

# Imatest Functional Interface

## The Imatest Functional Interface

Imatest's standard interface is a versatile and powerful tool for interacting with its many modules, but it can be complex and intimidating, especially for lab technicians or personnel with limited training in Imatest's many capabilities.

Also, there is always a chance that one or more settings may be different from the expected values. Imatest settings are stored in **imatest.ini** and updated whenever modules are run. They can be saved to a named file or loaded or merged from a named file. This gives users strong control the settings, but it can still be challenging to obtain the correct settings for production or lab environments— very important since erroneous settings can have unpredictable effects on test results.



These issues have been solved with the Imatest Functional Interface (introduced in Imatest 3.9)— a simple but highly-customizable alternative to the standard Imatest interface that ensures that the correct settings will always be used.

***The functional interface is of special interest to anyone who needs to run a large number of tests with consistent settings.***

## The Imatest Functional Interface

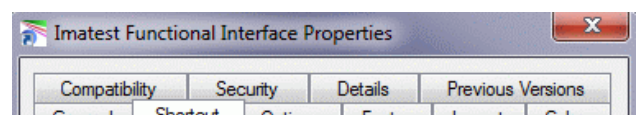
The Imatest Functional Interface is a separate interface for running Imatest modules that is

- **Simple and uncluttered:** Only the necessary functions are displayed.
- **Consistent:** You can specify a merge file that ensures that settings are correct and consistent.
- **Easy to use by personnel with limited training:** Operators don't need to worry about details of settings.

You can always return to the standard Imatest interface (main window) when the need arises.

### It can be opened

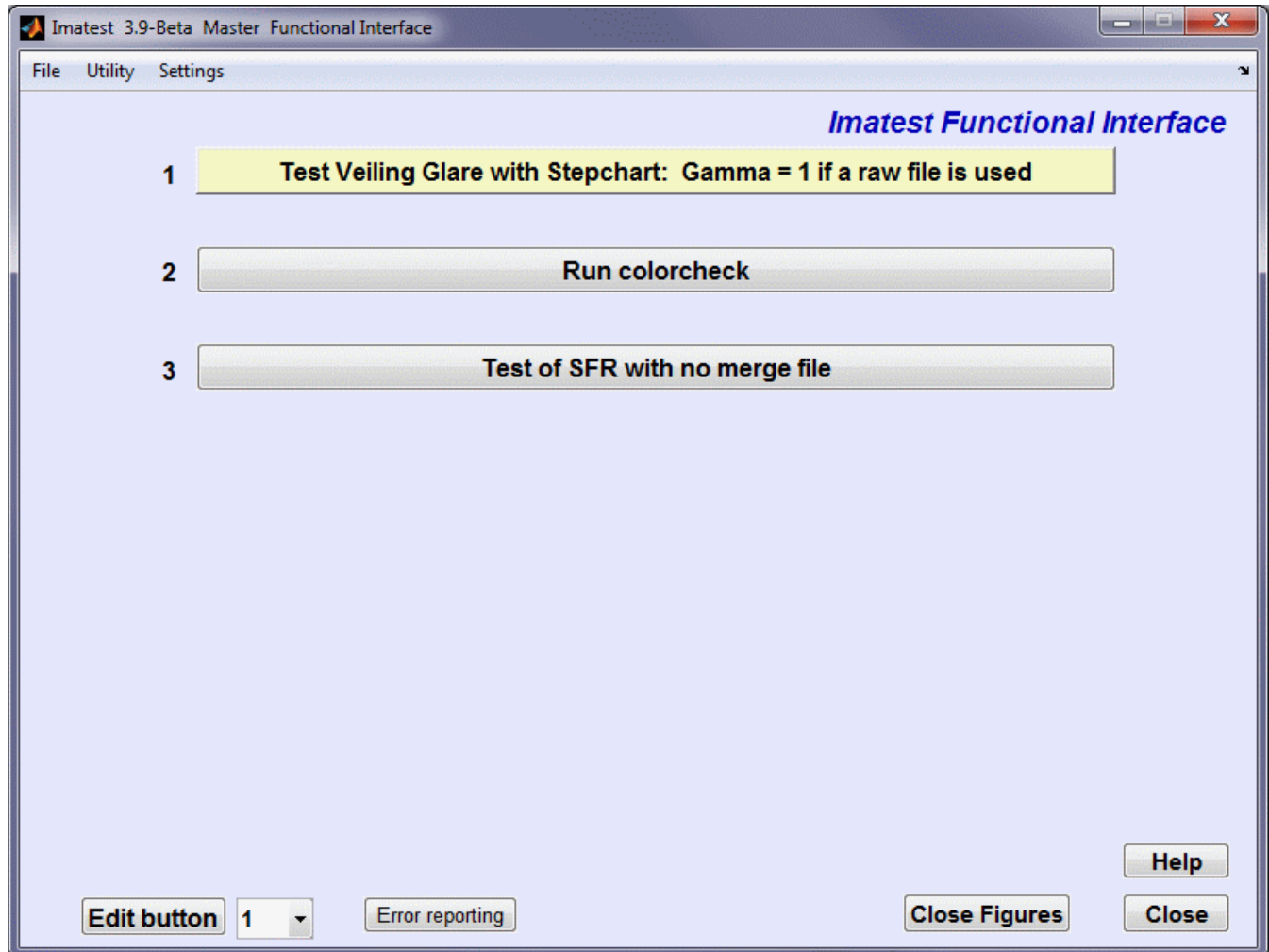
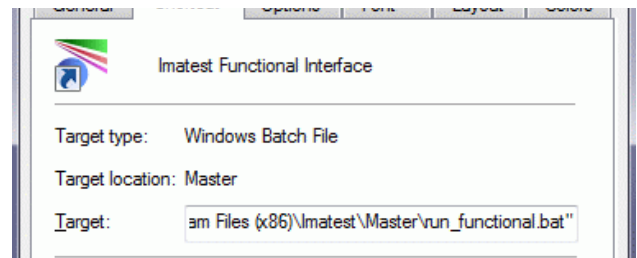
- (from Windows) by clicking on **All Programs, Imatest**, (edition if



