

GraPL.net Release Notes

Updates to 6th December 2005 (Release 2.1h)

IndexOrigin – when set as zero, we should not prepend a dummy X-label to those the user has provided. Fixed.

Clipping – the July adjustment to the clipping rect has the bad side-effect of throwing away any markers which exactly hit the boundary. Now we apply the full cliprect to markers and risers and only trim lines slightly within the available space.

Curves style – applies to line and polar charts. Points are joined with a set of smooth (Bezier) curves rather than with straight lines. The flexibility setting is used to control how ‘bendy’ the lines are allowed to become, and can range from 0 (straight lines) through 3 (the default should be reasonable in most cases) to around 12 where things may start to go a little wild. See the temperature data in *PolarCurves* for a fairly legitimate use of this style.

Pie wedges ignored ‘Edge attribute’ colour in EPS output – always black! Fixed.

Polar charts gain style ‘hollow’ to stop the radial grid at the innermost x-tick leaving the centre empty.

Boxplots gain style ‘horizontal’ which draws the bars from left to right instead of vertically. Useful if you have fairly wordy bar labels. Several examples in the ‘Barley’ sample set now use this style.

HTML support in SVG – we now handle `<sup>` and `<sub>` correctly in SVG output. Also `<code>` switches to a monospace font here. Multi-line text including HTML failed if an open tag spanned several lines. We now correctly close and re-open the generated `` XML tags so everything pairs up as it should. The basic HTML4 entities are all supported, for example “`∑<i>x</i> + <i>y</i>`” will set a simple formula correctly. However not ALL these characters are in the standard Times and Arial fonts, so be careful what you use!

Tick styles – new alternative `XTickstyle = “Inside|Outside|Crossing|None”` for all axes. Will supersede `PTICK` and friends in the C# build, so included here for forward compatibility. NB ‘Plain’ stays as an axis style also.

Linespacing – removed from all text styles and added as a separate numeric property (`HeadingLinespacing = 120`) and so on. The numbers can be factors like 1.2 or percentages like 150 or simply absolute values in pts.

Effects – rather than being buried in the style, these are now added explicitly to the known fixed-text items such as:

```
mygrap1.AddEffect'heading' '<animate attributeName="x" ... />'
```

... which should be much clearer and easier to use. Also allows us to call ALL the style elements 'flags' in C# which makes VisualStudio very happy.

For this reason, *EquationStyle* loses the 'index=23' option which is now a separate *EquationPosition = 23* setting, taking ints from 1-100.

Weighted regression – adds a new 'Weighting' property which defaults to 1 and may be used to weight the points in a model fit. This would be appropriate if the data was aggregated, but with differing numbers of measurements contributing to each point. Also it may be used to exclude selected points from a model by simply weighting them as zero. For points with errorbars, you might consider a weight varying as the sqrt of the error to have more certain points counting higher in the model.

BubbleChart – simple cover on a *chScatter* with sized points. It works out a nice size for the markers and also sets the regression weights, as bigger bubbles should exert more pull in proportion to area.

ContourPlot – also a thin cover. It seems much more intuitive to pass x,y,z values here rather than use the Value tags to hold the height data.

HTML entities – support for text like "Hello Ω World" is improved. The semicolon is not treated as a linebreak in text following an ampersand until we hit either a semi-colon or a blank.

Paper – added as a property to shade the entire bounding-box. Useful for images where we often need to go right to the edge. Takes Colour,Fill,Edge as usual but note that only half the linewidth of the edge will be drawn.

Dialchart – adds the 'anticlockwise' style to run backwards.

Markers – a few changes to make a better spread of marker types. Two of the circles (markers 12,13) are drawn unfilled and six new markers are added to support hi-lo-close charts. These run from 16-21 and are simple ticks running left,right,up,down from the data point. For completeness we also have 20=- and 21=| to make cross-ticks in either direction.

Keys – allows the frame to be given as a percentage of the chart area now. the default is still 'absolute' as this makes no reference to the chart axes. the frame is given as (x1,y1,x2,y2) relative to the lower left corner of the chart area (so (0,0) is typically 4pt from the corner as the default gutter is set at 4).

KeyNudge – allows an arbitrary adjustment (in points) to any key position.

Wrapping text – YLabels and YCaption (Atend style) now wrap automatically into the available margin width if no formatter is given. The only other auto-wraps are the 'atend' Xcaption where the width is also known, and the heading, subhead and footer.

YLabels – moved a little right when set 'between' tick marks which improves the visual style of horizontal barcharts and also allows a touch more wrapping width.

Headings – were actually centering between the Y-axis and the right edge of the frame. They should centre over the chart (the X-Axis) which normally looks close enough that the discrepancy was never spotted. Until now.

ValueNudge – new property to allow fine adjustment of data value placement.

Spanned Xlabels – applied in general, not just for table headers.

Picture format – extended to allow '~' to mask out characters as well as digits. Most obvious use is to get JFMAM labelling on monthly charts using the MMM-yy date format and a picture like 'X~~XXX' to mask all but the first character of the month name.

Stepchart – now defaults to the 'floating' style to be consistent with linegraphs. To get a filled surface, use 'Surface' in the style setting.

Value tags – can now be 'angled' (default rotation at 34deg) and left-aligned at the data point.

Key frame – keys with an explicit frame and style 'boxed' now draw around the text, which is wrapped into the width available if required. Autofit works as before, but may be constrained differently, as the wrapped text may force more vertical space to be used. You can still use the key format picture to set a smaller width if required.

NextNote – could fail with chart co-ordinates and a specified YRange. Fixed.

Heading – a single-line heading was vertically centered within the top margins. Now just goes at the top which is probably a more predictable behaviour and inline with SharpPlot.

HBars – bars with all negative range (not forced through zero) drew the bars off the right end of the chart to where zero would have been. Fixed.

MinMax – value labels were misplaced if the lines were draw from hi to low, typically when showing the 'drift' style when they could go either way. Fixed.

YCaptions – were wrapped when 'atend' using the margin width even when you specified an explicit Y-axis position. This case now looks at the nearer edge and allows twice this distance as the wrapping width.

