

# Iris DVA

---

**Author:** [Khris Griffis](#)

---

**Official release date:** **TBD**

---

## *Contents:*

- [Introduction](#)
  - [Some Commentary](#)
- [Release Schedule](#)
  - [2020-09-15 Release 2.0.104](#)
  - [2020-04-10 Release 2.0.97](#)
  - [2020-04-01 Release 2.0.96b](#)
  - [2020-03-12 Release 2.0.80b](#)
  - [2020-02-24 Release 2.0.76b](#)
  - [2020-02-23 Release 2.0.75b](#)
  - [2020-02-21 Release 2.0.73b](#)
  - [2020-02-19 Release 2.0.70b](#)
  - [2020-02-14 Release 2.0.70a](#)
  - [2020-02-10 Release 2.0.60a](#)
  - [2020-02-02 Release 2.0.56a](#)
  - [2020-01-28 Release 2.0.53a](#)
  - [2020-01-21 Release 2.0.51a](#)
  - [2019-12-16 Release 2.0.50a](#)
  - [2019-12-13 Release 2.0.49a](#)
  - [2019-12-12 Release 2.0.48a](#)
  - [2019-11-19 Release 2.0.45a](#)
  - [2019-11-15 Release 2.0.42a](#)
  - [2019-10-22 Release 2.0.40a](#)
  - [2019-09-17 Release 2.0.32a](#)
  - [2019-09-15 Release 2.0.31a](#)
  - [2019-09-13 Release 2.0.3a](#)
  - [2019-09-05 Release 2.0.29a](#)
  - [2019-09-04 Release 2.0.24a](#)
  - [2019-09-03 Release 2.0.22a](#)
  - [2019-09-02 Release 2.0.21a](#)
  - [2019-08-29 Release 2.0.18a](#)
  - [2019-08-27 Release 2.0.10a](#)
  - [2019-08-26 Release 2.0.8a](#)
  - [2019-08-25 Release 2.0.7a](#)
  - [2019-08-22 Release 2.0.2a](#)

- [2019-08-19 Release 2.0.1a](#)
- [< 2019 PreRelease](#)

# Introduction

---

Iris DVA is a MATLAB based tool for visualizing and analyzing electrophysiology data. Iris was originally created to standardize the process for viewing and performing offline analysis of physiological data acquired via [Symphony](#) in the Sampath Lab, UCLA. While Iris is still in development stages, a secondary goal was to make the software extensible and modular to allow for any data and any analysis to be easily created by the end-user. Documentation is essentially non-existent except which resides in code comments, UI tooltips, and this document, some of which is likely to be out-dated or simply inadequate... Sorry about that! Eventually, documentation will reside in this [gitbook](#).

*Usage Note:* Iris was developed in MATLAB 9.6+ (>=2019a) on windows (x64) platform using both 1080p and 4K displays. That said, it has ~~not been~~ been minimally tested on Mac and Unix distributions of MATLAB and if screen scaling (typically handled by the OS) is other than 100%, visual issues may be present. This app utilizes MATLAB's newer `uifigure` (app designer) which has some inherent bugs. One of which occurs sometimes when `drawnow()` is invoked, causing MATLAB to hang during a "coalescer flush" (see: `matlab.ui.internal.controller.FigureController/flushCoalescer` (line 473) ) and the app ui to stop responding. The best way to resolve this is to issue a `drawnow()` command from the command window with the debugger in "pause on error" mode and pressing `cmd+c` or `Ctrl+c` when the console hangs. It should take you to the `flushCoalescer` method. Hit continue in the editor toolstrip (may have to press more than once). Iris will resume after that. I've also noticed that this happens in other appDesigner apps. Hopefully the next MATLAB release will resolve this.

*Update (January 2020):* As of MATLAB 2019b, the coalescer issue persists, though less frequently. Iris is compatible with 2019b and I recommend updating to the latest release (Release 3 at the time of writing).

*Update (April 2020):* As of MATLAB 2020a, the coalescer issue appears to have resolved, at least to the point where it hasn't been encountered thus far. We recommend updating to the latest version of MATLAB (2020a U1 at time of writing) and to the latest edition of Iris DVA (>=2.0.9x).