needed...), **Imatest Functional Interface**,

- (from the Imatest main window) by clicking on **Utility, Functional Interface**.

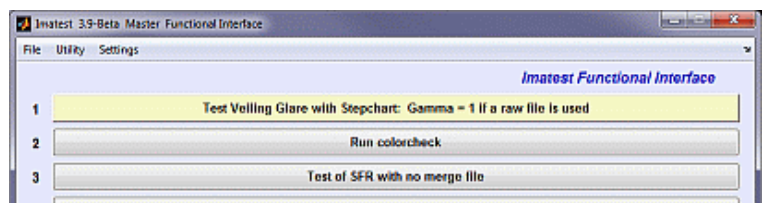
If you are working with shortcuts, change the Target (shown on the right) from **run\_imatest.bat** for standard Imatest Master runs to **run\_functional.bat**.



Imatest Functional Interface showing three populated buttons (out of 12 maximum); buttons 80% of maximum width; Large fonts selected for visibility

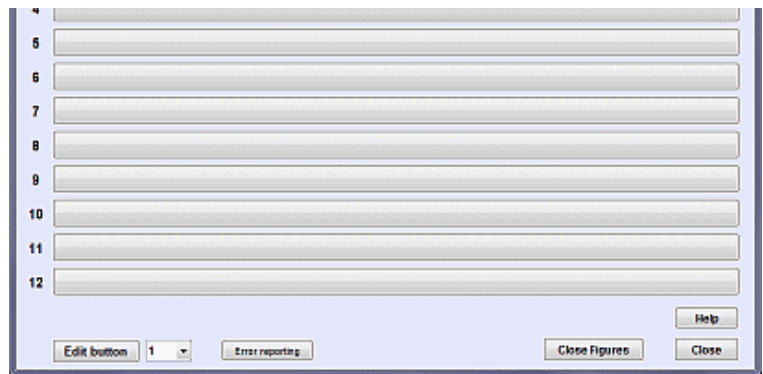
When you press a button in the Functional Interface,

- the contents of an ini file that contains the important settings is merged into imatest.ini, and



- A module is run. ***If the module has been set up correctly according to the instructions below, the correct settings will always be used.***

