# FinalCalc.hyper

Khalid Aldoseri

**COLLABORATORS**

| | *TITLE* :<br><br>FinalCalc.hyper | |
|---|---|---|

| *ACTION* | *NAME* | *DATE* | *SIGNATURE* |
|---|---|---|---|
| WRITTEN BY | Khalid Aldoseri | January 17, 2023 | |

**REVISION HISTORY**

| NUMBER | DATE | DESCRIPTION | NAME |
|---|---|---|---|
| | | | |

# Contents

# Chapter 1

# FinalCalc.hyper

## 1.1 Graphs & Animations...

```
 Graphs & Animations:


>
The Graph

>
Viewing, Saving and Printing Graphs

>
Animations

>
Colors, Patterns and Symbols Lists
```

## 1.2 The Graph...

```
 Graphs:


>
What is a graph?

>
The Edit Graph Requester

>
Chart Types

>
Data Points, Series and Sets

>
Special Data Series Definitions
```

## 1.3 The Graph...

A graph shows the values of cells in a project in a visual presentation that makes them easier to understand.

Values from the cells (called data points) are shown as lines, bars, pie slices or other graphic objects in a graph.

You can group values (data points). These are called data series, which can be shown in different colors or patterns in a graph.

Graphs can be shown on their own screen, in a view on the current project, and printed. A graph can also be animated (shown to change in time).

What is a graph?

A graph is a definition of a graphical representation of a set of values (data points) grouped into one or more sets (data series).

A project can have as many graphs as you like. These are called the Graph List

and are managed by the 'Graphs List' requester which is called by the 'Graphs - Graph List' menu.

The Graphs List is a collection of graph definitions. From the 'Graphs List' requester you can view a graph on its own screen, define a new graph, edit a graph, clone a graph or delete a graph.

The 'up', 'down', 'top' and 'bottom' buttons rearrange the order of graphs in the Graphs List. This has no effect on the graphs themselves, just how they are listed in the requester.

To define a new graph use the 'New' button, or to edit a graph use the 'Edit' button. Both of those bring up the 'Edit Graph' requester, which controls everything in a graph.

## 1.4 The Edit Graph Requester

The Edit Graph Requester:
The Edit Graph requester controls all aspects of a graph. It has several sub-requesters that it calls (described below). In it, you define the type of the graph, where it gets its data, its titles, and how it looks like.

The Edit Graph requester has several main areas:

a. Chart Type:
This is a list of different
chart types
. The list contains many types of
charts, and you can select one by clicking on it or by using the keyboard up and down cursor keys to scroll through the list.

b. 3D switch:
The 3D toggle switches between 2-dimensional mode (switch is off) and 3-dimensional mode (switch is on).

In some cases, when you select a chart type that only supports a single mode (not both 2D and 3D modes) this switch will be disabled and you will not be able to change it.

c. Info button:
The
Info
button provides information on the currently selected chart type.

d. Data Definitions:
The data definitions area is a box at the bottom left of the requester. It contains the following entries:

- Data Set 1:
  This defines Data Set 1, which is a collection of one or more
  data series
  .
  When a new graph is defined, this automatically filled in with the current range in the sheet the graph requester was called from.

- Data Set 2:
  This defines Data Set 2, which is needed for some chart types.

- Data Set 3:
  This defines Data Set 3, which is needed for some chart types.

- Labels Data Set:
  This defines one or more data series which select where the graph axis

labels
come from in the project. These are handled differently by each
  chart type.

- Labels Data Set 2:
  This defines additional labels which some chart types require.

If any of the above entries (except for Data Set 1) is not needed by the
currently selected chart type, it will be disabled.

- Default Sheet:
  This defines the default sheet name. (e.g. 'A', 'B', etc.) This defines
  which sheet the graph data should be taken from. For example, if you
  define Sheet as 'B', then all graph data will be taken from sheet B. You
  can still override that by specifiying a full 3D name for a cell or range
  in the data sets or labels.

  When a new graph is defined, the 'Sheet' automatically defaults to the
  current sheet the graph requester was called from.

e. Graph Preview:
The graph preview window is at the top left of the requester. It shows a
quick preview of what the graph will look like using the current settings.
This is a close representation and will sometimes show less
data points
than there are in the graph in order to speed up its display.

You can turn off the preview by turning off both 'Preview Axis' and
'Preview Data', or turn off just the axis preview or the data preview by
turning off on of those two controls.

f. Sub-requesters buttons:
These are a group of buttons at the bottom right of the requester. Each one
calls a requester that controls a certain aspect of the graph:

- Axis:
  Controls how the graph axis looks like.

- 3D Axis:
  Only available for 3D graphs. It controls graph rotation and scaling.

- Limits:
  Defines the numerical limits for each
data set
.

- Colors:
  Controls the color of all graph object (except for data points).

- Titles:
  Defines the titles in the graphs, as well as their font and colors.

- Legneds:
  Defines the
                Legends
                 box.

- Overlays:
  Allows you to add more charts on top of the current one.

- Auto:
  Controls automatic color, pattern and symbol generation in graphs.

g. Title:
A simple title for the graph.  It defaults to something like 'Graph # 1',
but you can edit it to anything you like.

At any time while editing a graph, you can use the 'View' button to get a full screen view of what the graph will look like using the current settings.  Once you are happy with the graph, click on the 'Use' button to store it in the Graphs List.

The Edit Graph requester does not lock you out from the sheet view that called it.  You can go back to the sheet view and do anything you like while the Edit Graph requester is up.  If you try to bring up the 'Graphs List' requester, however, FinalCalc will complain that it cannot access the Graphs List while a graph is being edited.

## 1.5 Chart Types

                Chart Types:
A Chart Type is a method of displaying data in a graph.  Each type has different requirements of data, in number of
                data points
                , number of

                data series
                , and number of
                data sets
                .

A chart type can also support one or two labels sets and support either 2D or 3D rendering, or both.