GroupBy and SplitBy – added as simple covers to set the ‘Groupby’ and ‘CategoriseBy’ properties so that the Grapl desktop interface can generate SharpPlot examples nicely! Probably adds a touch of readability anyway.

Boxplot – adds style ‘Tabular’ so we can still pass a bunch of categorised data as a matrix or array of arrays. This is assumed anyway if there are >3 columns but the 2 or 3-column case is ambiguous otherwise.

Pie – now simply uses the ‘explosion’ for the *Rose Diagram* with the addition of style ‘Rose’ – simplifies the C# call a bit and harmless here.

HBars – barcharts which could go either left or right from a centre axis got badly confused when the user had given text value labels. Now the missing value boolean is passed down along with ALL the bars (one each for left and right) so everything is compressed in synch.

Baseline – was not being applied consistently on both chart orientations. Also the interaction with intercept axes was not well-defined. Now the rule is:

- A *baseline* is an explicit instruction, so use it if given
- Failing that, if there is an *intercept axis*, use that as the baseline – N.B. Top or Right XYaxis counts as a special intercept setting here!
- Otherwise use *Zero (or the extreme ticks)* to run the bars from

Duration formatting – X or Y style ‘Duration’ will trigger base-60 formatting. No assumptions are made about the units, except that base 60 is used for magnitudes and : as the separator. ValueStyle ‘Duration’ will apply the same formatting rules to numeric value tags.

Piecharts – with VML extrusion the sorting for ‘Farthest first’ failed when the last segment wrapped over from 5:00 to the end. Fixed. Also JavaScript handlers set for specific segments were not re-targetted correctly (simple vectors of hints etc were OK). Fixed.

Multi-line Labels – no change if these were created with newline characters, but now we take one item per row of labels, rather than one item per label (with multiple lines). This makes much more sense of the labels have come from columns in a database (such as Months and Years) as the syntax is now just ch.Set ‘Xlabs’ (Months Years) which is very much cleaner. BUT this may break existing code, so watch out!

Histogram – rebuild of Frequency using the Stepchart which gives lots more formatting options and may be extended in future to handle variable width categories.

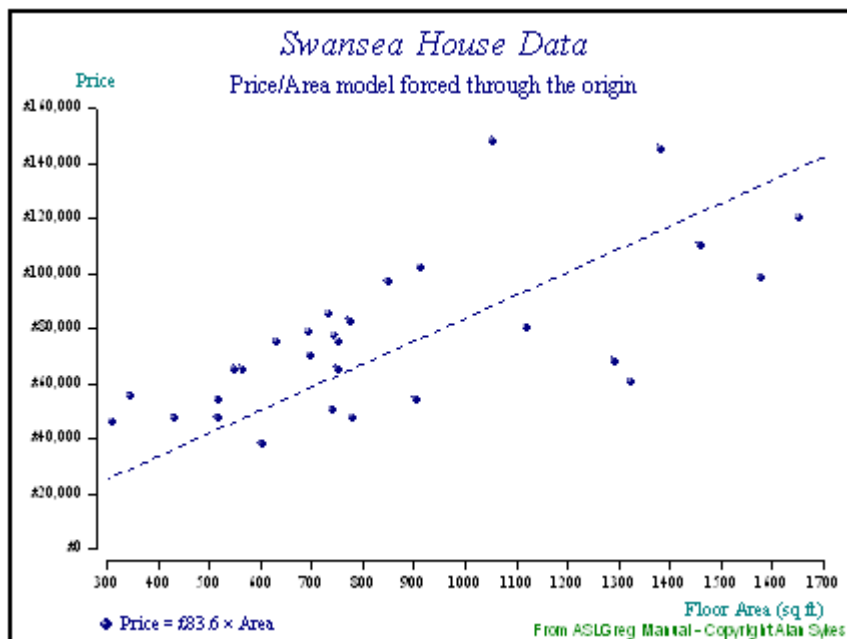
Updates to 12th Sept 2004 (Release 2.1g)

The last y-gridline was being removed with ‘framed’ axes even when the frame was above the final major tickmark. Fixed.

MinMax – calling this with a 3-element vector of (x)(max)(min) failed as it incorrectly patched on an extra column of x-values. Now works correctly for the vector case as well as the 3-column matrix.

Clipping – the clipping rectangle was set to the tick-mark range, not the total extent of the plot region. This is fixed, and also adjusted inwards for thick lines (or heavy axes) so that the clipped lines don't barge into the chart frame at the edges. For thin lines the effect is to leave a little white-space at the boundaries, which looks more like a hand-drawn plot, so is probably an improvement!

Dissected axes – a popular style in the RSS Journal, so we should do it.



This simply backs off all axes one tick-length from the plotting area. Applies to any of the rectangular 2D charts.

Datum lines – it seems reasonable to kill any datumlines and/or baseline if they fall outside the plotting region and the appropriate axis is 'clipped'.

Framed axes – should complete frame correctly if the Y-intercept is set so the x-axis falls below the bottom y-tickmark. Fixed. Also we should not remove the leftmost X-label if the y-axis has been moved downwards! Also fixed. Grid lines, zones and datum lines now extend down to the x-axis in this case.

SVG paths – could sometimes go negative and generate invalid SVG code. Fixed.

Updates to 29th June 2004 (Release 2.1h)

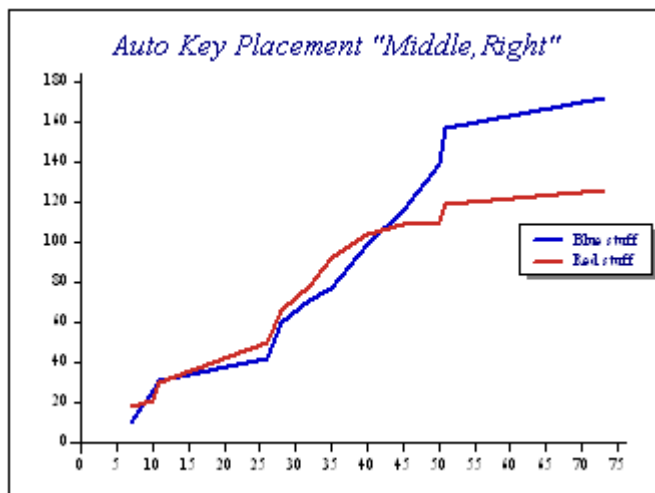
Dates – graphl now accepts integers in the form 20040625 as valid dates in addition to Excel-style ordinals from base 1/1/1900. The date range still begins at 1900 and runs to 2099 but this will be extended at both ends in future.

```

Arr_Entry_Date = Array(20021225,20030314,20040101,20041225)
Arr_X = Array(12,14,19,123)
graphl.Versus      = Arr_Entry_Date
graphl.XStyle      = "date"
graphl.Linegraph Arr_X

```

Keys – you may use the ‘left/center/right’ and ‘top/middle/bottom’ styles to get reasonable placement of the key box in the appropriate chart margin. A typical usage would be ‘middle,right,vertical’ which is the same as the Excel default key placement.



