

iVision-Mac 4.5 Release Notes

What's New in iVision-Mac 4.5.0

iVision 4.5 is the first Intel-native version of iVision. It includes the following changes and features:

Faster Processing on Intel

Here are the times for a simple image processing script that reads 20 indexed files, changes them, and writes them back to a different folder:

	G5 (2.3GHz)	MacPro (3GHz)
iVision 4.0	19 seconds	21
iVision 4.5	18	10

Note that camera frame rates are not noticeably improved. The faster processing speeds may speed up scripts that do other processing during the capture loop, though.

PCIe support on Intel

Cameras which use PCI-Express (PCIe) boards are now supported on Intel-based machines.

Scripts and non-device preferences are not convertible.

Sorry about that.

File/Image names: 255 Unicode characters

Unicode allows the use of "international" (non-Roman alphabet) characters (e.g. Japanese).

There are some items where Unicode is still not fully supported including:

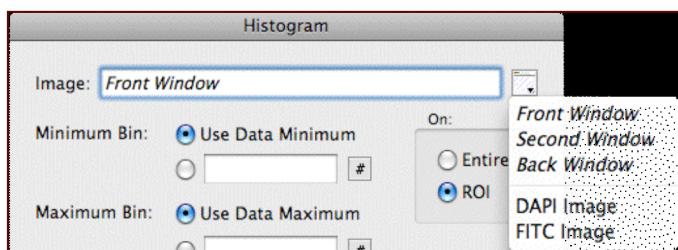
- Row and Column Labels

- Drawing Text

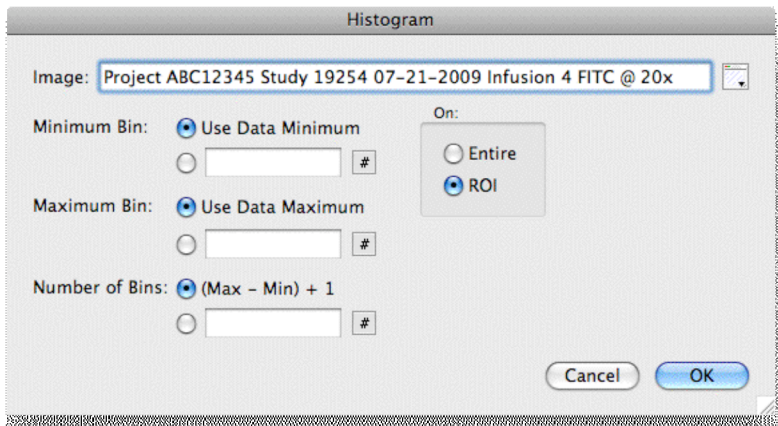
To fully support these a new file format will have to be developed, because the older IPLab format does not store these items in Unicode.

New dialog window name field & resizable dialogs.

The window name field in dialogs is now a single edit field with a menu to the right of it. The menu includes "relative" window names that let you refer to windows based on their position, for use in scripts. The relative names are drawn in italics. To put in a specific name (e.g. "DAPI Image"), pick from the menu, or just type into the field and any italic name will go away.



Since window names can potentially be very long now, dialogs that contain window name fields can be resized horizontally to see the full name.



Commands

More commands allow you to specify the image to use.

All commands that reference image windows now include a window name field, rather than only working with the front image. This should make scripting easier, since you don't have to worry as much about which image is at the front.