## Some Commentary

A major issue with the web-based `uifigure` is that it is remarkably ~~unstable~~ slower, especially when compared to the traditional Java-based figures. Where MATLAB is strongly linked to Java, at least when it comes to UI components, the 'new' web-based figures are linked to JavaScript. The new web-UI leverages React and Dojo toolkits to create and manage elements in the DOM, this is presented in a modern Chromium web browser, all of which means we can make edits using CSS, HTML and JavaScript directly from MATLAB (see: [this](#), [this](#), [this](#), [this](#) and [that](#)). From there, we observe a slowdown when MATLAB has to send data to the DOM, or if we send too much data to the DOM, but we see cleaner looking graphics and get nice, responsive control elements. The tradeoff is real and we are working hard to optimize for performance wherever possible.

### **Update** since MATLAB Release 2020a

Extensive testing of Iris DVA (Release 2.1+) has revealed that The Mathworks Inc. (TMW) have resolved and improved a number of performance issues related to

the App Designer (e.g. `uifigure`). Although TMW have made these considerable efforts, the issue of displaying many data points on a `uiaxes` is unfortunately slow and hinders performance of other components displayed on the canvas. For Iris, we always optimize with the data in mind and aim to improve useability and accuracy, sometimes at the cost of aesthetic. To this end, we have employed a number of "*hacks*" that may not be compatible in future releases, though we have taken care to ensure if the hack breaks, functionality will not be lost.

---

# Release Schedule

---

## 2020-09-15 Release 2.0.104

Compatibility with 2020a and minor updates.

- Iris:
  - Redesigned main view with grid layout managers (2018b+ compatible) and added resize behavior.
  - Improved plotting mechanism.
  - Analysis Interface:
    - Refresh feature to load new / drop removed analyses and load defaults for the selected analysis.
    - Added edit button to open the current analysis in the editor.
    - Added Update button to set the current analysis input settings as defaults.
    - Fixed some minor bugs on analysis cleanup.
  - Dropped beta status (this is now a release candidate)
  - Updated Symphony H5 reader function.
- IrisData:
  - Added device centric methods:
    - Renamed `AppendToDatums` method to `AppendDevices`.
    - Added `GetDevice`, `RemoveDevice` methods.
    - Updated `Split` method to now allow `'devices'` as a `'groupBy'` parameter.
- Iris > Builtin Modules > ResponseFamilies:
  - BUG: This module encounters issues when MATLAB tries to automatically update it from 2019b to 2020a compatibility which causes the module to fail to load properly in some cases. The next release will solve this.
  - Major update and redesign in the works.

## 2020-04-10 Release 2.0.97

Minor updates. Added command line tools to manage Iris.

## 2020-04-01 Release 2.0.96b

Major updates and compatibility with MATLAB 2020a release. Entered final preparations for release 2.1.

## 2020-03-12 Release 2.0.80b

Minor bug fixes, updates and improvements.

- Iris:
  - Fixed issue when loading multiple IrisData objects from the same original file.
- IrisData:

- Added grouping information ascertained for Aggregation to the UserData field of the returned IrisData object.
- Added ability to "update" inclusion list via a set method.
  - User can now easily reassign inclusions by using:
    - `iData.InclusionList = lst;` or `iData = set(iData, 'InclusionList', lst).`
    - `lst` must be a boolean vector with `iData.nDatums` length.
    - When using the `iData.set()` method, an output **is required**.
    - When using the `set.InclusionList` method (`iData.InclusionList = lst;`), no output is required as the new object is overwritten in-place.
- Iris > Builtin Modules > ResponseFamilies:
  - Added ability to edit inclusion of datums from the fit data.
  - Fixed units strings for IrisData outputs.

## 2020-02-24 Release 2.0.76b

Minor bug fixes.

## 2020-02-23 Release 2.0.75b

Minor bug fixes and enhancements.

- Iris:
  - Modules are now tracked.
    - New module requests send data to the module's `setData(iData)` method.
    - The Module's `setData(iData)` method should handle 3 cases:
      1. `iData` is an IrisData object.
      2. `iData` is a file path to a `*.idata` file.
      3. `iData` is an empty array, i.e. `[]`.
  - Updated error handling.
- Iris > Builtin Modules > ResponseFamilies:
  - Added preferences for exporting figures.
    - Set whether 95%CI's will show on fit datums.
    - Set whether to show the original datums along with aggregates on export.
- IrisData:
  - Fixed bug where using the `EditDatumProperties` method would ignore changes if the datum was not switched after changing.