The following is a quick description of the chart types available:

1.  Bar
  Draws one or more series of data as bars.

2.  Stacked Bar
  Draws multiple series of data as stacked bars.

3. Line
   Draws one or more series of data as lines.

4. Area
   Draws a series of data as a filled areas.

5. Stacked Area
   Draws multiple series of data as stacked filled areas.

6. Pie
   Draws pie charts made up of slices.

7. Spline
   Draws a curve approximation of a data series as a smoothed curve line.

8. High-Low-Open-Close
   Shows the high, low, open and close values for stock charts.

9. Scatter
   Draws a single symbol for each data point.

10. X-Y Line
    Draws an X/Y line chart of two sets of data.

11. X-Y Spline
    Draws a spline approximation of two sets of X/Y data.

12. X-Y Scatter
    Draws a single symbol at each X/Y data point described by two sets of data.

13. X-Y-Z Line
    Draws a line in 3D space described by three sets of data representing
    the X, Y and Z coordinates of each point.

14. X-Y-Z Scatter
    Draws a scatter graph in 3D space, described by three sets of data
    representing the X, Y and Z coordinates of each point.

To get detailed information on a chart type, select the chart type you want in the Edit Graph requester and click on the 'Info' button. You will get the 'About Chart Type' requester, which shows the following information about the chart type:

a. Chart:
   The name of the chart type.

b. Description:
   A simple description of the chart type.

c. Type:
   The modes the chart type can run in. This can be one or more of the
   following:

   - 2D         Chart type supports 2-dimensional rendering mode.
   - 3D         Chart type supports 3-dimensional rendering mode.
   - Stacked    Chart type stacks data series.

```
  - XY         Chart type uses 2 data sets.
  - XYZ        Chart type uses 3 data sets.

d.  Axes:
  The axes that the chart uses.  This can be 'none', 'X', 'X Y', or 'X Y Z'.
  Chart types with no axes (like Pie charts) do not render any axes on the
  display.

e.  Class:
  The 'class' of the chart type.  This describes the basic class of this
  chart type in relation to other chart types.  This is used for
                overlaying
                  charts.

f.  Overlays:
  Describes if this chart type supports
                overlays
                , and what type of overlays
  it supports.  This is used for overlaying charts.

g.  Data 1 Points:
  The number of
                data points
                 in Data Set 1 this chart type requires.  This can
  be a specific number, a range of numbers, a minimum number or a maximum
  number.

h.  Data 1 Series:
  The number of
                data series
                 in Data Set 1 this chart type requires.

i.  Data 2 Points:
  The number of data points in Data Set 2 this chart type requires.  If this
  chart type does not need a Data Set 2, this will be clear.

j.  Data 2 Series:
  The number of data series in Data Set 2 this chart type requires.  If this
  chart type does not need a Data Set 2, this will be clear.

k.  Data 3 Points:
  The number of data points in Data Set 3 this chart type requires.  If this
  chart type does not need a Data Set 3, this will be clear.

l.  Data 3 Series:
  The number of data series in Data Set 3 this chart type requires.  If this
  chart type does not need a Data Set 3, this will be clear.
```

## 1.6 Data Points, Series and Sets

```
                A graph must have one or more data sets made up of one or more  ↩
                   data series
which are, in turn, made up of one or more data points.

The Data Point:
```

A graph requires a set of numeric values to draw. Each value is called a data point, that is a single point on a graph.

Each data point must be specified as a specific number, (e.g. 1, 12, 25) or a cell from the project. (e.g. A1, B7)

You can use a range from the current project to define a sequence of data points. (e.g. A1:D10, B1:C5)

By default, any cell or range used to describe a data point is taken from the
Default Sheet
for the graph. You can, however, specify the sheet name of each individual cell or range by entering its full 3D name (e.g. B_A1, A_A1:B_C5). This will override the Default Sheet for the graph.

A single data point can be either a single number, two numbers describing the data point, or three numbers describing the data point, depending on the current chart type.

Data points that are infinity values (positive or negative) are automatically drawn to the maximum or minimum axis value.

The Data Series:

A Data Series is one or more data points defined as a group. You can group data points by using the comma ',' to define additional data points. For example: A1,B7,C1:C10 This defines a data series made up of cells A1, B7, and the cells in the range C1 to C10 (a total of 12 cells).

When a chart type supports or requires more than one data series, you can define multiple series by using the semi-colon ';' character to separate the data series. For example: A1:A10;B1:B10. This defines two data series made up of A1:A10 and B1:B10.

You can use comas and semi-colons to build complex lists of data series in a data set.

The Data Set:

A Data Set is one or more Data Series defines as the entire set of data to draw in a graph. The main data set for a graph is entered in the 'Data Set 1' entry in the Edit Graph requester to define the data for the graph.

If the chart type requires two values to define a data point, you will have to enter a second Data Set with the same number of data series and points as Data Set 1 into the 'Data Set 2' entry. The same applies for 'Data Set 3', which is needed if the chart type requires 3 values per data point.

A graph can also require a Labels Set. The Labels Set uses the same format as a Data Set (one or more data series) but treats the values as strings for display in the graph. Some chart types also require a second labels set, which is entered into the 'Labels 2' entry.

If you enter the wrong number of data sets, series or points for a graph, it will complain by showing a message in the graph preview telling you where the problem is.