Obviously, you should adjust the margins appropriately for this style of key!

Updates to 26th May 2004 (Release 2.1g)

VML Markers – improved rendering using antialiasing with an adjusted line-weight in the same way as SVG. Also the diamond (mkr-3) was incorrectly filled with white. Requires a VML workaround to draw it unfilled! Fixed.

WriteKey – now accepts keystyle ‘percentage’ as a 3rd alternative to ‘absolute/relative’ to place the key on an arbitrary scale from 0-100 on each axis. The top-left corner of the boxed key is placed at the co-ordinate you give. y=100 will align exactly with the topmost tickmark. For an unboxed key, the text baseline is positioned at the requested y-value. Negative numbers (or numbers >100) will place the key in the chart margins.

Keys – positioned keys notice ‘right/centre’ and ‘bottom/middle’ and place the required corner of the keybox at the co-ordinate you give. This allows for sensible combinations like x=95% with style ‘percent,right’ and so on.

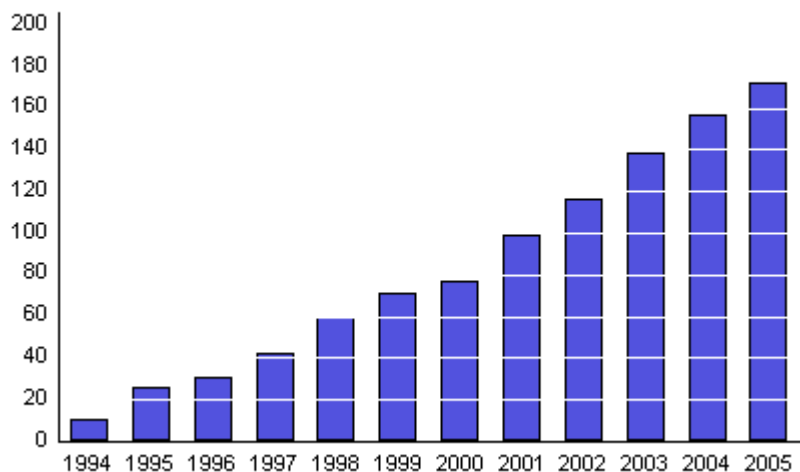
Duplicate X-axes – the code to thin out overcrowded labels was not being used on the upper axis when the ‘duplicate’ x-style was set. Fixed.

Value tags on horizontal bars – tags with style ‘inside’ are supposed to skip out of the bar if the space is too small for the text. This was not working correctly for negative bars (drawn left of the Y-axis). Fixed.

Updates to 6th April 2004 (Release 2.1f)

Overlay grid – new style to get the x/y grid drawn on top of the bars. Can be set as style, xstyle or ystyle. Tufte-style white grids are getting popular so we really should do them. Currently barcharts only – will propagate to all charts as time permits.

Overlaid Gridlines



JavaScript fragments on categorised charts were being replicated by the number of categories! It worked fine, but made the file stupidly large. Fixed.

Annual scale – tickmarks were always defaulting to 1-year. They now thin out ticks as per any normal X-axis. Ticks never get drawn at intervals less than one here – the ‘Annual’ style implies this.

Secondary Y-axes – should reset ‘Baseline’ to default. The incorrect behaviour shows as an extra baseline which is drawn by the secondary plot. Fixed.

Notes/Key – shadows done rather better using opacity, as the newer IE versions draw these very nicely in VML now. SVG also updated to match. This also applies to the optional drop-shadow behind the entire chart.

Updates to 3rd March 2004 (Release 2.1e)

Trellis – should log outer chart area as boundingbox if called when we have not drawn anything. Otherwise the SVG and VML output does not get the correct pagesize unless you use a full row and column of cells.

Indexed data – data indexed to 100 did not respect missing values (unless the missing value was 0). Fixed.

Tables – Rounded corners now look at the shorter side and use a proportion of its length as radius. Works better for really big tables!

Dialcharts – ticks now come less far into the centre if you don't have annular warning zones. Makes the labelling a lot less crowded. The final outline is drawn last to make sure it goes over any ticks etc.

Hyperlinks from the chart frame were becoming a pain, and were badly implemented. Removed, and will be done right next time they are wanted!

Scripts – Javascript fragments were not being treated in quite the same way as hints, tips and hrefs. This showed when you had script fragments on a categorised chart, or used a 'where' setting to mask out selected data items. Fixed.

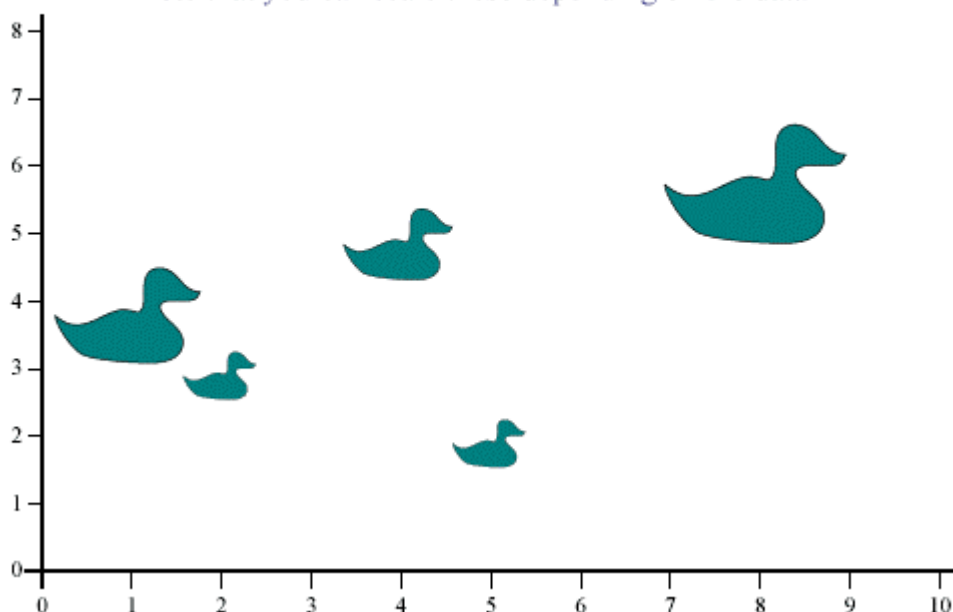
Include – may be called repeatedly to include several items. Makes for much more readable code.