## 2020-02-21 Release 2.0.73b

Minor bug fixes and minor feature updates.

- Iris:
  - Added screenshot keyboard shortcut (actually just connected the keypress to its method)
    - Screenshots are stored in the user defined output directory as PNG (portable network graphics) files.

- IrisData:
  - Added new method for appending new data onto existing IrisData object: `IrisData.AppendToDatums(...)`.
    - See `doc IrisData`, or `idata.help('AppendToDatums')` for usage syntax.
- Iris > Builtin Modules > ResponseFamilies:
  - Added ability to switch between calculating isomerizations and photon density.
    - If photon density is selected, the fit X-axis values are mapped from the light intensity volatge through the calibration table to the calibration intensity units and then attenuated by parameters configured on the ND tab. Stimulus duration, collecting area and spectral template (absoprtion probability) are not taken into account when "Photon Density" is selected for Intensity Type (on Fit tab).
  - More settings will persist between calls to the module.
    - Aggregation processing settings, e.g. filter, baseline parameters etc.
    - Fit equation parameter estimates and labels

## 2020-02-19 Release 2.0.70b

Release is now in beta! Key change from alpha version is that all utility functions that originally were in +kg, i.e. `kg.*`, are now in +utilities, e.g. `utilities.uniqueContents()`.

## 2020-02-14 Release 2.0.70a

Critical update for Reading Symphony H5 files. Other minor bug fixes and some new features.

- Iris:
  - Fixed issue with reading Symphony files where multiple sources were nested.
  - Updated SessionConverter utility for use with new readers
  - Added new library utility function, `kg.isWithinRange()`, see the entry below for the IrisData static method of the same name.
- IrisData:
  - Added feature to `view()` method that allows the user to view 'data' properties graphically.
  - Added static method, `IrisData.isWithinRange()`, which accepts a vector of values to check against a single set of extents (`[lower,upper]`) with the option to set the lower and upper bounds as exclusive or inclusive (default).
  - Added feature to edit datum properties. New method, `EditDatumProperties()`, opens a graphical interface to navigate Datum properties and modify values as needed. This method attempts to recast modifications to same class as previously, e.g. if you edit a cell which originally contained a value of type `uint8`, then the modification will be casted back to `uint8` type. For the most part, this new feature is for making small corrections, such as correcting an ND filter label or changing the datum inclusion status. Use responsibly!
- Iris > Builtin Modules > ResponseFamilies:
  - Updated layout further.
  - Added export feature for modifying the time values based on the stimulus time (on display tab).

- Now, if a value is entered on the display tab for stimulus presentation time, both the plot and results output will have the time-code corrected by the stimulus start time. If no entry, or 'none', is made then the time-code is the same as the original data.
- Added View menu entry, Datums, for making changes to metadata on the datum level.
  - View menu entry, Data, opens a second window which allows the user to alter data properties on a per-datum level.
  - To reset changes made during editing, re-import the idata file. All changes are volatile.

## 2020-02-10 Release 2.0.60a

### Minor updates

- Iris
  - Added support for reading IrisData objects (`*.idata`).
    - `*.mat` files containing `IrisData` objects can also be imported.
    - To support the above, a new utility function is introduced, `kg.r_isFielda()`, which recursively searches a struct for a field type and returns two outputs, a boolean and the first located field contents to match the requested type.
- Iris > Builtin Modules > ResponseFamilies:
  - Updated layout.
- IrisData:
  - Fixed typos.

## 2020-02-02 Release 2.0.56a

### Minor improvements and bug fixes

- Iris
  - Added access to modules independent of data readiness. If a module is requested while Iris does not have data loaded, the module is initialized with an empty input argument (`[]`).
- Iris > Builtin Modules > ResponseFamilies:
  - Fixed bugs related to persisting selected properties, introduced in Release 2.0.53.
  - Completed validation for "Custom" entries in the stimulus calibration table (template `$\lambda_{\max}` and collecting area).

