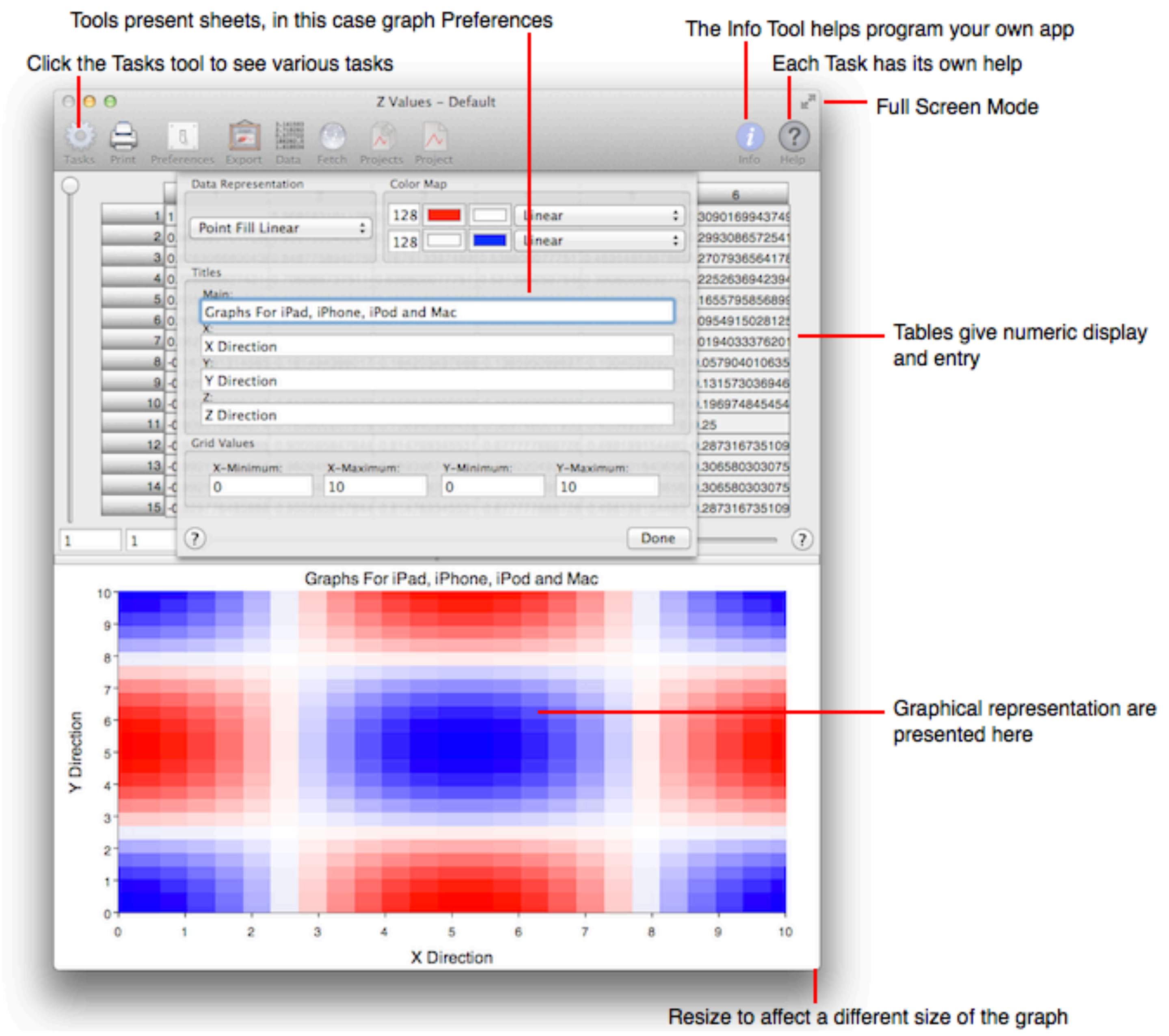
Graph And Vvidget User Manual

Also applicable to Chart Tasks in Vvidget Builder
For Mac

Revision 1.2.13 (10.8.3)
— August 26, 2012 —
By VVI ®
888-VVI-PLOT
www.vvi.com

Graph is an application with one design objective: *"You give it data and it gives you a graph"*. The figure below diagrams the essential components of Graph and is explained in the section Main Idea.


Tools present sheets, in this case graph Preferences
Click the Tasks tool to see various tasks
The Info Tool helps program your own app
Each Task has its own help
Full Screen Mode
Tables give numeric display and entry
Graphical representation are presented here
Resize to affect a different size of the graph
Z Values – Default
Tasks
Print
Preferences
Export
Data
Fetch
Projects
Project
Info
Help
Done
Graphs For iPad, iPhone, iPod and Mac
Y Direction
X Direction

Data Representation: Point Fill Linear

Color Map: 128 🟥 Linear; 128 🟦 Linear

Titles
Main: Graphs For iPad, iPhone, iPod and Mac
X: X Direction
Y: Y Direction
Z: Z Direction

Grid Values
X-Minimum: 0 X-Maximum: 10 Y-Minimum: 0 Y-Maximum: 10

	6
1	3090169943749
2	2993086572541
3	2707936564178
4	2252636942394
5	1655795856899
6	0954915028125
7	0194033376201
8	.057904010635
9	.131573036946
10	.196974845454
11	.25
12	.287316735109
13	.306580303075
14	.306580303075
15	.287316735109

Once you learn how Enter Data then you may wish to Print, Export to the powerful Vvidget Builder layout app or learn about key value dictionary encodings using the Info tool. You might want to manage Projects or Fetch data from other sources. Notice how you are not asked to save data or navigate to find previously entered data, that is all automatic. In fact, pretty much everything, except entering new data, is optional.

Important: If you are wondering how to make a "line graph" and other common graphs then you will need to first read the Main Idea section. That is because there is one simple idea you need to understand first. Once you understand a few basic concepts then you shall be on your way.

Graph > Overview

The following sections describe Graph features.

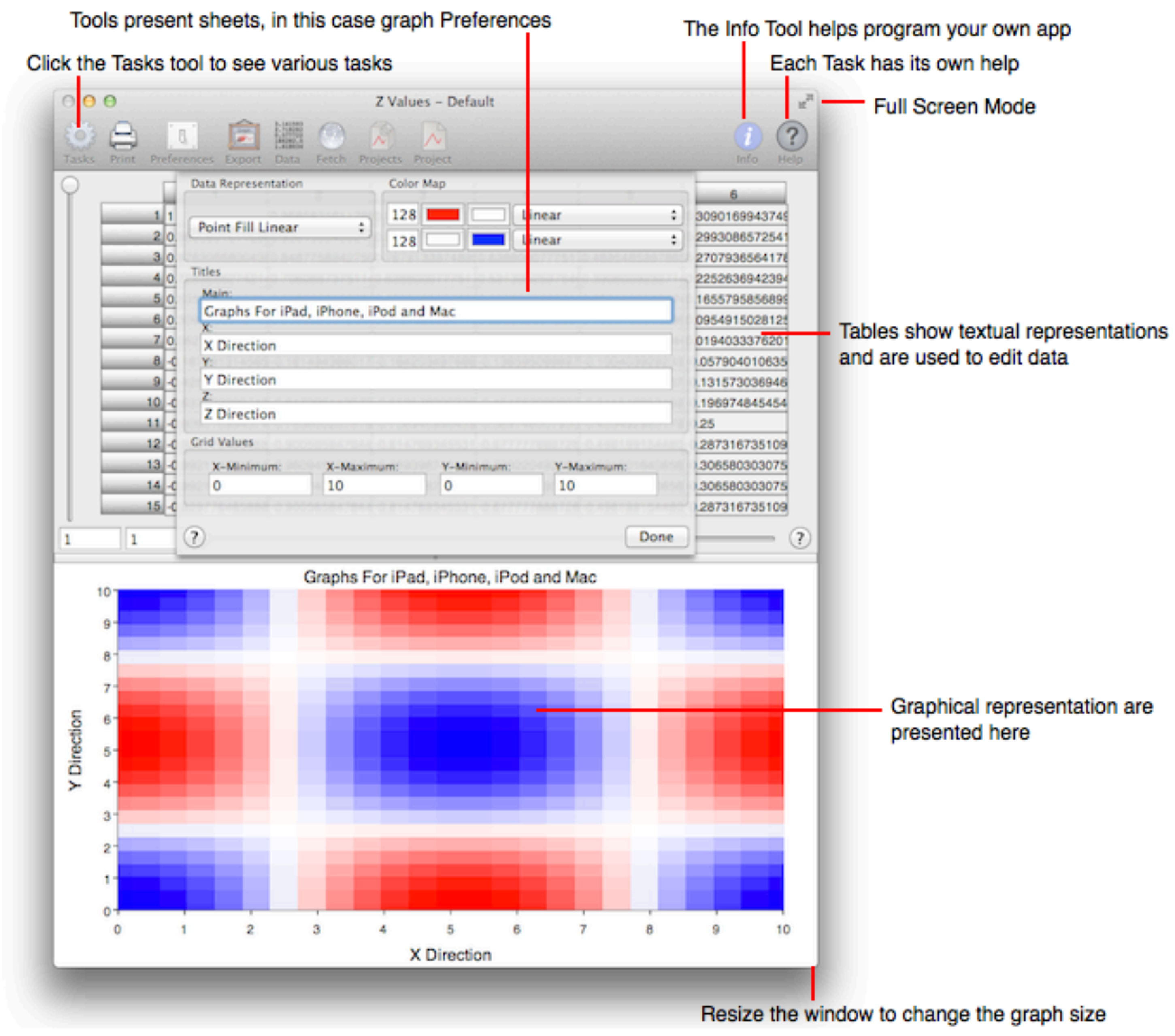
Overview	Description
Main Idea	Hits upon the main idea of Graph.
Tables	Explains how to enter data using the table interface. With tables you can paste rows, columns, individual cell values and edit individual cell values. You can also view and scan numeric values. Tables are manual data entry facilities, while Fetch is more automatic.
Popover	Describes the popover.
Preferences	Describes the app-wide preferences.
Glossary	Defines some basic terminology.

Graph > Overview > Main Idea

The figure below diagrams Graph's interface. It is a single window with access to various tools, in a toolbar, at the top of the window. The main portion of the window is comprised of a table, which shows the numeric representation, and the graph, which shows the graphical representation. You can hide the toolbar, graph or table by using the normal toolbar hide button (upper right of window) or the split view slider. Thus you are able to focus on the major component you desire, and then defocus to use auxiliary facilities. The figure below shows the Preference tool in a sheet. All tools are implemented as sheets because they are thought of as momentary access to auxiliary features that are not part of the main purpose of Graph. Tools are described in the Tools section, graphs are described in the Tasks section and tables are described in the Tables section.

Notice, for example, that you will not be able to find a direct link to a "line graph". That is because tasks are oriented to working with types of data, not types of representations. A "line graph" plots 2D Points so for that see the Set Of 2D Points task and choose the "line graph" from its Preferences tool. Likewise for other representations. See the Tasks sections for further relationships between data type (tasks) and representations.

Probably the best starting point for learning how to use Graph is to actually use graph. However, if you wish to read about using graph then perhaps a good starting point is the tutorial Enter Data.


Tools present sheets, in this case graph Preferences
Click the Tasks tool to see various tasks
The Info Tool helps program your own app
Each Task has its own help
Full Screen Mode
Z Values – Default
Tasks
Print
Preferences
Export
Data
Fetch
Projects
Project
Info
Help
Tables show textual representations and are used to edit data
Graphical representation are presented here
Resize the window to change the graph size
Graphs For iPad, iPhone, iPod and Mac
Y Direction
X Direction
Done

Data Representation: Point Fill Linear

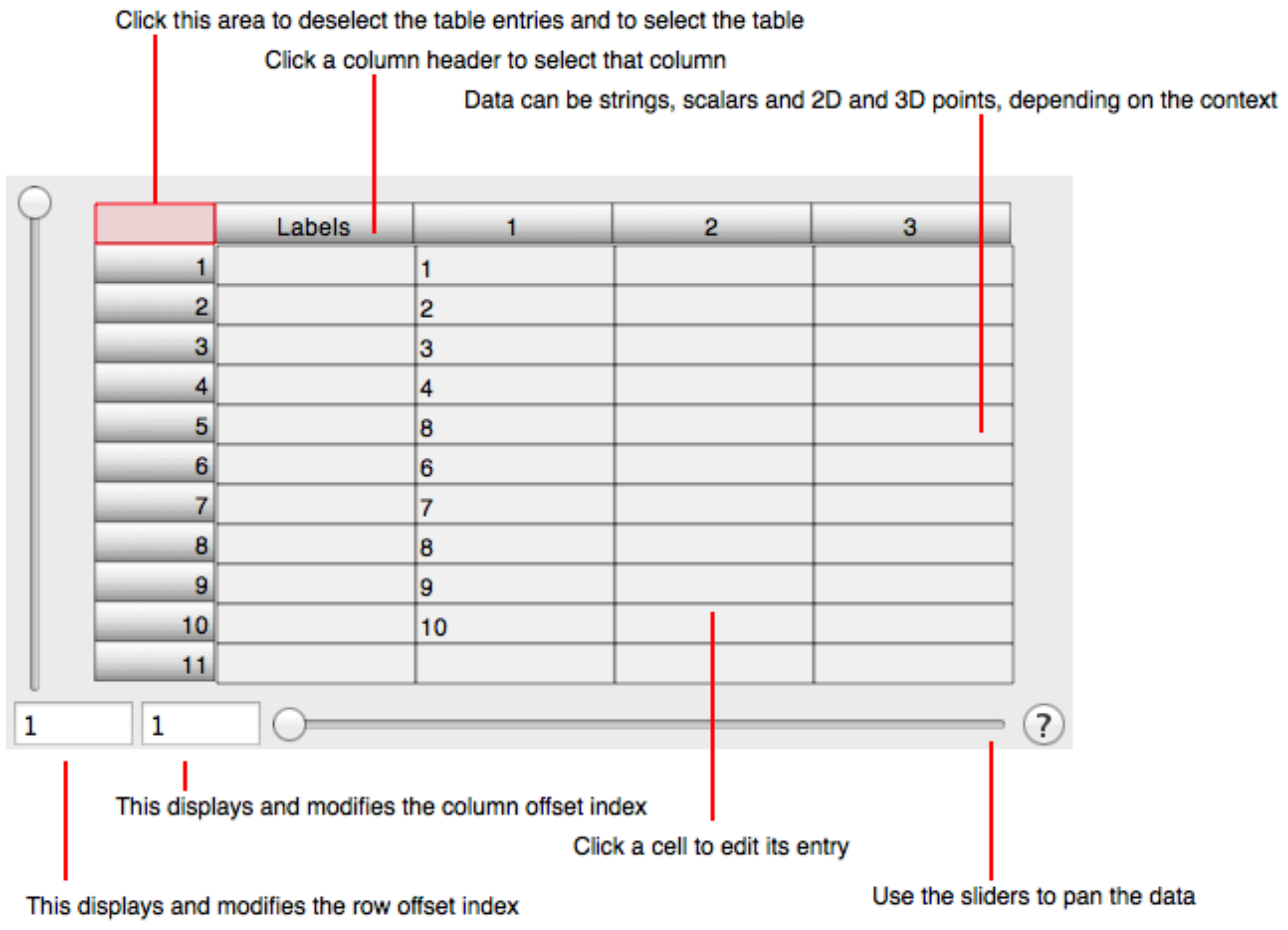
Color Map: 128 Linear; 128 Linear

Titles
Main: Graphs For iPad, iPhone, iPod and Mac
X: X Direction
Y: Y Direction
Z: Z Direction

Grid Values
X-Minimum: 0
X-Maximum: 10
Y-Minimum: 0
Y-Maximum: 10

Graph > Overview > Tables

The table component of a Task is used to enter and display the decimal and character representation of data. The cells of the table can represent a label (text), scalar (a single decimal number) or point (2 or 3 dimensional point) entries. A typical table is diagrammed below.



Click this area to deselect the table entries and to select the table

Click a column header to select that column

Data can be strings, scalars and 2D and 3D points, depending on the context

	Labels	1	2	3
1		1		
2		2		
3		3		
4		4		
5		8		
6		6		
7		7		
8		8		
9		9		
10		10		
11				

1 1

This displays and modifies the column offset index

Click a cell to edit its entry

This displays and modifies the row offset index

Use the sliders to pan the data

The following is an itemization of table features.

Columns

A column of a table is the vertical collection of cells of the table. Clicking on the column header (the grey portion at the top of each column) selects cells in that column. Alt-click edits the description for the column which is useful for setting labels for a legend. See Legend for additional information.

Cell Type

Each cell in a table is comprised of indivisible pieces of data. There are two types of "indivisible pieces of data" as defined here:

- Atomic: For a bar graph the atomic element is a single number. For a line graph the atomic element is two numbers representing a x and y value at a 2D point. Thus, for a 2D point, each cell shows, and can can be used to edit, two numbers. The notion of an atomic cell is consistent with the idea that there is no way to specify a 2D point without specifying two numbers so the two numbers are considered indivisible (atomic) and must be located within one cell.
- Component: It is often the case that numbers come from other sources that do not represent a 2D point and those numbers need to be inserted into the table. Those numbers are often a sequence of scalars that represent only one dimension of a sequence of 2D points. As such, they incompletely specify an atomic element and are not usable by themselves. They need to be combined with another sequence of scalars to form a sequence of 2D points (and hence a curve). The component cell type is used to facilitate this need. In component mode, the table columns are arranged by interleaving dimension. Specifically, the x-dimension is specified in a column and then the y-dimension is specified in the adjacent column.

The type of cell used is specified in the Task Edit tool. By switching between types, dimensions can be resorted. For example, for 2D points the data is serialized in the format: x1 y1 ... xN yN. That is the atomic representation. By using the component representation the data is serialized in the format: x1 ... xN y1 ... yN. It is often the case that both serializations are useful and as a result, tables have two cell types.

The above discussion is in terms of 2D points but is equally applicable to 3D points in the obvious way. Notice that bar data is scalar and scalars only have one dimension so the atomic and component cell type are identical.

It is hard to overemphasize the fact that the atomic cell is the more consistent representation, but that the component cell is the more conventional representation. When you use the component cell type you will notice this fact. For example, deleting an interval of y-values will not remove the y-values but rather zero them. That is because those y-values are intrinsically (and implicitly) associated with the x-values. To remove the y-values you must also select the respective interval of x-values and then delete both of them simultaneously. Only then will the points be removed. By using the atomic cell, this is a non-issue because the x and y values are in the same cell and deleting a cell deletes both the x and y value pair of a point. This same issue applies to the many other operations applied to tables. By utilizing a sequence of operations that combine atomic and component representations the resulting composite operation can be fairly complex and useful. For example, in component element type paste in a column of x-values and then a column of y-values, switch to atomic element type and then copy the column of 2D points. In that way you have just transposed the matrix of data a.k.a.: changed serialization from x1 ... xN y1 ... yN to x1 y1 ... xN yN.

Component Selection

- Click on a column header to select a column. Shift-click twice to select an interval of columns. A single click on a column selects only that column for use and hence resulting operations are bounded to that column. A shift-click selects intervals of columns and hence resulting operations span columns.
- Click on a row header to select a row. Shift-click twice to select an interval of rows. A single click on a row selects only that row for use and hence resulting operations are bounded to that row. A shift-click selects intervals of rows and hence resulting operations span rows.
- Click to the left of the column headers, right above the row headers, to focus on the table. Clicking on that area deselects any cells of the table. Then select all cells in the table by typing command-a.
- Click-drag on the cells to select a group of cells and to scroll the table.
- Click-hold to select a single cell and also to bring up a menu of options. Click again to dismiss the options menu while keeping the cell selected.

Operations On Selection

Once a component is selected then choose one of the following:

- Delete (the delete key or del numeric keypad key) to delete the selected component.
- Copy (command-c, Copy main menu item, or click-hold menu) to copy the selected component.
- Paste (command-v, Paste main menu item, or click-hold menu) to paste the selected component.
- Cut (command-x, Cut main menu item, or click-hold menu) to cut the selected component.
- Click-hold on a selection to bring up the click-hold menu of options.

It should be noted that if you select only one column or row of data (with a click) then that column or row is extended with a subsequent paste. However, if you shift-click an interval of columns or rows then the subsequent paste wraps over the rectangular cell selection.

Cell Edit

When you click on a cell and then release, without dragging or holding, then the cell editor is brought forward. Once forward the following applies.

- Edit the cell text using the normal keyboard edit facilities.
- Click Return to enter the data, tab to enter and proceed to the next row in the column or shift-tab to enter and proceed to the previous row in the column.
- Use the arrow keys to enter the text and proceed to an adjacent cell in the direction of the arrow.
- Click the ESC or command-. to cancel cell editing and dismiss the cell editor.
- Use the on-board buttons to enter the data and dismiss, revert or proceed to adjacent cells.

Pop Over

- Move the cursor over a cell to see a popup. The table popup shows the cell indices and value. In the case of a blank cell it shows the cell data entry instruction.

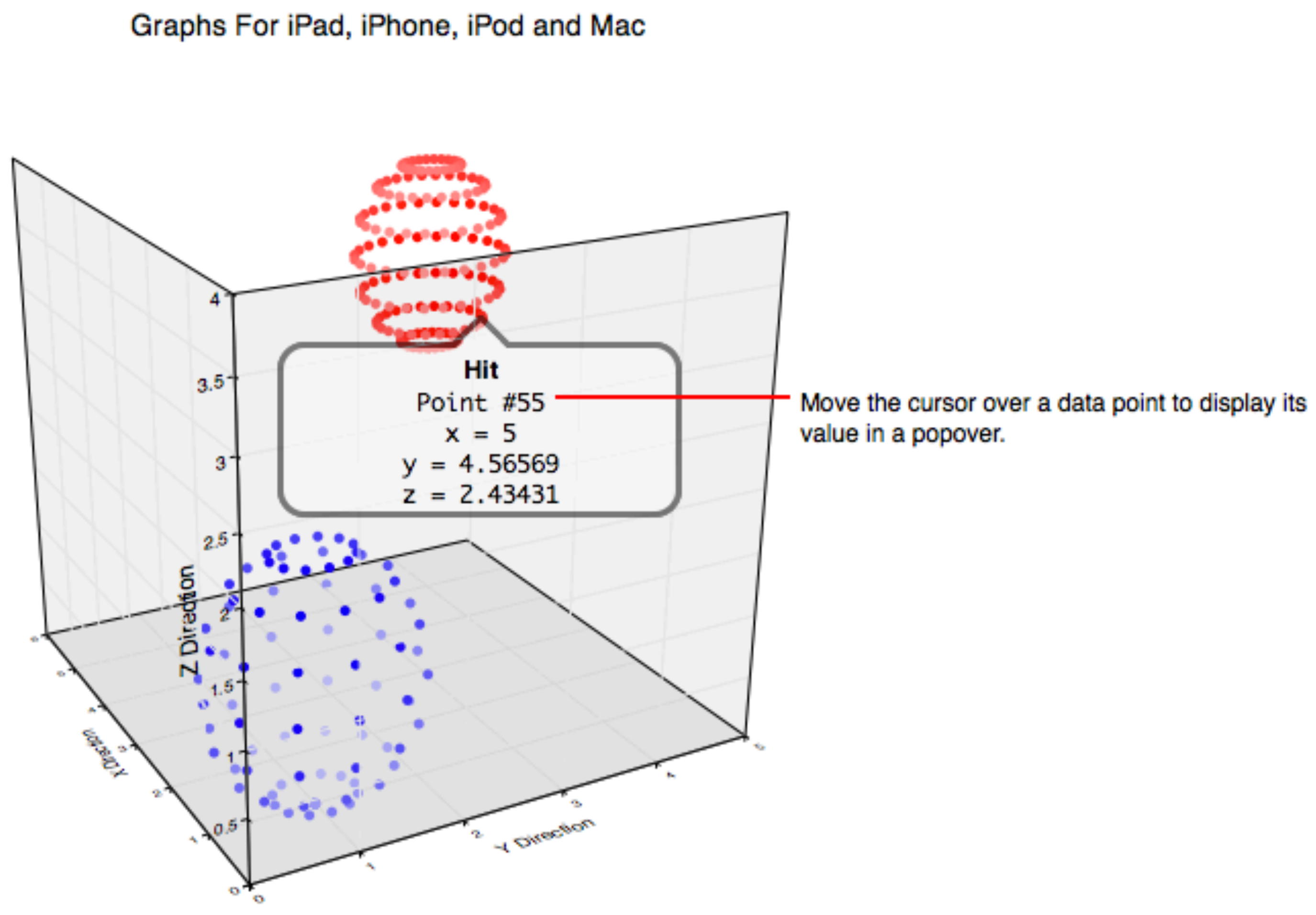
Data Format

The data format for each import operation is define in the respective task section. Generally:

- For scalar data the column import (paste) format is a list of numbers, for point values it is a list of point components (for example, for 2D points: x1 y1 x2 y2 ... xN yN) where numbers are separated by a blank.
- When importing into numeric columns the format can be much more liberal. Any non-numeric ASCII delimiter can be used such as comma, semicolon, space, tab, Return, etc.
- When importing into a label row the delimiter is a Return character.
- When importing into a 2D point column or cell the format can be a x y numeric pair or a date number pair. The date is formatted as: MM/DD/YY HH:MM:SS.fraction and a numeric number representing the y-value follows. The date input is most appropriate for a date graph (see Set Of 2D Points).
- If you paste to the entire table then the table import sheet comes forward because pasted data is formatted without explicit delimiters and you need to supply additional information as follows.
- Choosing the double-return delimited format defines column ends as two consecutive return characters in the string serialization of the data.
- Under some circumstances, you can explicitly define the table dimensions and other data attributes.
- If the data dimensions is symmetric then you can transpose the data as needed. Non-symmetrical data can not transpose by inherent limitation.