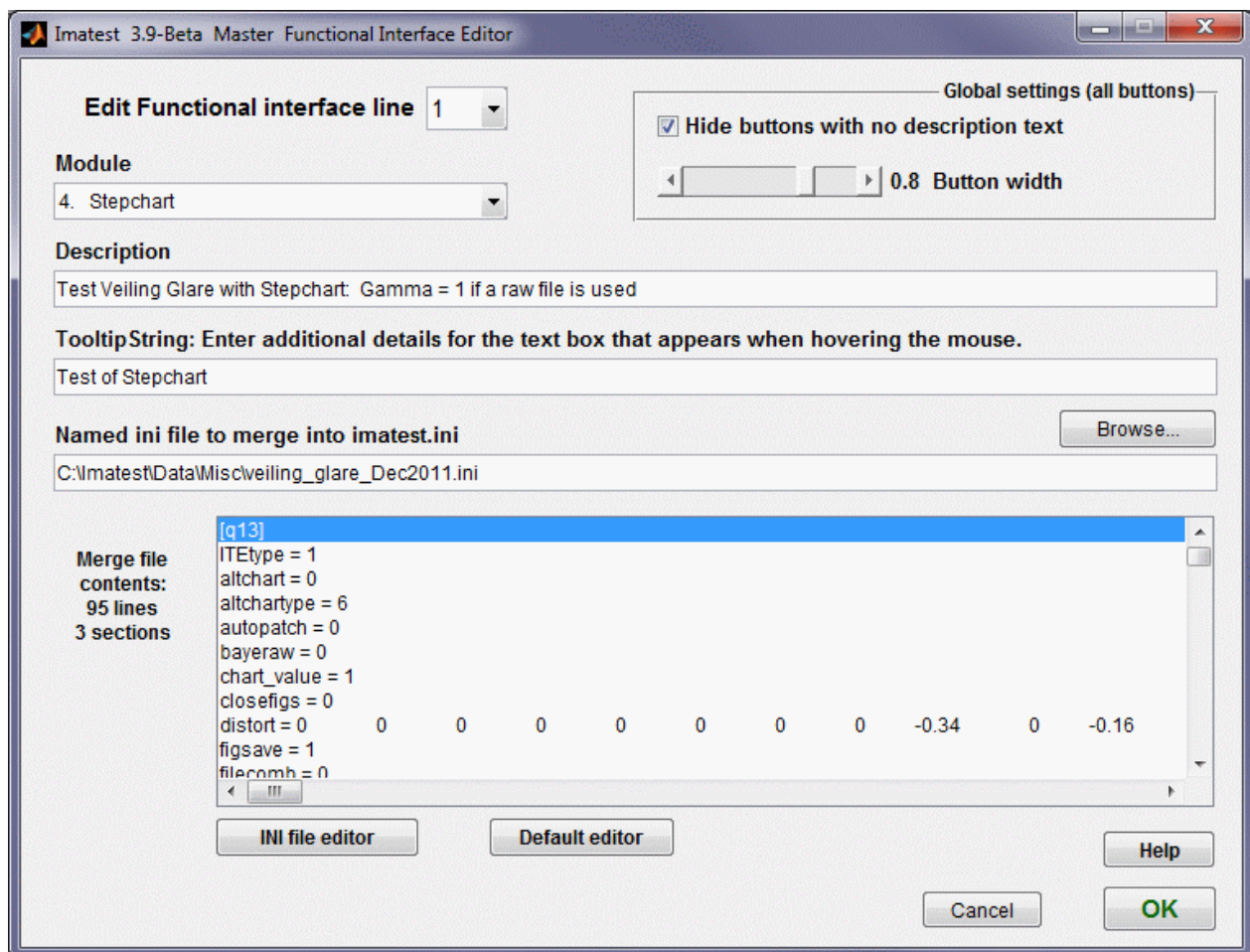
The button text, merge file, and module are specified in an editor that is opened with **Edit button** on the lower-left. (This menu is not displayed when the Functional Interface is opened from Windows, but it can be displayed by pressing **Settings, Show edit menu.**)



**Imatest Functional Interface showing all 12 buttons; full width**

## The Functional Interface Editor

The editor is opened by clicking on the popup menu just to its right, on the lower-left of the Functional Interface. (If this menu is not displayed, click **Settings, Show edit menu.**)



Functional Editor window (for Button 1 of the Functional Interface)

## Functional Editor settings:

**Global settings:** These affect all the buttons.

- **Hide buttons with no description text** If checked, only display buttons that contain text in the description field. If not checked, display all twelve buttons.
- **Button width** (slider): Set the (relative) button width: 1 is the maximum width. Can improve button appearance if description text is short.

**Module:** Choose the module to run: SFR, SFRplus, Light Falloff, etc. The long list includes Rescharts and Multicharts modules.

**Description:** This is the text that appears on the button (1 in the above example). It should be clearly descriptive.

**Tooltip string:** This is the text that appears in a box when the mouse hovers above the button. Enter additional detail and instructions here, if needed.

**Named ini file to merge into imatest.ini:** This is the ini file that contains the saved settings to be used. A complete copy of imatest.ini should generally *not* be used. The ini file should include

- **the section for the module to be run** ([q13] in the above example— an old name for Stepchart kept for backwards-compatibility),
- **[imatest]** (a few general settings),
- **[dcraw]** (settings for demosaicing the image using dcraw), and
- **[rdraw]** (if needed: settings for General Read Raw).