## 2020-01-28 Release 2.0.53a

### Minor updates and improvements.

- Iris > Builtin Modules > ResponseFamilies:
  - Added ability to export data and metadata to csv,txt,tsv.
  - Added toggle to let user decide to fit to normalized or non-normalized values.
    - This should allow, for example, a 3-parameter equation without forcing a [0,1] range.
  - Setting smooth span (in the peaks menu) to 0 now disables smoothing.
  - Some properties now persist, such as the stimulus and nd calibration tables and some of the related dropdowns, the smoothing parameters, and the fit equation.



- A reset method is available in the View menu.
- Expanded and clarified collecting area used for selected cell types. Now, the calibration tab contains the lambda max as well as the collecting area. Both the spectral template lambda max and the collecting area can be set, independently, to **'Custom'** and the table can be modified.
- **TODO: values entered into the cell are not strictly validated and invalid values will throw errors elsewhere which may not be clear. I created an empty method for validation but the logic needs to be written.**

## 2020-01-21 Release 2.0.51a

Minor bug fixes and optimizations.

- Iris > Analysis > Export > Session now exports an IrisData file rather than a struct.
- Iris > Builtin Modules > Response Families:
  - Added stimulus arrow selectable by data property or "custom" with an edit box.
  - This stim triangle auto orients opposite of the response direction (roughly). and is copied to the output figure.
- IrisData:
  - Updated docs some more.
  - Plot method allows new arguments to control x and y limiting. Use **'computeYLimitBy'** and **'computeXLimitBy'** with the value of either **'Data'** (default), **'Axes'** or **'None'** to use the data being plotted, the data existing in the axes or no limit modification at all.

## 2019-12-16 Release 2.0.50a

Minor updates to builtin modules and IrisData.

- Iris > Builtin Modules > Response Families:
  - Added support for ND tables stored in character delimited files (csv,tsv,etc.).
  - Fixed typos and minor improvements
- IrisData:
  - Updated some help doc

## 2019-12-13 Release 2.0.49a

Minor updates and new features.

- IrisData:
  - Added new method, **view(['notes','info','properties'])**, to view the associated notes, meta-information or data properties, respectively, in a graphical table similar to those as seen in the main application.
  - Updated some help docs
  - Fixed some minor typos.
- Iris > Builtin Modules > Response Families:
  - Updated menus and file name display string

- Added View menu to allow access to notes, file information and datum properties as popup figures containing tables.
- Minor bug fixes and optimizations.

## 2019-12-12 Release 2.0.48a

Stability updates and minor improvements. **Please update MATLAB to 2019b Update 2.**

- Iris:
  - Minor improvements related to Iris paths.
  - Enhanced module: Response Families
    - Added fit from curve fitting toolbox (previously depended only on optimization toolbox)
    - Added ability to select fit pre-processing method:
      - Dot Product (Sampath lab default processing method)
      - Peak Detection (Use with restricted analysis window, see Display tab of the module)
    - Exported figures will use the whole data set rather than just the analysis window portion
    - Minor bug fixes and enhancements pertaining to Data handling (especially regarding IrisData object with multiple devices)
- IrisData:
  - Added public method, `getGroupBy(...)`, for generating a grouping table, such as the one used in the `Aggregate()` method for the 'groupBy' parameter. This is useful when determining grouping vectors outside of the IrisData object.
  - Added public method, `getDomains([devices])`, for quickly getting the X and Y domains for each device (optionally provide device name to get only that device data domains).
  - Updated some help documentation
  - Minor bug fixes and general enhancements

## 2019-11-19 Release 2.0.45a

- Iris:
  - Optimized filtering.
  - Bug fixes and improvements for modules
- IrisData:
  - Bug fixes and improvements for methods that don't return values
  - Added `help()` and `explore()` methods to see doc or code for specific methods.
  - Optimized filtering.
  - Improved save to session method to allow easier importing IrisData objects into Iris.

## 2019-11-15 Release 2.0.42a

- This release fixes a few bugs related to opening modules on various platforms.
- Optimized the butterworth filtering.