Updates to 19th December 2003 (Release 2.1b)

Inline markers in SVG – symbols defined with `grap1.Include` may be referred to by `#id` in the marker definition and will be placed and scaled in the same way as external image files. The marker height and width default to the `viewBox` of the symbol, but may be given to set the 'base' size of the marker, which will be used in the chart key and for unscaled markers in the chart.

SVG Symbols as Markers.

Note that you can scale these depending on the data



The aspect ratio will be preserved under scaling, but you should try to keep it the same in both places to avoid clipping. The Lineplot sample set now includes an example.

Markers in keys were always given the 'halo' effect. Normally harmless, but shows when your chart is drawn transparently over a pretty backdrop JPEG! Now the key follows the 'halo' attribute of the current marker set.

Image markers in SVG – these were assumed to be GIF images, so were sized assuming pixels, and not scaled with markerscale. All markers are now scaled, as the SVG viewer handles this very well, even for images. You may, of course, refer to an external SVG graphic here as an image.

Updates to 16th November 2003 (Release 2.1)

Stepcharts – these are designed to expect one more X-value than Y-value, but if you gave them vectors of the same length, there was an inconsistency between the filled and floating types (floating offset to the right by one X-value). Fixed.

vml/svg bitmaps – updated to use the second part of a split filename as the content for the Xlink tag, as this is how the client should ask the server for the file. Clearly it is not likely to be the same as the local path to the file, and could even be a different filetype. Filenames are separated by the pipe character here.

Clipped lines – a clipped line with a trend failed when the entire line was off the plotting region. Fixed.

Notes – will use the 'Background' property as an alternative to 'FrameAttr' now that we have it. Will be documented as the correct way to do this in future, but old code will keep working.

Values – value tags are now run as a single group (rather than one group per possibly multi-line tag). This is faster to render and improves the SVG filter handling, as the effect is now applied to the entire group. Multi-line X-labels are also now grouped as above.

Zero pie sectors – drew a 360 arc for the PNG format. Fixed.

Tables – if `grap1.Table` is called before anything else, it now sets the chart area to fit the table. This means that margins, headings, footnotes and so on are not used in this case. However the 'gutter' is respected. This can be used to make simple tabular output in SVG for inclusion in webpages with `<embed>` tags correctly sized. To get the old behaviour, insert a call to `grap1.Null` before the call to make the table.

Tickmarks – scaling with axis line-weight now more friendly (square-root) relationship so ticks stay reasonable with very lightweight axes and do not get stupidly long with very heavy lines (possible for graphs rendered on REALLY small devices like mobile phones).

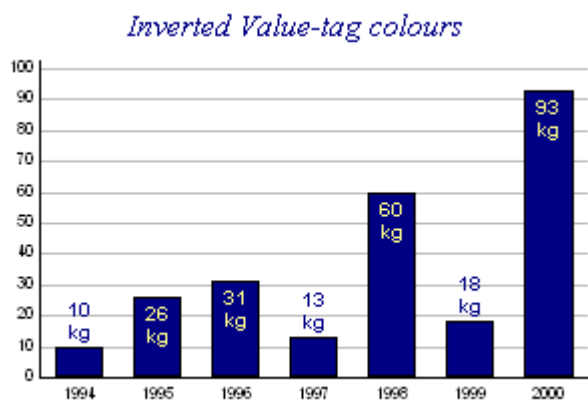
Cloud with modelfit – if forced through the origin, failed when the x-fit was linear and the y-fit quadratic or higher. Fixed.

Table – now uses style 'Rows' to decide if a simple vector is treated as a column vector (default) or a row-vector.

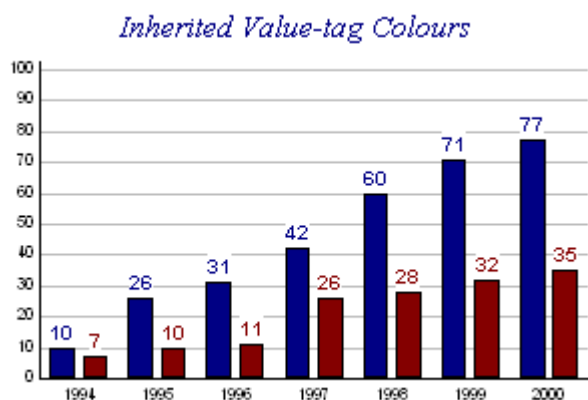
Xbar – failed when used to plot a matrix with a scatter of missing values. Also did not handle missing values in the 'stacked' case correctly. Fixed.

Secondary Y-axes – left-aligned caption on a secondary (or intercept) y-axis stuck to the left edge of the plotting space, rather than following the axis. Fixed. Also multiline y-labels broke if we used the XIntercept setting, as the code to eliminate the label at the crossing point assumed one text item per y-tickmark. Fixed.

Values on barcharts – when an 'inside' label hops ontop of its bar (due to lack of space in the bar) we now have the value-style 'invert' which switches the label to the bar colour. This makes sense if you have been using light-coloured labels on dark bars. Also some bugs fixed in this area. Multi-line labels only hopped up by one line. Fixed. Vertical labels on stacked bars ALL hopped up – only the ones on the topmost bar should do this – the rest should just disappear if there is no space in the bar. Fixed.