To learn more about entering data see the The Enter Data tutorial section. Tables can be an intensive manual data entry operation. The Fetch section describes an automated data entry facility and the Programming section gives reference to the programming facilities.

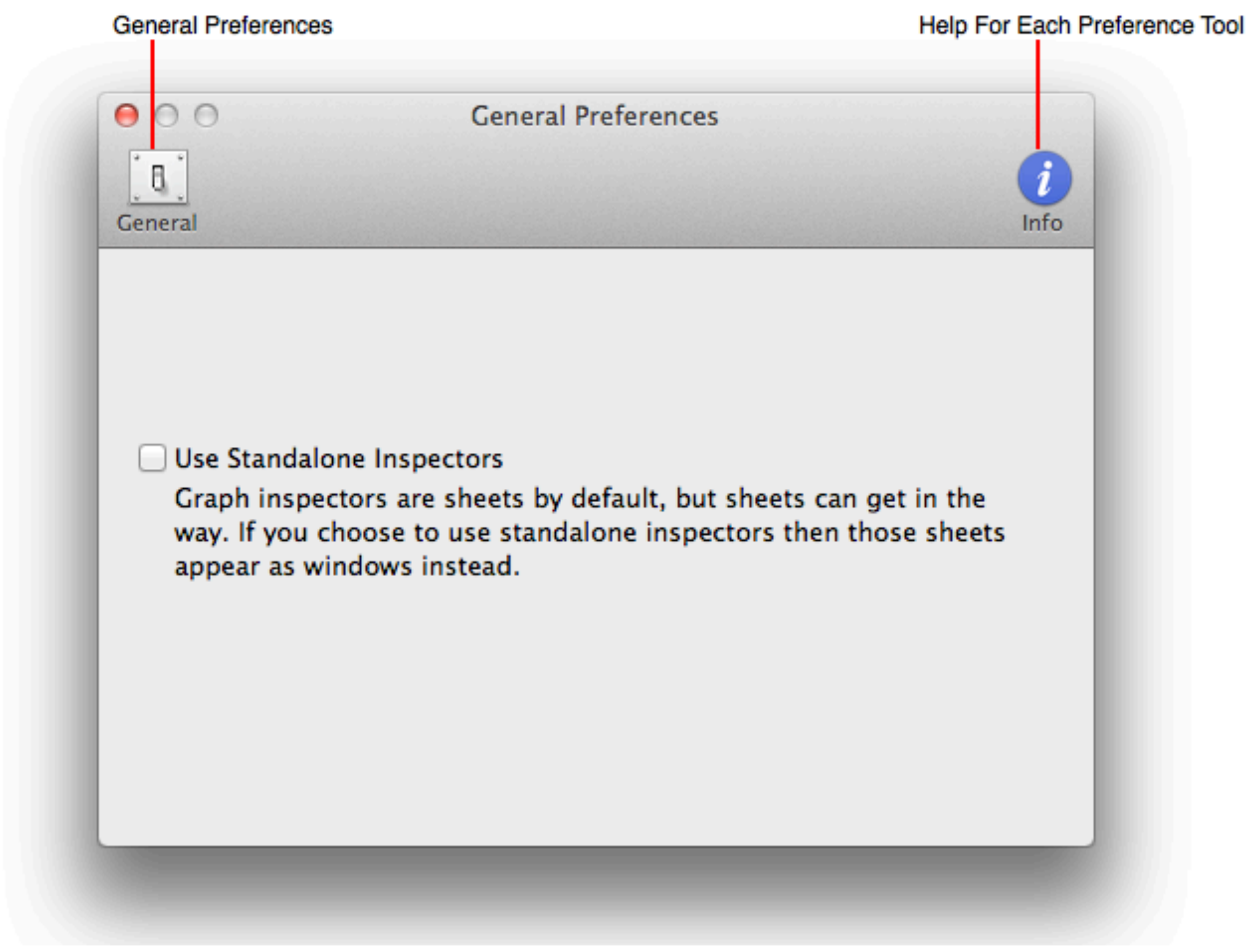
The Popover window is shown below.



The popover window appears when you move the cursor over a data component which is one of a point, line or map component or some such thing. Each task's graphical representation can have distinct responses to the cursor but in general, the popover gives information regarding the data.

Graph > Overview > Preferences

The Graph Preference panel is shown below.



Use Standalone Inspectors

Graph inspectors are sheets by default, but sheets can get in the way. If you choose to use standalone inspectors then those sheets appear as windows instead.

It has one preference setting to make Task Graph Preference Inspectors standalone panels or sheets. Sheets are desirable to keep a clean interface, however a separate panel may be more practical. Do not confuse this app-wide preference with the task Preferences tool which is task specific.

Graph > Overview > Glossary

Below is a glossary of words used in this manual.

Terminology	Definition
Atomic	An atomic is an indivisible element of a task's data. For example, for the Set Of Scalars task an atomic is a scalar and for the Set Of 2D Points task an atomic is a 2D Point (x and y value pair). An atomic is also referred to as an element. See also: Component, Element.
Column	A Column refers to a column of a table (in the vertical direction) or a bar of a bar chart oriented in the y-direction. This dual meaning leads to a bit of confusing use of the word column in the Set Of Scalars task.
Component	A component is a piece of an atomic value. For example, for the Set Of Scalars task a component is a scalar (identical to an atomic) and for the Set Of 2D Points task a component is either the x or y value of the point. See also: Atomic, Element
Element	An element is any single indexed part of a set. For example, for the Set Of 2D Points task an element is either a single 2D point, or is a list of points (since the task uses a set of set of points as data).
Scalar	A scalar is a single number such as `3.1415`. A scalar can also be called a number which is more common but in this manual the word scalar is preferred.
Task	A Task is a specific class of problems that is associated with a class of data. For example: The line graph task operates on 2D points and that is why the line graph task is called "Set Of 2D Points" because the data is what binds the different representations of that task together (line, scatter, area, etc.).
Tool	Tools operate upon tasks and are represented by sheets.
2D Point	A 2D point is a pair of scalars, that is an ordered pair. The x-dimension is first and the y-dimension is second. The Set Of 2D Points task operates on 2D points.
3D Point	A 3D point is a triplet of scalars, that is an ordered triplet. The x-dimension is first, y-dimension second and z-dimension is third. The 3D Points task operates on 3D points.

Graph > Tasks

When you operate Graph you first select an appropriate task using the Tasks tool. Once selected then you operate upon a task using all the features described in this manual including all the other Tools. The following sections describe Tasks features.

Tasks	Description
Set Of Scalars	Describes the Set Of Scalars task, basically bar, column and pie charts.
Set Of 2D Points	Describes the Set Of 2D Points task, basically line graphs.
3D Points	Describes the 3D Points task. 3D points are represented on a 3D perspective graph in dot, label and line (trajectory) form.
Z Values	Describes the Z Values task. Z Values are a 2D grid of scalars interpreted as height in another orthogonal axis (the z axis). Such a representation can also be mapped onto a 2D point fill representation.
Density	Describes the Density task. Densities are interpreted as values between 0 and 1 that represent the density of a 3D object on a regular grid.
Least Squares	Describes the Least Squares task (linear regression).
Map	Shows how to work with a Map task.
Polynomial	Describes the Polynomial task.
Error Bars	Describes a task specific to error bar plotting.

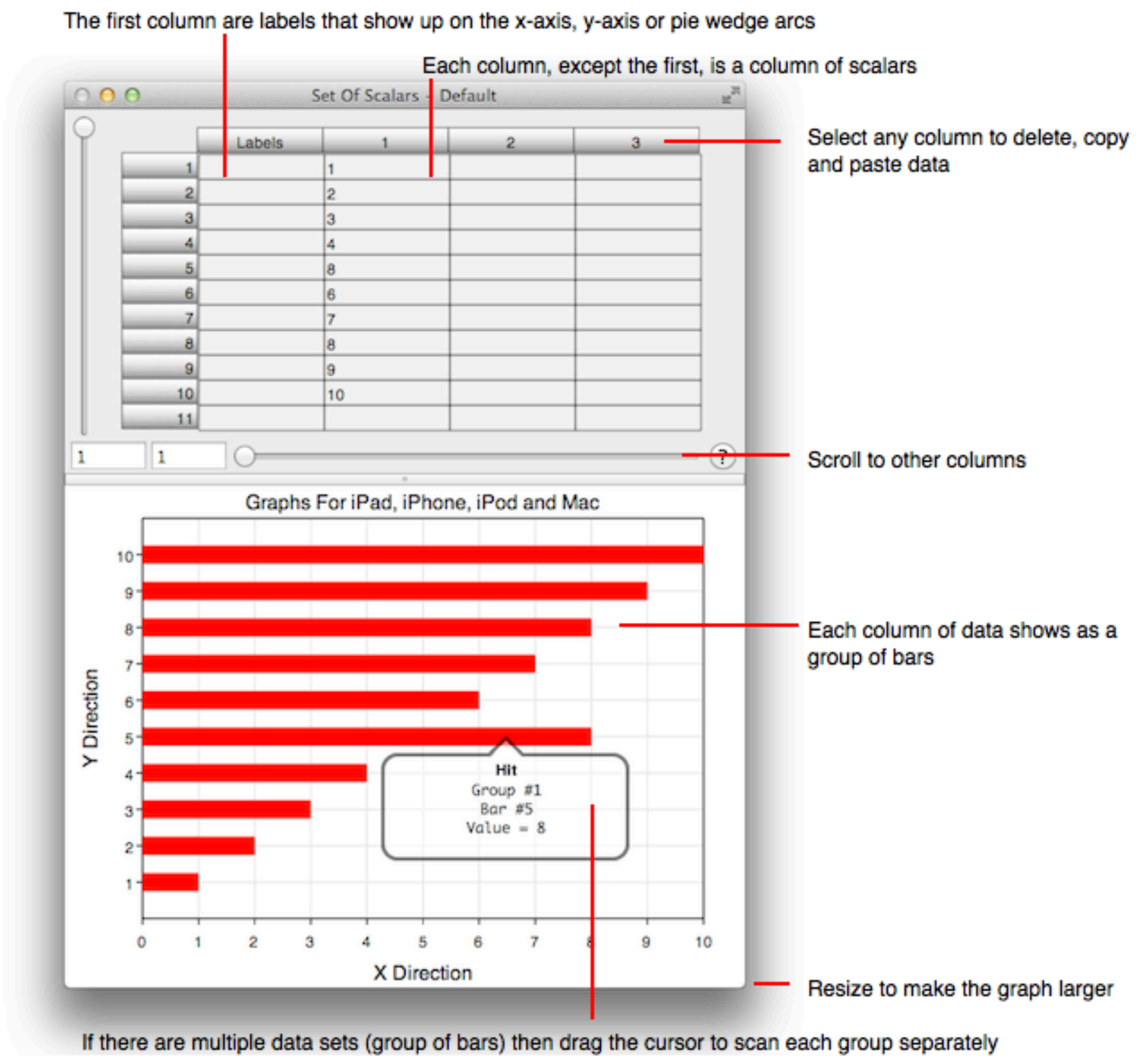
Graph > Tasks > Set Of Scalars

A scalar is a number, Scalars (plural) is multiple scalars in this case an ordered list and a set of scalars is a set of ordered lists of scalar values. Since that is a bit wordy lets just think about bars and columns of a table. A list of scalars is organized in one column of a table and bars or columns of a graph. When there are multiple columns on the table then the numbers are organized by stacking the bars or offsetting the bars on a chart.

The presentation may seem conventional and without further need of explanation, however there are some concepts that I mention here even though they are "obvious":

- Each column of a table is though of as contiguous, that is a list of numbers without interruption. However, when there are multiple columns in a tables then the bars on a chart are organized in an interleaved (non-contiguous) fashion. Essentially, the rows and columns of a table are transposed on a bar or column chart. This issue only seems uninteresting because it is conventional.
- Stacked bar representation is related to the tables through a successive sum of row values. Again, uninteresting because it is conventional however the implicit nature of this mapping and taking it for granted makes me want to point out this issue.
- There could be some interesting deviant pie chart representations for sets of scalars, however the implementation in this task simply displays only one column of data at a time. You can scan each column by dragging the cursor horizontally across the pie chart.

The figure below diagrams the Set Of Scalars task.


The first column are labels that show up on the x-axis, y-axis or pie wedge arcs
Each column, except the first, is a column of scalars
Set Of Scalars - Default
Labels
1
2
3
Select any column to delete, copy and paste data
Scroll to other columns
Graphs For iPad, iPhone, iPod and Mac
Each column of data shows as a group of bars
Y Direction
X Direction
Hit
Group #1
Bar #5
Value = 8
Resize to make the graph larger
If there are multiple data sets (group of bars) then drag the cursor to scan each group separately

	Labels	1	2	3
1		1		
2		2		
3		3		
4		4		
5		8		
6		6		
7		7		
8		8		
9		9		
10		10		
11				

While importing data using Tables or Fetch keep in mind that the format is an ordered list of numbers separated by a blank. The chart type (bar, column or pie) can be changed using the Preference tool. The Preference tool is also used to alter some other attributes. Skins are used to add distinction to the presentation and the Export tool can be used for specialized layouts and post-production editing.

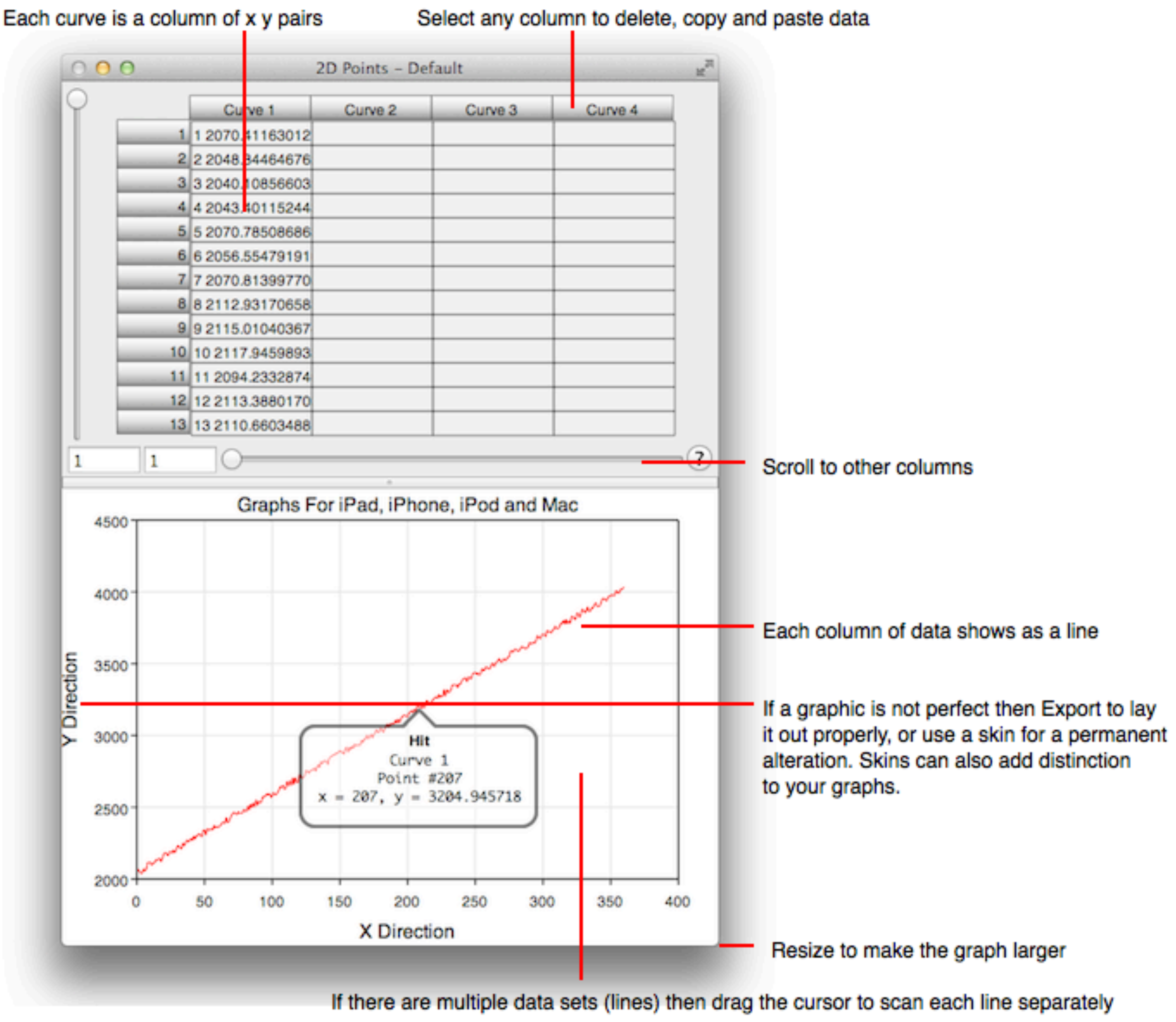
Moving the cursor over a pie wedge or bar shows the data value. Dragging the cursor over the chart scans table columns one at a time.

Graph > Tasks > Set Of 2D Points

The Set Of 2D Points task plots curves, well not really. The longer explanation is that it plots sets of ordered sequences of pairs of scalars, but saying it plots curves is a lot easier. It also makes scatter plots and area graphs. With a little Skins manipulation it can also plot trajectories. Here are a few points regarding this task:

- The purest will not banter about the term "curve" since that is a continuous concept and computers can only do things like plot discrete points and connect those points with discretized line and spline segments but not much more. A "line graph" is equally inapplicable because a line is a straight curve without bound. Even though "curve" and "line" are really not valid terms for Set Of 2D Points tasks, in practice we tend to use those terms anyways because it is conventional and to describe things accurately can take a lot of words which garble the main ideas. But, keep in mind that there is really no plotting of lines, curves, functions or anything of that nature even though we say there is. We just plot things that approximate those other things.
- Curves are formed from the column data in a table. Area graphs are curve graphs with the portion from y = 0 filled in for each curve.
- There is no enforcement of ascending x values in the table so for line graphs you need to make sure the x values are ordered by index (they ascend in value). This caveat does not apply for scatter graphs.
- The Preference tool gives access to the 21 different representations.
- For date entry see Date Graph.

The figure below diagrams the Set Of 2D Points task.


Each curve is a column of x y pairs
Select any column to delete, copy and paste data
2D Points – Default
Scroll to other columns
Graphs For iPad, iPhone, iPod and Mac
4500
4000
3500
3000
2500
2000
Y Direction
0
50
100
150
200
250
300
350
400
X Direction
Hit
Curve 1
Point #207
x = 207, y = 3204.945718
Each column of data shows as a line
If a graphic is not perfect then Export to lay it out properly, or use a skin for a permanent alteration. Skins can also add distinction to your graphs.
Resize to make the graph larger
If there are multiple data sets (lines) then drag the cursor to scan each line separately

	Curve 1	Curve 2	Curve 3	Curve 4
1	1 2070.41163012			
2	2 2048.84464676			
3	3 2040.10856603			
4	4 2043.40115244			
5	5 2070.78508686			
6	6 2056.55479191			
7	7 2070.81399770			
8	8 2112.93170658			
9	9 2115.01040367			
10	10 2117.9459893			
11	11 2094.2332874			
12	12 2113.3880170			
13	13 2110.6603488			

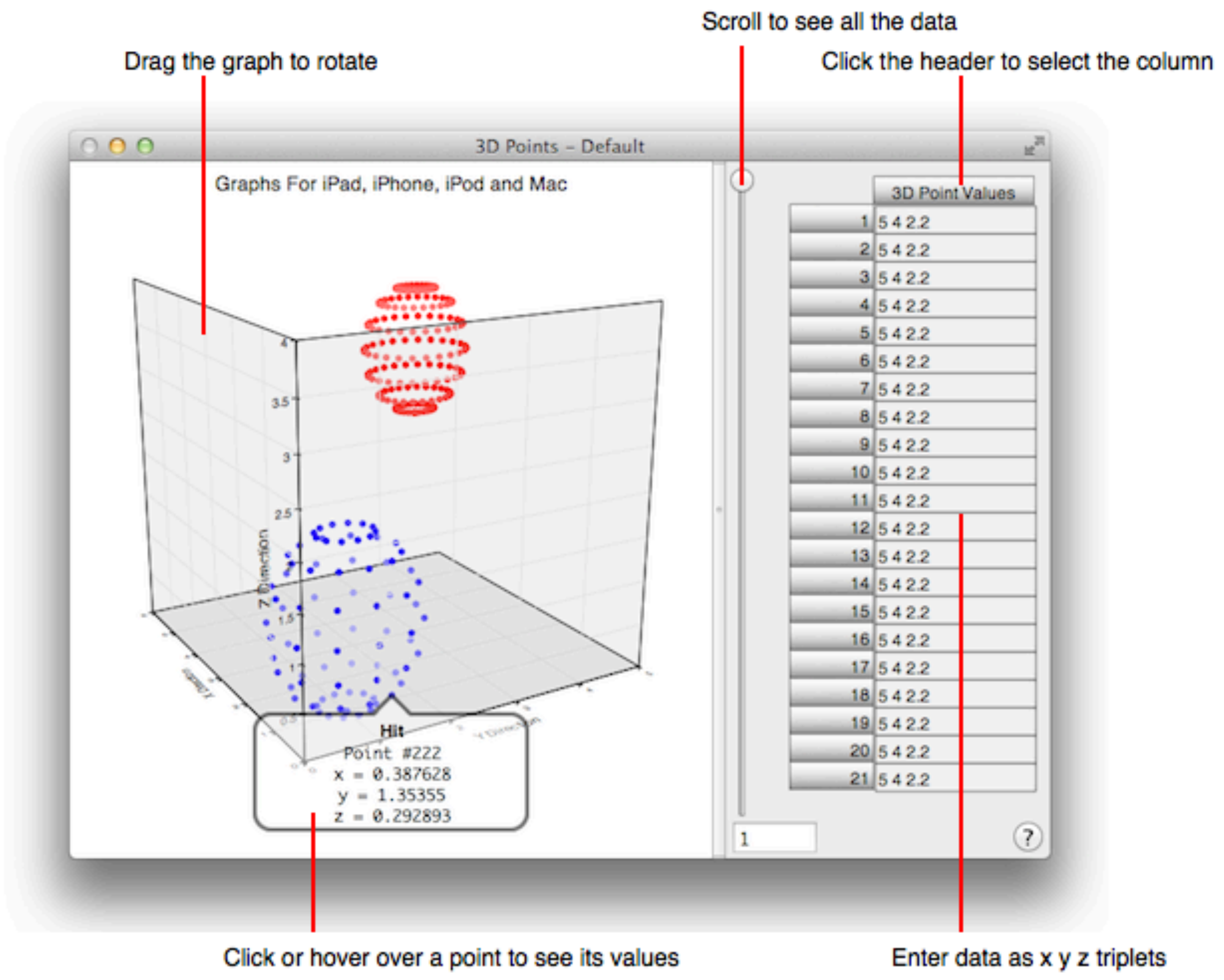
1 1

While importing data using Tables or Fetch keep in mind that the format is a list of pair of numbers (2D points) separated by a blank.

Hovering the cursor over a curve or data point shows its values. Dragging the cursor over the chart scans one column of data (curve) at a time.

Graph > Tasks > 3D Points

The 3D Points task makes 3D scatter plots of a sequence of triplet of numbers. By using the Preference tool it also makes 3D trajectory plots. The figure below annotates the 3D Points task user interface.


Scroll to see all the data
Drag the graph to rotate
Click the header to select the column
3D Points – Default
Graphs For iPad, iPhone, iPod and Mac
Z Direction
Click or hover over a point to see its values
Enter data as x y z triplets

Hit
Point #222
x = 0.387628
y = 1.35355
z = 0.292893

	3D Point Values
1	5 4 2.2
2	5 4 2.2
3	5 4 2.2
4	5 4 2.2
5	5 4 2.2
6	5 4 2.2
7	5 4 2.2
8	5 4 2.2
9	5 4 2.2
10	5 4 2.2
11	5 4 2.2
12	5 4 2.2
13	5 4 2.2
14	5 4 2.2
15	5 4 2.2
16	5 4 2.2
17	5 4 2.2
18	5 4 2.2
19	5 4 2.2
20	5 4 2.2
21	5 4 2.2

1

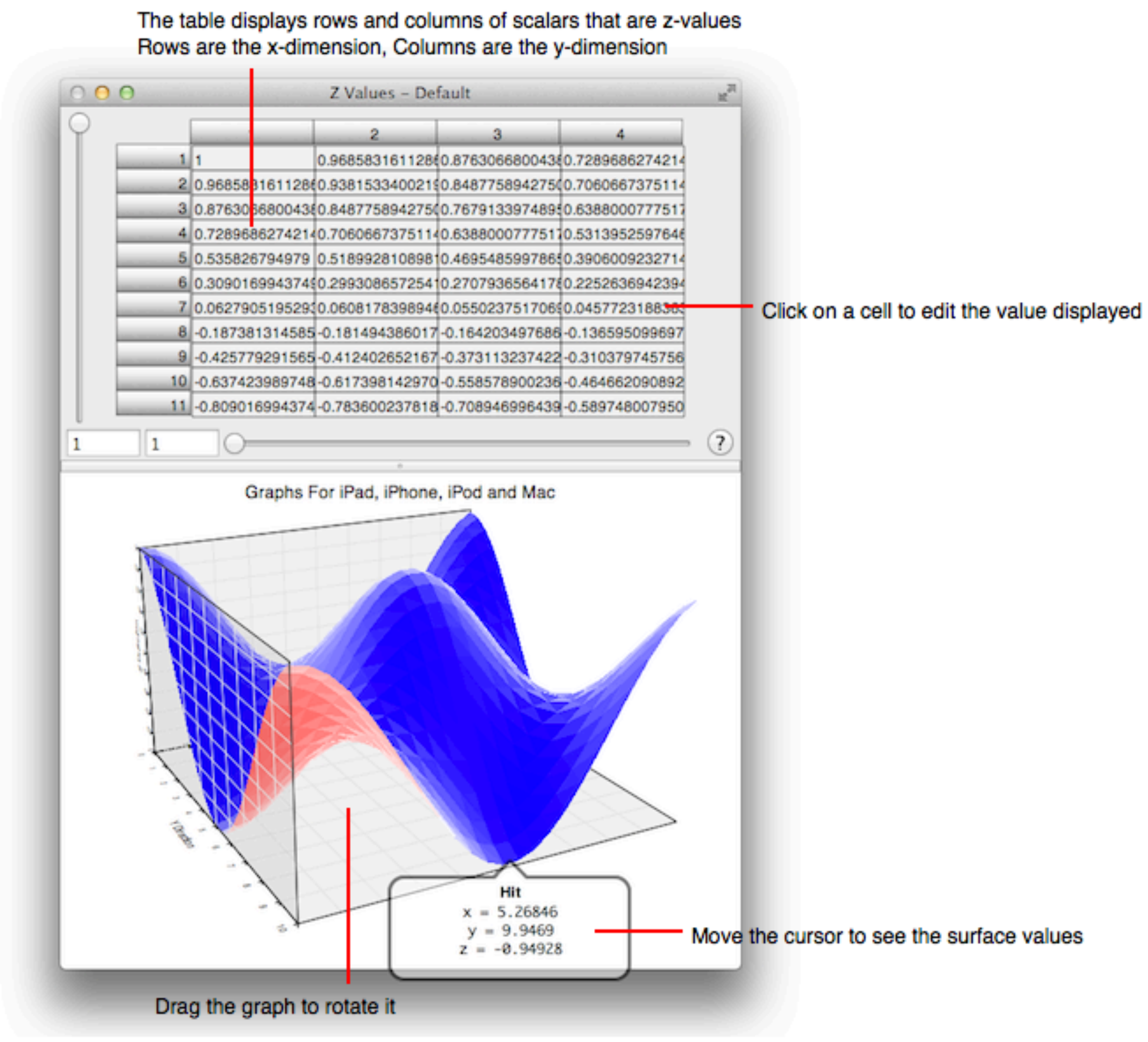
While importing data using Tables or Fetch keep in mind that the format is a list of triplets of numbers (3D points) separated by a blank such as x1 y1 z1 x2 y2 z2 ... xN yN zN.