## 1.7 Special Data Series Definitions

```
                Special Data Series Definitions:
For each
                data point
                 in a series you can specify a specific color or pattern to
show it in, or a specific symbol to show for it.  The definition of a color,
pattern, or symbol value is called a token.

You can directly specify a color or pattern to show a
                data series
                 in, or a
specific symbol to display for its data points.

To directly a
                color value
                 for a data series all you need to do to is to enter
the color value at the end of the range surrounded by '(' and ')'.  The color
can be defined as a number from 1 to 255, or a cell (or range) that contains
a value from 1 to 255.

For
                patterns
                , you just do the same but use the '[' and ']' brackets to  ↩
                   surround
the numbers, and for
                symbols
                 you use the '{' and '}' brackets.


Examples:

  a1:a4(7)
  This will show the series a1:a4 in color 7.

  a1:a4(7,4,5,10)
  This will show the series a1:a4 with a different color for each item.

  a1:a4(b4)
  This will show the series a1:a4, and will use the value of the
  cell b1 as the color for the series.

  a1:a4(b1:b4)
  This will show the series a1:a4, and will use the values of the
  cells in the range b1:b4 for the colors for each item.

  a1:a4(b4,7,4,5)
  This will show the series a1:a4, and will use the value of the cell
  b1 as the color for first item, and 7, 4 and 5 for the remaining items.
```

```
  a1:a4[7]
  This will show the series a1:a4 in pattern 7.

  a1:a4(5)[7]
  This will show the series a1:a4 in color 5 and pattern 7.

  a1:a4(b1:b4)
  This will show the series a1:a4, and will use the values of the cells
  in the range b1:b4 for the patterns for each item.

  a1:a4{1}
  This will show the series a1:a4 with symbol 1.

You can define the colors, patterns or symbols for each
                 data point
                  either at
end of each data point, or at the end of the data series:

  a1,a2,a3,a4(7)
  a1,a2,a3,a4(7,4,5,6)
  a2,a2,a3,a4(b1:b4)
  a1(7),a2(4),a3(5),a4(6)
  a2,a2,a3,a4[b1:b4]
  a2,a2,a3,a4[b1:b4]{2}


There is another token which can be used in a data series.  It is called the
'specials' token, and it is used differently by each chart type.  It is
normally a numeric value (negative or positive).  Read the details on Specials
for each chart type in the FinalCalc manual.

The specials token is entered in the same way as the color, pattern and symbol
tokens, but is surrounded by '<' and '>'.

The last token you can use in a data series is the
                Legends
                 token.  It is a
simple string, surrounded by double-quotes, which defines the title for the
data series.

For example:

  a1:a10"Net Profit"
  b1:b10"Sales"

Legends tokens are only accepted in Data Set 1.  Once you enter a Legend token
for any data series this legend is shown in the Legends Box.

The
               next section
                describes an easier method of entering data series tokens.
```

## 1.8 Editing a Data Set

Editing a Data Set:

To edit a Data Set, you can simply enter the data set by typing into the Edit Graph's Data Set entries box which represents that data set. Each entry also has two small buttons next to it:

The [?] button: (keyboard shortcut is Alt + the hotkey for that entry)

This brings up a requester listing the current range in the current sheet as well as all defined range names. You can select one of them to paste back into the data set entry.

The [e] button: (keyboard shortcut is Control + the hotkey for that entry)

This brings up the 'Edit Data Set' requester which provides several tools that help in editing and building data sets.

The 'Edit Data Set' requester breaks down the data set into its component data series and shows them as a list of entries. It allows you to work on each data series separately.

From it you have access to the following tools:

a. Rearrange Data Series:

The 'Up', 'Down', 'Top' and 'Bottom' buttons rearrange the order of the data series in the data set.

b. Edit a Data Series:

The 'Edit' button brings up the 'Edit Data Series' requester which breaks down the data series into its components allowing you to edit each one of them separately.

The data series components are the 'Range' which defines the actual data points in the data series, and the color, pattern, symbol, and special tokens and the legend text for this data series.

For the range, color, pattern, and symbol entries, there is a [?] and a [+] gadget next to each one. The [?] brings up a requester that allows you to select a range, color, pattern, or symbol to paste into the entry. The [+] also allows you to select the same thing, but the selection is added to the end of the current entry.

c. Define a New Data Series:

Adds a new data series and brings up the Edit Data Series to define it.

d. Clone a Data Series:

Creates a copy of the currently selected data series.

e. Delete a Data Series:

Deletes the currently selected data series.

f. Split a Data Series:

Hitting the 'Split' button brings up the 'Split Range' requester, which allows you to break up a data series into multiple data series broken by sheets, columns and/or rows.

The Split Range requester has the following settings:

- Range:
  The range to split. You get a chance to edit it here if you like.

- Split Sheets:
  If turned on, a range of multiple sheets will be split into sheets. e.g. "A_A1:C_Z10" would be split into "A_A1:A_Z10", "B_A1:B_Z10" and "C_A1:C_Z10"

- Split Columns:
  If turned on, a range of multiple columns will be split into columns. e.g. "A1:C10" would be split into "A1:A10", "B1:B10" and "C1:C10"

- Split Rows:
  If turned on, a range of multiple rows will be split into rows. e.g. "A1:C4" would be split into "A1:C1", "A2:C2", "A2:C2" and "A2:C2"

- Generate 3D Ranges:
  If turned on, the ranges generated will always be in 3D format. (e.g. "A_A1:A_C4"). If turned off, ranges will be generated in 3D format only if the source range contained sheet references.

When you click on 'Split' the data series will be split according to these settings and any new data series generated will be inserted into the Edit Data Set requester.

You can turn on more than one split mode, i.e. sheets, columns and/or rows. This will have the effect of breaking the range into smaller units. If you turn them all on, you will get it split into single cells.

Only the first range in the Data Series range string is used. The remainder of the string is copied as is to all the generated data series during splitting.

Note that all changes done the Data Set are not applied until you hit 'Use'. If you hit 'Cancel' you will be informed that your changes will be lost.

## 1.9 The Graph Axis

The Graph Axis:

Most chart types draw an axis for the graph. The axis can be drawn as either 2-dimensional or 3-dimensional.

The 'Edit Graph Axis' requester controls how the graph axis is shown, and comes up a little differently for 2D and 3D axis modes. It is called by the 'Axis' button in the Edit Graph requester.

2D Axis mode:

If the current chart type supports 2D rendering and the 3D button is off in the Edit Graph requester, the current graph is in 2D mode. The Edit Graph Axis requester will look like this when called:

3D Axis mode:

If the current chart type supports 3D rendering and the 3D button is on in the Edit Graph requester, the current graph is in 3D mode. The Edit Graph Axis requester will look like this when called:

The Edit Graph Axis requester contains the following settings:

a. Borders:
This controls whether a left, right, top or bottom border line is drawn in a 2D graphs. It has no effect on 3D graphs.

b. 3D Planes:
This controls the rendering of 'planes' (axis surfaces) in a 3D graph. You can control all 6 planes individually.

c. Axis Grid and Ticks:

When an axis is drawn, lines can be drawn across the axis to show steps in the values across that axis. At each 'step' you can define a line (either solid or dotted) to be drawn and a tick (a small line at the edge of the axis) to be drawn inside or outside the border, or centered at the border.

Between each step, you can also draw smaller steps, called sub-steps. The number of sub-steps between each step is defined by the 'Number of sub-steps' control as described in section d. below.