## 2019-10-22 Release 2.0.40a

It is sometimes useful to split and merge different data objects, perhaps after some analysis has been performed on individual groups one wants to perform a new analysis. Now we have a few methods for updating IrisData objects (though due to the behavior of MATLAB's [Value Class](#), here I mean 'update' but each method returns a new IrisData instance with the desired modifications). This release brings a few fixes and some new functionality.

- IrisData:
  - Plot method now plots devices on separate axes
  - AppendUserData method introduced, allows one to add custom data, meta, etc. to the object (see above note)
  - Concat method now allows you to seamlessly merge IrisData objects.
  - Static method subtractBaseline now returns baseline values in a cell array rather than appending them to each datum. In methods utilizing this, we append baseline values to the UserData property of the returned IrisData instance.
- Iris
  - fixed issue when loading multiple session files which originated from a single data file. Now, datums are concatenated while other salient properties are merged uniquely.
  - Some minor warning and bug fixes pertaining to compatibility with the new 2019b release.
    - On this note, *due to the different usages of `Ls()` on mac vs win, this release has modified the way folders are queried and created, it should have no appreciable effects compared to prior releases.*
  - fixed horzcat error when loading symphony .h5 files that contained nested sources.

## 2019-09-17 Release 2.0.32a

Release 2.0.30 introduced the idea of Modules, where one can extend Iris functionality in a customized manner. Just like creating a custom Analysis script, creating a module allows the user to send the currently selected data, the whole session or nothing at all to a GUI created using MATLAB's appDesigner. A functional example of how a module could work has been included. To allow for future builtin modules, the actual \*.mlapp files are copied from source folders (the builtin directory is in the Resources folder and the user defined modules directory, settable in the preferences) and stored in a package directory, `iris.modules.*`. This allows the user to make modifications to their custom modules and then use the refresh option in Iris to update the accessible modules.

- fixed improper grouping in IrisData when supplying multiple grouping vectors where some cases are empty.

## 2019-09-15 Release 2.0.31a

- minor bug fixes and typo corrections to Iris, IrisData and responseFamily module.
- fixed issue with handler subsetting multiple files. Notes indices are now properly re-allocated when using copySubs() method. (mostly for export to IrisData);

## 2019-09-13 Release 2.0.3a

Things are coming together. There is still a persistent hang that occurs seemingly without pattern. Sometimes, simply waiting a minute will be enough to resolve the issue. Other times you have to kill the app by deleting the

hidden handles then restarting MATLAB. Unfortunately, the hang never causes an error and isn't a result of a runaway loop (on the app side).

- fixed minor bugs and typos
- enabled modules and added ResponseFamilies module as a builtin module
- fixed plotting mechanism which is aware of devices with singular responses. These are now forced into "markers" mode to make them visible.
- added zero-lines to the axes for visual reference. These are currently always "on" until I make a preference toggle for it.
- fixed dataOverview inclusion switching. Now icons in the datum tree correctly indicate inclusion status.
- updated symphony reader to include more parameters from available devices. Now, you can find the "background" `'mean'` of a stimulus or the `'value'` of a passive device (i.e. not a stimulus device) in the Specifications tables/cells. These can be located in the Specs table with the [MATLAB compatible variable name](#) using the following patterns (though depending on how you setup your Symphony Rigs, this may be different): `'stimulus_<DeviceName>_mean'` (usually for the builtin pulse generator) or `'device_<DeviceName>_value'`, which should be the same no matter how your setup is configured.

## 2019-09-05 Release 2.0.29a

- Updated Iris stats switch to use manual aggregation loop, which enhanced plotting speed a little. *Still need to optimize the plotting mechanism, I don't like how slow it is.*
- fixed timestamp accuracy when reading files in timezones other than pacific time. This applies to notes and datum times in Symphony h5 files.
- updated IrisData `Aggregate` method to check if builtin function was supplied as character array / string and then explicitly defines the function to operate only on columns. This prevents the need to pull out singleton datums and also allows functions like `var` to be computed for those singletons (note that `var(<scalar>) == 0`). But this removes the need to supply an anonymous function to manage singletons, e.g. `@(x)mean(x,1)` or `@(x)max(x,[],1)`.
- fixed other minor bugs and typos

## 2019-09-04 Release 2.0.24a

- updated AverageEpochs builtin analysis tool to include the filter example.
- modified how IrisData.Aggregate performs statistic. Now the entered statistic function MUST operate down columns (this has always been true when using `grpstats` but now you have to be more careful). E.g. if you want to compute averages across datums, supplying: `'statistic', 'nanmean'` to `IrisData.Aggregate()` will produce desired effect. Should you provide a custom function, create it to operate as though each datum is a row vector and the aggregate group will have  $n_{\text{groups}}$  rows and  $m_{\text{samples}}$  columns, e.g. `@(x)mean(x,1)` is appropriate to calculate the point-by-point mean of a group of datums. In the future, this may change to accommodate summarizing statistics.