Clicking or hovering the cursor over a point shows its value while dragging the cursor over the chart rotates it.

Graph > Tasks > Z Values

The Z Values task interprets the columns of a table as x-contiguous z-values and the rows of a table as y-contiguous z-values. The z-values correspond to height of cells on a 3D perspective plot or as cell amplitude on point fill plots. Some things to note about this task follows:

- This task makes surface graphs, but of course, not really. It actually interpolates z-values on a regular grid with a bi-linear function as a basis over that cell.
- The z-values are not actually plotted, that is a misnomer. What is actually plotted is a color gradation related to the z-values. For the flat (2D) point fill representation that gradation corresponds with the color map defined in the template. For the 3D perspective plot, the color mapping is remapped using a normal-to-observer amplitude to give a sense of reflection and hence 3D quality. Skins can be used to change some of the properties of the color mapping that are not available from the Preference tool settings.
- The grid dimensions are inferred from the table's column and row lengths. Notice that deleting columns in the table may give unexpected results as the grid cells are considered contiguous. Also, each column length should be the same.


The table displays rows and columns of scalars that are z-values
Rows are the x-dimension, Columns are the y-dimension
Z Values – Default
Click on a cell to edit the value displayed
1
1
Graphs For iPad, iPhone, iPod and Mac
Hit
x = 5.26846
y = 9.9469
z = -0.94928
Move the cursor to see the surface values
Drag the graph to rotate it

		2	3	4
1	1	0.968583161128	0.876306680043	0.728968627421
2	0.968583161128	0.938153340021	0.848775894275	0.706066737511
3	0.876306680043	0.848775894275	0.767913397489	0.638800077751
4	0.728968627421	0.706066737511	0.638800077751	0.531395259764
5	0.535826794979	0.518992810898	0.469548599786	0.390600923271
6	0.309016994374	0.299308657254	0.270793656417	0.225263694239
7	0.062790519529	0.060817839894	0.055023751706	0.045772318836
8	-0.187381314585	-0.181494386017	-0.164203497686	-0.136595099697
9	-0.425779291565	-0.412402652167	-0.373113237422	-0.310379745756
10	-0.637423989748	-0.617398142970	-0.558578900236	-0.464662090892
11	-0.809016994374	-0.783600237818	-0.708946996439	-0.589748007950

While importing data using Tables or Fetch keep in mind that the format is scalars separated by blanks.

Grid Definition

The numbers (a string representation) in the table are mapped onto a regular grid in the normal way in an x-contiguous and column-contiguous fashion. That is, lets say there are "n" z-values:

$z_1\ z_2\ z_3\ \ldots\ z_n$

On a grid with definitions and relationships:

n_x	The number of z-values in the x-direction (column length)
n_y	The number of z-values in the y-direction (row length)
n	Must equal $n_x\ n_y$
x_{min}	The value of x-minimum set in the preferences
x_{max}	The value of x-maximum set in the preferences
y_{min}	The value of y-minimum set in the preferences
y_{max}	The value of y-maximum set in the preferences
Δx	$= (x_{max} - x_{min})/(n_x - 1)$

Δy	$= (y_{max} - y_{min})/(n_y - 1)$
i	The index of the i-th element in the list of z-values (starting at 1)
j	The index of the j-th z-value in the x-direction (starting at 1)
k	The index of the k-th z-value in the y-direction (starting at 1)
i	$= (k-1)\ n_y + j$

then for the z-value $z_{j,k}$ (which is $z_i = z_{(k-1)\ n_y + j}$) the x and y values are defined as:

$x_j = x_{min} + \Delta x\ (j - 1)$

$y_k = y_{min} + \Delta y\ (k - 1)$

to form the triplet (3D point): $\{x_j, y_k, z_{j,k}\}$

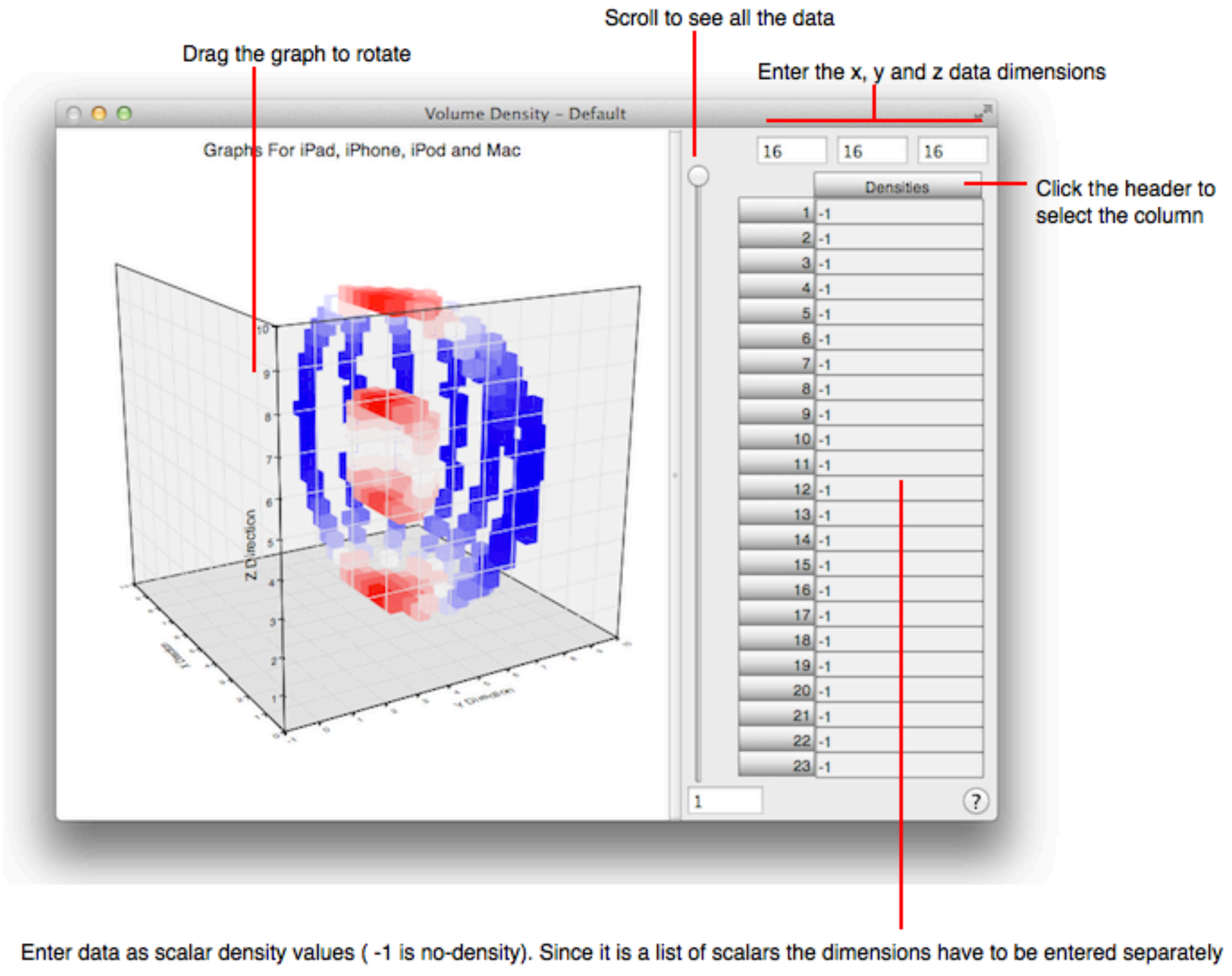
which is to say that "successive column-contiguous z-values are ordered in a x-contiguous way on a regular grid in the x and y direction" and the units in the x and y direction are specified by separate parameters (x_{min}, x_{max}, y_{min}, y_{max}).

Graph > Tasks > Density

The Density Task takes a sequence of scalars and presents them as boxes on a uniform 3D grid. The scalars have values between 0 and 1 and any value outside that range (especially -1) means "no density". By selecting Z Slice in the Preference tool the densities can be scanned in the z-direction one x-y plane at a time. Notice these peculiarities:

- The number of densities must equal the dimension entries on the task multiplied together.
- This task plots density values, a.k.a: densities, but is called Density (singular) because it is thought of as a continuous density "field" even though that terminology is not particularly right or appropriate.
- The color map in the template is key to making a good density plot. In particular, transparency is important to see within the density field. Many times a particular density is of interest and the color map can be adjusted so that that particular density value can show up as a unique color in the color map.
- Experimenting with the Volume 3D Data Graphic in Vvidget Builder might help with understanding the Density Task.

The figure below diagrams the Density task.


Scroll to see all the data
Drag the graph to rotate
Enter the x, y and z data dimensions
Volume Density – Default
Graphs For iPad, iPhone, iPod and Mac
Z Direction
Y Direction
Densities
Click the header to select the column
?
Enter data as scalar density values (-1 is no-density). Since it is a list of scalars the dimensions have to be entered separately

	Densities
1	-1
2	-1
3	-1
4	-1
5	-1
6	-1
7	-1
8	-1
9	-1
10	-1
11	-1
12	-1
13	-1
14	-1
15	-1
16	-1
17	-1
18	-1
19	-1
20	-1
21	-1
22	-1
23	-1

16 16 16

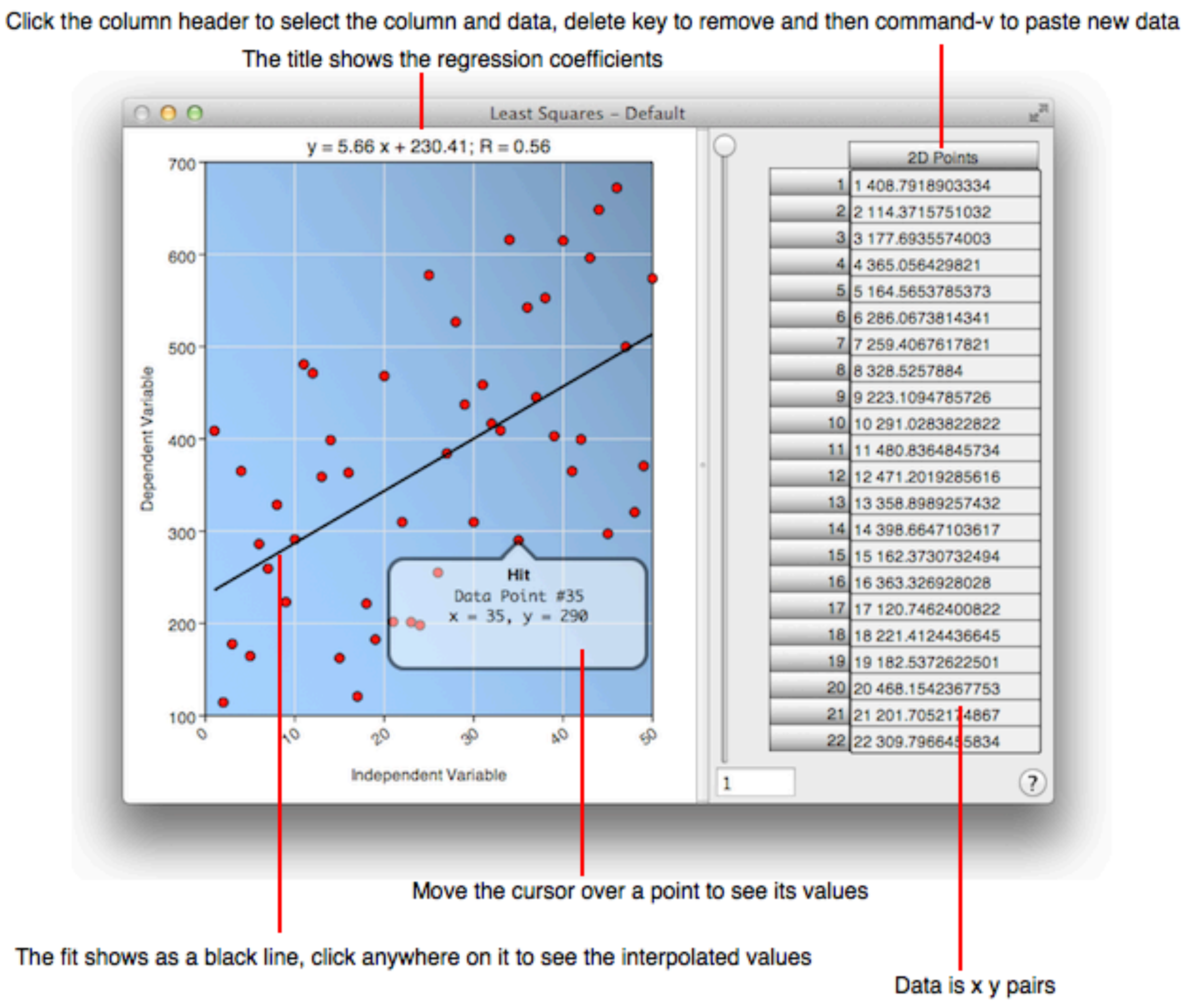
1

The 3D perspective graph is intended to give an overall qualitative indication of the data. The Z Splice point fill representation gives a more quantitative representation. Dragging the cursor over the 3D graph rotates it. While in Z Splice representation hovering the cursor over a density (in the x-y plane) shows its value and dragging the cursor scans each x-y plane successively.

Graph > Tasks > Least Squares

Least Squares takes 2D Points and plots them as a 2D scatter plot and also plots a linear regression of those 2D points.

The figure below annotates this task's user interface.


Click the column header to select the column and data, delete key to remove and then command-v to paste new data
The title shows the regression coefficients
Least Squares – Default
700
600
500
400
300
200
100
0
10
20
30
40
50
Dependent Variable
Independent Variable
1
Move the cursor over a point to see its values
The fit shows as a black line, click anywhere on it to see the interpolated values
Data is x y pairs

y = 5.66 x + 230.41; R = 0.56

Hit
Data Point #35
x = 35, y = 290

	2D Points
1	1 408.7918903334
2	2 114.3715751032
3	3 177.6935574003
4	4 365.056429821
5	5 164.5653785373
6	6 286.0673814341
7	7 259.4067617821
8	8 328.5257884
9	9 223.1094785726
10	10 291.0283822822
11	11 480.8364845734
12	12 471.2019285616
13	13 358.8989257432
14	14 398.6647103617
15	15 162.3730732494
16	16 363.326928028
17	17 120.7462400822
18	18 221.4124436645
19	19 182.5372622501
20	20 468.1542367753
21	21 201.70521[illegible]4867
22	22 309.79664[illegible]5834

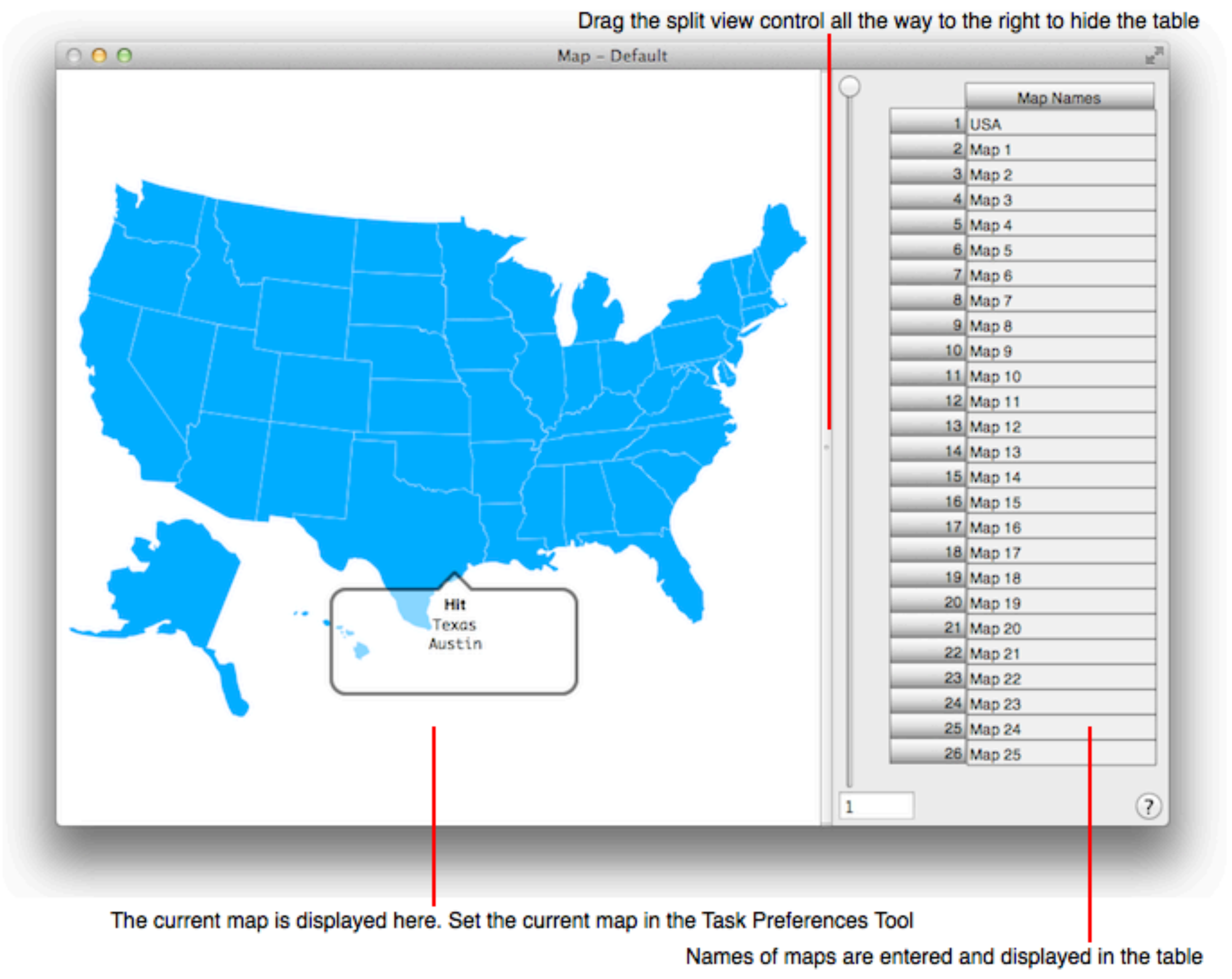
While importing data using Tables or Fetch keep in mind that the format is a list of pair of numbers (2D points) separated by a blank.

Hovering the cursor over a data point shows its values, hovering over the line segment shows the interpolated value of the regression. If you select the Labels representation in the Preference tool then you can see the sequence number of each data point on the chart.

Graph > Tasks > Map

The Map task interprets the columns of a table as names of maps. This is an unusual task because the "data" is really the Skin itself and the table entries are auxiliary. Hence, with the exception of the default USA map, a skin must be made to use the map task. Here are a few points regarding this task:

- The Making A Map Tutorial shows how to make a Vvidget Builder document and skin that is then imported by the Skins tool.
- The current map is selected by the Preference tool.
- The table is somewhat superfluous to using a map and you can hide it by moving the split view bar all the way to the right.

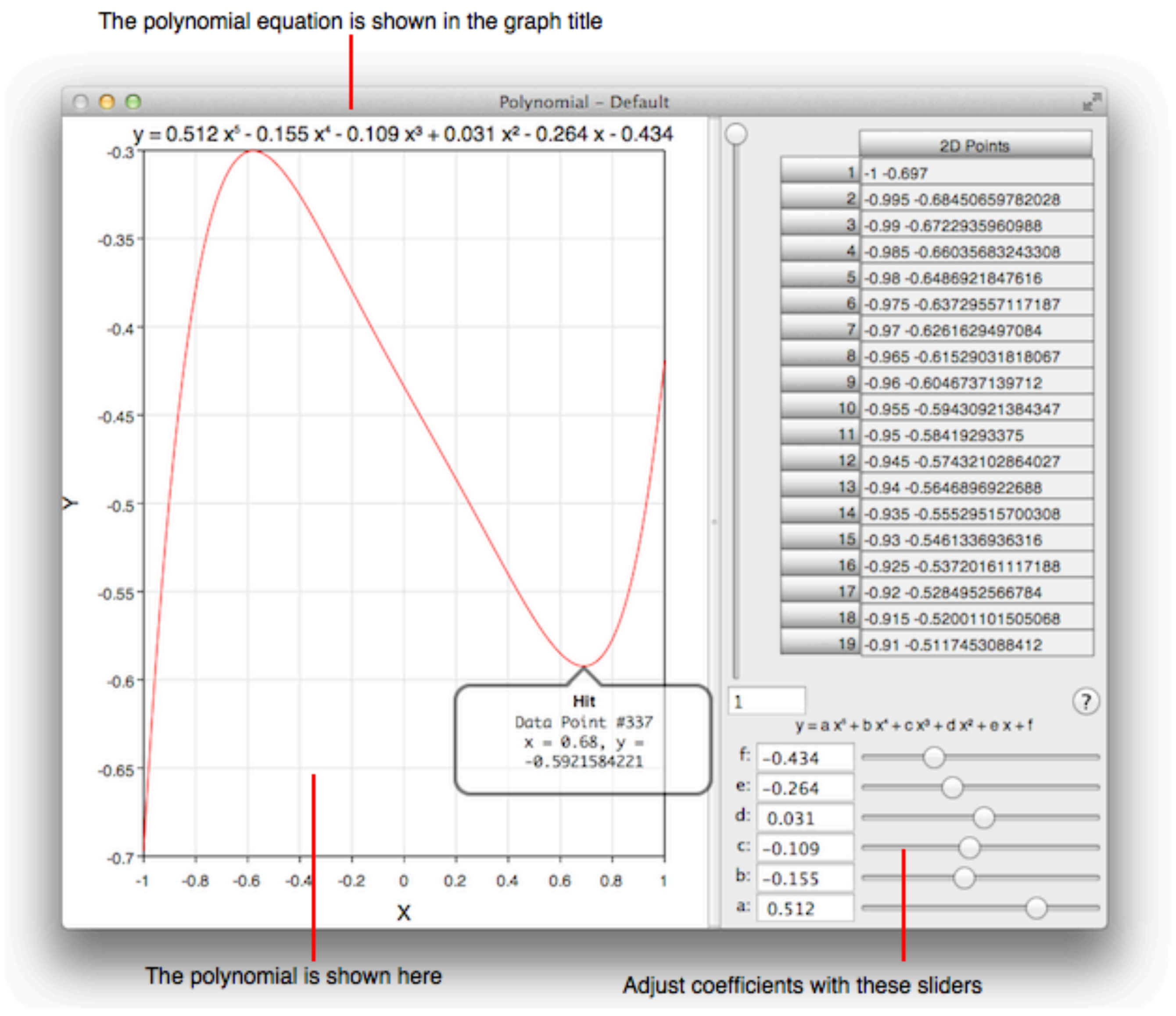


While importing data using Tables or Fetch keep in mind that the format is a list of names with one name per line.

Graph > Tasks > Polynomial

The Polynomial task utilizes six sliders to set the coefficients of the polynomial and then maps those entries into a table of x y pairs and a graph. Hence, in this task the "data" is the 6 coefficient scalars and the table is merely a mapping of those coefficients. Some things to note about this task follows:

- You can use the table to define the data, but if you then move a slider the table values will change as the sliders override the table entries.
- This task is really the beginning of a "graphing calculator" and many functions can be used instead of a polynomial. However, using a single equation has the advantage of controlling that equation with predefined UI (sliders) instead of a script type input so that the UI is easy.