You can control how the grid lines and ticks are drawn at each step and sub-step from the 'Axis Grid & Ticks' box in the Edit Graph Axis requester:

- Grid Steps and Sub-steps can be set to:

  1. A solid line.
  2. A dotted line.
  3. None. (no line drawn)

- Grid Ticks and sub-ticks (ticks at sub-steps) can be set to:

  1. Inside: the tick is drawn inside the graph border.
  2. Outside: the tick is drawn outside the graph border.
  3. Centered: the tick is centered on the graph border.
  4. None: no tick is drawn.

These can be set separately for the X, Y and Z axes.

d. Number of sub-steps:

You can define the number of sub-steps for each axis. The X, Y, and Z 'Number of Sub-steps' entries control how many sub-steps are shown for each axis. They default to 0, which means no sub-steps. To enable sub-steps for an axis, simply enter a number in its entry. Note that you must also have that axis's grid sub-steps rendering type selected (normal or dotted) otherwise nothing will show up.

Sub-steps are not available for 3D graphs.

e. Steps & Labels Skip Count:

The Steps & Labels Skip Count is a number for each axis that allows you to display only every X axis steps and labels. This reduces the number of steps and labels shown and is useful when the number of
data points
is
large.

For example, if the number of data points is 100, and you set the skip count to 5, all the data points will be shown, but only 20 steps, ticks and labels will be shown. (i.e. 1 step for every 5 data points)

## 1.10 Labels Data Sets

Labels Data Sets:
When drawing an axis, some chart types (like bar charts, line charts and area charts) can get the text shown at each axis step from cells in the project.

To set labels for a graph, simply enter a
data set
into the Labels
entry in the Edit Graph requester.

The cells defined in a labels data set are read directly from the project and rendered exactly as they appeared on the sheet. Cells referenced in the Labels Data Set can be numeric or string.

Some chart types (like Pie charts) can use two labels data sets. To use the second labels data set simply enter it in the 'Labels 2' entry in the Edit Graph requester.

Most 3D chart types accept multiple
data series
in the Labels Data Set. The
first series will be used for the X axis titles, and the second range will be used for the Y axis.

You can specify a different color for each individual label by using color

tokens
.

## 1.11 Graph Colors, Patterns and Symbols

Graph Colors, Patterns and Symbols:
When drawing an object that corresponds to a
data point
in a graph (like a
line, bar, pie slice, etc.) FinalCalc shows that object in a certain color, and possibly a pattern or a symbol. (A symbol is a graphic shape drawn for an object, e.g. a small box, cross, or a circle)

Colors, patterns and symbols in graphs are defined by the Graph Colors List, the Graph Patterns List and the Graph Symbols List.

The color, pattern and symbol used for an object is normally automatically decided by FinalCalc (as described below). You can, however, override the automatic selection by defining a specific color, pattern or symbol for an object (or an entire data series) via a
token
.

The 'Edit Graph Auto Color, Patterns & Symbols Settings' requester is called from the 'Auto' button in the Edit Graph requester. It allows you to control the automatic colors, patterns and symbols selection for rendering data points in a graph.

Patterns:

Use Patterns:
Defaults to off. If set to on, then patterns are used in graph objects that support patterns such as bars, pie slice, and areas. Line charts do not support patterns as the object has no thickness.

First Pattern:
The first pattern to use in a graph.

Use Single Pattern for Series:
Defaults to on. If set to off, then each object in a
data series
will get
a different pattern. If set to on, then each data series will get a single pattern.

```
Reset Pattern for each set:
Defaults to off.  If set to on, then the pattern will be reset to the
first pattern after each new
                data set
                .

Colors:

First Color:
The first color to use in a graph.

Use Single Color for Series:
Defaults to on.  If set to off, then each object in a
                data series
                 will get
a different color.  If set to on, then each data series will get a single
color.

Reset Color for each set:
Defaults to off.  If set to on, then the color will be reset to the first
color after each new
                data set
                .

Symbols:

Use Symbols:
Defaults to off.  If set to on, then symbols are used in graph objects
that support symbols, for example line charts and scatter charts.

First Symbol:
The first symbol to use in a graph.

Use Single Symbol for Series:
Defaults to on.  If set to off, then each object in a
                data series
                 will get
a different symbol.  If set to on, then each data series will get a single
symbol.

Reset Symbol for each set:
Defaults to off.  If set to on, then the symbol will be reset to the first
symbol after each new
                data set
                .

Symbol Color:
Forces symbols to be of a specific color.  If no color is defined, symbols
will have the same color as the data series they follow.
```

## 1.12 Graph Axis Limits

```
                Graph Axis Limits:
When FinalCalc draws a graph, it first looks at the value of all the data
points in all
```

data sets
and decides the maximum and minimum values of each
axis' scale as well as the number of steps in the axis and other settings.

You can override the automatic scale limits as well as control other aspects of the drawing scale for each axis by using the 'Edit Graph Limits' requester, which is called by the 'Limits' button in the Edit Graph requester.

The scale of each of the 3 data sets in a graph is calculated separately, and you can control each of them separately too.

The Edit Graph Limits requester allows you control over the following settings for each of the data sets:

Minimum:
Defines the lowest value shown in the graph. Any
data point
lower than this
value will be cropped and either not shown at all or shown at the minimum value's level.

If the Auto Limits Minimum option is turned on (in the Auto Limits) box, this value is automatically decided by FinalCalc every time the graph is drawn. If you turn it off by either entering your own minimum value, or by turning off the Auto Minimum button, the value shown will be used for drawing the graph.

Maximum:
Defines the highest value shown in the graph. Any data point higher than this value will be cropped and either not shown at all or shown at the maximum value's level.

If the Auto Limits Maximum option is turned on (in the Auto Limits) box, this value is automatically decided by FinalCalc every time the graph is drawn. If you turn it off by either entering your own maximum value, or by clicking on the Auto Maximum button, the value shown will be used for drawing the graph.

Floor:
Defines the 'floor' of the data. The defaults to 0. This only affects where the main line of the axis is rendered, and where some chart types (like bar charts) base their objects at.

If the Auto Limits Floor option is turned on (in the Auto Limits) box, the floor is set to zero. If you turn it off by either entering your own floor value, or by turning off the Auto Floor button, the value shown will be used for drawing the graph.

Step:
Defines the value of each step in the axis grid and ticks. You can use any value here, even if it causes the last step not to be the same size as the rest of the steps.