New value style 'inherit' colours the tags to match the (solid) colour of the bar the tag goes with. Also applies to marker values on linecharts and on Max-min plots and Stepcharts.



SVG titles – were not being given the &# treatment which upset the viewer for certain German characters like ü which is now ü and works fine.

ID generation – series left to the default name (Series1 etc) were incorrectly creating id="Series1_C1" attributes in VML and SVG output. These should only be included if the series has been explicitly named by `grap1.Series`. Bug introduced with the April 2003 build. Fixed.

Adds support for hint/tip/href/jscript settings on several new rectangles. Not all of these work on all environments yet (PDF does href only) but you can have jumps and tips in SVG and VML on (for example) the chart frame and XZones.

Character markers – using hi-ascii dingbats in PDF output (e.g. the Heart symbol) should represent the character as \octal in the PDF stream. Fixed. Also the hi-bit characters were slightly misplaced horizontally in VML. Fixed.

SVG keys – adding effects to lines really should put the effect on the key. Due to an SVG viewer oddity, if a polyline has zero y-dimension (even though it is quite a thick line) adding a filter simply hides the line. Now we make a little undrawn T-piece at the end of the line in the key, and it works nicely.



Tables – adds a new style 'filled' which will use the current settings of Colours and Patterns to shade individual cells. This could easily be used to implement data-driven colouring by setting up either Colours or Patterns as arrays matching the shape of the data.

Table Cells with Colour and Pattern Fill

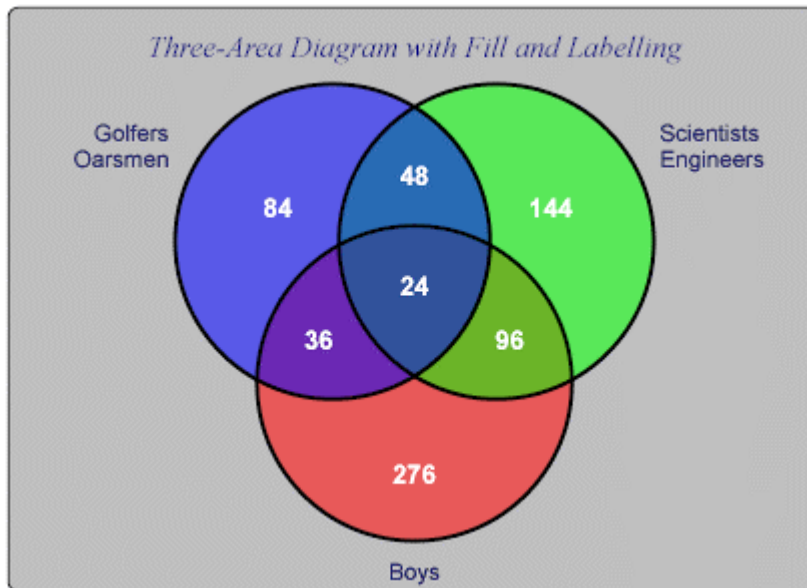
	North	South	East	West
Coats	25.00	58.00	88.00	110.00
Hats	22.00	43.00	60.00	75.00
Gloves	19.00	22.00	25.00	28.00

HeadingNudge – simple addition to heading placement. Nudges the heading (and subhead) the given (x,y) distance in points.

Trendline – was a bit of a memory hog when used on a large timeseries. Now loops rather than using outer product to get weights for Gaussian. Seems at least as fast and avoids "out of memory" errors.

Venn diagrams – extended to 3-areas with a non-proportional diagram simply showing 3 overlapped circles, appropriately labelled with up to 7 values. Circles may be labelled with X-labels (radial placement) or keyed in a legend. Filled circles are allowed, but only

work well in SVG and VML formats as they rely on opacity to colour the overlapped regions correctly.



Tables – hints, tips, hyperlinks and Javascript expressions may now be set up on table cells in the same way as for normal charts. An array of links (matching the shape of the table data) will be applied to the individual table cells. Note that the entire cell rectangle is set as the ‘hotspot’ for the action here, not the cell text (which could be blank for a zero value).

Multiple – was ignoring ‘CategoriseInto’ for the splitting of the charts into the trellis (or pages). Fixed.

SVG styles – the original SVG viewer required any CSS properties to be buried in a `style="font-size:nn; ... "` tag. This is ugly, and makes it really hard to modify selected attributes with JavaScript. The newer viewers are happy to take all these as first-class SVG attributes, so they are now recoded as `font-size="nn"` in approved XML style.

Event-handlers – extended to any ‘boilerplate’ elements such as keys, headings, axis labels and so on. A reasonable use of this would be to have mouse-over on each key highlight the corresponding line in a timeseries.

Formatting – the picture-format capability is now extended to cover basic scientific formats such as ‘0.000E00’ or ‘-#0.00e+000’ and so on. This applies wherever a picture format can be set, so axis labels, values etc.

Gantt chart – notices ‘Categorise By’ and ‘Categorise Into’ settings and cycles through colours and patterns appropriately. Key entries are auto-generated as usual if no key has been set by the user.