The polynomial equation is shown in the graph title
Polynomial – Default
y = 0.512 x⁵ - 0.155 x⁴ - 0.109 x³ + 0.031 x² - 0.264 x - 0.434
-0.3
-0.35
-0.4
-0.45
-0.5
-0.55
-0.6
-0.65
-0.7
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1
X
Hit
The polynomial is shown here
Adjust coefficients with these sliders
?

```
Data Point #337
x = 0.68, y =
-0.5921584221
```

	2D Points
1	-1 -0.697
2	-0.995 -0.68450659782028
3	-0.99 -0.6722935960988
4	-0.985 -0.66035683243308
5	-0.98 -0.6486921847616
6	-0.975 -0.63729557117187
7	-0.97 -0.6261629497084
8	-0.965 -0.61529031818067
9	-0.96 -0.6046737139712
10	-0.955 -0.59430921384347
11	-0.95 -0.58419293375
12	-0.945 -0.57432102864027
13	-0.94 -0.5646896922688
14	-0.935 -0.55529515700308
15	-0.93 -0.5461336936316
16	-0.925 -0.53720161117188
17	-0.92 -0.5284952566784
18	-0.915 -0.52001101505068
19	-0.91 -0.5117453088412

1

$y = a x^5 + b x^4 + c x^3 + d x^2 + e x + f$

f:	-0.434
e:	-0.264
d:	0.031
c:	-0.109
b:	-0.155
a:	0.512

Graph > Tasks > Error Bars

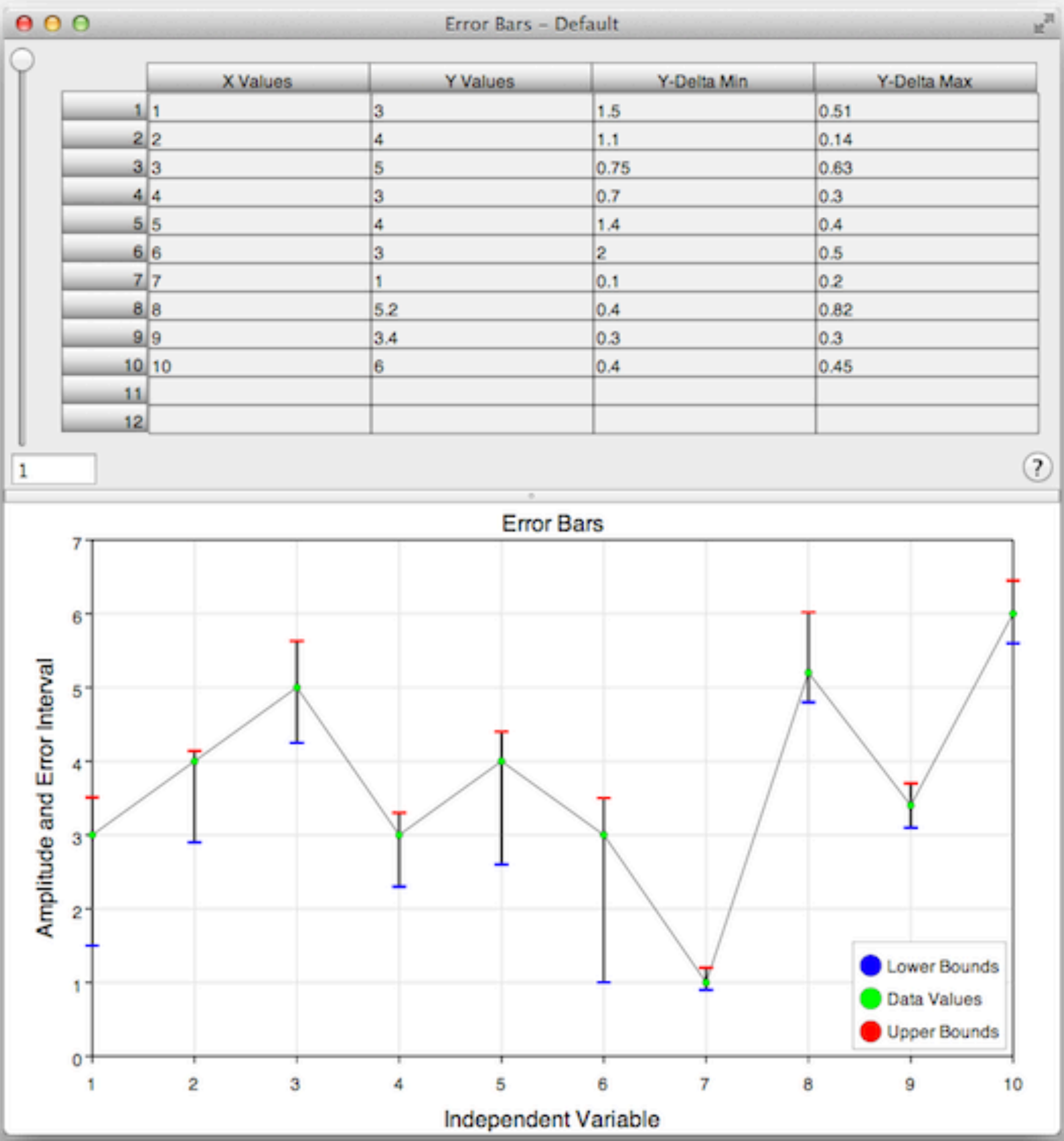
The Error Bars task helps make an error bar graph. Enter the data according to the following table.

Column Name	Explanation
X Values	The x-value for the data point and error bars. The data and error bar x-position are always identical.
Y Values	The y-value of the data (the y-location of the green dot)
Y-Delta Min	The length from the y-value of the data point to the lower part of the bar (the distance between the green dot and the blue bar).
Y-Delta Max	The length from the upper part of the bar to the y-value of the data point. (the distance between the red bar and the green dot).

Some things to note about this task are:

- Click on a data graphic component (green dot, red bar or blue bar) to edit its value.
- The data value (represented by the green dot) is entered in absolute value. However, the error bar data (y-delta min and y-delta max) are entered as the distance from the data point (relative values).
- To assign legend labels alt-click on a column header. See Tables for additional information.

The figure below shows the error bar task.


Error Bars – Default

	X Values	Y Values	Y-Delta Min	Y-Delta Max
1	1	3	1.5	0.51
2	2	4	1.1	0.14
3	3	5	0.75	0.63
4	4	3	0.7	0.3
5	5	4	1.4	0.4
6	6	3	2	0.5
7	7	1	0.1	0.2
8	8	5.2	0.4	0.82
9	9	3.4	0.3	0.3
10	10	6	0.4	0.45
11				
12				

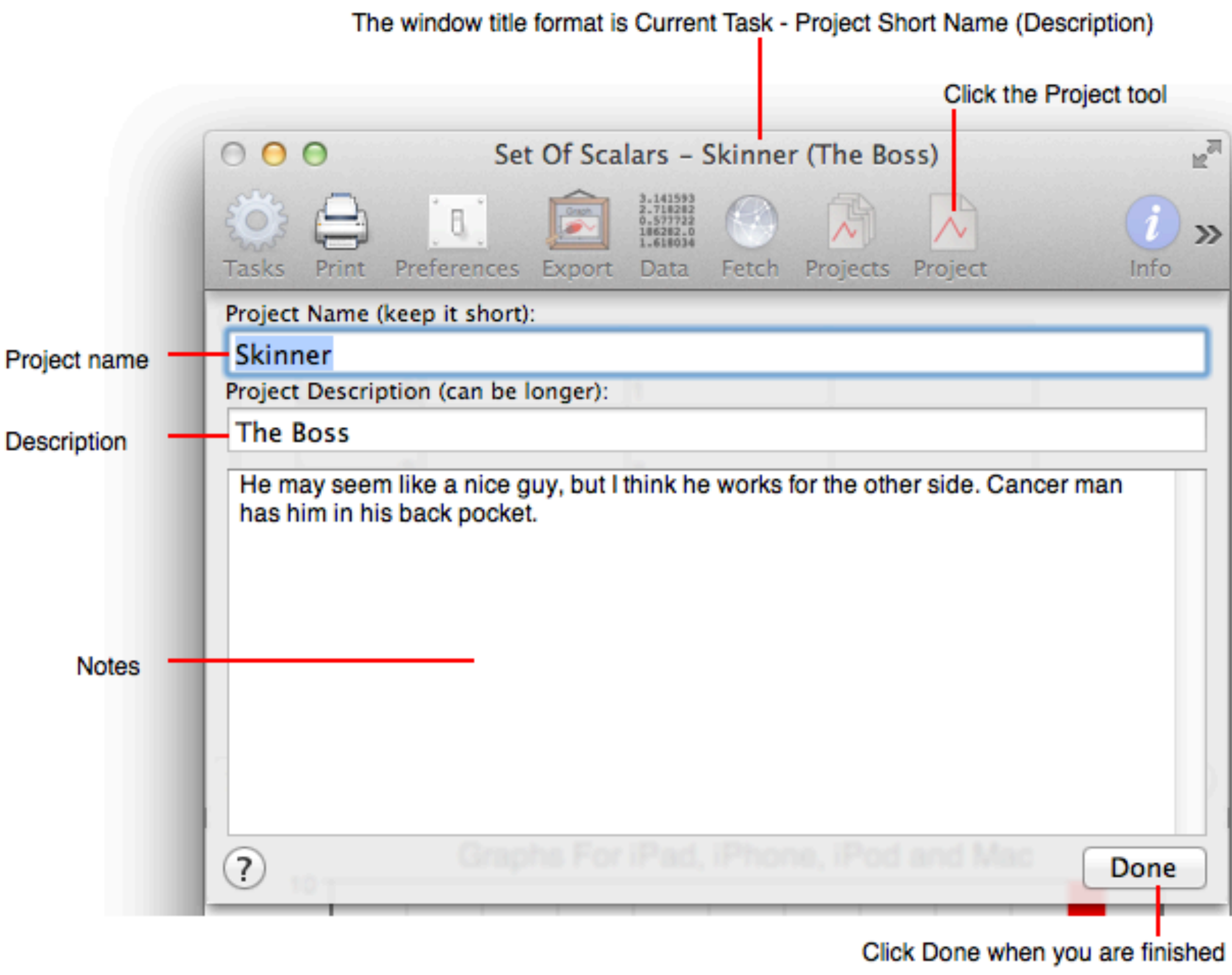
Graph > Tools

Tools operate on Tasks and are implemented as sheets accessed by clicking a toolbar tool icon. The following sections describe Tools.

Tool	Description
Tasks	Brings forward the tasks selector. Tasks are organized by data type and are described in the Tasks section.
Project	Brings forward the project sheet.
Projects	Brings forward the projects sheet. Data and preferences are stored on a project basis and you can define new projects or remove projects as needed. You can also set a project to the current project. Only the current project's data, preferences and skins are used by the tasks.
Export	Click this tool icon to export the current task's graph to Vvidget Builder for further layout and to use the full power of Vvidget Builder.
Print	Use this tool to print the current task's graph. The print layout is set to fill the page while maintaining aspect. If you need finer control of the graph layout then first export to Vvidget Builder, modify the document layout size (for example to the size of the print page) and then modify the graph as needed.
Data	Use this tool to set basic data-related attributes.
Task Edit	Use this tool to set task interface attributes.
Preferences	This tool brings forward the current task's graph preferences. When these preferences are not sufficient then you can use Skins or export to Vvidget Builder.
Project	This tool brings forward the current project's editor. For additional information consult Project.
Fetch	This tool is used to define Fetch parameters. Fetch is used to acquire external data and insert it into the current project and chart task.
Text	The text tool is used to show and edit data using familiar textual data methods.
Data	This tool is used to define basic data parameters.
Skins	This tool is used to define skins for the graph of the current project, task and data representation type.
Info	This tool shows the key value pairs (dictionary) that generates the graph displayed in the task.
Help	This tool shows help for the current task. Help is built into the Tasks, however this manual provides much more extensive descriptions of Tasks.
Customize	Use this tool to customize the toolbar. For instance, to add the Skins tool or remove the Help tool.

Graph > Tools > Project

The following figure diagrams the Project tool. It is associated with the current project only. You can set the project's short name, description and notes. The project name and description show in the task window.


The window title format is Current Task - Project Short Name (Description)
Click the Project tool
Set Of Scalars – Skinner (The Boss)
Tasks
Print
Preferences
Export
Data
Fetch
Projects
Project
Info
Project name
Description
Notes
Done
Click Done when you are finished

Project Name (keep it short): Skinner

Project Description (can be longer): The Boss

He may seem like a nice guy, but I think he works for the other side. Cancer man has him in his back pocket.

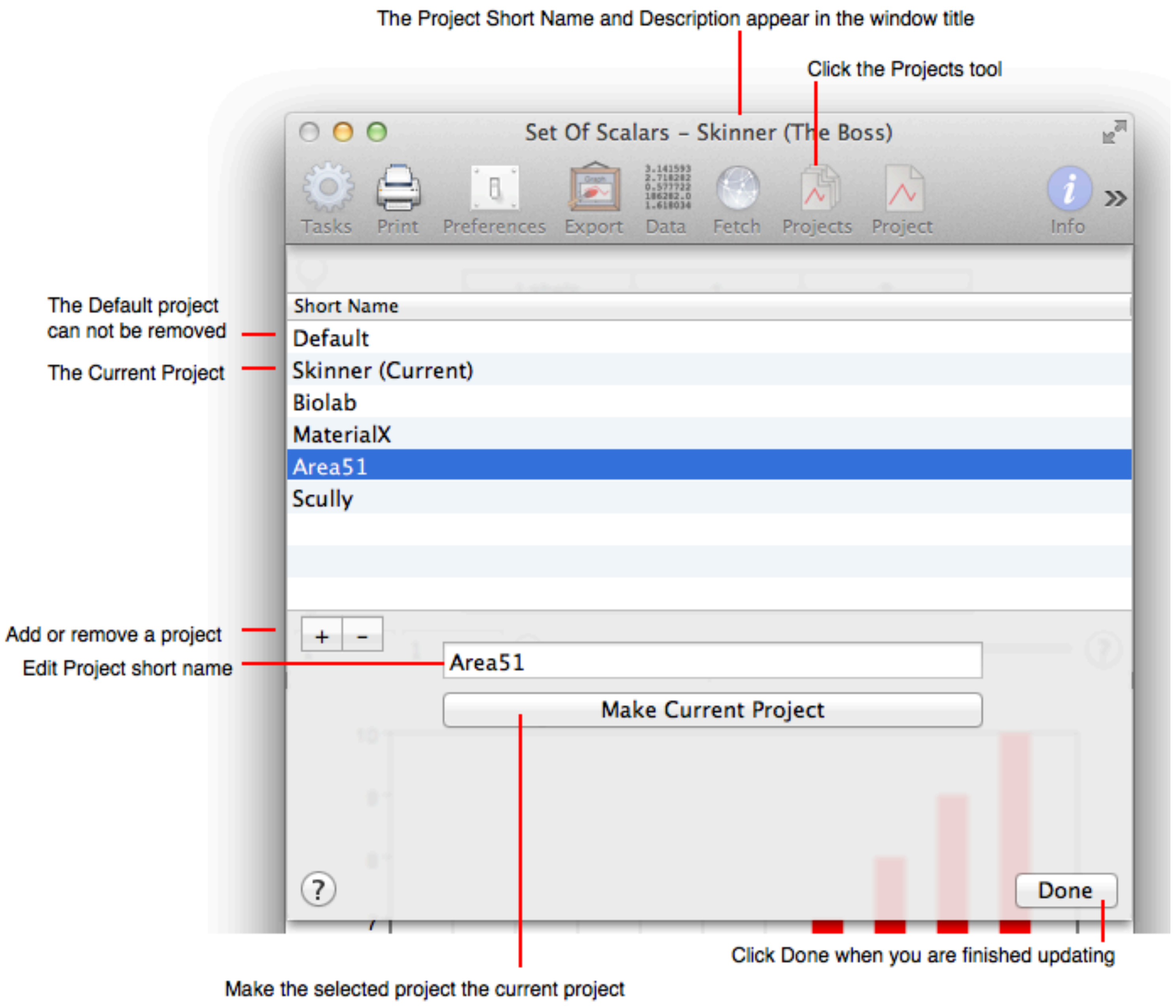
There really isn't much to the explicit project settings. All of the other settings and operations in the Graph are to manipulate the current project's data, preferences, skins and other attributes only. To reaffirm, projects other than the current project are not accessible by the chart tasks and to gain access to a project's settings you must make it the current project. For that see the Projects (plural) tool.

Graph > Tools > Projects

Graph's data, preferences, skins and other information are stored in a project. When you first use Graph the current project is the Default project. That project can not be renamed or removed. You can add new projects and delete and rename projects other than the Default project. This section describes how to use projects. First note that a project is a document, but unlike the Vvidget Builder document and other documents you may be use to, a project is only referenced and navigated by the project interfaces shown here. Projects are not accessible in the Finder and can not be navigated using the system Open and Save panels. Because of that, projects are very easy to use and, in fact, are almost implicit in nature, that is: Project preferences, data and other parameters are retrieved, saved and used without any explicit action on the part of the user.

Use the Projects (plural) tool, as described below, to add and remove projects and set the current project and the Project (singular) tool to define current project attributes.

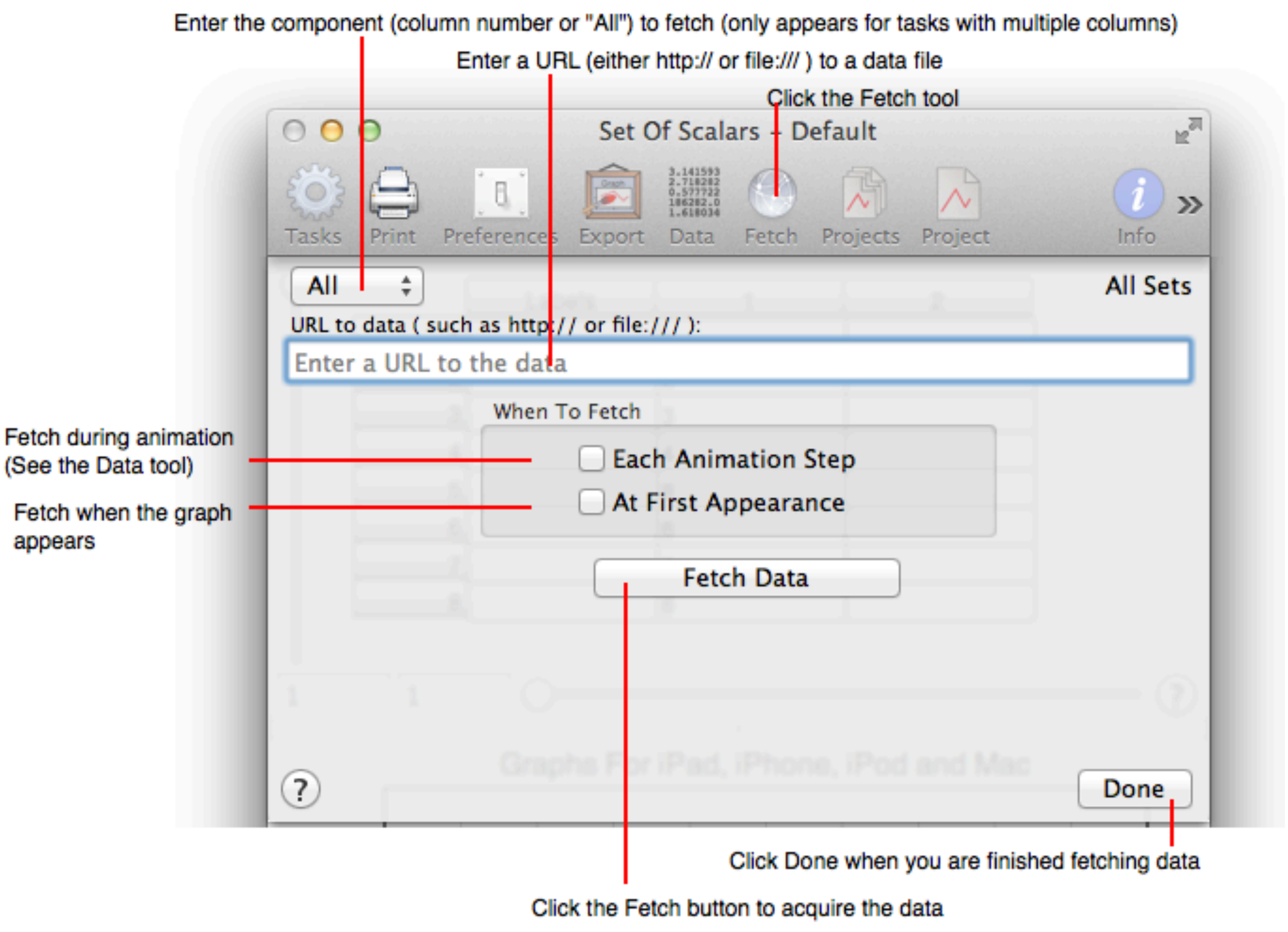
The following figure diagrams the Projects tool. As you can see you can add projects to a list, select projects and make a project the current project. Once a project is the current project its Tasks (or more accurately its data associated with tasks) are available for use.


The Project Short Name and Description appear in the window title
Click the Projects tool
Set Of Scalars – Skinner (The Boss)
Tasks
Print
Preferences
Export
Data
Fetch
Projects
Project
Info
The Default project can not be removed
The Current Project
Add or remove a project
Edit Project short name
+
-
Make Current Project
Done
Click Done when you are finished updating
Make the selected project the current project

Short Name
Default
Skinner (Current)
Biolab
MaterialX
Area51
Scully

Area51

Use the Fetch tool to retrieve data from a web server or the file system on your computer as referenced by a URL. The Fetch tool is diagrammed here:


Enter the component (column number or "All") to fetch (only appears for tasks with multiple columns)
Enter a URL (either http:// or file:///) to a data file
Click the Fetch tool
Set Of Scalars - Default
Tasks
Print
Preferences
Export
Data
Fetch
Projects
Project
Info
All
All Sets
URL to data (such as http:// or file:///):
Enter a URL to the data
When To Fetch
Each Animation Step
At First Appearance
Fetch Data
Done
Fetch during animation (See the Data tool)
Fetch when the graph appears
Click Done when you are finished fetching data
Click the Fetch button to acquire the data

URL Types

There are two types of URLs that can be used as defined here:

- File: A file URL begins with "file://" and the rest of the URL is the path to the data file. Since paths are absolute there are three leading slashes, such as: "file:///Users/steve/data/mypoints.txt" (without the quotes). In this example, the path extension is txt however in practice it can be many things or the file can be without extension.
- Web: A web URL begins with "http://" and the rest of the URL is the path to the data file, such as: "http://www.vvi.com/data/mypoints.txt" (without the quotes). Notice that the path extension is "txt" in this case. Although the extension is arbitrary, txt is a safe extension as it informs a web server to use an ASCII MIME type. In practice, any extension, or no extension, also works. This is the same URL that you would type into a web browser to download the data. You can use the data directly in this manner, or first download it and then use the File type URL. Either way, the data file must be a resource of the web server that the URL points to, either as a static file or a dynamic URL that retrieves the data file bytes from other places.

Notice that the Web URL gives access to a programmable system. For example, you can turn on your Web Sharing in the system preferences and then use PHP, perl or other scripting languages to write algorithms to retrieve computed data. The URL would be of this form: "http://localhost/cgi-bin/myalgorithm?data=lab1&section=2&param1=5" and you use the scripting engine's built-in form facilities to parse the URL parameters as input to an algorithm. Alternatively, the URL could point to an existing web service to retrieve SOA type information from queries.

Content Format Type

If the component pop up button does not exist or a column number is selected on it then the content retrieved from a URL (the response bytes) is formatted according to the list below. Note that the fetch content format is simple, it is just a list of numbers with a blank delimiter. No fancy XML formatting and no dimension data. It is intended be as simple as possible in order to facilitate ease of use. Data formats are also explained in each Task's Help tool and section in this manual.

- A list of numbers for the Set Of Scalars task.
- A list of 2D points (two numbers) for the Set Of 2D Points task, such as: x1 y1 x2 y2 ... xN yN.
- A list of 3D points (3 numbers) for the 3D Points Task.
- A list of numbers for the Z Values and Density tasks.

If the component pop up button does not exist or "All" is selected then the content retrieved from a URL (the response bytes) can be XML. For an explanation of this content see the XML Fetch tutorial.