## 2019-09-03 Release 2.0.22a

- IrisData:

- fixed issue where dropping the first datum in a family would lead to disorganized grouping. Now we can add an boolean **inclusions** vector to **IrisData.determineGroups()** in order to correctly determine group order

## 2019-09-02 Release 2.0.21a

- IrisData:
  - fixed bug where dropping epochs sometimes led to error in **getDataMatrix**
- Iris:
  - added **Aggregate** method to manage in-app aggregation. *Functionality is not yet stable.*
    - When stats rocker is switched on, it is recommended to not navigate new epochs.
    - When rocker is on, changes in preferences are not redrawn sometimes. It is recommended to change preferences while rocker is OFF for most accurate results.
    - Note that the preferences checkbox baseline zeroing is only for aggregates, it does not affect the operation of the baseline rocker. When this option is checked, the baseline rocker effectively does nothing. If this checkbox is unchecked, the baseline rocker will then perform baseline zeroing for both original epochs and the aggregates.
  - modified plotting mechanism. Moved much of the data gathering to Datum, Handler and App.
- Future Bug fixes coming:
  - When modifying scaling values in "Custom" mode, values are reverted to 1 each time the scaling rocker is switched from OFF to ON. Currently, you need to switch the rocker to ON, then modify your custom scaling options.

## 2019-08-29 Release 2.0.18a

- IrisData:
  - fixed bug in Aggregate that left temporarily padded NaNs. Datums are now subset to shortest datum WITHIN aggregate group (i.e. if 100 sample vector is averaged with 200 sample vector, rather than have in 100 points and 100 nans, it will just be 100pts).
  - Method chaining now works well e.g, **IrisData.Filter().Aggregate()**
- Iris:
  - enhanced file info tree naming and info table column widths
  - enhanced protocols table view
  - enhanced Datum getProp method (to match IrisData Specs get method)
  - enhanced symphony reader to include source labels for each Epoch.
    - This enabled grouping by '**Sources**'
  - enhanced ReadToSession with verbose flag

## 2019-08-27 Release 2.0.10a

- IrisData:
  - added response and rig device configurations to the **Specs** table
  - fixed indexing bug when original indices overlapped during construction
- fixed an issue with parallel pool creation in ReadToSession utility

## 2019-08-26 Release 2.0.8a

- fixed typing error resulting from IrisData Aggregation
- fixed bug where saving iris session from IrisData caused empty inclusion status
- fixed bug with plotting mechanism to allow for NaNs
- added cell parsing support to unknowncell2str and unknown2cellstr helpers
- modified output of IrisData save session to match the Iris data handler `saveobj`

## 2019-08-25 Release 2.0.7a

- fixed typos
- added "Import Analysis" to analysis menu
- modified IrisData:
  - changed property `Filters` to `Specs`
    - to avoid typo issues with added method.
  - added `Filter` method to apply digital filtering to datums
  - Fixed bug associated with `Aggregate` which would fail to parse filenames correctly when more than 1 file was present.
  - Added public method `UpdateData` to easily create a new IrisData object from a modified data struct as retrieved from `IrisData.copyData()` method.
  - fixed some minor bugs with the `getOriginalIndex` method
- Attempted to add functionSignatures.json though it doesn't seem to work.

## 2019-08-22 Release 2.0.2a

- bandaid patch of `drawnow()` issue, not fully fixed. maybe matlab bug
- added save iris session from IrisData classdef
- added subset method for splitting irisdata
- fixed some typos.

## 2019-08-19 Release 2.0.1a

### First Release

*Expect Bugs!*

## < 2019 PreRelease

Development began after finalizing version 1, originally called: MetaVision.