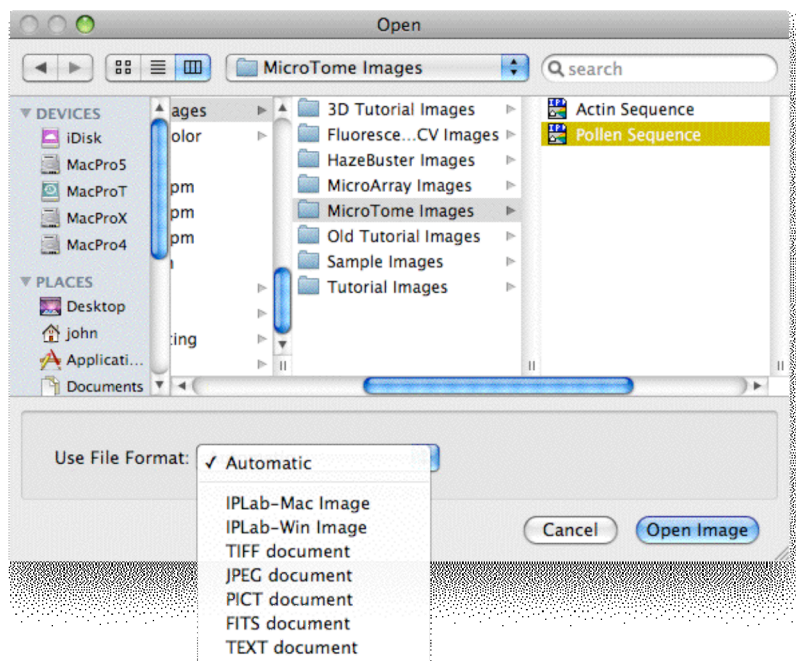
Open As replaced by enhanced Open command.

The "Open As" command let you tell iVision what format a file was (e.g. TIFF or PICT), so if the file had the wrong extension or file type associated with it you could still open the file. Since it was not used much and seemed to confuse people more than it helped them, the "Open As" has been removed. Its functionality has been included in the "Open" command.

To keep "Open" simple to use, the default file format is set to "Automatic". This will figure out the format of the file for you and open the image if it is in any format that iVision knows.

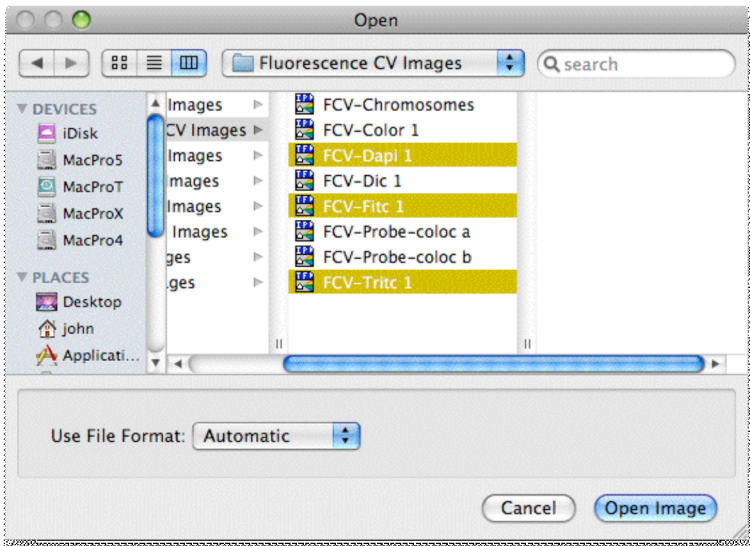
The "Automatic" format should be used in almost all cases.

The best reason to use a specific format is when using a script to open a large number of images that are all the same format but do not have an extension or other file type (e.g. from an old OS 9 CD). In that case telling iVision the format will speed up the file reading.



Open Multiple Files at once, interactively.

The Open command allows you to select more than one file at a time interactively. This is not scriptable.



File Sets are just Indexed and Folder (no more List).