Fetch Timing

Fetching data can occur at different time intervals as defined here:

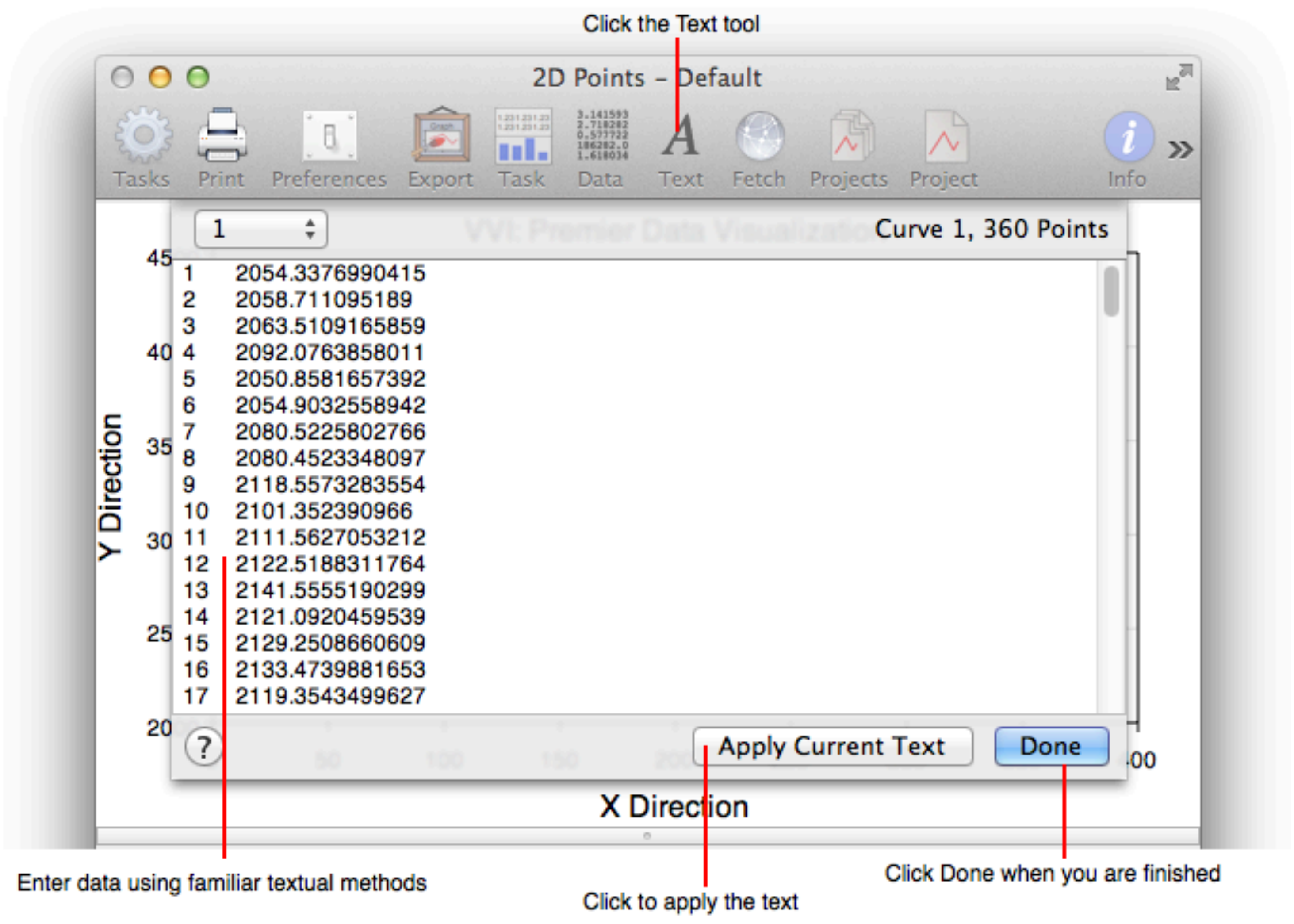
- Fetch Data: Clicking the Fetch Data button fetches the data immediately.
- Each Animation Step: Selecting the Each Animation Step switch causes the data to be fetched upon each animation, which is at an interval of one second. If you use this button then go to the Data tool and select Animate.
- At First Appearance: Selecting the At First Appearance switch causes the data to be fetched when the task is made visible. Tasks are only made visible when switching tasks so this operation does not occur too frequently.

Fetch Operation

In a nutshell, when a fetch occurs the data is retrieved, entered into the task and then displayed. Usually the fetch appears to be atomic, that is: it seems to be a single operation. However, fetching is asynchronous and non-atomic. You may notice this fact if the fetch takes a long time. Also, for tasks that have multiple columns the fetch is on a per-column basis and the fetch retrieves each column individually until all columns are fetched and then updates the task with the fetched data. Multiple column fetches are done in parallel and asynchronously and each column fetch rendezvous to amalgamate the fetch into a single result.

Graph > Tools > Text

Use the Text tool to work with data using familiar textual methods. The Text tool is diagrammed here:



```
1    2054.3376990415
2    2058.711095189
3    2063.5109165859
4    2092.0763858011
5    2050.8581657392
6    2054.9032558942
7    2080.5225802766
8    2080.4523348097
9    2118.5573283554
10   2101.352390966
11   2111.5627053212
12   2122.5188311764
13   2141.5555190299
14   2121.0920459539
15   2129.2508660609
16   2133.4739881653
17   2119.3543499627
```

Here are a few facts about the text tool:

- The text tools is all about familiarity. If you find the Fetch tool and the Table interface to be too unfamiliar then the Text tool may be appropriate.
- Obviously you can edit, copy and paste textual sequences of data. However, the textual methods are somewhat difficult for large data sets because there is no way to know which row you are at and also editing very large sets may be slow.
- Textual methods are generic whereas the Table interface is specialized. Consider using the table interface and becoming accustom to its features instead of falling back on textual entry.

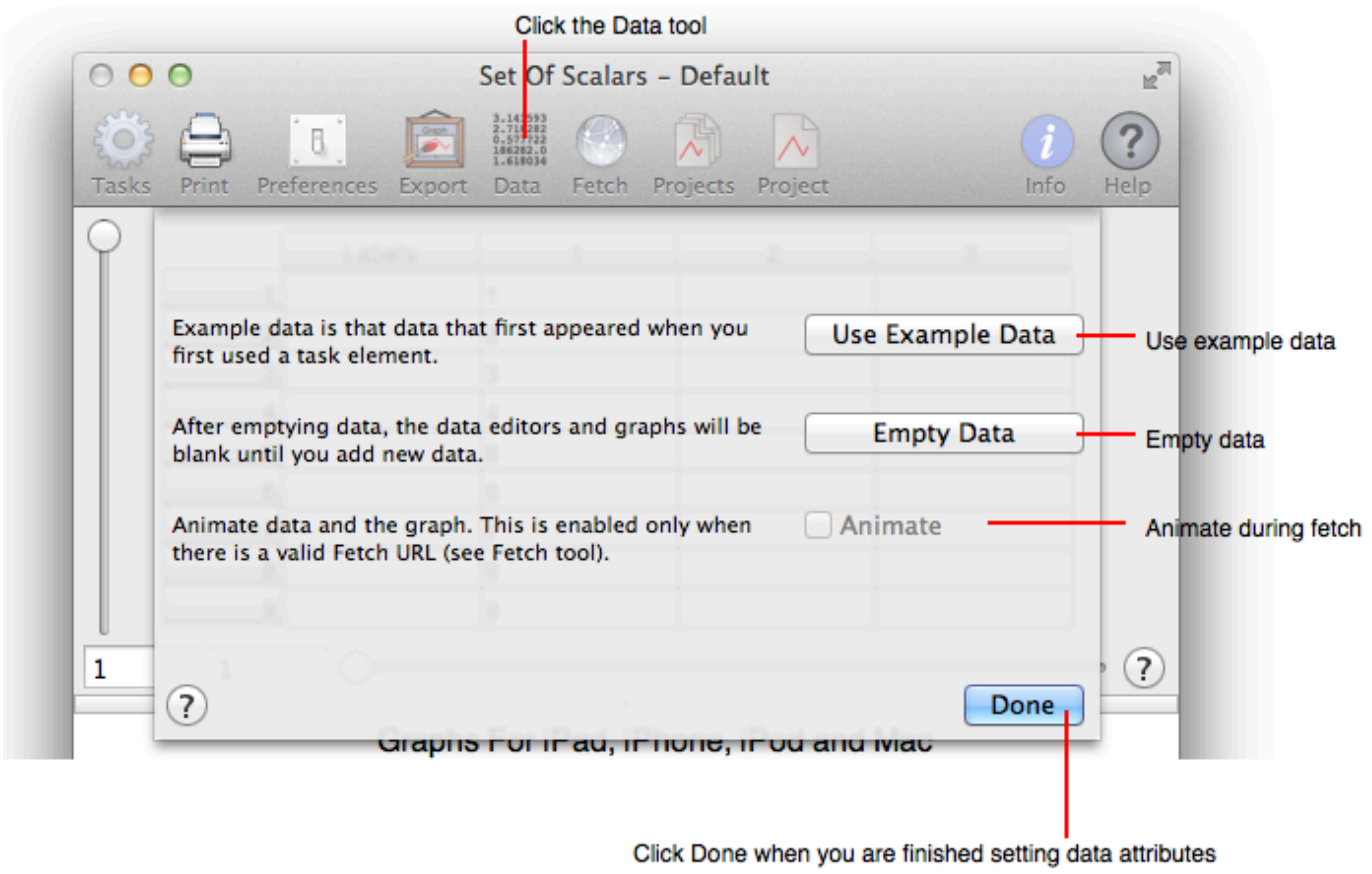
When the data represents more than one column then select the column entry from the pop up button. The text is free formatted and does not represent all of the data parameters so may not be appropriate for all needs.

Graph > Tools > Data

Use the Data tool to set these basic data-related parameters:

- Use Example Data: Resets the data content to example data, which is that data that first appeared when you used the task.
- Empty Data: Empties the data content. After emptying you can then insert data and be assured that no previous data is being used.
- Animate: Animates the fetch of the data. This is only enabled if the Fetch tool is set with a valid URL. Animation occurs every one second at which time the fetch URL is executed and new data is retrieved.

The figure below diagrams the Data tool.



Example data is that data that first appeared when you first used a task element.

After emptying data, the data editors and graphs will be blank until you add new data.

Animate data and the graph. This is enabled only when there is a valid Fetch URL (see Fetch tool).

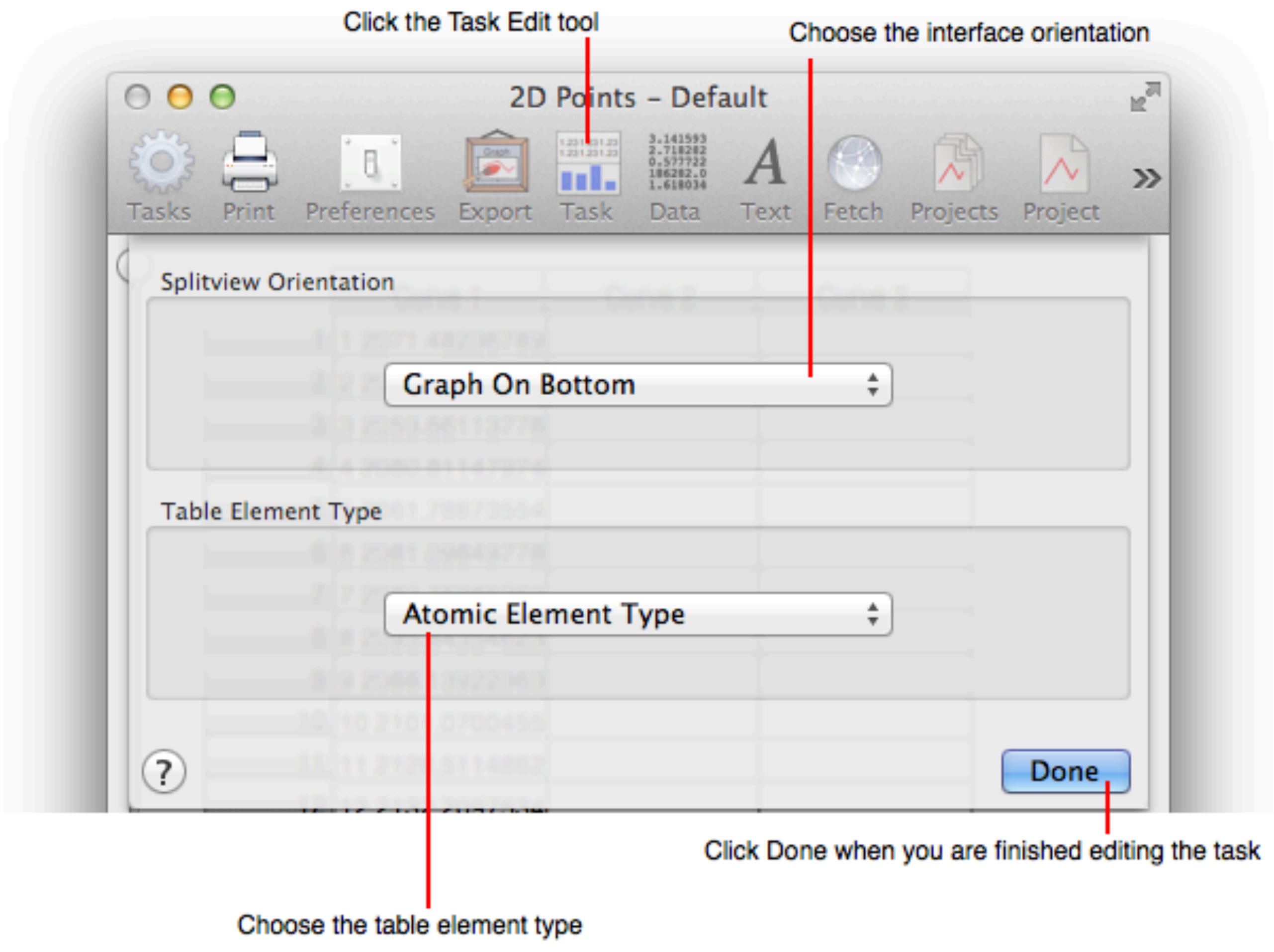
Notice that the Data tool has nothing to do with the data content but rather just some data-related parameters that define data content. To define the data content see the Fetch and Tables sections.

Graph > Tools > Task Edit

The Task Edit tool alters the main states of the task interface according to the following:

- **Orientation**: Chart tasks usually have two main interface areas, a table and a graph. The orientation pop up button can be used to set the graph to the bottom, right, top or left of the table.
- **Table Element Type**: The table element type pop up button is used to set the table cell type to either atomic or component. For additional information see the Tables section.

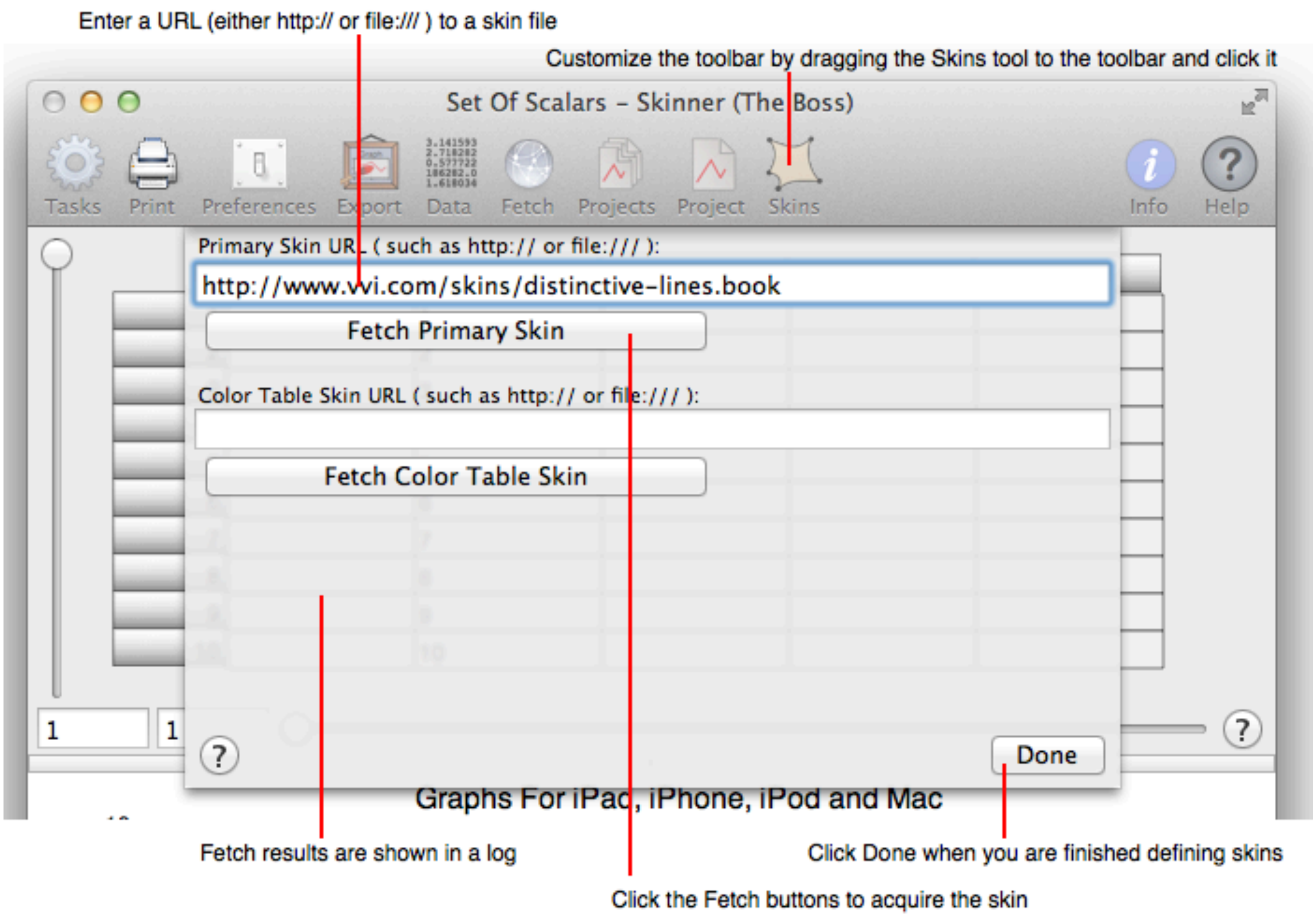
The figure below diagrams the Task Edit tool.



Graph > Tools > Skins

Each graph within the Tasks is defined by a specialized Vvidget Builder document called a skin. By having a task use your own skin you can make the graph have distinctive qualities that are otherwise unavailable with the limited controls of the task. Making a skin is described below, but first lets explain how to get a skin into a task.

Click the Skins tool to bring forward a sheet like that shown below.


Enter a URL (either http:// or file:///) to a skin file
Customize the toolbar by dragging the Skins tool to the toolbar and click it
Set Of Scalars – Skinner (The Boss)
Tasks
Print
Preferences
Export
Data
Fetch
Projects
Project
Skins
Info
Help
Primary Skin URL (such as http:// or file:///):
http://www.vvi.com/skins/distinctive-lines.book
Fetch Primary Skin
Color Table Skin URL (such as http:// or file:///):
Fetch Color Table Skin
Done
Graphs For iPad, iPhone, iPod and Mac
Fetch results are shown in a log
Click Done when you are finished defining skins
Click the Fetch buttons to acquire the skin

The Primary Skin defines the graph attributes while the Color Table Skin defines colors associated with distinct elements of the data graphics, such as curves. To import a skin type its location as a URL into the corresponding field and then click the respective Fetch button, then click Done. The graph of the current task will be updated to reflect the new skin.

URL Types

There are two types of URLs that can be used as defined here:

- File: A file URL begins with "file://" and the rest of the URL is the path to the skin file. Since paths are absolute there are three leading slashes, such as: "file:///Users/steve/skins/distinctive-line.book" (without the quotes). Notice that the path extension is "book".
- Web: A web URL begins with "http://" and the rest of the URL is the path to the skin document, such as: "http://www.vvi.com/skins/distinctive-line.book" (without the quotes). Notice that the path extension is "book". This is the same URL that you would type into a web browser to download the skin. You can use the skin directly in this manner, or first download it and then use the File type URL. Either way, the skin file must be a resource of the web server that the URL points to, either as a static file or a dynamic URL that retrieves the skin file bytes from other places.

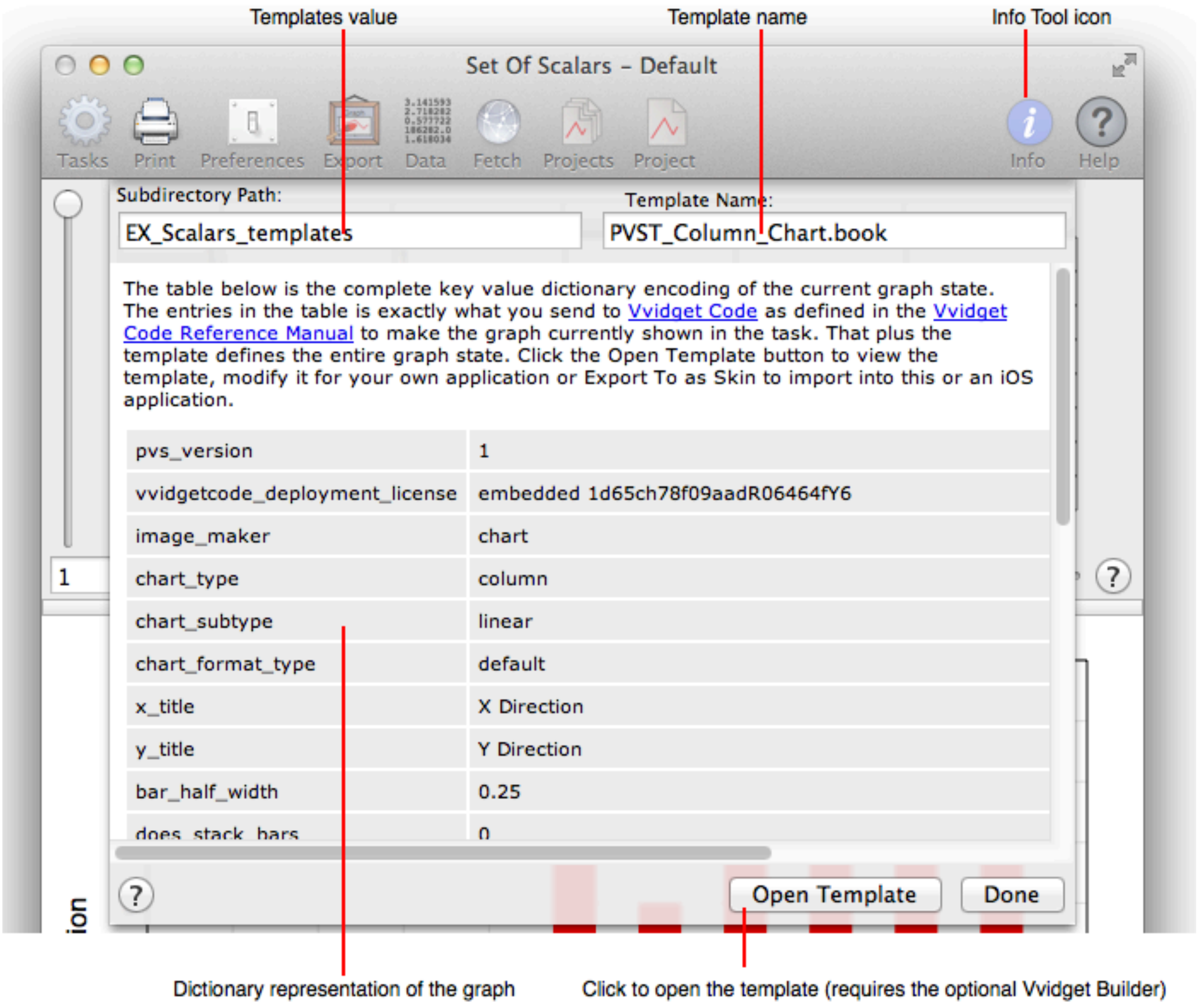
Constructing Skins

A Skin file is simply a Vvidget Builder document that has specialized graphical elements. To facilitate the retrieval of the document it should be exported as a Skin type using the menu item Vvidget Builder > File > Export To ... and choosing the Skin type on the resulting panel. That exported file must have a book extension. The export converts the Vvidget Builder document, which is a bundle of resources, into a compressed binary flat archive which can be efficiently transferred.

The hard part in making a skin is understanding the "specialized graphical elements" on the document. For that consult the Vvidget Code Reference Manual.

Graph > Tools > Info

The Info tool shows the key value pairs (dictionary) that is used to make the graph of the task. This information is very useful for constructing XML Fetch content and for developing applications with Vvidget Code. The figure below shows the Info sheet.

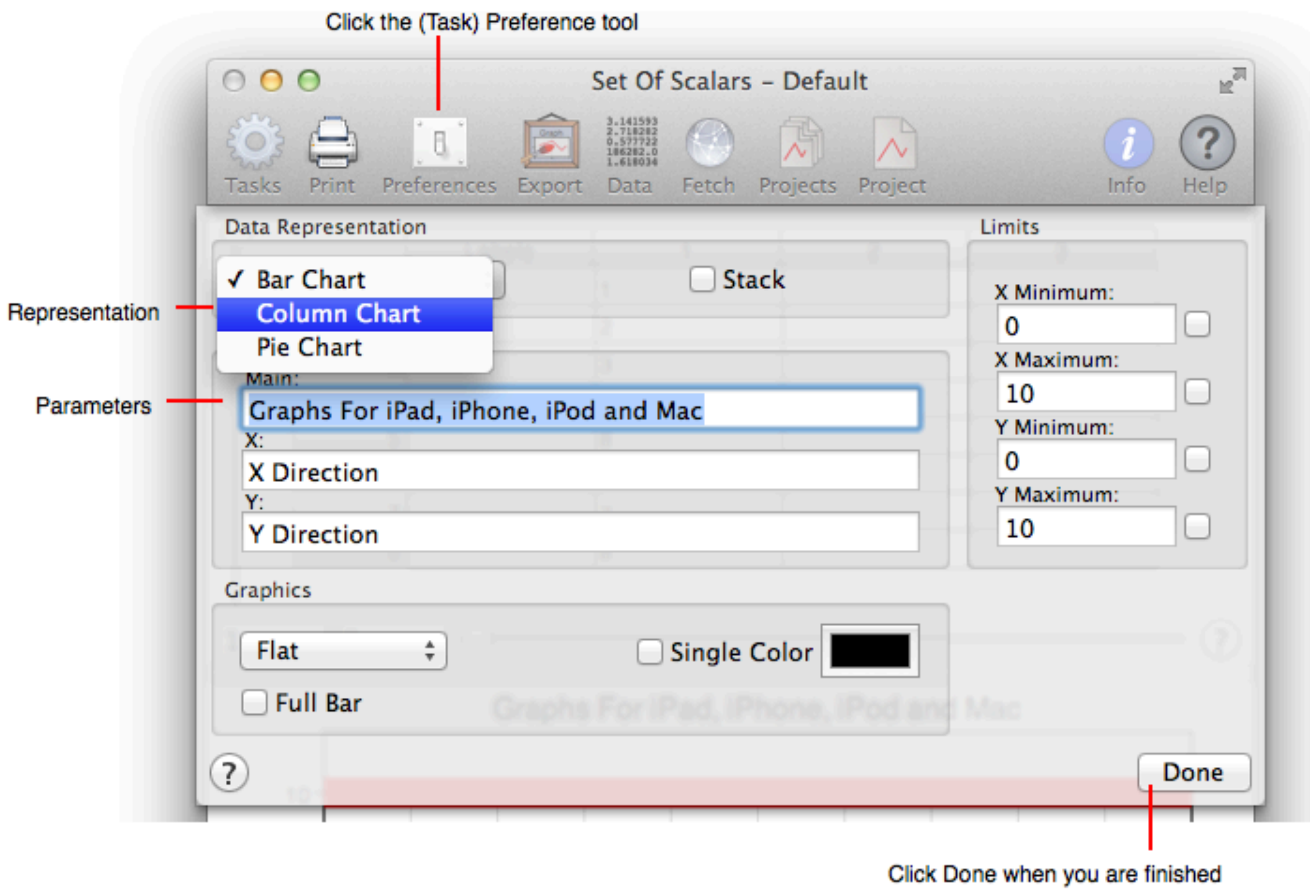


The table below is the complete key value dictionary encoding of the current graph state. The entries in the table is exactly what you send to Vvidget Code as defined in the Vvidget Code Reference Manual to make the graph currently shown in the task. That plus the template defines the entire graph state. Click the Open Template button to view the template, modify it for your own application or Export To as Skin to import into this or an iOS application.

pvs_version	1
vvidgetcode_deployment_license	embedded 1d65ch78f09aadR06464fY6
image_maker	chart
chart_type	column
chart_subtype	linear
chart_format_type	default
x_title	X Direction
y_title	Y Direction
bar_half_width	0.25
does_stack_bars	0

Graph > Tools > Preferences

The following figure diagrams the Task Preferences tool. It is associated with the current task only and usually is used to affect changes to the task's graphical representation.