If the Auto Limits Steps option is turned on (in the Auto Limits) box, this value is automatically decided by FinalCalc every time the graph is drawn. FinalCalc tries to get between 4 and 8 steps in a graph. If you turn it off by either entering your own step value, or by turning off the

Auto Steps button, the value shown will be used for drawing the graph.

Scale:
Defines a scale value which is applied to the axis and all data points in the
data set
. The scale value defaults to a value of 1, which shows all points in their original values.

For example, if you enter a scale of 1000, then the values of all data points and the axis will be divided by 1000, effectively making it show the values in thousands.

Log Scale:
The Log Scale, if turned on for a
data set
, scales all data on a logarithmic scale (of log 10). All values for the data set, including the data points, axis minimum, maximum, floor, steps an scale, will be converted to a log 10 value when rendering. Also, all values less than or equal to zero get a log value of 0.

The only thing that will not be shown in a log scale will be the values shown on the axis.

Axis Numbers Format:
The Numeric Format button brings up the 'Edit Numeric Format' button, which defines the way numbers along the axis are displayed. You can define the format in a similar way to defining the format for a numeric cell.

## 1.13 Graph 3D Axis Control

Graph 3D Axis Control:
When FinalCalc draws a 3D graph, it builds it as a real 3-dimensional object in a 3D world. This allows you to rotate the graph and zoom and shrink it along any axis, even move it in closer or further away.

The '3D Axis Control' requester, which is called by the '3D Axis' button in the Edit Graph requester, allows you to manipulate a 3D graph using the following settings:

Rotate:
This rotates the graph along the center of its axis in 3 directions: Roll (left/right), Pitch (up/down), and Yaw (diagonal). Each of these can range from -180 degrees to +180 degrees, which defines degrees of rotation.

Axis Center:
Shift the center of the graph away from the center of the display along the X, Y and Z axes, allowing you to move around the graph.

Axis Scale:
Shrinks and expands the graph in the X, Y and Z directions, allowing you to customize how the graph looks like.

Distance:

Sets the distance from the graph to the viewer. Defaults to 0. Changing it
to a negative value moves the graph away from the viewer and changing it to
a positive value brings the graph closer in to the viewer. As all graph
objects are clipped to fit in the display, you can zoom right into the
graph and even on a single graph object if you like.

## 1.14 Graph Titles

Graph Titles:
The Graph Titles requester allows you to define titles to be shown for a graph.
It is called from the 'Titles' button in the Edit Graph requester.

You can define many titles for a graph, each of them goes in a specific place.
You can also define the color, font and size the text shows up in. The
requester also controls the text titles shown for graph axes.

Text Titles:

These are basic text strings that are shown in specific places in a graph.

  Heading 1:
  Main graph heading, top center.

  Heading 2:
  Graph sub-heading, below Heading 1.

  Note 1:
  Main graph note, bottom center.

  Note 2:
  Sub-note, below Note 1.

  Left:
  Vertical text, left margin.

  Right:
  Vertical text, right margin.

To enter any of those titles, enter the text in the appropriate entry in the
Edit Graph Titles requester. You can also select its color and font.

You can enter any text for a title.
Special tokens
(that start with a %
sign) are replaced by specific entries like the current date or time.

To access a list of these tokens in FinalCalc, click in the [?] gadget next
to the title string gadget. This will bring up the 'Edit Graph Text String'
requester, which allows you to directly paste tokens into the title strings.

Axis Titles:

The bottom box in the Edit Graph Titles requester controls how text in axes
is drawn. This includes text generated by the graph itself (numbers along
the axis) or titles from the

Labels Data Set(s)
.

1. Pie and X Axis Labels:

Defines the font and color for text shown along the X Axis (bottom axis in 2D graphs) and pie labels (text shown for each pie slice).

The Enable Left button enables showing labels along the left of the X axis in 2D and 3D graphs. The Enable Right enables graph text titles along the right side of the X axis in 2D and 3D graphs.

2. Y Axis Labels:

Defines the font and color for text shown along the Y Axis (vertical axis in 2D graphs).

The Enable Left button enables showing labels along the Y axis in 2D and 3D graphs (text is shown along the left side of the axis in 3D graphs) as well as text for pie slices in pie charts.

The Enable Right enables graph text titles along the right side of the Y axis in 3D graphs. It has no effect in 2D graphs.

The Sideways button shows the Y axis labels in 2D graphs sideways.

3. Z Axis Labels:

Defines the font and color for text shown along the Y Axis in 3D graphs.

The Enable Left button enables showing labels along the left of the Z axis in 2D and 3D graphs. The Enable Right enables graph text titles along the right side of the Z axis in 2D and 3D graphs.

## 1.15 Graph Objects Color Control

Graph Objects Color Control:
The Edit Graph Colors requester allows you to define the colors for all graph objects other than
data points
and
legends
. It is called by the 'Colors'
button in the Edit Graph requester.

Axis Objects:

These are parts of the X, Y and Z axes rendered:

- Plane
The background of the X, Y, or Z plane.

- Borders
The color of the lines used to draw a border around a graph.

- Grid
  The color of the grid lines.

- Sub-grid
  The color of the sub-grid lines.

- Ticks
  The color of the axis ticks.

- Sub-ticks
  The color of the axis sub-ticks.

In 2D graphs, the Z axis colors and the Y axis plane color are ignored.

Screen Background:
The color to set the screen background to. This only applies if the graph is being shown on its own screen and not in a graph view or preview.

Graph Background:
The background color to draw the graph on. If the Graph and Screen background colors are the same, the graph background color is never actually drawn in order to speed up graph rendering. However, when printing, it is important to define the graph background color if you want it printed.

Default Lines Color:
The default color for any line object in the graph that is set to color 0.

Default Text Color:
The default color for any text object in the graph that is set to color 0.

## 1.16 Graph Legends Box

Graph Legends Box:
The Graph Legends Box shows what each data series in a graph represents. A legend consists of a title for a data series, and a small icon showing what color, pattern and/or symbol the data series is shown as in the graph.

To define a legend for a data series, add it to the data series definition surrouned by double-quotes (e.g. a1:a10"Total Sales").

If a graph has any legends defined, the Legends Box will automatically appear in the graph. You can control how the box looks like and where it will be placed by using the 'Legends Box Control' requester, which is called by the 'Legends' button in the Edit Graph requester.