It is best to remove file names from the ini merge file.

opens the merge file in the INI File Editor described below: it is a very powerful tool for preparing merge files for the Functional Interface as well as Imatest IT (the Industrial Testing editions).

opens the merge file in a standard editor (Notepad in Windows).

You can use the **Edit Functional Interface line** menu at the top to edit a different button.

## INI File Overview

Imatest settings are saved in INI files: **imatest.ini**, **rescharts.ini**, and **multicharts.ini**. INI files are updated when settings are changed in input dialog boxes, in the **Settings** menu of the Imatest main window, or when the ini file is edited (see below). Settings can be edited and saved in named files for

use as Functional Interface merge files or IT (Industrial Testing) control files.

***You don't need to know the exact location** because the INI File Editor will open `imatest.ini` (or `rescharts.ini` or `multicharts.ini`) from the correct location, which derived from the system environment variable `$APPDATA$` (`C:\Users\user_name\AppData\Roaming` In Windows Vista/7 (English)). The full English-language Windows Vista/7 path name for `imatest.ini` is `C:\Users\user_name\AppData\Roaming\Imatest\imatest.ini`. The name may be different in other operating systems and languages.*

<p><b>INI files</b> are text files structured so that section titles are enclosed within brackets ( [ ... ] ) and lines have the form</p> <p>name = value, e.g., <code>nwid_save = 2601</code>. Value may be a number, a character string, or empty. The meaning is often obvious. The equal sign (=) <i>must</i> be present.</p>	<pre>[section] name1 = value1 name2 = value2 ... [next_section]</pre>
<p><b>Example:</b></p> <p><code>dispwelcome = 1</code> means display the Welcome screen when <code>Imatest</code> is started.</p> <p><code>edfile</code> is the image editor location, used to show saved screen displays in interactive modules (Rescharts, etc.)</p>	<pre>[imatest] dispwelcome = 1 evalvl = 1 exiftool = 2 expandplt = Min fontsize = 10 readexif = 1 shrink_largef = 1 acquire = 0 edfile = C:\Program Files (x86)\IrfanView\i_view32.exe [sfr] filecomb = 0 folder = C:\Imatest\Data\SFR\ save_dir = C:\Imatest\Data\SFR\Results\</pre>

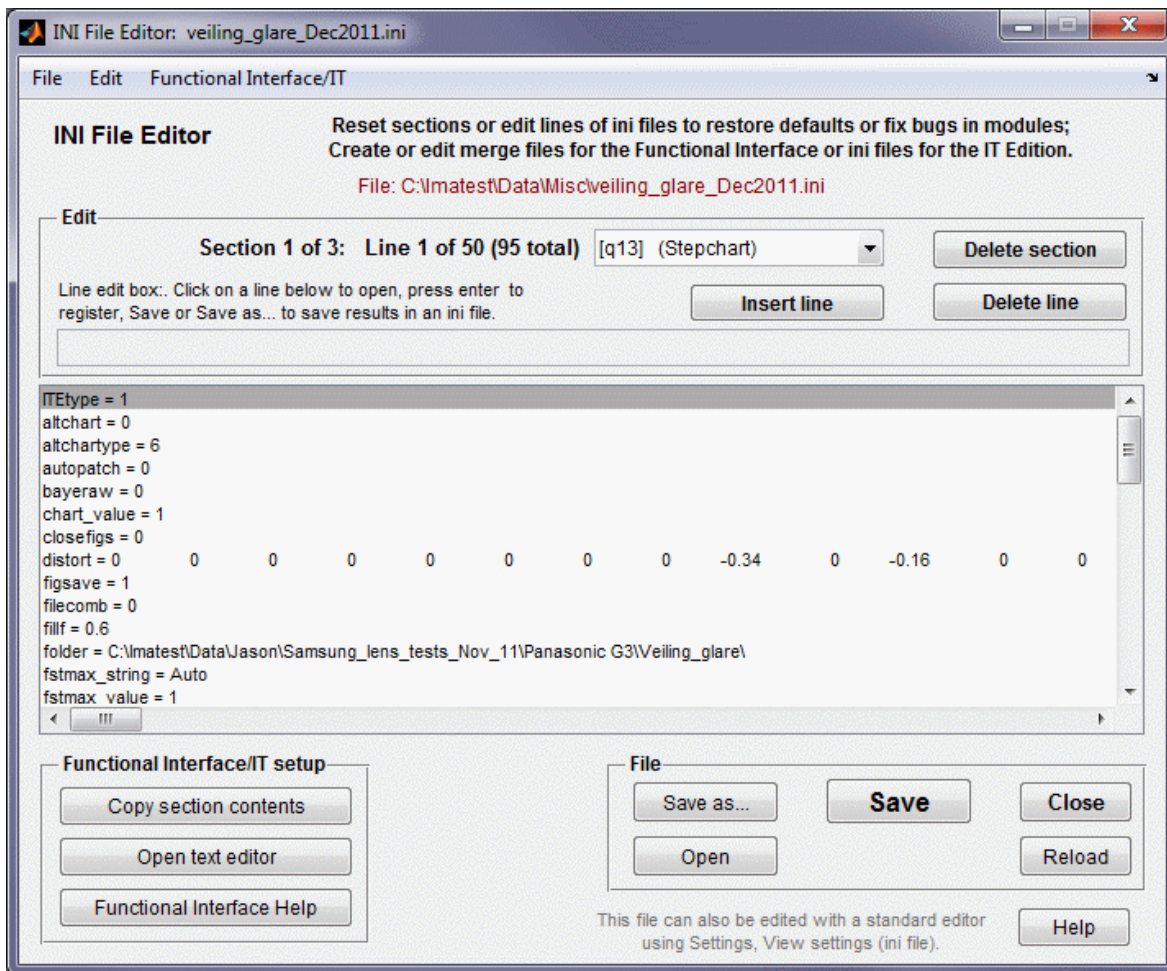
For Functional Interface merge files or IT control files, non-essential sections and data can be removed by following one of the procedures described below.