The Preferences settings are fairly self-explanatory and can vary according to the current task, but generally fall into these categories:

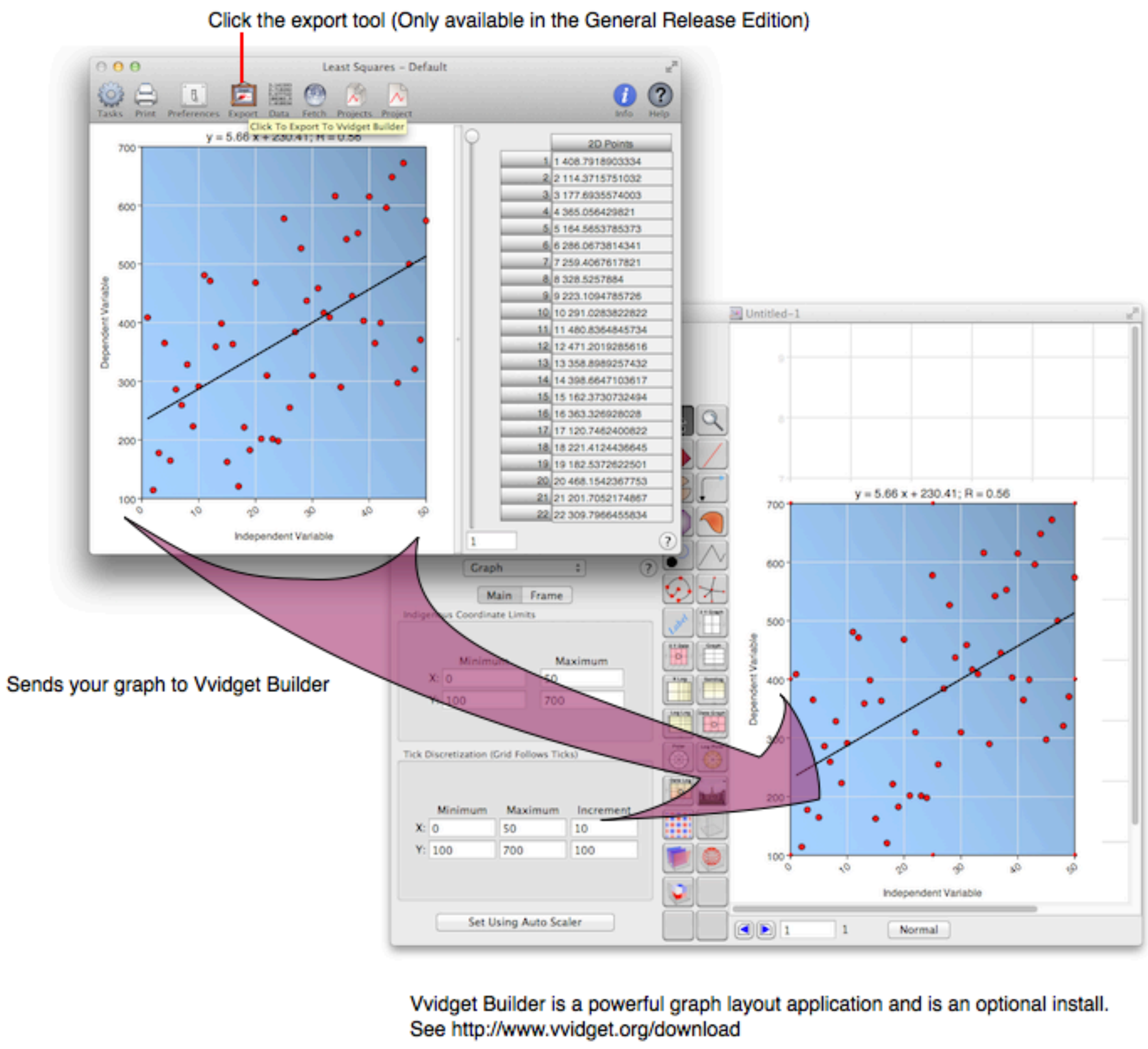
- Data Representation: Sets the way the data is represented on the graph.
- Titles: Sets the respective graph titles.
- Graphics: Any effects are set in this area.
- Limits: The graph's limits are on autoscale, however a limit may be explicitly set by entering the limit value and then turning on that option with the switch next to the limit value.

The Preferences are purposely frugal since the intent is to focus on data representation, discovery and presentation and not necessarily graphical presentation. If you need more options (vastly more!) then consider the Export tool or the Skins tool.

Graph > Tools > Export

Any graph can be sent to the powerful graph layout tool Vvidget Builder (an optional application) where it can be altered in fine detail. Click the Export tool, diagrammed in the figure below, to do so.

Note: The Export tool is only available in the General Release Edition of the Graph application and requires the optional Vvidget Builder application (see: Download).


Click the export tool (Only available in the General Release Edition)
Least Squares – Default
Tasks
Print
Preferences
Export
Data
Fetch
Projects
Project
Info
Help
Click To Export To Vvidget Builder
y = 5.66 x + 230.41; R = 0.56
Dependent Variable
Independent Variable
Graph
Main
Frame
Untitled-1
Sends your graph to Vvidget Builder
Set Using Auto Scaler
Normal
Vvidget Builder is a powerful graph layout application and is an optional install.
See http://www.vvidget.org/download

	2D Points
1	1 408.7918903334
2	2 114.3715751032
3	3 177.6935574003
4	4 365.056429821
5	5 164.5653785373
6	6 286.0673814341
7	7 259.4067617821
8	8 328.5257884
9	9 223.1094785726
10	10 291.0283822822
11	11 480.8364845734
12	12 471.2019285616
13	13 358.8989257432
14	14 398.6647103617
15	15 162.3730732494
16	16 363.326928028
17	17 120.7462400822
18	18 221.4124436645
19	19 182.5372622501
20	20 468.1542367753
21	21 201.7052174867
22	22 309.7966455834

Indigenous Coordinate Limits

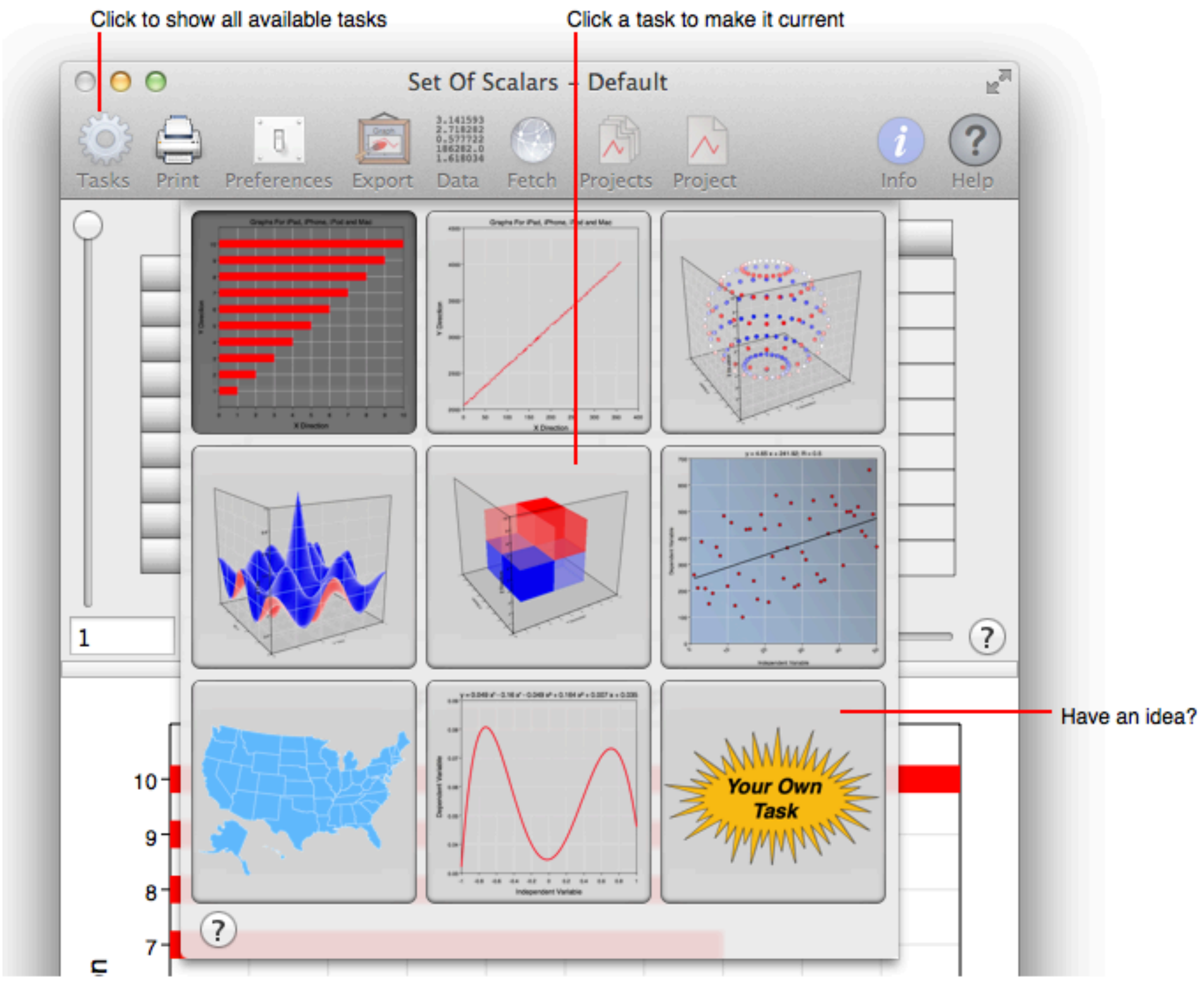
	Minimum	Maximum
X:	0	50
Y:	100	700

Tick Discretization (Grid Follows Ticks)

	Minimum	Maximum	Increment
X:	0	50	10
Y:	100	700	100

Once in Vvidget Builder, see its manual (also available at: Vvidget Builder User Manual) for further information. Vvidget Builder is very powerful and flexible and, unlike the Graph application, takes some experience to use well.

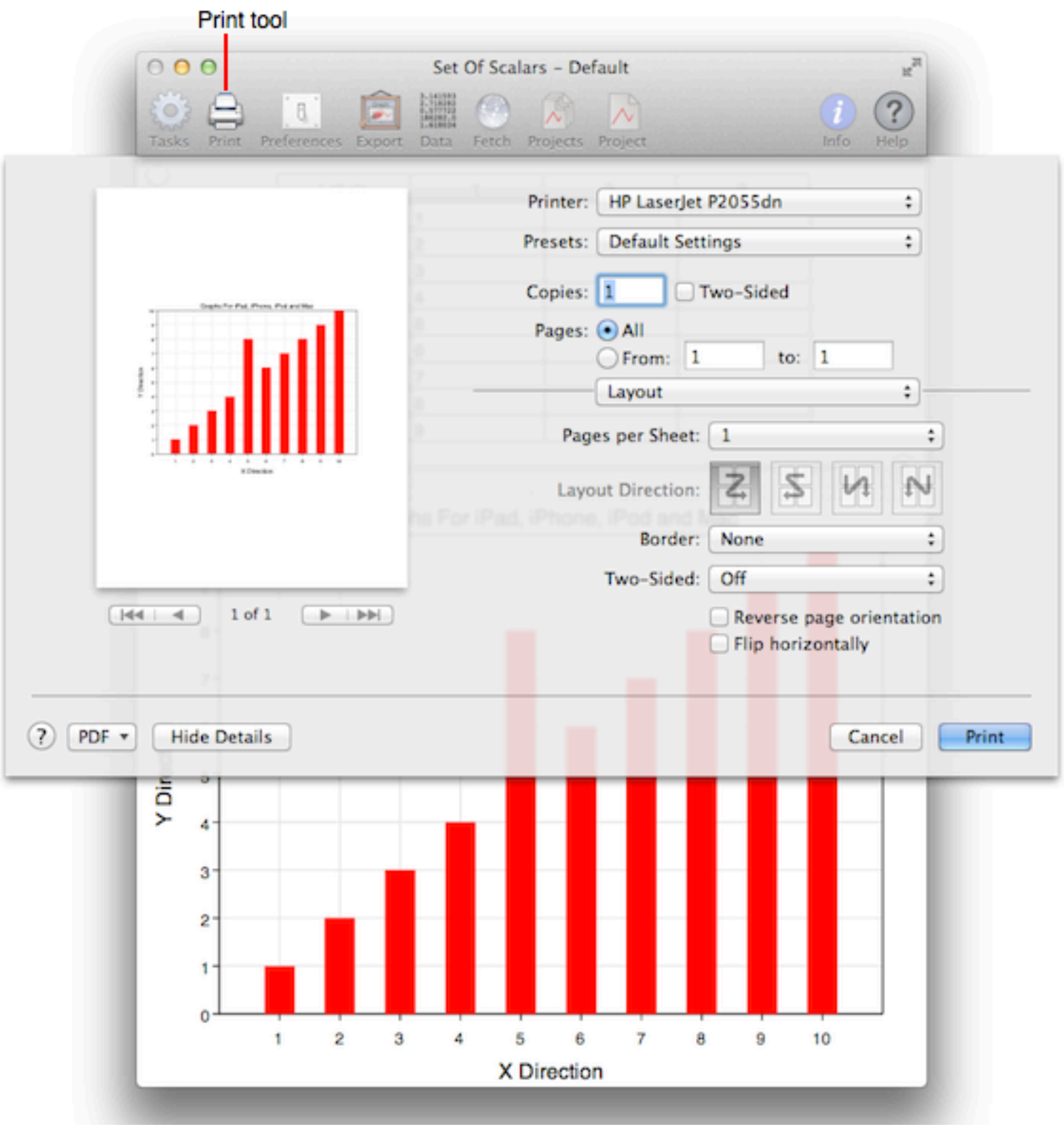
The task tool is where it all begins. Click the tool and select the type of task you are interested in. The figure below shows the task sheet.



Each task is explained in the Tasks section.

Graph > Tools > Print

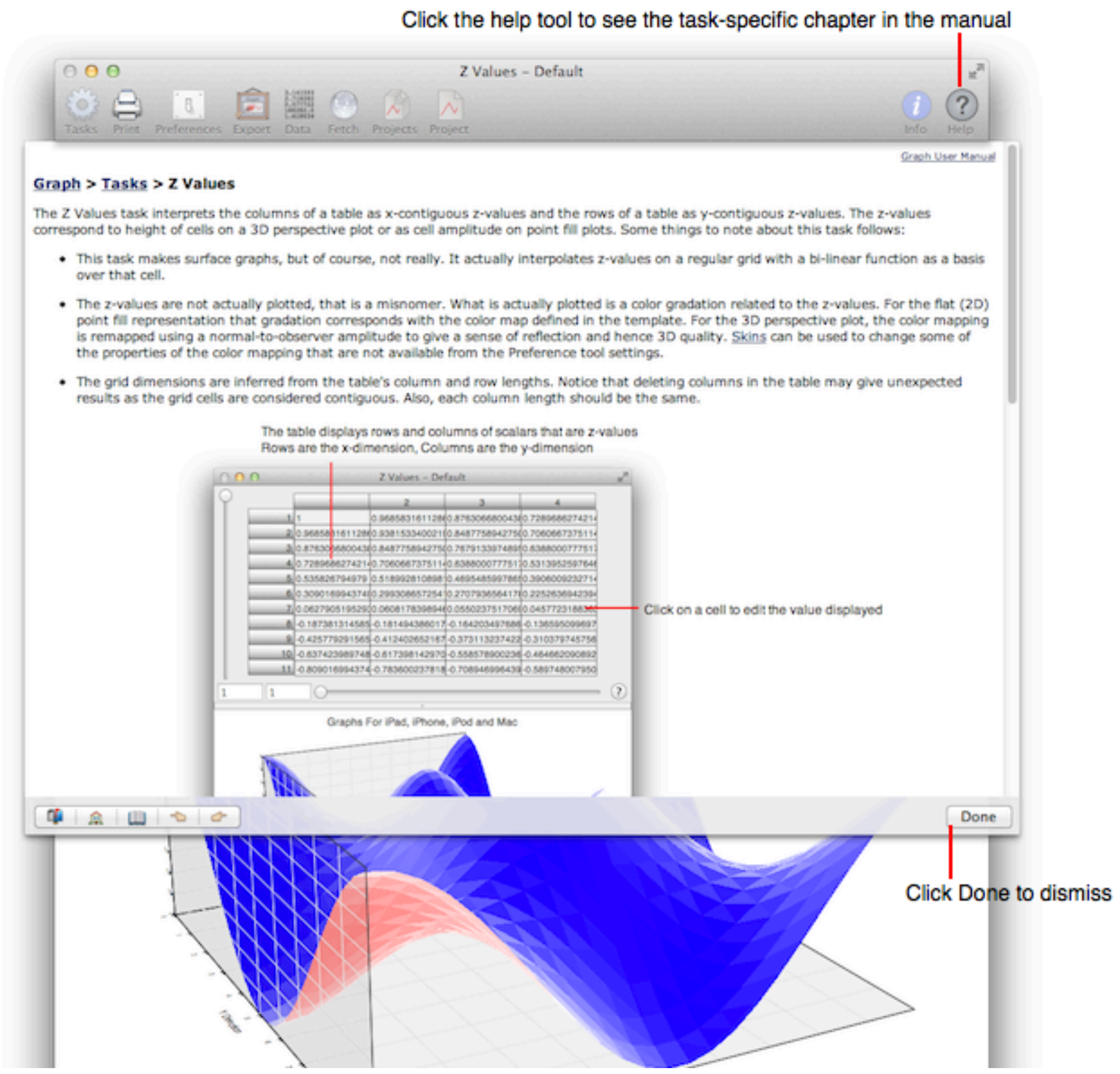
The print tool, shown below, brings forward a print sheet loaded with the current graph paginated to fit the print page. If you need more control over printing then Export to Vvidget Builder first.



Printer: HP LaserJet P2055dn
Presets: Default Settings
Copies: 1 Two-Sided
Pages: ● All ○ From: 1 to: 1
Pages per Sheet: 1
Border: None
Two-Sided: Off
Reverse page orientation
Flip horizontally

Graph > Tools > Help

Help is built into each task. To access it click the Help tool as shown below.



Graph > Tasks > Z Values

The Z Values task interprets the columns of a table as x-contiguous z-values and the rows of a table as y-contiguous z-values. The z-values correspond to height of cells on a 3D perspective plot or as cell amplitude on point fill plots. Some things to note about this task follows:

- This task makes surface graphs, but of course, not really. It actually interpolates z-values on a regular grid with a bi-linear function as a basis over that cell.
- The z-values are not actually plotted, that is a misnomer. What is actually plotted is a color gradation related to the z-values. For the flat (2D) point fill representation that gradation corresponds with the color map defined in the template. For the 3D perspective plot, the color mapping is remapped using a normal-to-observer amplitude to give a sense of reflection and hence 3D quality. Skins can be used to change some of the properties of the color mapping that are not available from the Preference tool settings.
- The grid dimensions are inferred from the table's column and row lengths. Notice that deleting columns in the table may give unexpected results as the grid cells are considered contiguous. Also, each column length should be the same.

The table displays rows and columns of scalars that are z-values
Rows are the x-dimension, Columns are the y-dimension

Z Values – Default

		2	3	4
1	1	0.9685831611286	0.8763066800438	0.7289686274214
2	0.9685831611286	0.9381533400211	0.8487758942750	0.7060667375114
3	0.8763066800438	0.8487758942750	0.7679133974895	0.6388000777513
4	0.7289686274214	0.7060667375114	0.6388000777513	0.5313952597646
5	0.535826794979	0.5189928108698	0.4695485997866	0.3906009232714
6	0.3090169943749	0.2993086657254	0.2707936564176	0.2252636942394
7	0.0627905195293	0.0608178398946	0.0550237517068	0.045772318836
8	-0.187381314585	-0.181494386017	-0.164203497686	-0.136595099697
9	-0.425779291565	-0.412402652167	-0.373113237422	-0.310379745756
10	-0.637423989748	-0.617398142970	-0.558578900236	-0.464662090892
11	-0.809016994374	-0.783600237818	-0.708946996439	-0.589748007950

Click on a cell to edit the value displayed

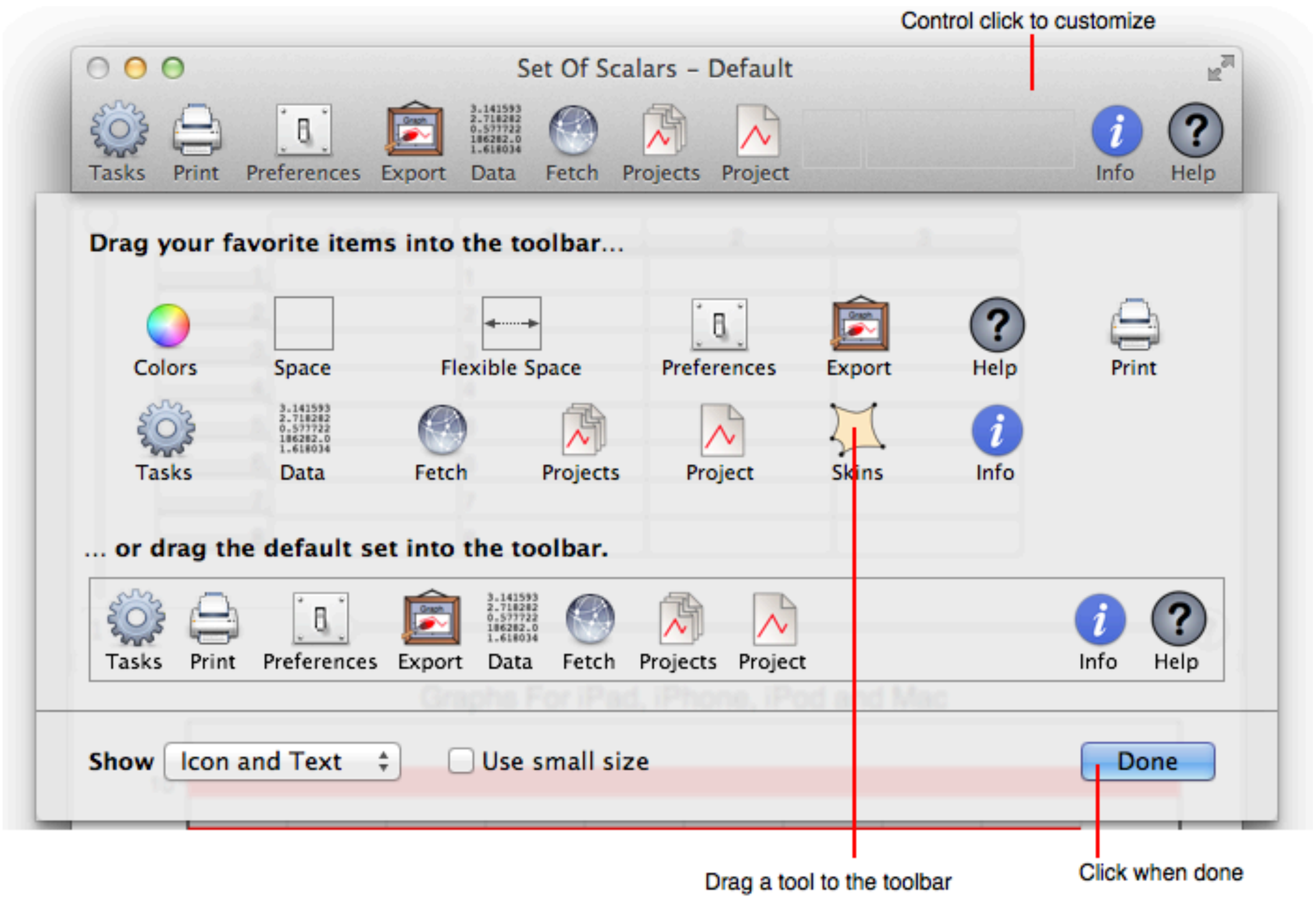
Graphs For iPad, iPhone, iPod and Mac

The Help tool actually brings up this manual with the section specific to the current task. Once the manual is brought forward then you can navigate it using the usual hyperlinks and buttons.

In addition please email support@vvi.com for help.