The Legends Box Control Requester control the position and appearance of the legends box. It contains the following controls:

Box Position:
The position where the Legends box will show up:

- Bottom
  Below the graph Y axis, above the notes.

- Right
  Right side of the graph

- Top
  Above the graph, below the heading.

- Left
  Left side of the graph.

Box Border:
The look of the border around the legends box:
- None
  No border is drawn.

- Thin
  A thin (one pixel) line is used.

- Thick
  A thick (2 pixel) line is used.

- Shadow
  A shadow effect is drawn.

Legends Order:
Controls how the legends are shown in the box:
- Horizontal
  Legends are shown in one line.

- Vertical
  Legends are shown each one its own line.

Legend Font:
The font to use for legends text.

Legend Color:
The color to use for legends text. If set to 0, each legend will use the same color as its data series.

Box Color:
The background color of the legends box.

Border Color:
The color of the legends box border.

User Comments:
A single line user comment to be shown at the top or bottom of the legends box. You can also set its font and color. You can enter two comment lines, one shows up at the top of the legends box, and the other shows up at the bottom of the legends box.

If a comment is entered, the legends box will show up in the graph even if no legends were defined for any data series.

Show Symbols:
The Show Symbols option, if turned on, shows the actual symbol used for the data series in the legends box. It defaults to on.

Show Patterns:
The Show Patterns option, if turned on, shows the actual pattern used for the data series in the legends box. It defaults to on.

## 1.17 Graph Overlays

Graph Overlays:
An overlay is a chart overlayed on top of the main graph chart. A graph can have multiple overlays.

An overlays consists of a chart type (bar, line, area, etc.), 3
data sets
,
and a title. The title is only used for identification purposes.

You can overlay a graph with different chart types, allowing graphs of mixed chart types, like a bar chart overlayed with a line chart for example.

To add an overlay to a graph, click on the 'Overlays' button in the Edit Graph requester. This will bring up the 'Edit Graph Overlays' requester which lists the overlays on the current graph.

Select the 'New' button to define a new overlay. This will bring up the 'Edit Overlay' requester.

You have to define the three Data Sets and chart type for the overlay. If any of the data sets is not defined, (i.e. left empty) it will inherit the same data set as the main graph.

Each Chart type (bar, stacked bar, line, pie, area, etc.) is defined as being of a specific 'class'. The class is used to define whether the chart can be overlaid on another chart type.

The 'Overlay Info' window shows information about how the selected overlay chart type mixed (successfully or not) with the main graph chart type depending on its class and overlay setting.

The 'Info' button in the Edit Graph requester gives information about the current chart type's class and overlay setting.

A chart type's Overlay setting can be one of the following:

1. Overlay on any other chart.
2. Overlay on same chart class only.
3. Overlay on same chart type only.
4. Never overlay.

If an overlay chart type is not compatible with the main chart type, FinalCalc will warn you that the overlays are not compatible, but will show them anyway in case you really want to get that effect.

Graph Overlays are only allowed in 2D graphs. The 'Overlay' button in the Edit

Graph requester is disabled if the graph is in 3D mode.

## 1.18 Viewing, Printing and Saving Graphs...

Viewing, Printing and Saving Graphs:

Once a graph has been defined, you can then view it on its own screen, show it in a graph view, or print it.

Viewing a graph:
To view a graph on its own screen, simply use the 'View' button in the Graphs List requester, which is called from the 'Graphs - Graphs List' menu.

You can also view a graph on its own screen while you are editing a graph by hitting the View button in the Edit Graph requester or any requester it calls.

The Graph Screen:
The Graph Screen is the screen a graph shows up on when it is shown on its own screen.

When a graph screen is shown, you can hit the Escape key to close the screen. You can also use the screen's 'Cancel' menu to close the screen.

At any time, while a graph is being drawn on the screen, you can hit the Escape key and the graph will stop drawing and close the screen.

The type of screen opened for a graph is defined by the current Graph Screen Mode. To select a graph screen mode, use the 'Settings - Display - Graph' menu to bring up the 'Graph Screen Mode Settings' requester.

In this requester, you can select any Amiga screen mode of any width and height. You can also select the number of colors by setting the Bitplanes setting. It is recommended that you have at least 3 or more bitplanes (i.e. 8 or more colors).

Saving a Graph as an IFF file:
When a graph is shown on its own screen, you can save the graph to a file as an image in IFF format (the standard format for Amiga image files).

Use the 'Save As IFF' menu on the graph screen. You will get a file requester with the name of the graph as the file name. Simply select where you want to save the image, and under what name, and click on 'OK'.

The image will be saved in the same screen mode as the screen it is on.

You need 'iffparse.library' in your system Libs: directory for IFF save to work. If it is not available, or if the IFF Save failed, you will be informed.

Printing a Graph:
To print a graph, you must first define a Graph Print Job and then print it as a normal Print Job.

When printing graphs, the Color mode setting in the Print requester has the

following effect:

- Color:
  All graph objects obey their color setting.  The CMYK values in the

                Color List
                 are used.

- Greyscale:
  All graph objects obey their color setting, but use the Greyscale value in
  the Color List.  Graph text is always printed in black.

- B&W:
  All graph objects are rendered in black.  The graph and screen backgrounds
  and the axis plane background colors are ignored.

Putting a Graph in a View:
You can put a graph in a project view by opening a graph view.

The Graph View is upated as the project data is changed.

When a graph is being drawn into a graph view, that view is locked while the view's contents are being updated.  You cannot resize the view during that time.  Once the update is done, the view's size is unlocked and you can resize it.

## 1.19 Animations...

Animations

>
What is an Animation?

>
Creating an Animation

>
How Animations Work

>
Playing an Animation

>
Recording an Animation

## 1.20 Animations...

What is an Animation:
FinalCalc can 'animate' a graph.  That is, it will show a graph and change how the graph looks and/or the data it displays over a set of steps (frames) in time.

An Animation is a definition of how a graph is to be animated. A project can contain many animations. Animations are managed by the 'Edit Animations' requester, which shows you the currently defined animations for a project and allows you to define, edit, playback, and record animations to disk.