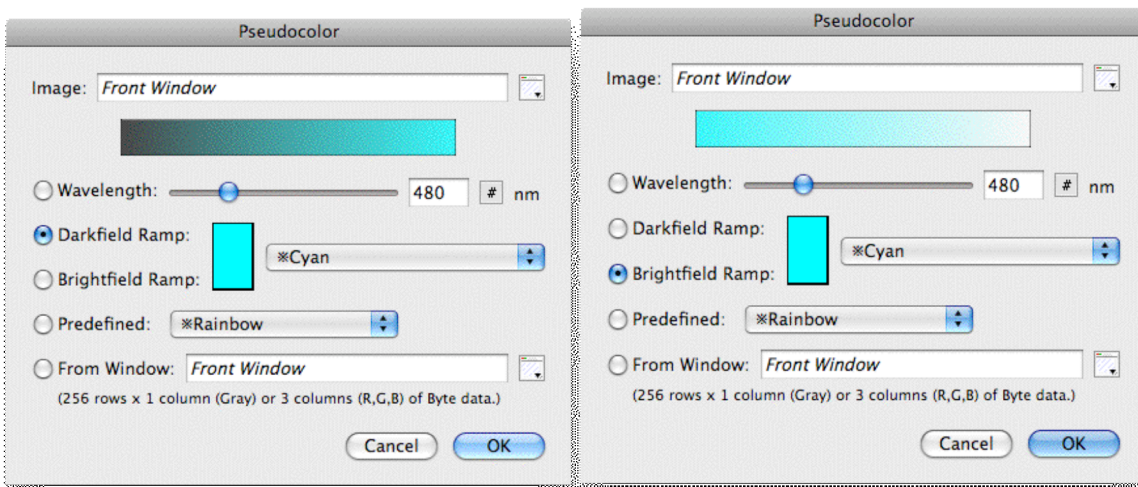
Segmentation sets gone.

You can just place the segmentation command into the script without creating a segmentation set. This required that the number of colors saved for the Color Cube feature be limited to a maximum of 20.

Pseudocolor dialog expanded:

The "Predefined" menu is reduced to the "complex" color ramps, like Rainbow, Classify, and System, as well as the commonly used "Monochrome" and "Reverse Mono".

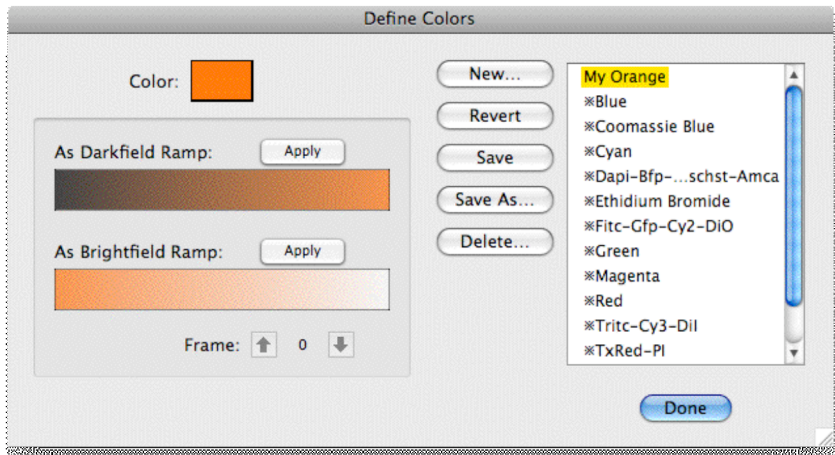
There are also two new options to create simple color ramps:



The menu to the right of the ramp choices lists "single color" color definitions (e.g. Red, Green, Blue, Cyan, etc.). You can add to this list with the new "Define Colors" command.

Edit CLUT replaced with Define Colors:

The Define Colors command lets you specify a single color and give it a name:



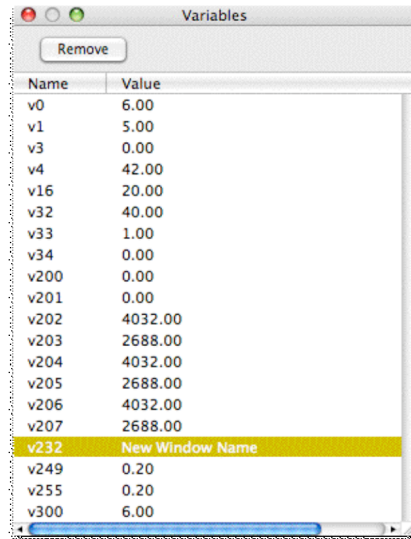
The dialog show what a darkfield and brightfield ramp made from this color will look like. If there is an image open when this dialog is open, you can apply either of the ramps to the image to see how they look. You can also step through the frames of the image to preview what the change will look like.

