

# Contents

- 1 ... [Contents](#)
- 2 ... [General File Structure](#)
- 5 ... [Scaling Information](#)
- 6 ... [Using Templates](#)

## General File Structure

RLPlot stores graphs in plain text files usually with the extension `.rlp`. The file starts with some comment lines followed by a list of sections, each section defining a graphic object. Any line starting with a semicolon is treated as a comment and ignored. An exception is the first line of the file starting with `‘;RLP ...‘`. This line must be present to indicate a valid RLPlot graph file.

An example section defining a graphic object may look like the following:

```
[26=Axis]
sAxLine= 0.2
sAxTick= 2
sTickLabel= 4
LbDist= 0 8
TickLbDist= 0 4
Color= 0x00000000
AxisDef=0x000002c1 0 100 25 90 0 145 90 0 0 20 0 0 0 0
GridLine= 0.1 8 0x00808080 0xfcfcfcfc
Ticks=(6){14 16 18 20 22 24}
Label= 25
```

More generalized this may be summarized as:

```
[<Ref>=<name>]
<item1>= <record1>
<item2>= <record2>
...
<itemn>= <recordn>
```

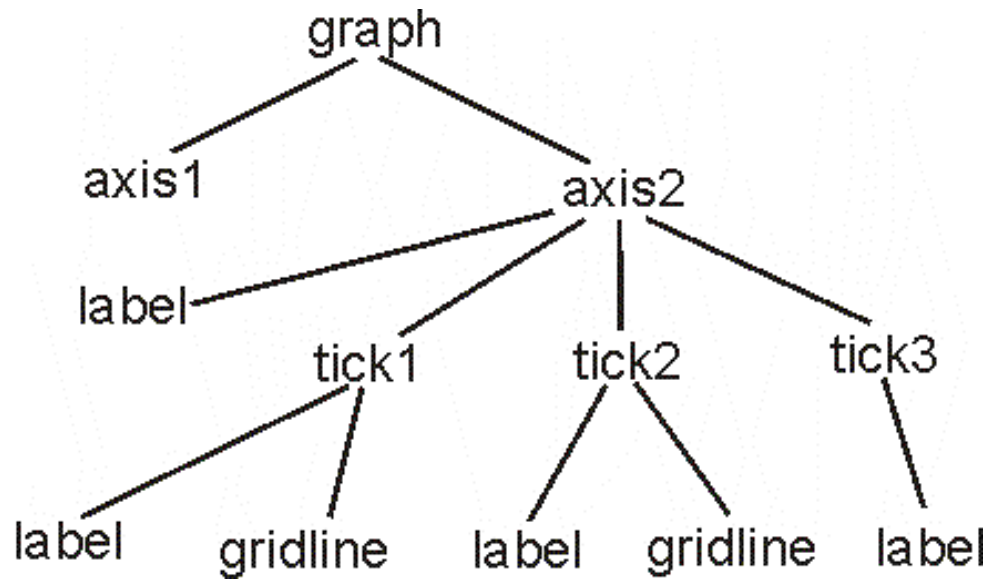
where

- <Ref> Is a unique number occurring only once in a file. It is used as handle and identifies this specific object. Any reference to this object by another object uses this handle.
- <name> Defines the type of the object (i.e. the object class). The meaning of all following lines is dependent on this name.
- <item<sub>1</sub>, item<sub>2</sub> ... item<sub>n</sub>> These are symbolic names for variables and structures filled with the values of record<sub>1</sub>, record<sub>2</sub> until record<sub>n</sub>.
- <record<sub>1</sub>, record<sub>2</sub> ... record<sub>n</sub>> Records contain the values assigned to each item.

Some general considerations apply to objects defined in a RLPlot file:

- Every object in the file must be defined within the file before it is referenced by another.
- Any missing item in an object is replaced by default values.
- The order of items within an object does not matter. However, it is suggested that the order is maintained for faster performance used by RLPlot or given in this document in a later section.