The requester is called by the 'Graphs - Animations List' menu. From it you can do the following:

a. New:
   Define a new animation. Calls the Edit Animation requester.

b. Edit:
   Edit the selected animation. Calls the Edit Animation requester.

c. Clone:
   Clone the selected animation.

d. Delete:
   Delete the selected animation.

e. View:
   Play the animation on the screen.

f. Record:
   Record the animation to disk an image frames.

## 1.21 Creating an Animation...

Creating an Animation:

To create an animation, you must first create a graph. Once you have a graph defined, you can then create an animation by using the 'New' button in the 'Edit Animations' requester.

This will bring up the 'Edit Animation' requester, which defines what the animation does. In in, you have to enter the following:

a. Graph:
   The name of the graph to animate. Hitting the [?] gadget will bring up a list of graphs defined.

b. Start Value:
   The start value (number) for the animated variable X.

c. End Value:
   The end value (number) for the animated variable X.

d. Interval:
   The interval value (number) for the animated variable X. If not defined, it defaults to 1.

e. Script Command:

The script command to execute on every iteration of the loop.

f. Command Source:
Selects whether the Script Command is a simple command as entered (source set to 'String Command'), or is a file name that points to a script on disk (source set to 'External File').

g. Show Animation Screen:
Turns on Animation Screen viewing. It defaults to on. The only useful reason to turn it off is to view an animation in a graph view.

h. Smooth Animation:
Turns on smooth animation mode. In this mode, two screen are used for the animation. While one screen is visible, the other one is used to prepare the next 'frame' in the background, making the animation very smooth.

i. Base filename:
The base file name to use for output files when recording animation frames. (e.g. "ram:myanim")

## 1.22 How Animations Work...

How Animations Work:

The
Script Command
is the basis for what an animation does.

The start, end and interval value define a loop which is run multiple times depending on their values. The Script Command is executed every time the loop is repeated.

To use animations effectively, you will need to learn the basics of scripts, however, here are a few examples of script commands:

Example 1:

```
cell a1;store x;calc wait
```

This will move the cursor to cell a1, store the value of 'x' (which is the current value of the loop) and then wait for the background recalc to end. (It is important to use 'calc wait' if you change a cell in the project that will cause the values of other cells to change.)

Example 2:

```
cell a1;store x/100;calc wait
```

This does the same as above, but stores the value of x divided by 100 into cell a1.

Example 3:

```
graph gr roll x
```

Sets the 3D roll setting for the current graph ('gr' is an automatically defined variable that points to the graph being animated). If you set the start value to -180 and the end value to 180, this will rotate a 3D graph around its axis from left to right in 360 steps. You can reduce the number of steps by increasing the interval.

You can basically do anything in the Script Command as long as it is a valid ARexx script. This can include acquiring data from a another program.

Here is a more detailed description of what is happening in an average script:

Finalcalc takes the start, end and interval values and the script command and builds a script out of them. For example, if you set start to 0, end to 100, and interval to 5, and the script command to "cell a1;store x;calc wait", the following script will be generated:

```
an = '"Animate Growth Rate"'
gr = '"Bar chart"'
anim start an
do x=0 to 100 by 5
cell a1;store x;calc wait
anim show
end
anim end
```

This is what each line does:

```
an = '"Animate Growth Rate"'
```

Defines the script variable 'an', and stores the name of the animation to run it in.

```
gr = '"Bar chart"'
```

Defines the script variable 'gr', and stores the name of the graph to be animated in it.

```
anim start an
```

Puts FinalCalc into Anim mode. It reads the settings of the animation called 'Animate Growth Rate' (as stored in variable 'an') and opens one or two screens depending on whether Smooth Animation was off or in the the animation requested.

```
do x=0 to 100 by 5
```

The loop defined by start, end and interval. This is executed multiple times as defined by the settings in the animation. Note that the variable

'x' is used as the master variable. Its value will change on every run of the loop, which can be used in the 'script command' as defined below.

cell a1;store x;calc wait

The user defined Script Command. This will be executed multiple times depending on the loop defined above. Note the use of 'calc wait' at the end of the command. This makes sure that the effects of the change made to the sheet data have finished recalculating before the graph is drawn.

anim show

Shows the current frame of the animation on the screen. It will automatically handle dual-screen (smooth) animation. If the 'anim save' command was generated (when running in Save mode, as defined below), the frame will then be saved to disk according to the animation output settings.

end

Marks end of the above loop.

anim end

Ends the animation, shuts down all screens and closes all files.

The actual script generated is longer than the example above, but it just has code to handle the user aborting the animation.

For full details on the script ANIM command used in animation scripts, and the GRAPH command used to change graph settings during an animation, see your FinalCalc user manual.

## 1.23 Playing an Animation...

Playing an Animation:

To play an animation, simply click on the 'View' button in the 'Edit Animations' requester. A status window will be opened that shows information about the animation, which frame is currently being drawn, which frame is currently on the screen, and an estimate of the time it will take to run the animation.

After that, the animation screen will open and the animation will start running. If you want to abort the animation at any time, hit the Escape key, or return to the FinalCalc screen and click on the 'Abort' button in the status window.

While an animation is running, the system pointer (mouse pointer) is turned off.

If you move away from the animation screen and back onto the FinalCalc screen, as soon as you do anything on any FinalCalc view, the animation is automatically aborted.

Important notes:

An animation will possibly change a project's data while running, depending on the script command(s) defined in the animation. It is recommended that you save the project before you run an animation if you have not saved the project. However, any changes made to the project data can be undone if the Undo System is on.

Animations require that ARexx is available on your system and is currently running. If ARexx is not available on your system, animations will not work.

## 1.24 Recording an Animation...

Recording an Animation:

Recording an animation simply plays an animation and saves every frame of that anim to disk as an IFF image file.

To record an animation, click on 'Record' in the Edit Animations requester. If you defined a Base filename for the animation, it will start running immediately. If you didn't define a base filename, a file requester will come up to ask you for one.

Each frame of the animation is saved as the base filename with ".0001" added to it for the first frame, ".0002" for the second frame, and so on.

You can abort recording an animation in same way as aborting playback of an animation.

Once the animation is recorded, a file will be generated with the base filename with ".list" added to it listing the names of all the frame files generated. This is useful when using the MakeAnim tool (a public domain program) to generate an Anim file. The shell command to use for this is:

```
makeanim anim.list our.anim
```

where anim.list is the ".list" file generated, and our.anim is the file name for the Anim file to be generated.