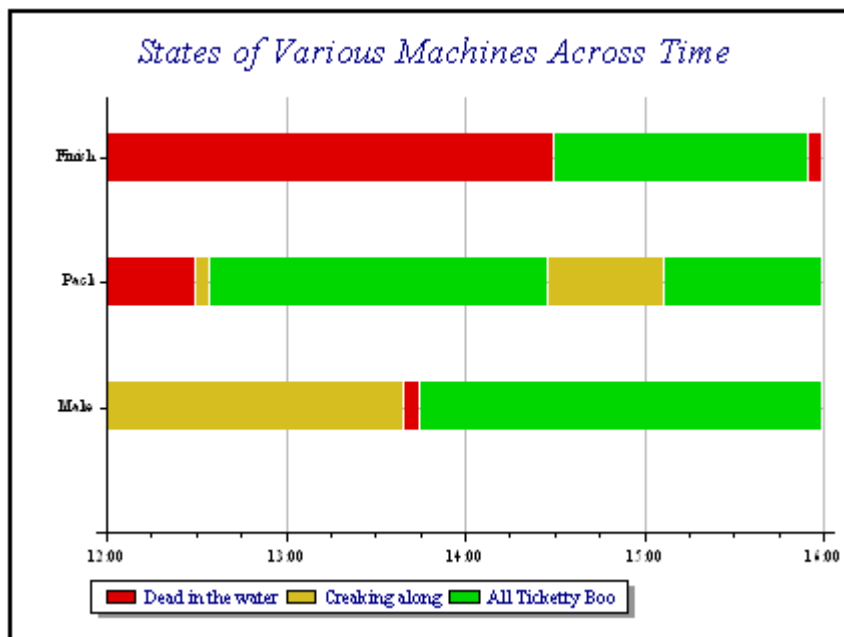


Table chart – was treating a 1-row vector of enclosed elements as a column. This should disclose to 1-row matrix. Fixed. Also this is now more tolerant of a scatter of text cells in mostly numeric input. These were treated as if they were missing values, but now they simply pass through verbatim. Currently there is no way of aligning text and numbers differently – valuelstyle ‘left’ applies to all cells.

XLabels – labels with multiple lines were not always getting ‘thinned’ correctly for the second and subsequent lines when the thinning on the first line eliminated nearly all the labels (the first line labels were much wider than the interval between tickmarks). Fixed.

Updates to 14th May 2003 (Release 2.0p)

Boxed notes – the box depth was not adjusted for font pitch! Fixed.

Plotting null arrays – the code to notice these jumped out too early and failed to clock the various cycles of colours, styles etc by the number of columns passed. It should also generate an entry for the key, even though nothing is actually plotted. If you have a ‘trend’ or ‘modelfit’ set on an empty chart, these clock forward similarly to keep the key consistent. Same issue with singletons when no markers were requested. Fixed.

CaptionFormat – just to let us set a ‘picture’ in X/Y/Z axis captions to force these to get wrapped if required. Only applies to ‘attend’ captions.

Stepchart – vertical value labels were not centred on the steps. Fixed.

Series names – nasty issue on scatters and cloud charts where if you plot two independent sets of data with hints and tips, you always see the hints from the first series unless you use `grapL.Series 'ONE'` and `grapL.Series 'TWO'` to give them unique names. Fixed.

Span=x1-x2 style on model fitting. In the simple case ('style' 'model,span') this sets the fitted line to span the range of the data, rather than the X-axis. If you give one number, it is taken as the top value and the regression line is drawn from the first data point to this. Two numbers 'span=1:7' will set both ends.

Cloudchart – markers were being scaled (by projecting a unit cube and using the longest projected edge) for perspective, but then the computed scale was being rounded to 1dpl which meant that they were stepped in size rather than evenly graduated. Also, each series was being normalized independently so several sets of data (say one cloud at the far corner and another nearby) were being drawn at the same size. Now it all scales absolutely relative to the unit cube at the (0,0,0) point.

Barchart – if a barchart had no labels, tickmarks and possible gridlines were not drawn. Applied to vertical bars only – HBar was fine. Fixed.

NewFrame – scales fonts and lineweights down in proportion for small charts. However, it was just averaging x and y scale here which has the nasty side-effect that for a stack of 'letterbox' charts of slightly differing heights the axis lines can vary in weight, which looks really messy. Now we use the old scaling rule for fonts, and the line-weights get scaled by whichever of the x and y-ratios is bigger.

WriteKey – if called with a text argument (such as '') it simply flushes any existing key definitions without writing anything to the chart.

Effects – allow JScript handlers on headings and other labels using defined effects and `headingStyle = "effect=nn"` and so on.

Captions and Labels – 6 new properties allow selective setting of the fonts for these by axis. This is effectively an over-ride, so for example:

```
LabelFont = "arb,14"  
YLabelFont = Lime"
```

This will first set the x and y fonts to 'arb,14' (black by default) then selectively change the colour of the y-labels.

Table charts – simple table-formatter using X-labels for the coltitles and Y-labels for the row-stubs. The value style and format picture control the appearance of the cells.

Summary Table of Barley Data

This is a typical use of a table to add a cross-tab as a chart element.

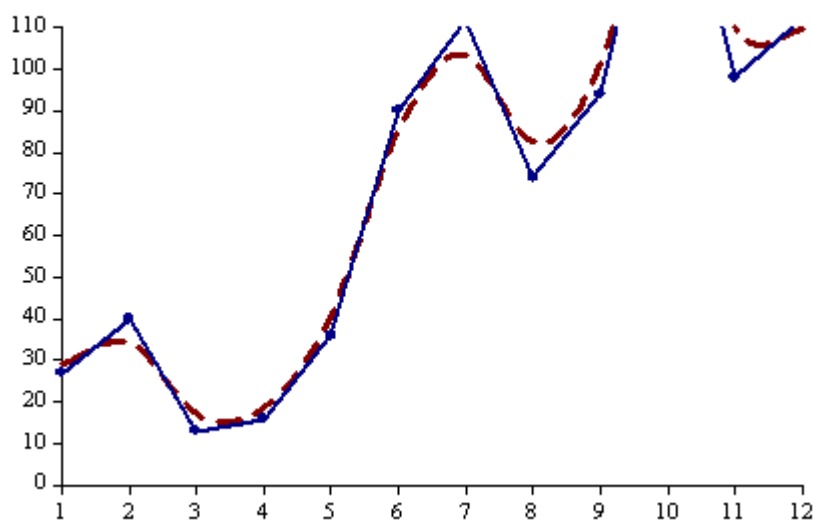
Mean Yield by Variety & Year	1931	1932
<i>Glabron</i>	37.3kg	29.4kg
<i>Manchuria</i>	34.2kg	28.7kg
<i>No. 457</i>	40.3kg	31.4kg
<i>No. 462</i>	39.1kg	31.7kg
<i>No. 475</i>	31.8kg	31.7kg
<i>Peatland</i>	36.6kg	31.8kg
<i>Svansota</i>	34.0kg	26.7kg
<i>Trebi</i>	42.5kg	36.3kg
<i>Velvet</i>	34.5kg	31.6kg
<i>Wisconsin No. 38</i>	40.6kg	38.2kg

Both cells and labels may be wrapped and the coltitles may be spanned over more than one column. Intended use is to add simple summary tables to charts, for example see Barley which tabulates the mean yields to go with the Boxplot of the raw data.