## Preparing a merge file

The **INI File Editor** is the same editor that opens when you press **Edit/Reset Module(s)** from the **Settings** dropdown menu of the `Imatest` main window, `Rescharts`, or `Multicharts`. It has been enhanced to facilitate creating of INI files for the Functional Interface as well as the IT (Industrial Testing) edition.



You can use it to set up a merge file fairly quickly following the steps below.



## INI file editor

- From the Imatest main window (or the [Rescharts](#) or [Multicharts](#) windows), **run the module** you want to add to the Functional Interface, making sure you have the settings you want to save (the correct ddraw demosaicing parameters, units, displays, etc.).
- **Click Settings, Edit/Reset Module(s)** to open the INI editor.

## Method 1: Use an external editor (Notepad).

- **Click Open text editor** in the Functional Interface/IT setup section on the lower-left. This opens an empty editor screen in Notepad.
- **Select the section** that corresponds to the module you just ran ([a13] in this example), then click on **Copy section contents**.
- **Paste (control-V) the clipboard contents** into the Notepad window.
- **Repeat** this procedure for [imatest], [dcraw], [rdraw], and any other relevant sections. For example, you may want to add [sqf] to [sfr] or [sfrplus] if you plan to

display Subjective Quality Factor.

- **Edit the merge file contents.** Go through the contents of Notepad, removing path names that might interfere with the run. If you don't do this you'll always have to navigate from the saved folder. (Keep certain path names such as color reference files for Colorcheck or Multicharts). It's also a good idea to remove image size and crop settings; these would tend to overwrite settings from recent runs, which should generally be saved. Typical names are `nht_save`, `nwid_save`, and `roi_mult` or `roi` (depending on the module).
- **Save the Notepad contents**, taking care with the location and path name. You can always open the file for editing later.

## Method 2: Use the INI file editor.

- **Click Save as...** to save the file to the intended merge file location: Do this first so you don't accidentally overwrite the original file.
- **Select a section to remove** from the merge file. (There may be quite a few.) In general you'll want to keep the section for the module of interest and related modules (for example, keep `[sqf]` for `[sfr]` or `[sfrplus]`). Most of the time you'll want to keep a few others, especially `[imatest]`, `[dcraw]`, and `[rdraw]`.
- **Edit the merge file contents.** Go through the remaining sections, removing path names that might interfere with the run. If you don't do this you'll always have to navigate from the saved folder. (Keep certain path names such as color reference files for Colorcheck or Multicharts). It's also a good idea to remove image size and crop settings; these would tend to overwrite settings from recent runs, which should generally be saved. Typical names are `nht_save`, `nwid_save`, and `roi_mult` or `roi` (depending on the module).
- Click **Save**. You can always open the file for editing later.

## To complete the button setup,

- Click **Utilities, Functional Interface** to switch to the Functional Interface.
- Click the **button number** you want to add or edit on the lower left of the Functional Interface window.
- **Edit the settings buttons** as needed. You'll need to add the ini merge file location to the **Named ini file to merge into imatest.ini** box. Then click **OK**.

The button should be ready to boogie. :-) Of course you'll want to test it.

We are preparing an ini file reference page, which should be helpful with the Functional Interface as well as the Industrial Testing (IT) edition.