The objects of a graph or plot are organized as a tree. Some root object owns child objects which in turn may own their own children. The following example illustrates how a total of 12 objects are arranged as a tree:



The representation of this two dimensional structure in a RLPlot file is shown on the next page. Arrows indicate the references to objects.

```

;RLP 1.0

[1=Axis]
[2=Label]
Dist=-4 0
Flags= 0x00000022
TxtDef= 0x00000000 0x00ffffff 4.000 0.000 0.000 5 1 0 0 "0"

[3=Tick]
Val= 0
Flags= 0x00000292
Label= 2

[4=GridLine]
Type= 3
Line= 0.1 8 0x00808080 0xfcfcfcfc

[5=Label]
Dist=-4 0
Flags= 0x00000022
TxtDef= 0x00000000 0x00ffffff 4.000 0.000 0.000 5 1 0 0 "50"

[6=Tick]
Val= 50
Flags= 0x00000296
Grid= 4
Label= 5

[7=GridLine]
Type= 3
Line= 0.1 8 0x00808080 0xfcfcfcfc

[8=Label]
Dist=-4 0
Flags= 0x00000022
TxtDef= 0x00000000 0x00ffffff 4.000 0.000 0.000 5 1 0 0 "100"

[9=Tick]
Val= 100
Flags= 0x00000296
Grid= 7
Label= 8

[10=Label]
Pos= 13 50
Dist=-12 0
Flags= 0x00000002
TxtDef= 0x00000000 0x00ffffff 4.800 90.000 0.000 9 1 0 0 "y-axis"

[11=Axis]
Ticks=(3)(3 6 9)
Label= 10

[12=Graph]
Type= 1
Xaxis=0x000000c0 70 96 25 90 0 145 90 0 70 5 0 0 0 0
Yaxis=0x00000092 0 100 25 10 0 25 90 0 0 20 0 0 0 0
DefXAxis= 0
DefYAxis= 1
Axes=(2)(1 11 )

```

## Scaling Information

Scaling information is stored at different locations in the File. Each plot or layer maintains a record where minima and maxima are stored, both for x- and y-values. Also axes store scaling information, the minimum and maximum in their direction. During creation of a graph each layer collects size information from each of its children. The size information of the layers is collected by the root object (graph) and the values calculated for the default axes. Upon changes to values, addition of plots, deleting objects etc. part of this process is repeated if automatic scaling is necessary.

Automatic scaling is not necessary if files are read from the disk. All values are usually present in the file.

### Automatic scaling for files

Automatic scaling during file read is implemented as a kind of error recovery. That means that automatic scaling is initiated if the information in the file is missing or not valid. If you want automatic scaling of files after read the best point to achieve that is the basic graph object. Setting the 'Bounds' member of this object to '0 0 0 0' (that means x-values have a range from zero to zero, y-values have a range from zero to zero) will force RLPlot to rescale the graph after reading the file. Be sure that the flag-values of the default axes are set to autoscale. Probably automatic ticks should also be checked.

### Read ticks from different file

You may specify a file to define the ticks of the axes. See the Axis object description for more details. The file is a text file, each line defining a tick. The format is as follows:

File:

```
50
100= "hello world"
150=
200 a label
250
300= another label
350=""
```

Action:

```
minor tick at 50
major tick at 100, label 'hello world'
minor tick at 150
major tick at 200, label 'a label'
minor tick at 250
major tick at 300, label 'another label'
minor tick at 350
```

## Using Templates

The information in each RLPlot file contains all data to draw the graph. Thus, it is independent from any data file and every RLPlot file may be used as a template to create similar graphs from other data or to update a graph with changed values.

Data are loaded into the RLPlot-spreadsheet either by opening a file or by copy and paste from some other application. The RLPlot file to be used as template is opened. It will be in a separate window than the data automatically. Choosing [Edit/Update Values] from the graphs menu bar will update all items from the current graph with values from the current spreadsheet.

This behaviour is controlled by the 'ssRef', 'ssXrange' etc. items of the graph file. If these items are missing or invalid, values remain unchanged.

Exmple:

```
[3=Bar]
Type= 257
ssRef=(2){ 0 2 1 2}
Pos= 20 21
Size= 60
Org= 0 0
Line= 0.2 6 0x0 0x0
Fill= 0 0x00ffffff 1
FillLine= 0 6 0x0 0x0
```

This section describes a bar where the x- value is taken from spreadsheet cell A3 (index 0, 2) and the y- value from B3 (index 1, 2). Note that indices start at zero.

```
[4=Bar]
Type= 257
ssRef=(2){ -1 -1 1 3}
Pos= 30 31
Size= 60
Org= 0 0
Line= 0.2 6 0x0 0x0
Fill= 0 0x00ffffff 1
FillLine= 0 6 0x0 0x0
```

The bar described by this section is only partially updated from the spreadsheet. The y- value is updated from B4 (index 1, 3) while the x- value remains unchanged (index -1, -1) which is outside any valid data range.