The colors defined in this command are also available in FCV when assigning colors to images in the Color Join command.

Changes to Variables

For the most part variables still work the same as before. You still use a number with the "Var" button in a dialog to specify a variable.

But the Variables window has changed from an image window to a palette:



The screenshot shows a window titled "Variables" with a "Remove" button at the top. Below the button is a table with two columns: "Name" and "Value". The table contains a list of variables, each with a name starting with "v" followed by a number and a numerical value. The variable "v232" is highlighted in yellow and has the text "New Window Name" in its value field.

Name	Value
v0	6.00
v1	5.00
v3	0.00
v4	42.00
v16	20.00
v32	40.00
v33	1.00
v34	0.00
v200	0.00
v201	0.00
v202	4032.00
v203	2688.00
v204	4032.00
v205	2688.00
v206	4032.00
v207	2688.00
v232	New Window Name
v249	0.20
v255	0.20
v300	6.00

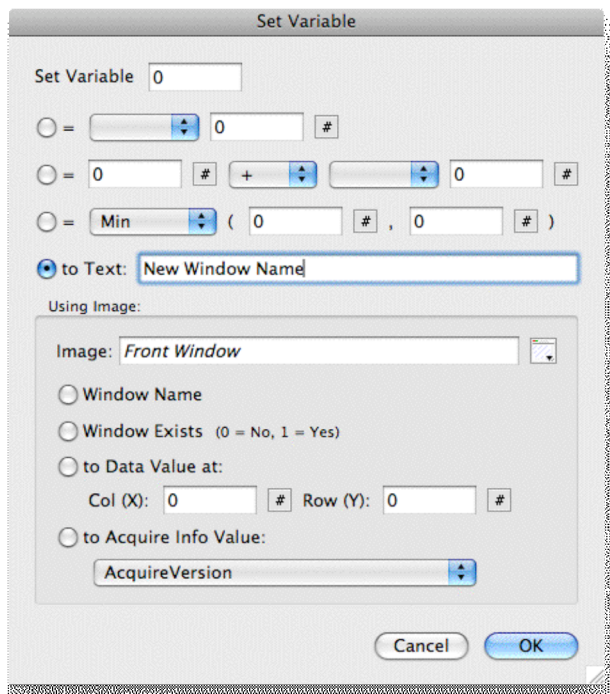
In this window the name of the variable is preceded by a lower case "v".

The Variable window only shows those variables that have been used at some point (and have not been removed). You can change the value of a variable in this window by clicking on the Value column and typing a new value. In addition, variables can contain text as well as numbers.

Updated Set Variables dialog.

The Set Variables dialog includes several ways to place text into variables:

- A new "Text" option to set any arbitrary text string.
- A "Window Name" option to place the name of an existing window into a variable.
- The "Acquire Info" menu now includes non-numeric fields such as Filter Names and dates, which will be placed into the variable as a text string.



Updated Enter Variables dialog.

The Enter Variables dialog has been updated to add some formatting features, and to allow text to be entered into variables:

Usage:	Prompt:	Variable to use:	Display Constant:
Text	Name of Experiment ?	0	<input type="checkbox"/> 0
Line	-----	1	<input type="checkbox"/> 0
Title	Camera Settings	2	<input type="checkbox"/> 0
Number	Exposure?	10	<input checked="" type="checkbox"/> 0.200
Number	Binning?	11	<input checked="" type="checkbox"/> 2
Filter Name	CRI Filter	5	<input type="checkbox"/> 0
Unused	-----	6	<input type="checkbox"/> 0
Number	-----	7	<input type="checkbox"/> 0
Text	-----	8	<input type="checkbox"/> 0
Checkbox	-----	9	<input type="checkbox"/> 0