Venn diagram – simple Venn Diagram for the 3-area case. May eventually be extended for more complex examples.

Clipped axes – style 'clip' used on the chart or on either axis will now clip the data at the chart boundary as well as truncating the axis line.

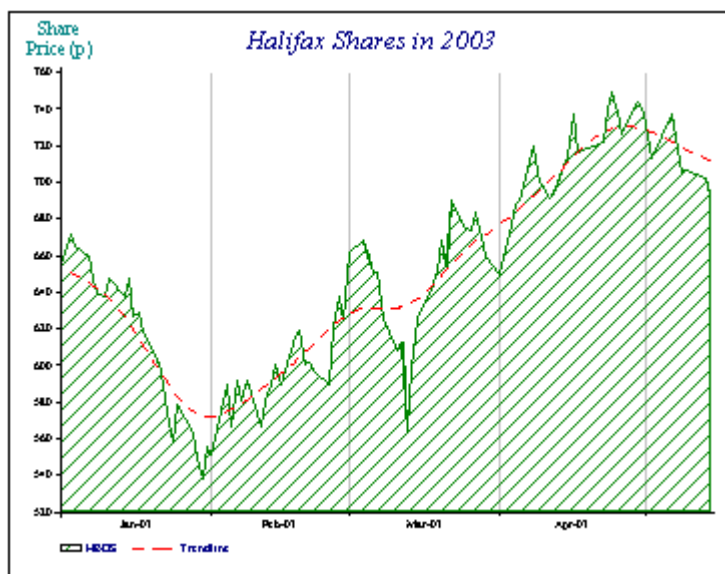
Data Clipped at Axis Range



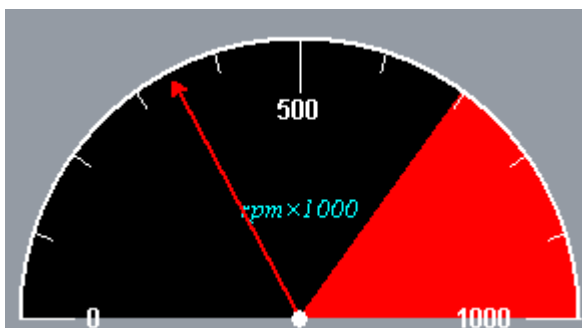
This is what it was always meant to do, but the poly-line clipping code is quite tricky! Where a line exits the plotting region and returns, it is drawn as two separate line segments. Any markers which fall outside the clipping region are scaled to zero and will not be drawn. If you have 'surface' set, the line is simply 'clamped' to the boundary and the region filled as usual.

Missing values – Scatterplots were not screening out missing values in the X-dimension (col-1 when style is XYPlot). Fixed.

Autorangeing – styles 'monthly,exact' and 'annual,exact' now work as they should on the X-axis (only complete months/years are labelled). We treat a month as complete when data is available for the last week, and a year when data is available for December. A bit subjective, but seems to give reasonable behaviour with typical timeseries.



Background – new property sets the background attributes of the plotting area (bounded by the axes) with colour, pattern, edge-weight as normal. This is drawn underneath any zones, axes and so on. It extends to the last tick-mark in each direction, so looks neatest when used with styles 'clipped/framed'.



Examples *Polar* and *Dial* updated. Dial charts now use Background rather than 'FrameAttributes' for the colour inside the dial. Much more logical, as you may want to set both, but could affect existing charts, so check carefully.

Barcharts – the ‘zones’ and ‘background’ were drawing after any gridlines, so could partially or totally obscure them. Fixed.

Using calculations in charting expressions

This could lead to problems when the ‘engine’ mistook text data for an expression, an example being a string such as “1/2/2003” set as the X-label property. Unfortunately, this can be evaluated, and hence give rise to a very unexpected result.

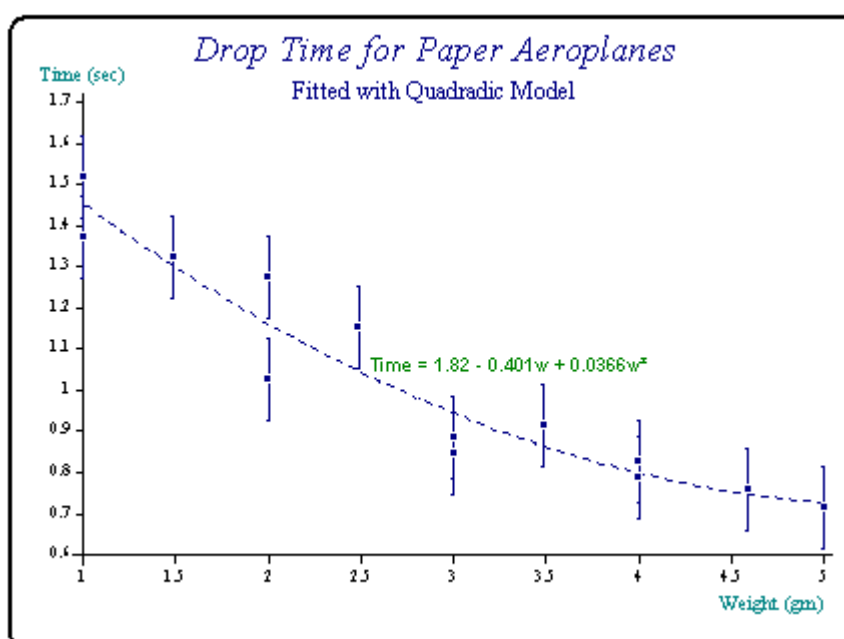
Rather than force the programmer to quote strings such as this, the rule is now that *only expressions beginning with the backquote ` character* are evaluated. This changes the examples in the help file, for example to plot a moving average of data stored as ‘MySeries’ with `grap1.SetValue`, you must now use `grap1.Linegraph "`MySeries;12 Mav MySeries"` whereas you could previously omit the backquote here.

