases the graph.
- **Zoom to fit** – Increases or decreases the graph to fit the size of the graph area.
- **Export to file** – Saves the graph to a file. The formats: *.png*, *.jpg*, and *.bmp*.

You can also click and drag your mouse to pan the graph in any direction.

4.6 Log pane

The Log pane displays results of **Generate Sources** or **Validate**.



5 Version changes

This section lists and describes the changes for each version of this tool.

Version	Change Descriptions
1.0	New tool.
1.10	Added **Open in System Explorer** in **File** menu. Changed how the target/feature columns are specified in Validation dialog.
1.20	Added **Edit** menu and **Undo**/**Redo** commands. Added **Advanced scratch memory optimization** check box. Added support for Validate on Target.
2.0	Changed GUI to use different validation tabs instead of separate dialogs. Removed Validate in Desktop and Validate on Target buttons. Added tflm inference engine. KERAS renamed to ifx. Added support *.tflite models (only for tflm inference engine). For tflm inference engine: • Only float quantization available (8x8 if calibration data presented). • Enable model quantization check box. • Enable sparsity check. • TFLM interpreter-less check. Added Model calibration section. Added notice list for show errors and info. Added cancel button for each tab.
3.0	Added support of PSOC™ Edge target devices. Added Chart and Log commands to the View menu. Removed the IFX inference support and Inference engine field. Moved the validation data section to the General Settings tab. Added new controls: Type, Visual Samples , Verbose output, and Baud rate. Added the Vela options section.

Revision history

Revision	Date	Description
**	2021-03-16	New document.
*A	2021-04-30	Updated to version 1.10.
*B	2021-09-08	Updated to version 1.20.
*C	2022-09-29	Updated to version 2.0.
*D	2023-10-26	Updated to version 3.0.
*E	2025-09-03	GUI style update.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Edition 2025-09-03

Published by

Infineon Technologies AG

81726 Munich, Germany

© 2025 Infineon Technologies AG. All Rights Reserved.

Do you have a question about this document?

Email: erratum@infineon.com

Document reference

002-32756 Rev. *E

Important notice
The information given in this document shall in no event be regarded as a guarantee of conditions or characteristics (“Beschaffenheitsgarantie”)

With respect to any examples, hints or any typical values stated herein and/or any information regarding the application of the product, Infineon Technologies hereby disclaims any and all warranties and liabilities of any kind, including without limitation warranties of non-infringement of intellectual property rights of any third party.

In addition, any information given in this document is subject to customer’s compliance with its obligations stated in this document and any applicable legal requirements, norms and standards concerning customer’s products and any use of the product of Infineon Technologies in customer’s applications.

The data contained in this document is exclusively intended for technically trained staff. It is the responsibility of customer’s technical departments to evaluate the suitability of the product for the intended application and the completeness of the product information given in this document with respect to such application.

Warnings
Due to technical requirements products may contain dangerous substances. For information on the types in question please contact your nearest Infineon Technologies office.

Except as otherwise explicitly approved by Infineon Technologies in a written document signed by authorized representatives of Infineon Technologies, Infineon Technologies’ products may not be used in any applications where a failure of the product or any consequences of the use thereof can reasonably be expected to result in personal injury.