When the script is run, the result looks like this:

Usage:	Prompt:	Variable to use:	Display Constant:
Text	Name of Experiment ?	0	<input type="checkbox"/> 0
Line	-----	1	<input type="checkbox"/> 0
Title	Camera Settings	2	<input type="checkbox"/> 0
Number	Exposure?	10	<input type="checkbox"/> 0
Number	Binning?	11	<input type="checkbox"/> 2
Filter Name	CRI Filter	5	<input type="checkbox"/> 0
Unused	-----	6	<input type="checkbox"/> 0
Number	-----	7	<input type="checkbox"/> 0
Text	-----	8	<input type="checkbox"/> 0
Checkbox	-----	9	<input type="checkbox"/> 0

Using Variables in Text Fields

To use any variable (numeric or text) in a text field, you use a special formatting rule that lets you insert the value of the variable into the text.

Place the name of the variable inside braces: {v123}.

You can optionally use one of two formatting options for numbers: zero fill, or decimal places.

For zero fill you can specify a total length of 2 to 6 digits.

For decimal places you can specify 0 through 6 digits after the decimal place.

You specify the option by following the variable name with a comma and then a 'z' or a 'd', followed by the number.

Example:

if v42 = 12.547000000

[v42}	=> 12	(truncate)
[v42,z2}	=> 12	(truncate)
[v42,z3}	=> 012	(truncate)
[v42,z4}	=> 0012	(truncate)
[v42,z5}	=> 00012	(truncate)
[v42,z6}	=> 000012	(truncate)
[v42,d0}	=> 13	(round)
[v42,d1}	=> 12.5	(round)
[v42,d2}	=> 12.55	(round)
[v42,d3}	=> 12.547	(round)
[v42,d4}	=> 12.5470	(round)
[v42,d5}	=> 12.54700	(round)
[v42,d6}	=> 12.547000	(round)

You can put more than one variable substitution in a text field.

For example, if v10 = 4, and v12 = 32.4112:

"Experiment {v10,z3} temperature: {v12,d1} F" would produce:

"Experiment 004 temperature: 32.4 F".

Variable substitution is available in:

Window Names

Draw Text: text

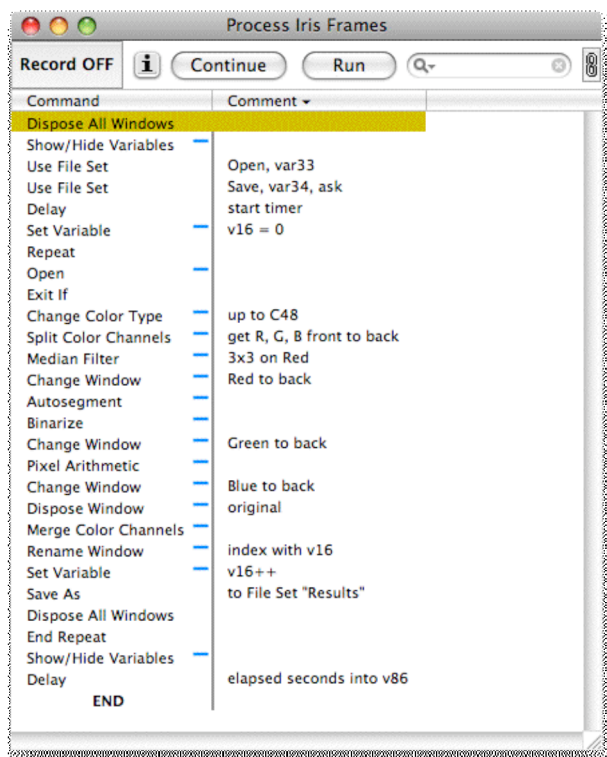
Enter Variables: title, description, prompt(s) [for Enter Var and Title uses]

