**MetaMetrics Documentation**

# Overview of MetaMetric Tabs

Each tab represents some stage of using the app, flowing in order of use from left to right.

The program structure will be outlined in the following style:

1. Tab Name
   1. [Program name of component], [Displayed name of component to user], (Component type)
      1. Purpose
      2. Result
      3. Approach
      4. Functions used
      5. Notes
   2. Name of 2nd UI element interacted with
      1. Purpose
      2. Result
      3. Approach
      4. Functions used
         1. Function 1
         2. Function 2
         3. …
      5. Notes
   3. …
2. **Data Import**
   1. [app.Button\_ImportTable], [Import Table], [Button]
      1. Purpose Display the spreadsheet of SPDs the user has on the app.
      2. Result app.UITable\_ImportedFile ui table is populated with the spreadsheet
      3. Functions
      4. Approach
      5. Notes
   2. [app.EditField\_ImportedFileName], [Imported File Name], [Edit Field]
      1. Purpose Notify/remind user which file they loaded
      2. Result
      3. Functions
      4. Approach The value property is assigned during 1a
      5. Notes The component is called an edit field, but it is not editable. It is a misnomer. Interactivity is disabled.
   3. [app.UITable\_ImportedFile], [EMPTY], [UI Table]
      1. Purpose The user selects the entire library of spds by drag-selecting the power values from 380nm to 780nm. This method of importing data enables the user to tell the program interactively where the spd power values begin. Alternatively, there could be strict standards such as “1st row of table are the names and the other rows are spd powers from 380 to 780nm”
      2. Approach UITables have a selection property that can be enabled. The selection is only used whenever the Use Selected SPDs button is pressed
   4. [app.Button\_UseSelectedSPDs], [Use Selected SPDs], [Button] (Button)
      1. Purpose Load the library of SPDs into the app for later selection
      2. Approach Query UITable\_ImportedFile.Selection property.
      3. Result An app property is assigned to the library.
3. **Data Validation**
   1. [app.UIAxes\_ImportedSPDs], [Imported SPD Channels], [UIAxes]
      1. Purpose Plot the entire library as dashed lines and the selected channels as solid lines
      2. Functions
      3. Approach Upon pressing app.Button\_UseSelectedSPDs, all spds will be plotted as dashed lines. Upon any updating of app.UITable\_ChannelSelection, the dashed lines of channels selected will become solid lines.
   2. [app.UIAxes\_ChromDiagram], [CIE Diagram], [UIAxes]
      1. Purpose To display a chromaticity diagram of the library and selected SPDs.
      2. Functions
      3. Approach
         1. The chromaticity diagram will be plotted on startup.
         2. Upon app.UserSelectedSPDs button being pressed, the chromaticities will be plotted.
         3. Upon updating of app.UITable\_ChannelSelection the chosen chromaticites will be filled in.
      4. Notes Right now only 2 degree is available.
   3. [app.UITable\_ChannelSelection], [Channel Selection Table], [UITable]
      1. Purpose Enable user to see their library with columns as channels and rows as metrics that can help them choose. Full-width half max, centroid wavelength, peak wavelength, XYZ2, XYZ10, alpha opics, LER2, LER10.
      2. Result
      3. Functions
      4. Approach Columns will be spd channels. The first row will be a checkbox system. The following rows will be metrics of those SPD channels
      5. Notes Should the row/column be flipped?
4. **Optimization Settings**
   1. [app.UITable\_Constraints ], [(panel above named: Optimization Constraints) ], [UITable ]
      1. Purpose Enable user to interactively select which metrics to use as constraints, what type of constraints to use, and what value they are constrained to.
      2. Result Constraints are passed to optimization function
      3. Functions
      4. Approach A dummy spd (equal energy) is passed into spdToTM30 and then into spdToAlphaOpics where a structure is created and assigned fieldnames and value corresponding to the metrics associated with those functions. These fieldnames become the rows.

The columns of the uitable are merged horizontally with the fieldnames column by programmatically creating the constraint defaults via the numel function.

* + 1. Notes There should be a parent function for all the functions that create metrics. Right now the metric functions are spdToTM30 and spdToAlphaOpics, but those two should be under something like spdToMyMetrics. This same function (creating all the metrics from an spd) needs to be used in the optimization function.
  1. [ app.DropDown\_MaxOrMinGoal], [Maximize or Minimize? ], [ Drop Down]
     1. Purpose Let user select whether the metric chosen to optimize is to be minimized or maximized under the given constraints
     2. Result
     3. Functions
     4. Approach
     5. Notes
  2. [app.DropDown\_MetricGoal ], [Which Metric? ], [Drop Down ]
     1. Purpose Enable user to select which metric to be optimized
     2. Result
     3. Functions
     4. Approach A dummy spd (equal energy) is passed into spdToTM30 and then into spdToAlphaOpics where a structure is created and assigned fieldnames and value corresponding to the metrics associated with those functions. The list of goals, as well as constraints, is determined via using fieldnames function of that dummy structure.
     5. Notes There should be a parent function for all the functions that create metrics. Right now the metric functions are spdToTM30 and spdToAlphaOpics, but those two should be under something like spdToMyMetrics. This same function (creating all the metrics from an spd) needs to be used in the optimization function.

1. **Optimization Results**
   1. [app.UIAxes\_OptimSPD ], [Optimized SPD ], [UIAxes ]
      1. Purpose Display the optimized spd on a radiant watt vs wavelength
      2. Result
      3. Functions
      4. Approach
      5. Notes
   2. [app.UITable\_OptimzationResults ], [ ], [UITable]
      1. Purpose Display all of the metrics of the best optimization inside of the app, so the excel sheet does not need to be created.
      2. Result
      3. Functions
      4. Approach
      5. Notes
   3. [app.Button\_ExportOptimization ], [Export Optimized SPD ], [Button ]
      1. Purpose Enable user to save the optimized SPD if they like the results.
      2. Result
      3. Functions
      4. Approach
      5. Notes