The graph animation screens use the screen size and depth as defined in the Graph Screen Mode settings, however, the screen width and height are rounded down to the closest 8 pixel boundary to ensure compatibility with animation tools like MakeAnim.

## 1.25 Graph Colors, Patterns and Symbols...

Graph Colors, Patterns and Symbols:
You can edit the colors, patterns and symbols used in graphs and can also

define new ones.

Colors List:

The Graph Colors List is a user-defined list of colors that are used in graphs. The colors list has a standard set of entries, but you can modify any of these and add more colors to the list.

Any color entry in requesters that edit a graph (requesters called by the Edit Graph requester) has a [?] gadget attached to it that allows you to pick a graph color for that entry.

To edit the color list use the 'Graphs - Colors List' menu. This brings up the 'Edit Colors' requester which shows a list of all graph colors currently defined.

You can select any color for editing or define a new color. This will bring up the 'Edit Color' requester.

A graph color entry consists of the following settings:

a. Color Name:

   The color's name. It is used by some functions to find the color.

b. Display Color Values:

   The red-green-blue (RGB) values for the color when being displayed to the screen. The R, G, and B value is defined as a value of 0 to 255 (8 bits), but is scaled down to the range the machine it is running on can handle. (for example, non-AGA Amiga machines scale it down to a range of 0 to 15)

c. Print Color Values:

   The cyan-magenta-yellow-black (CMYK) values for the color when being printed to a color printer. The CMYK values can be different from the RGB values. This allows you to control exactly how a color is printed.

   The 'Calc' button in the Edit Graph Color requester calculates the CMYK values from the RGB values.

d. Print Greyscale Value:

   The greyscale value for the color when printing to a black & white printer in greyscale mode. It defines a percentage of black for the current color.

Graph Screen Colors Handling:

When a graph is being drawn on its own screen, FinalCalc first looks at the colors that the graph will be using. If the number of colors the graph needs is less than the number of colors the screen can display, all those colors are allocated directly to the screen, and any remaining colors in the Color List are then allocated to the screen (as many as will fit).

If the number of colors the graph needs is more than the number of colors the screen can display, FinalCalc will allocate as many colors as it can, and then

replace the missing colors by substituting them with the closest colors from the colors that were allocated to the screen.

FinalCalc will not allow an object to have the same color as the graph screen background.

Patterns List:

The Graph Patterns List is a user-defined list of patterns that can be used in graphs. The patterns list has a default set of entries, but the user can modify these entries and add more entries to the list.

To edit the patterns list use the 'Graphs - Patterns List' menu. This brings up the 'Edit Patterns' requester.

You can select any pattern for editing, or define a new pattern by using the 'Edit' or 'New' buttons. They both call the 'Edit Pattern' requester, which defines what a pattern looks like.

A pattern basically consists of two 16x16 pixel definitions of a pattern. One is used to show the pattern on the display, and the second is used when printing.

To edit a pattern, click on the pixels in the pattern box to turn them on or off. Or, use one of the following buttons:

| | |
|---|---|
| Copy | Copy the pattern from the other box. (display or print) |
| Invert | Invert all pixels in the patterns. |
| Rotate | Rotate the pattern 90 degrees. |
| Clear | Clear all pixels in the pattern. |
| Flip V | Flip pattern vertically. |
| Flip H | Flip pattern horizontally. |

The pattern Print Scale control allows you to control how big each pixel of the pattern is rendered on your printer. You can control it from 1 to 100, 1 is a very small dot (about 0.001 inches) and 100 is a big dot (about .1 inches). This defaults to a setting of 18. (Due to the way FinalCalc prints patterns, the larger the scale value, the quicker the pattern will render on your printer.)

Symbols List:

The Graph Symbols List is a user-defined list of symbols that can be used in graphs. The symbols list has a default set of entries, but the user can modify these entries and add more entries to the list.

Symbols are only used in 2D graphs. The chart types that currently support symbols are Line, Scatter, and X-Y Scatter charts.

To edit the symbols list use the 'Graphs - Symbols List' menu. This brings the 'Edit Symbols' requester.

You can select any symbol for editing, or define a new symbol by using the 'Edit' or 'New' buttons. They both call the 'Edit Symbol' requester, which defines what a symbol looks like.

A symbol basically consists of two 16x16 pixel definitions of a symbol. One is used to shown the symbol on the display, and the second is used when printing.

The symbol image is centered on the center pixel (8,8) in the symbol image. The center of the symbol is drawn exactly at the join between lines in a chart.

To edit a symbol, click on the pixels in the symbol box to turn them on or off. Or, use one of the following buttons:

| | |
|---|---|
| Copy | Copy the symbol from the other box. (display or print) |
| Invert | Invert all pixels in the symbols. |
| Rotate | Rotate the symbol 90 degrees. |
| Clear | Clear all pixels in the symbol. |
| Flip V | Flip symbol vertically. |
| Flip H | Flip symbol horizontally. |

The symbol Print Scale control allows you to control how big each pixel of the symbol is rendered on your printer. You can control it from 1 to 100, 1 is a very small dot (about 0.001 inches) and 100 is abig dot (about .1 inches). This defaults to a setting of 24.

## 1.26 Graph Text Special Codes

The following is a list of all supported tokens for graph text:

| Token | Description |
|---|---|
| %d | The current date, formatted with the global date format |
| %t | The current time, formatted with the global time format |
| %d"format" | The current date, formatted with a custom date format |
| %t"format" | The current time, formatted with a custom time format |
| %f | The file name of the current project |
| %F | The full path name of the current project |
| %cA1 | Contents of a cell. |
| %c"TEST" | Contents of the top-left cell in a range |
| %l | The sheet user title |
| %g | The current graph's name. |
| %% | Output a % sign |

| Example | Result |
|---|---|
| %d | 21 Jan 1993 |
| %t | 11:15:00 |
| %D"Wwww, d Mmmm yyyy" | Monday, 21 January 1993 |
| %t"hh:mm aa" | 11:15 a.m. |
| %f | test.sheet |
| %F | work:sheets/test.sheet |
| %cA1 | <contents of cell A1> |
| %c"TEST" | <contents of top left cell in range TEST> |