Script Query: title, message, button titles

Script Alert: title, message

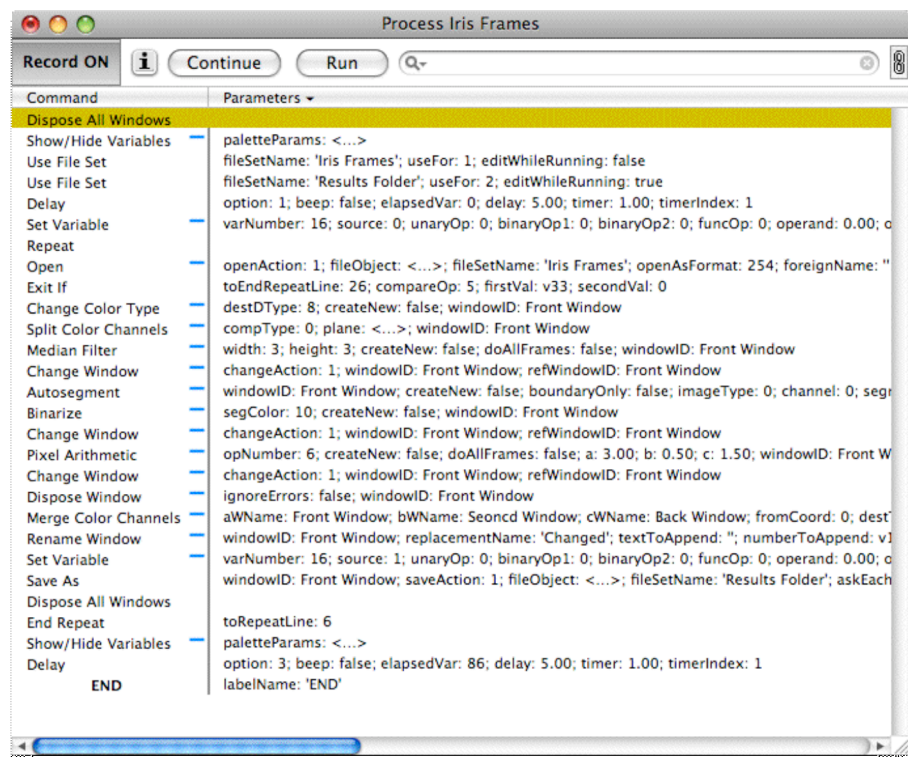
Scripts:

The Script window has been updated. It is resizable horizontally.



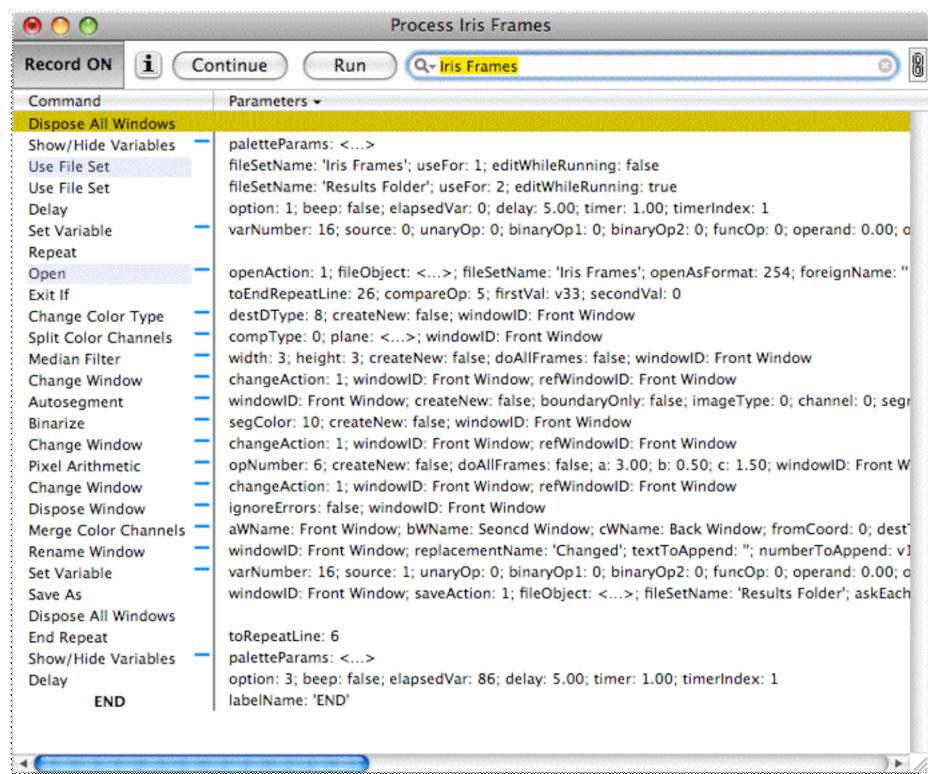
The END symbol now acts like a label,; you can jump to END without adding a label just above it.