Graph > Tools > Customize

In keeping with the focused nature of Graph, you can customize the toolbar to add and remove tools you want access to. The figure below diagrams the customize tool.



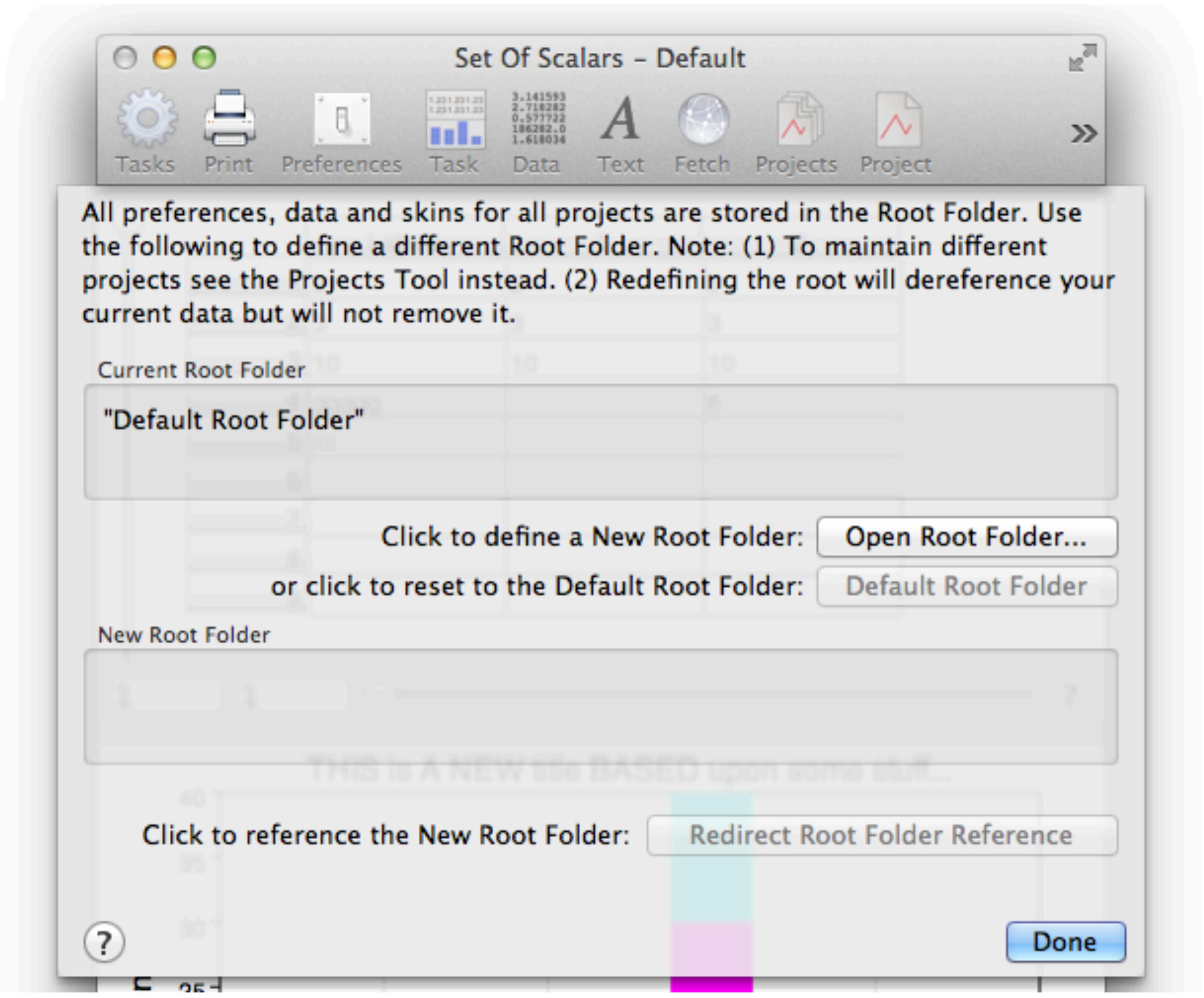
Each icon in the sheet refers to Tools, a specific tool, that operates upon the current task and project and permits settings upon the current task and project.

Graph > Tools > Root Folder

The information for all projects is stored in a folder called the "Root Folder". The Root Folder tool is an infrequently used tool to redirect data storage for all projects. If, instead, you wish to work with multiple data sets (projects) then use the Projects tool. The following are some reasons to use the Root Folder tool:

- You can redirect data storage to a folder that you can see in the Finder. That way you can archive it (zip it), send it to others, share it amongst different computers, define it to be located at a network mount point, etc.
- The Root Folder can potentially be used by different applications. But, in a sandbox environment, you must dereference the Root Folder for one application so that you can then instruct another application to use the Root Folder.
- The Root Folder tool does not move your data and it does not delete your data. It merely redirects (points to) a different location (a Folder) on your disk to store information. You can have multiple Root Folders, but it is up to you to keep track of where the Root Folders are and only one Root Folder can be used at a time.
- The default location for the Root Folder is in the application's Container which is a somewhat obscure location on your disk that you might accidentally caused to be removed if you remove the application. By defining a new and well known and seen Root Folder you can explicitly make it more noticeable and disassociate it with an application.

The Root Folder tool is accessed via the application menu item (`File > Root Folder...`). The figure below diagrams the Root Folder tool.



All preferences, data and skins for all projects are stored in the Root Folder. Use the following to define a different Root Folder. Note: (1) To maintain different projects see the Projects Tool instead. (2) Redefining the root will dereference your current data but will not remove it.

Current Root Folder

"Default Root Folder"

Click to define a New Root Folder: Open Root Folder...

or click to reset to the Default Root Folder: Default Root Folder

New Root Folder

Click to reference the New Root Folder: Redirect Root Folder Reference

Done

Use the Root Folder tool in the following ways.

- Click the "Open Root Folder..." button and use the resulting Open Panel to select a new folder. Once done, click the "Redirect Root Folder Reference" button.
- Click the "Default Root Folder" button to use the application's Container Root Folder then click the "Redirect Root Folder Reference" button to complete the redirection.

When you redirect the Root Folder then all your previous data appears to be lost. That is because it is dereferenced. To retrieve a reference to previous data use the Root Folder tool to direct the Root Folder to the previous Root Folder. Remember: The Root Folder tool does not delete or move data, it simply defines a reference to where data is stored and retrieved.

Graph > Tutorials

The following is a brief list of tutorial sections:

Tutorials	Description
Enter Data	Describes various ways to enter data.
XML Fetch	Describes XML Fetching and schema.
Making A Map	Describes How to make a map for the Map task.

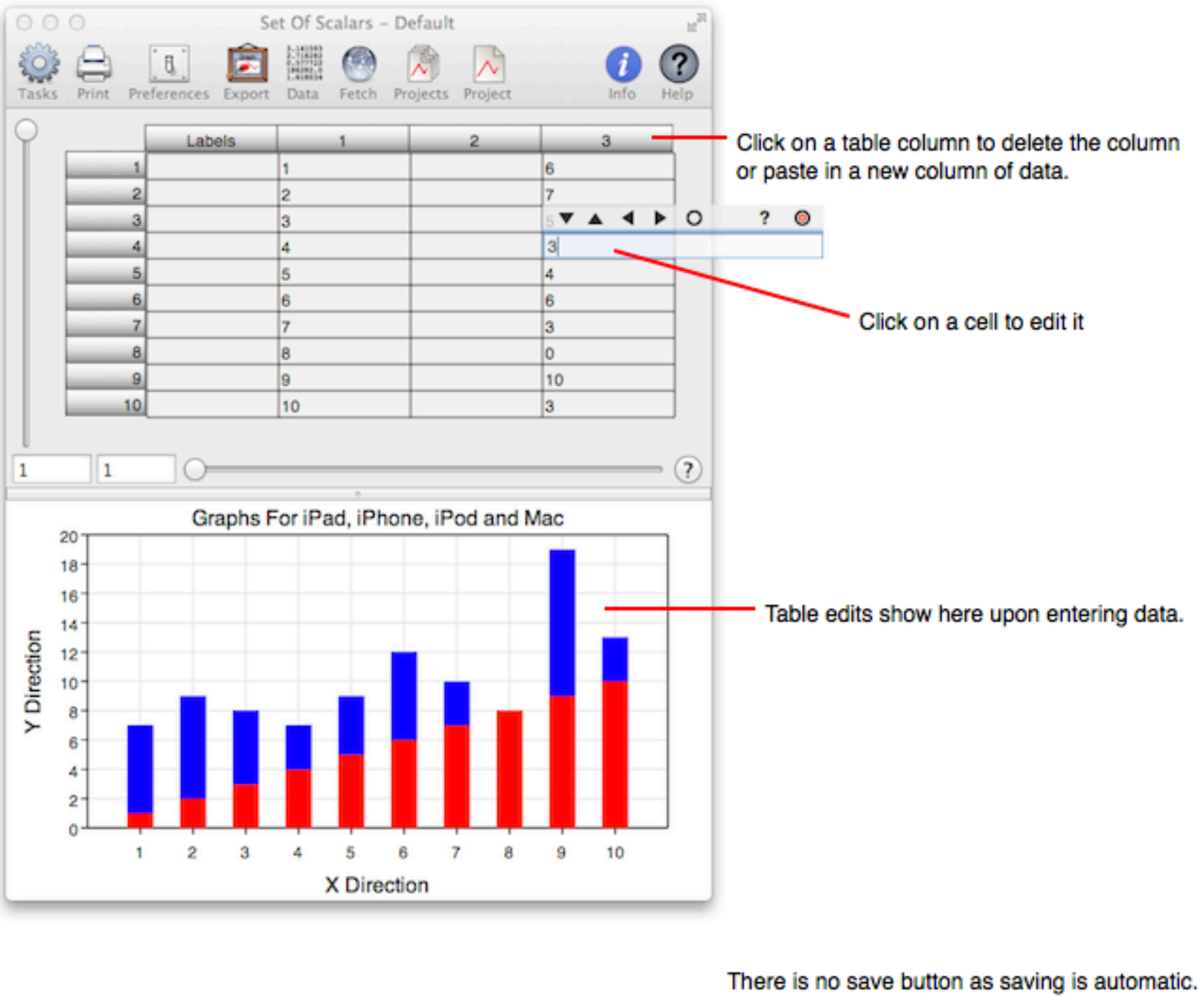
If you have a question that is not explained in this manual please contact support@vvi.com so that we may answer your question and update this manual as needed.

Graph > Tutorials > Enter Data

There are two main ways to enter data which are explained in this section.

With the Tables interface you can select a row, a column or a cell and paste data. You can also select the entire table and paste in data. The data format is defined by the task type and is specified in the particular task section.

By way of example, the figure below shows a column chart table being edited. The cells are formatted as single numbers (scalars). The figure shows the cell editor when clicked on the fourth column (the third scalar column), fourth row. Clicking the Return key dismisses the editor and updates the graph. The data is automatically saved. There is so little to do that this tutorial is short and mostly a referral to other sections of this manual.


Set Of Scalars – Default
Tasks
Print
Preferences
Export
Data
Fetch
Projects
Project
Info
Help
Click on a table column to delete the column or paste in a new column of data.
Click on a cell to edit it
Graphs For iPad, iPhone, iPod and Mac
Y Direction
X Direction
Table edits show here upon entering data.
There is no save button as saving is automatic.

	Labels	1	2	3
1		1		6
2		2		7
3		3		5
4		4		3
5		5		4
6		6		6
7		7		3
8		8		0
9		9		10
10		10		3

With the Fetch tool you can acquire data from a file or a web server (SOA facility). Fetching occurs one column at a time and the fetched URL contents are the same as pasted column data. In the case of a column chart, a list of numbers delimited by a space. Here are a few ideas when using Fetching:

- The URL may refer to a static file, which is easy enough. However, it may also refer to a CGI facility on a web server in which case the column number can be encoded into the URL and parsed by the CGI facility in the normal way.
- You can easily turn on web sharing on your Mac and drop in a PERL or PHP script to program the data content. That way the data access refers to an algorithm and not specifically numerical data, although ultimately numeric data is the result of the fetch.
- Of course, the URL facilities can use any SOA service. See XML Fetch for more information.

The ultimate facility for entering data is your own program. For that see the Vvidget Code Reference Manual.

Graph > Tutorials > XML Fetch

The Fetch section shows how to fetch content from a URL. The Info section shows how to view the key value dictionary description of a graph. This Tutorial amalgamates those sections and describes how to fetch XML content which defines the task state.

Line Graph

The following is the XML needed to make a line graph (curves). Line graphs are made with the Set Of 2D Points task so first select that task from the Tasks tool, click the Fetch tool, make sure the component pop up button is set to "All" and enter the URL to the XML content shown below. Then click the "Fetch Data" button.

Available at this URL: linegraph.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>title</key>
<string>Set Of 2D Points XML Test</string>
<key>x_title</key>
<string>XML-X-Title</string>
<key>y_title</key>
<string>XML-Y-Title</string>
<key>data_3</key>
<string>1 15 2 16 3 17 4 18 5 5</string>
<key>data_2</key>
<string>1 2 2 26 3 2 4 28 5 2</string>
<key>data_1</key>
<string>1 31 2 36 3 32 4 33 5 41</string>
</dict>
</plist>
```

Once the fetch is confirmed to work then you may wish to turn on "Each Animation Step" and "At First Appearance", click "Done", click the Data tool and turn on Animate then click "Done". The XML will be fetched and its results will be displayed every second. As such, when you change the content of the XML then the graph and table will change to reflect the changes to the XML. If the URL points to a SOA service then the XML it serves will be polled and displayed every second.

Now you have some XML and know how to fetch it, but perhaps you don't know what XML exactly is so lets hit some bullets on that issue:

- The type of XML shown above is called a "Property List". It is a pretty well documented format on the Apple system so I won't describe it too much. Note that if you install the developer toolset then there is a Property List Editor application that can help you view and edit property list type XML. But, this XML is so simple that it is probably best to use a text editor. It is pretty obvious to tell that the content has a header, footer and key value pairs that are delimited by HTML type tags where a key is delimited by the <key></key> pair and a value is delimited by one of <string></string>, <integer></integer>, <float></float> pair.
- The keys and values in the XML are gathered in the Info tool and explained in the Vvidget Code Reference Manual. For the most part, you can simply see a graph you want and then view the Info tool table to determine the keys and values to use in the XML.
- Titles values are <string> type, data arrays are also <string> type as the numeric values are not atomic, which saves a lot on using repetitive tags. Lengths are generally of type <integer> and numbers are of type <float>. However, since XML is simply a delimited string you can also use <string> type instead of <integer> and <float> type.
- Of course, this is not a tutorial on HTML or XML so I didn't explain the DOCTYPE, enclosing plist and dict types and other common knowledge items. Suffice it to say that if you don't know of such things then simply do what the pros do: Copy the XML above and only change the keys and values you need.

Column Chart

The following is the XML needed to make a column chart. Column charts are made with the Set Of Scalars task so first select that task from the Tasks tool, click the Fetch tool, make sure the component pop up button is set to "All" and enter the URL to the XML content shown below. Then click the "Fetch Data" button.

Remember to use the Preferences tool to select the Column representation. You can also make stacked and offset bar and column charts and pie charts by using different representations and key value pairs.

Available at this URL: columnchart.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>title</key>
<string>Set Of Scalars XML Test</string>
<key>x_title</key>
<string>XML-X-Title</string>
<key>y_title</key>
<string>XML-Y-Title</string>
<key>data_3</key>
<string>1 2 3 4</string>
<key>data_2</key>
```

```
<string>1 2 3 4</string>
<key>data_1</key>
<string>2 2 3 6</string>
<key>label_1</key>
<string>a kitty cat</string>
<key>label_2</key>
<string>dog eat dog</string>
<key>label_3</key>
<string>buddy</string>
<key>label_4</key>
<string>more</string>
<key>label_5</key>
<string>the sky</string>
</dict>
</plist>
```

Surface Graph

The following is the XML needed to make a surface graph. Surface graphs are made with the Z Values task so first select that task from the Tasks tool, click the Fetch tool, enter the URL to the XML content shown below. Then click the "Fetch Data" button.

Available at this URL: surfacegraph.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>title</key>
<string>Z2 Values XML Test</string>
<key>x_title</key>
<string>XML-X-Title</string>
<key>data_values</key>
<string>1 1 1 2 3 2 0 1 0 5 5 3</string>
<key>grid_x_length</key>
<integer>4</integer>
<key>grid_y_length</key>
<integer>3</integer>
</dict>
</plist>
```

3D Density Graph

The following is the XML needed to make a 3D density graph. 3D density graphs are made with the Density task so first select that task from the Tasks tool, click the Fetch tool, enter the URL to the XML content shown below. Then click the "Fetch Data" button.

Available at this URL: densitygraph.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>title</key>
<string>Density XML Test</string>
<key>x_title</key>
<string>XML-X-Title</string>
<key>data_values</key>
<string>0.2 0.2 0.1 0.2 0.2 0.2 0.2 0.2 0.2 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.3 0.6 0.6 0.6 0.6 0.6 0.6 0.6 0.6
0.6</string>
<key>grid_x_length</key>
<integer>3</integer>
<key>grid_y_length</key>
<integer>3</integer>
<key>grid_z_length</key>
<integer>3</integer>
</dict>
</plist>
```

Least Squares

The following is the XML needed for least squares. Least Squares graphs are made with the Least Squares task so first select that task from the Tasks tool, click the Fetch tool, enter the URL to the XML content shown below. Then click the "Fetch Data" button.

Available at this URL: leastsquares.plist

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
<key>title</key>
<string></string>
<key>x_title</key>
<string>XML-X-Title</string>
```