The Comment column can be toggled to show the parameters in the command:

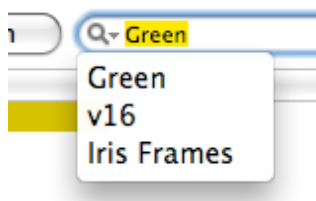


At the moment this is more confusing than helpful because the description shows every possible field even if they aren't in use, and in a hard to read format. This will be improved in the future.

The search field at the top of the window will highlight any commands that contain the given text:



In this example the File Set named "Iris Frames" is used by two commands "Use File Set" and "Open". The command name, the user comment, and the parameters are all searched for the given text. The search field remembers the last 10 searches:



The script commands palette includes two new commands: **Enable** and **Disable**.



These commands allow you to disable commands in a script, which causes them to be skipped over, as if they weren't in the script at all. Disabled commands are displayed in red:

Median Filter		3x3 on Red
Change Window	—	Red to back
Autosegment	—	
Binarize	—	
Change Window	—	Green to back
Pixel Arithmetic	—	

Scripts remember different size/placement than interactive dialogs.

If you are scripting a command where you want the dialog to be opened while the script is running (with the "+"), the size and position of the dialog used when the script is run will be saved independently from the size and position used when not running a script. So you can reliably place the dialog at a specific place on screen.

Miscellaneous:

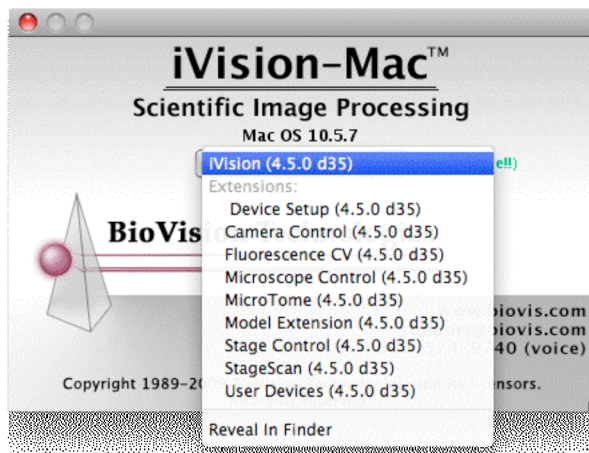
New File Extensions for iVision files:

iVision Script	.ivsc
iVision Preference	.ivpr
iVision Extension	.ivex
iVision Device Module	.ivdm

About Box:

The About Box contains a popup menu which shows the name and version of all Extensions in use. This does not include device modules, which are listed in the Device Setup dialog.

At the bottom of the popup menu is "**Reveal in Finder**", which will open a Finder window for the folder which contains the currently running iVision. This helps verify which copy of iVision is running (if there is more than one on the machine),



The Device Setup dialog has an **Open Folder** button to show the current Device Modules folder.