```
<key>y_title</key>
<string>XML-Y-Title</string>
<key>data_values</key>
<string>1 31 2 36 3 32 4 33 5 25</string>
</dict>
</plist>
```

Least Squares is unusual because the Info tool shows <key>data_1</key> and <key>data_2</key> as data keys, but the XML uses the <key>data_values</key> key. That is an intricate implementation detail of this task that I won't explain, but mention so that you know not to totally rely upon the Info tool for the Least Squares task.

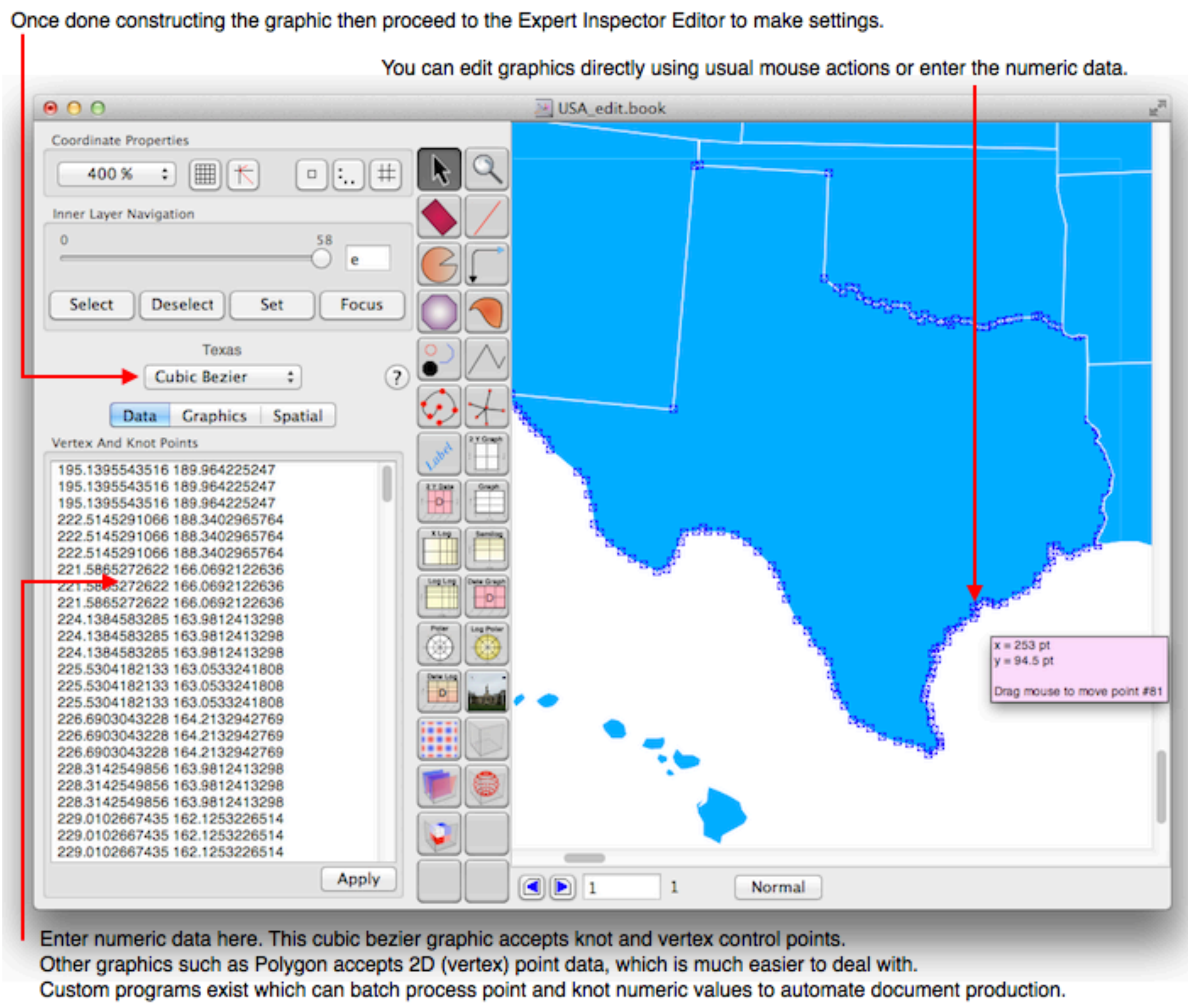
Conclusion

There are many graph types that can be updated with XML and those graphs can be accessed using the various tasks and their representations. With a little experimentation XML fetching may solve a lot of problems.

Graph > Tutorials > Making A Map

This section explains how make a document from a user perspective. The type of document is called a "Map" because it can be loaded into other applications and its components can be queried upon. Making a map is described first and loading the map into an existing application is described next.

First make a map using point and click methods of Vvidget Builder as demonstrated in the following diagram. Vvidget Builder has its own extensive manual, consult that manual for additional information.



Once done constructing the graphic then proceed to the Expert Inspector Editor to make settings.

You can edit graphics directly using usual mouse actions or enter the numeric data.

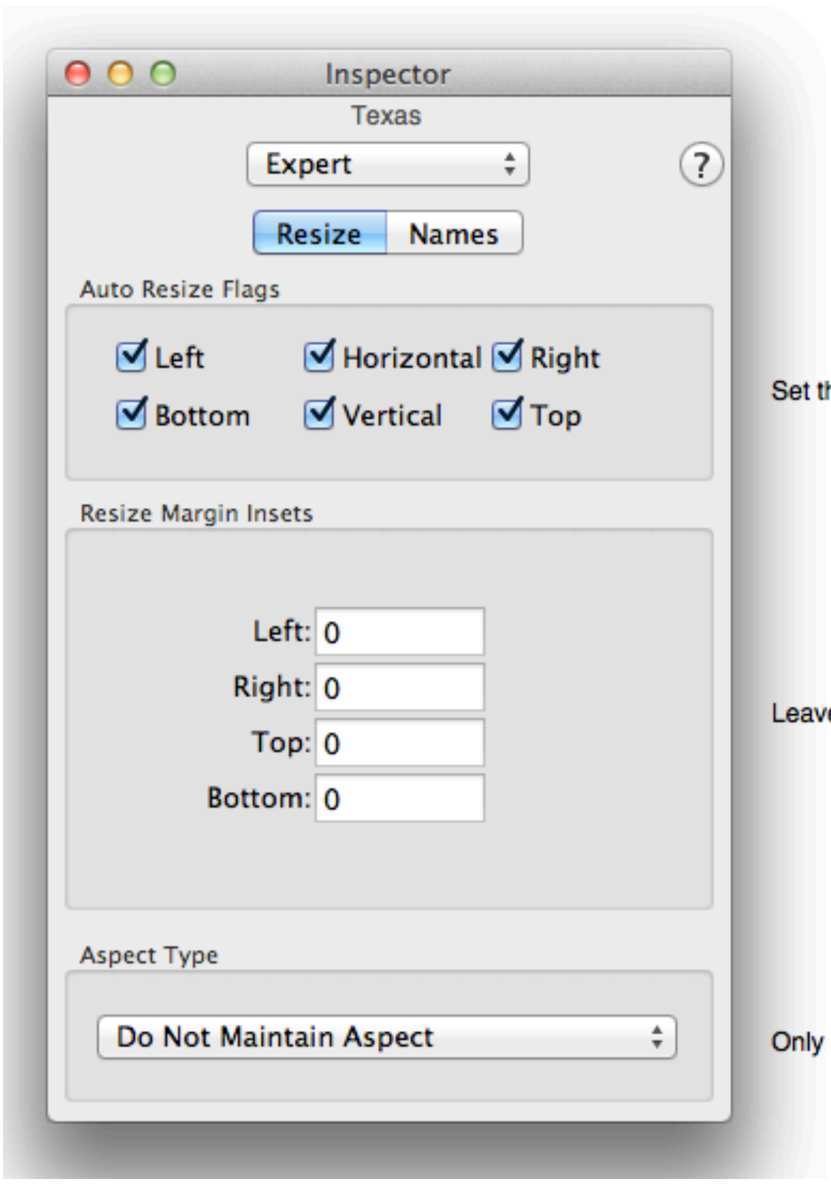
```
195.1395543516 189.964225247
195.1395543516 189.964225247
195.1395543516 189.964225247
222.5145291066 188.3402965764
222.5145291066 188.3402965764
222.5145291066 188.3402965764
221.5865272622 166.0692122636
221.5865272622 166.0692122636
221.5865272622 166.0692122636
224.1384583285 163.9812413298
224.1384583285 163.9812413298
224.1384583285 163.9812413298
225.5304182133 163.0533241808
225.5304182133 163.0533241808
225.5304182133 163.0533241808
226.6903043228 164.2132942769
226.6903043228 164.2132942769
226.6903043228 164.2132942769
228.3142549856 163.9812413298
228.3142549856 163.9812413298
228.3142549856 163.9812413298
229.0102667435 162.1253226514
229.0102667435 162.1253226514
229.0102667435 162.1253226514
```

x = 253 pt
y = 94.5 pt
Drag mouse to move point #81

Enter numeric data here. This cubic bezier graphic accepts knot and vertex control points.
Other graphics such as Polygon accepts 2D (vertex) point data, which is much easier to deal with.
Custom programs exist which can batch process point and knot numeric values to automate document production.

The diagram above shows how the map of the United States of America (USA) was constructed. That map, called USA, is available in the Graph application. It was made by point and click methods building up each state individually and then abutting the states to form the entire USA. Here is a hint: scan an image, drag out the Image graphic, load your scanned image into it, trace over that image with other graphics and then if desired delete the original scanned image from the document leaving only the live graphical components. Geographic maps are challenging whereas process maps, control consoles and flow charts can be very easy, being rectangles and circles connected by lines or simply connected by each other. The creation of your own map is a matter of ingenuity since there are many ways to make maps. If you need to bulk process node data into maps then please contact support@vvi.com so we can support your efforts. We have specialized tools to process point information into documents and hence maps.

Once the map is constructed then select each element and assign the resize flags as shown in the following diagram.


Inspector
Texas
Expert
Resize | Names

Auto Resize Flags
- [✓] Left
- [✓] Horizontal
- [✓] Right
- [✓] Bottom
- [✓] Vertical
- [✓] Top

Set the Auto Resize Flags accordingly.

Resize Margin Insets

Left: 0
Right: 0
Top: 0
Bottom: 0

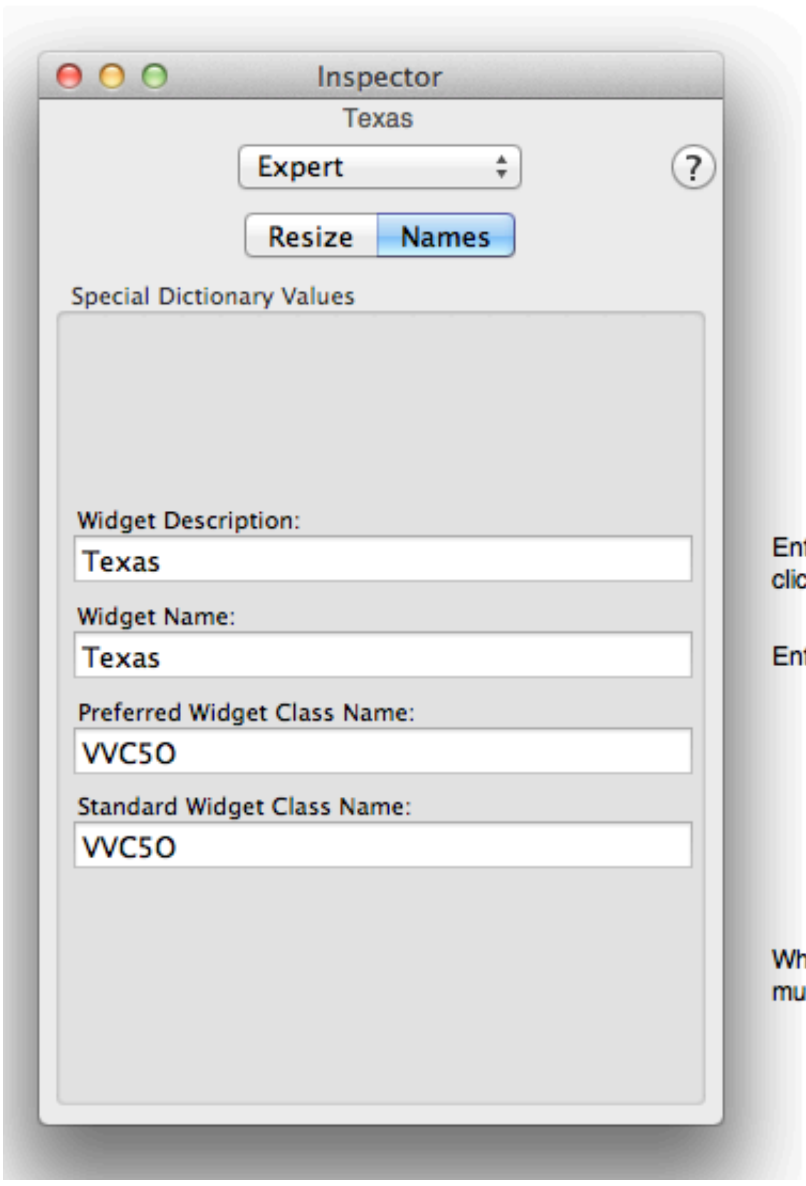
Leave all the Margins zero unless you intend otherwise.

Aspect Type

Do Not Maintain Aspect

Only maintain aspect if you need that feature.

and then assign the element name and description.


Inspector
Texas
Expert
Resize | Names

Special Dictionary Values

Widget Description: Texas

Enter the "Description", which shows up when you click or touch the component (in this case Texas).

Widget Name: Texas

Enter the "Name". The name can be anything for a component.

Preferred Widget Class Name: VVC50

Standard Widget Class Name: VVC50

When you group the components then the group name must be PVS_group.

When all the attributes described above are set then select all the elements (in this case the states) and group them (using the menu item

Format > Group and choosing a normal group) and give the resulting group the name PVS_group (very important). You should probably set the group to autoresize in width and height and to maintain aspect.

To load your newly created document into the Graph application you will need to make it "Internet ready". That means you will have to take the normal Vvidget Builder document and export it as a skin document using the menu item File > Export To ... and choosing the Skin export type. It is that skin file, which is a compressed flat file that gets imported into the Graph application. Place that skin file on a web server or on your disk and import it into Graph this way: choose the Maps task, in the preference tool select a map entry other than USA for example the Map 1 element or an element that you have renamed, choose the Skin tool, enter the URL (file or http) of the skin file, click the Fetch Primary Skin button and then finally click the Done button. Your skin shows in the resulting view.

In summary, the steps to making a map are:

- Make a Map document.
- Export that document to a Skin file.
- Place that Skin file on a web server or on your computer's file system.
- Import that Skin file into the Graph application or similar application that has Vvidget Builder Skin importing features.

Making a simple map can take a few minutes, making a hard map can take days, batch processing thousands of maps from existing map data can take a few CPU hours. Once you have a map and wish to do more with it than simply query component names then consult the Document section to program your own application. For example, a process map can be used to click a process element, for example schematic element representing a valve, to then issue commands to that valve to turn it on and off. Another example: Click a state or other jurisdiction such as sales territory to then call the sales representative for that territory. The possibilities with maps are very interesting.

Graph > Support

The following is a brief list of support sections:

Support	Description
Legend	Describes how to make a legend for a graph.
Date Graph	Describes date entry for a date graph.
Graph For iOS	A reference to the Graph User Manual For iOS (iPhone, iPad and iPod touch).
Programming	Gives a reference on how to make your own chart task.
Data Backstore	Describes the backstore and how to backup it up.
Question Answer	A brief list of question answers on subjects that are not covered elsewhere in this manual, or not obvious to find.

If you have a question that is not explained in this manual please contact support@vvi.com so that we may answer your question and update this manual as needed.

Graph > Support > Programming

The Fetch tool provides some capability to program data generation. Combined with the Data tool programming data can be animated. However, for true programming you may be interested in the Vvidget Code Reference Manual.

Graph > Support > Legend

A legend is a graph key that associates a color with a label describing each data graphic (curve, bar, etc.) The Preferences tool is used to set the position of a legend and Tables editing is used to set the labels for legend.

Follow these steps to make a legend:

- Alt-mouse-click on each table column header (the grey portion at the top) to edit the column description. The text you enter will also be entered into the label of the legend.
- Set the position of the legend using the Preference tool.

If you need more control over the legend then consider buying Vvidget Builder. Within Vvidget Builder legends are group graphics and hence can be altered using the powerful capabilities of Vvidget Builder.

Graph > Support > Date Graph

The Set Of 2D Points task supports date entry for the x-value. If you enter x-values in this format: MM/DD/YYYY hh:mm:ss.f where:

Date Fields

MM	Months, a number from 1 to 12
DD	Days, a number from 1 to 31
YYYY	Years, a four digit number
hh	Hours, a number from 0 to 24
mm	Minutes, a number from 0 to 60
ss	Seconds, a number from 0 to 60
f	Fraction of second, an integer

then the x-value is interpreted as a Gregorian date. The MM/DD/YYYY part is mandatory while the hh:mm:ss.f is optional. Here are some date entry examples:

Date	Explanation
12/1/2010	A valid calendar date
12/1/2010 8:0:0	A valid calendar date and time
12/1/2010 8	An invalid time, the time part must contain two colons and three numbers.
12/1/99	An invalid date, the year part must contain four decimals indicating the century.

The following are a few issues when working with dates:

- The date is simply the x-value and must be followed by a y-value scalar in order to complete the 2D point entry.
- Only the Gregorian calendar is supported and the month must be the first number in the date.
- Data entry supports date or scalar representation for the x-value but not both at the same time. For table cell entry, if you enter a date for a x-value then all x-values are then interpreted as dates. If you enter a scalar for a x-value then all x-values are then interpreted as scalars in which case any previously entered dates are shown as seconds since 1/1/1970.
- If you paste in a series of dates then each date y-value entry must be on one line only, i.e.: a delimiter of a return character must separate each 2D point. In this case, date entry is determined by looking for a slash (/) in the first 20 characters of the pasteboard string. If a slash is found then all subsequent x-values are assumed to be in date format.
- In order to appear on all the coordinate systems, the date is remapped to a second since 1/1/1970 as an intrinsic basis, hence dates can also be entered as seconds since January 1, 1970.
- Dates are most important for the Curve > Date graph where the x-axis is shown in Gregorian calendar format and scaled to any of the fields specified in the table above. For example, if your x-values span months then the x-axis will show increments in months, if your x-values span years then the x-axis will span years, if your x-values span seconds then the x-axis will span seconds, and so on and so on. This autoscale feature and the resulting x-axis label format is rather complex and can be altered with the use of Skins. Also note that the x-axis tick increments are uniform in date field, but not uniform in the second basis hence the ticks may appear non-uniformly spaced. A good date graph is particularly complex.

Graph > Support > Data Backstore

A data backstore is the area on your file system where Graph maintains all of its persistent information so that the next time you launch it your data and preferences are available to you. Normally you need not be concerned about the data backstore but there are a few instances where it is handy to know about the data backstore.

The default backstore is at the location:

```
Library/VVI/Graph
```

in your home folder or the application container if it is sandboxed. The reference to the backstore location is altered by using the Root Folder tool. The following lists some concepts regarding the data backstore.

Backup

If you feel compelled to backup the information you enter into a task then simply zip up the backstore location and rename the zipped file. Notice that if you backup your home folder then the data backstore is backed up along with everything else.

Manual Reacquisition

Rarely the backstore location is changed. With version 10.6.10 the backstore location changed. To reacquire your settings from a previous version please do this:

```
Launch the Terminal application.

cd ~/Library
mkdir VVI
cd VVI
mv ../com.vvi.GraphMac .
mv com.vvi.GraphMac Graph
```

Transport

To transport your task information from one computer to another zip up the data backstore, copy it to the new computer and then unzip it at the same location as on the original computer.

Manual Reset

Sometimes you may simply want to start anew. To do that simply drag the data backstore to the trash. The next time you launch Graph the backstore will be reinitialized and available for new task entries.

Graph > Support > Question Answer

Below are answers to commonly asked questions about Graph. If you have a question please mail support@vvi.com.

Question: How do I program (automate) graph generation?

- **Answer:** See the section Programming.

Question: Is there a running list of bugs?

- **Answer:** No. Instead of trying to figure out if you encountered a bug please email support@vvi.com so we can fix it either by documenting your misinterpreted issue or fixing the bug. Hopefully, effort can be applied to fix such issues expediently instead of maintaining a running list of bugs which soon becomes obsolete.

Question: What is the difference between the Vvidget application and the Graph application?

- **Answer:** The Graph application is currently free and has some informative ads regarding VVI services and different features available to you. The Vvidget application is the same as the Graph application but without ads and currently is not free.

Question: What is the difference between the Vvidget Builder application and the Graph and Vvidget applications?

- **Answer:** The Graph and Vvidget application are duplicated in the Vvidget Builder's Chart Task. In addition, Vvidget Builder includes a complete system to graphically layout graphs without data and also to embed data onto those graphs. Vvidget Builder is more complex and feature laden whereas the Graph and Vvidget applications are simple and direct. See the Vvidget Builder User Manual for more information.

Question: What is the difference between the General Release Edition and the Mac App Store Edition?

- **Answer:** The Mac App Store Edition is that which is available on the Mac App Store whereas the General Release Edition is that which is available at the Vvidget web site (see download). The General Release Edition includes everything in the Mac App Store Edition plus some other features such as the Export tool.

Graph > Administrative

The following is a brief list of Administrative sections:

Administrative	Description
License Agreement	The Graph End User License Agreement.
Trademarks	A list of trademarks used in this manual.

Below is a copy of the End User License Agreement that you received with your copy of Graph and was presented to you before installing Graph.

Graph v10.7.8

END-USER LICENSE AGREEMENT FOR VVI SOFTWARE

IMPORTANT-READ CAREFULLY: This End-User License Agreement ("Agreement") is a legal agreement between you (either an individual or a single entity) and VVimaging Corporation (VVI) for the VVI software product(s) identified above which may include associated software components, media, printed materials, and "online" or electronic documentation ("SOFTWARE PRODUCT"). By installing, copying, or otherwise using the SOFTWARE PRODUCT, you agree to be bound by the terms of this Agreement. If you do not agree to the terms of this Agreement, do not install or use the SOFTWARE PRODUCT. If the SOFTWARE PRODUCT was purchased by you, you may return it unopened to your place of purchase for a full refund.

The enclosed copy of the SOFTWARE PRODUCT is **never sold**. It is licensed by VVI to the original customer for his or her use only under the terms of this license agreement which follows:

- **1. SCOPE OF LICENSE**
  - This Agreement governs the use of the SOFTWARE PRODUCT and user documentation in printed and electronic forms (the "SOFTWARE PRODUCT"). The SOFTWARE PRODUCT also includes, and this Agreement also governs, later releases, IF ANY, of the SOFTWARE PRODUCT which VVI distributes without additional charge to licensees of your release of the SOFTWARE PRODUCT.
- **2. RESTRICTED USE**
  - STUDENT USE: If you licensed the SOFTWARE PRODUCT for student use you represent to VVI that you are a current member of an accredited educational institution's student body. ("Student User"). Student Users are licensed to use the SOFTWARE PRODUCT in accordance with this Agreement, and solely for the purposes directly related to satisfying the requirements of degree-granting programs ("Student Purposes"). Student Use is use of the SOFTWARE PRODUCT by Student Users and for Student Purposes only. Any use of the SOFTWARE PRODUCT for other than Student Use is expressly prohibited.
  - DEMONSTRATION USE: If you licensed the SOFTWARE PRODUCT for demonstration use then you may only use the SOFTWARE PRODUCT for Demonstration Purposes. Demonstration Purposes shall mean use of the SOFTWARE PRODUCT for the sole limited purpose of verify that the SOFTWARE PRODUCT performs in accordance with manufacture's representations as documented in the SOFTWARE PRODUCT online manuals. Demonstration Purposes shall not mean use for commercial, official university, government laboratory purposes, or any use other than Demonstration Purposes.

  THIS LICENSE WILL TERMINATE AUTOMATICALLY if you fail to comply with the terms and conditions set forth above if such terms are applicable to you.
- **3. LICENSEE's RIGHTS**

  YOU MAY:
  - A. Install and use the SOFTWARE PRODUCT on any computer, as long as it is used only on one computer by one user at a time and in a way which is consistent with this Agreement. If several persons use this SOFTWARE PRODUCT at the same time, or if one person uses it on more than one computer, you must pay one license fee for each copy being used as designated in the Purchase Agreement between you and VVI. You may only use this SOFTWARE PRODUCT on a computer network, if authorized under the Purchase Agreement and if you pay one license fee for each computer and terminal connected to the network.
  - B. Copy the SOFTWARE PRODUCT for back-up purposes only. You may make one (1) copy of the SOFTWARE PRODUCT for back-up purposes. All copies must contain the copyright notice contained in the original copy of the SOFTWARE PRODUCT.
  - C. Transfer the SOFTWARE PRODUCT permanently to another person ONLY IF:
    - (a) that person agrees in writing to accept all of the terms and conditions of this Agreement; and
    - (b) you notify VVI in writing that you have transferred your copy of the SOFTWARE PRODUCT; and
    - (c) you cease all use of the SOFTWARE PRODUCT.
  - D. Terminate this license by destroying the original and all copies of the SOFTWARE PRODUCT in whatever form.
- **4. PROHIBITED ACTIVITIES**

  YOU MAY NOT:
  - A. Loan, rent, lease, give, sublicense or otherwise transfer the SOFTWARE PRODUCT (or any copy), in whole or in part, to any other person.
  - B. Copy or translate the User Manual included with the SOFTWARE PRODUCT.
  - C. Copy, alter, translate, decompile, or reverse engineer the SOFTWARE PRODUCT, including but not limited to, modify the SOFTWARE PRODUCT to make it operate on non-compatible hardware.
  - D. Remove, alter or cause not to be displayed, any copyright notices or startup messages contained in the SOFTWARE PRODUCT or documentation.

THIS LICENSE WILL TERMINATE AUTOMATICALLY if you fail to comply with the terms and conditions set forth above.

- **5. OWNERSHIP**
  - SOFTWARE PRODUCT contains software proprietary to VVI. As a licensee, you own the media on which the SOFTWARE PRODUCT is originally or subsequently recorded, but VVI retains title and ownership to the SOFTWARE PRODUCT recorded on the media and all copyright and other intellectual property rights therein. This license is not a sale of the SOFTWARE PRODUCT or any copy.
  - Operations, features and methodologies of graph and data-oriented graphics by user interface oriented creation, manipulation and editing such as by or in relation with, but not exclusive to, a general purpose drawing system, application or software in any combination of the aforementioned is a trademark and tradedress of VVI with all rights reserved in the United States and all international locations.
- **6. WARRANTY**
  - VVI warrants, to you personally, for a period of ninety (90) days from the date of the Purchase Agreement for SOFTWARE PRODUCT subject to this License Agreement (the "Warranty Period"), that the media containing the SOFTWARE PRODUCT shall be free from defects in material and workmanship under normal use. VVI further warrants, for the Warranty Period, that the SOFTWARE PRODUCT shall operate substantially in accordance with the functional specifications in the accompanying documentation if properly used on a machine for which it was designed. If during the Warranty Period a defect in the SOFTWARE PRODUCT or media appears, you may return the SOFTWARE PRODUCT to your place of purchase for either, at the election of VVI, replacement or refund of the amounts paid by you for the license of the SOFTWARE PRODUCT. You agree that the foregoing constitutes your sole and exclusive remedy for breach by VVI of any warranties made under this Agreement.
- **7. DISCLAIMER**
  - Because it is impossible for VVI to know the purposes for which you acquired this SOFTWARE PRODUCT or the uses to which you will put this SOFTWARE PRODUCT, you assume full responsibility for the selection of the SOFTWARE PRODUCT, and for its installation and use and the results of that use.
  - EXCEPT FOR THE LIMITED WARRANTY SET FORTH IN SECTION 6 ABOVE, YOU AGREE THAT THE SOFTWARE PRODUCT IS FURNISHED ON AN "AS-IS" BASIS. NEITHER VVI NOR ANY THIRD PARTY SUPPLIER MAKES ANY WARRANTIES OF ANY KIND, EITHER EXPRESS OR IMPLIED, AS TO ANY MATTER WHATSOEVER, INCLUDING WITHOUT LIMITATION THE CONDITION, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE OF THE SOFTWARE PRODUCT. SOME STATES DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO THE ABOVE EXCLUSION MAY NOT APPLY TO YOU. IN THE UNITED KINGDOM THE ABOVE DISCLAIMER OF WARRANTIES DOES NOT AFFECT YOUR STATUTORY RIGHTS AS A CONSUMER. NEITHER VVI NOR ANY THIRD PARTY SUPPLIER WARRANTS THAT THE SOFTWARE PRODUCT WILL MEET YOUR REQUIREMENTS, THAT IT WILL OPERATE IN THE COMBINATIONS WHICH YOU MAY SELECT, OR THAT ITS OPERATION WILL BE UNINTERRUPTED OR ERROR FREE. NEITHER VVI NOR ANY THIRD PARTY SUPPLIER ASSUMES ANY LIABILITY REGARDING USE OF, OR ANY DEFECT IN, THE SOFTWARE PRODUCT.
  - VVI is not responsible for problems caused by changes in the operating characteristics of the hardware or operating system software you are using which are made after the release date of this version of the SOFTWARE PRODUCT, nor for problems in the interaction of the SOFTWARE PRODUCT.
- **8. LIMITATION OF LIABILITY**
  - IN NO EVENT SHALL VVI OR THIRD PARTY SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, EVEN IF SUCH PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. Some states do not allow the exclusion or limitation of incidental or consequential damages, so the above limitation or exclusion may not apply to you.
- **9. EXPORT**
  - You agree not to export or re-export the SOFTWARE PRODUCT, or any portion thereof, without the appropriate United States or foreign government licenses.
- **10. SEVERABILITY**
  - If any portion of this Agreement is held invalid or unenforceable for any reason, such invalidity shall not affect the validity of the remaining provisions of this Agreement, and the parties will substitute for the invalid provisions a valid provision which most closely approximates the intent and economic effect of the valid provision.
- **11. TERMINATION**
  - This Agreement shall be effective until terminated by Licensor. Any failure by you to comply with any of the provisions of this Agreement will be a material breach of this Agreement and entitle VVI to terminate this Agreement immediately. Upon termination, you are to immediately stop all use of the SOFTWARE PRODUCT and to return to VVI or erase all copies of the SOFTWARE PRODUCT in any form (including copies contained in any mass storage device).
- **12. ENTIRE AGREEMENT**
  - This Agreement shall constitute the entire agreement between the parties with respect to the subject matter hereof, and supersedes any prior agreements or understandings between the parties, whether written or oral, with respect hereto. No modification to this Agreement shall be of any force or effect unless made in writing signed by each party.
- **13. APPLICABLE LAW; JURISDICTION; VENUE**
  - This Agreement shall be governed and construed in accordance with the laws of the State of Pennsylvania. The parties agree that Centre County in the State of Pennsylvania shall be the proper venue for any action brought under the Agreement whether in state or federal court. You consent to the personal jurisdiction of such courts.

- **14. U. S. GOVERNMENT END USERS**
    - If the SOFTWARE PRODUCT is acquired by or on behalf of a unit or agency of the United States Government, the following provisions apply. The documentation and Software licensed under this Agreement is provided with RESTRICTED RIGHTS and constitute restricted computer software, under the Federal Acquisition Regulations, as set forth below. The SOFTWARE PRODUCT: (a) is existing computer software and, was developed at private expense, (b) is a trade secret of VVI for all purposes of the Freedom of Information Act, (c) is "commercial computer software" subject to limited utilization as expressly stated in this Agreement or as provided in the contract between the vendor and the government entity, (d) in all respects is proprietary data belonging solely to VVI and (e) is unpublished and all rights are reserved under the copyright laws of the United States.
    - The SOFTWARE PRODUCT is licensed only with "Restricted Rights" and use, reproduction or disclosure is subject to restrictions set forth in Alternate III(g)(3) of the Rights in Data - General Clause at 52.227-14 (June 1987) and subparagraphs (a) through (d) of the Commercial Computer Software - Restricted Rights clause at 52.227-19 (June 1987) of the Federal Acquisition Regulations and their respective successors. For units of the Department of Defense (DoD), this SOFTWARE PRODUCT is licensed only with "Restricted Rights" and use, duplication, or disclosure is subject to restrictions as set forth in subdivision (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013 (June 1988) of the DoD Supplement to the Federal Acquisition Regulations and its successors.
    - Contractor/manufacturer is VVimaging, Inc., 311 Adams Avenue, State College, Pennsylvania 16803.

EULA - 10.5.8 - 7/19/2009

Graph > Administrative > Trademarks And Legal

Manipulating graphs and data graphics and other manipulations unique to Vvidget is a Trademark and Tradedress of VVimaging, Inc (VVI) in the United States and world-wide.

VVI is a registered trademark of VVimaging, Inc (VVI). Peer Visual, Vving, Vvidget, OpenGraph, VVI, VVimaging, SAM, "State Automation", "State AutoMation", and Vvidget with any word combination are trademarks of VVimaging, Inc (VVI) in the United States and world-wide.

The OpenGraph logo, Vvidget logo, Vvidget Builder logo, VVI logo are registered trademarks or trademarks of VVimaging, Inc (VVI).

Apple, Power Macintosh, Macintosh, WebObjects are trademarks or registered trademarks of Apple Computer, Inc. Microsoft and Windows are registered trademarks of Microsoft, Inc. All other trademarks and service marks belong to their respective owners.

VVI has tried to make the information contained in this manual as accurate and reliable as possible. Nevertheless, VVI disclaims any warranty of any kind, whether express or implied, as to any matter whatsoever relating to this information, including without limitation the merchantability or fitness for any particular purpose. VVI will from time to time revise the software described in this manual and reserves the right to make such changes without obligation to notify the purchaser. In no event shall VVI be liable for any indirect, special, incidental, or consequential damages arising out of purchase or use of this manual or the information contained herein.