Updates to 24th January 2003 (Release 2.0m)

Piecharts – if the axis line weight is set at 0 the spider lines really should go away, not be shown as hairlines. Fixed. Also, either of NoTick/Plain should kill them off. It does now. Also we have a more flexible way to add values to the ‘xlabs’ here. If the ‘ValueFormat’ is “The XL is ###0.0” then the value of each xlabel will be substituted into the value format. This is rather neater than the default of “xlab : value” which is the current behaviour.

Model fit – some new properties to help write the equation nicely. These are to set a ‘picture format’ and to position the equation. For example in Aero we have:

```
grap1.EquationStyle = "index=40,above"
grap1.EquationFormat = "Time = C0 + C1w + C2wφ"
```

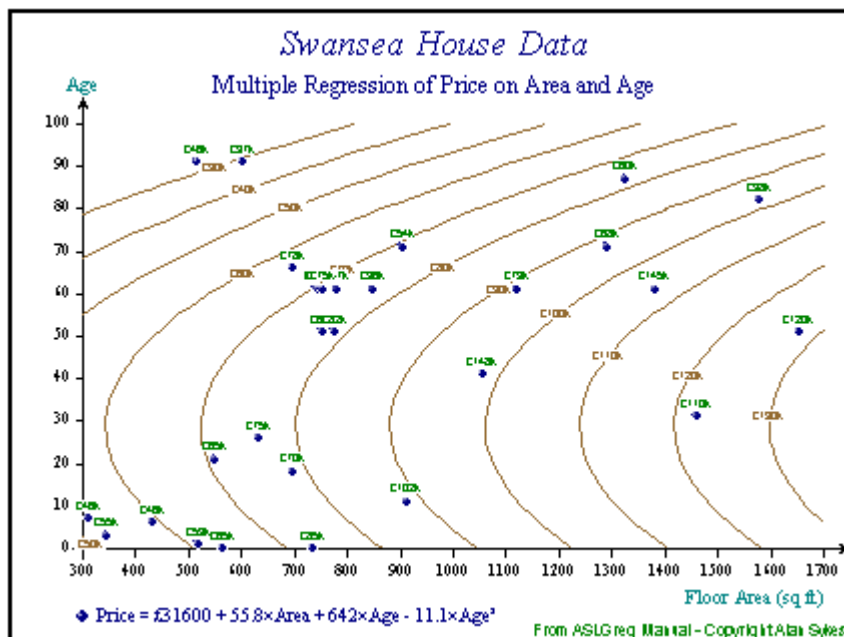


You can have 'left/middle/right/attend' for positioning, as well as explicitly setting `index=1-100` which goes `n/100` of the way along the fitted curve. The default is 'above' or you can set 'below/straddle'. The default is 'transparent' but 'opaque' is also allowed. The text is aligned left or right depending on the slope of the line at the point where you put the equation.

Hopefully the picture format is obvious! C0-Cn are placeholders for the coefficients and '+' is a placeholder for the sign. If a term is nearly 0 (within 1E-10) then it is omitted. If it is exactly 1 then the value is omitted. You can query the equation (after the chart has been run) with

```
eqn = graph1.Equation
```

This returns the formatted text with the coefficients inserted. An obvious place to put this would be the key! Note that for multiple regressions (Cloud charts and Scatterplots with the 'contour' option) the coefficients are generated in the order `Int,x1,x2,...,y1,y2,...,y4` depending on the order of fit chosen for x and y. Set the `equationstyle` to include 'forcezero' to cut out the constant term and force the fit through the origin.



Notes – you can set the vertical spacing of the text in the note-style with the syntax:

```
NoteFont = "arb,12"
NoteStyle = "left,pitch=15"
```

... to over-ride the default line-spacing which is `1.2×fontsize`. For fonts such as Arial, you may find a more open spacing looks better. As an alternative, you can set a percentage of the current fontsize (for example "pitch=130%"). This applies to anything which can be multiline, so headings, captions, x/y labels, keys, values and pagelabels. To cancel a previous setting, set 'pitch=auto' for any of these.

grapl.NextNote(vs) – returns the xy position (chart/absolute co-ords) where the next note should be written. This is the position of the last line of the previous note, spaced down by the current line-spacing (default is 120% of the font size). This allows you to write several blocks of free text as notes, spacing them down the chart correctly. To add extra vertical space (e.g. between paragraphs) pass the required vertical space (in points, always) as argument.

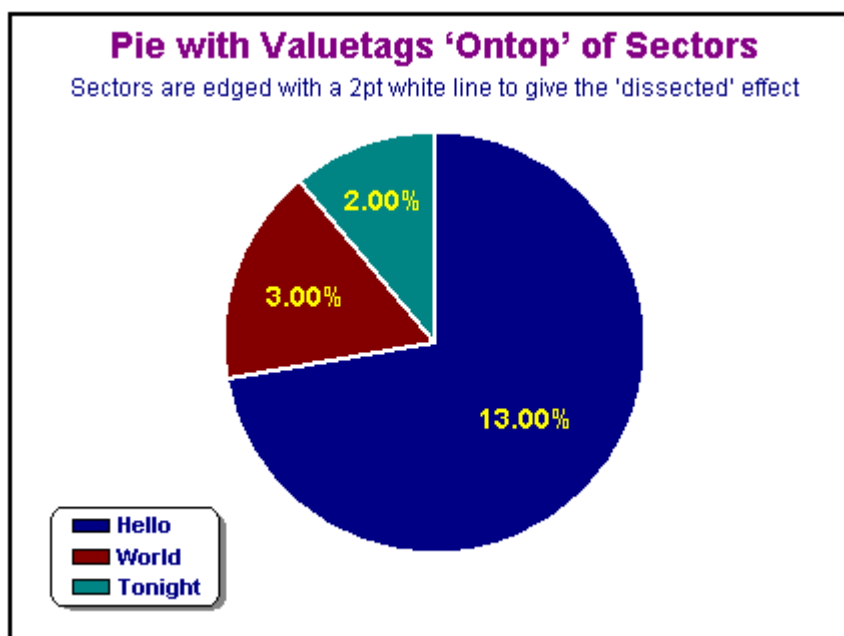
Contours – the labels are now formatted with the same picture setting as the value tags (since they must come from the same dimension).

Updates to 12th December 2002 (Release 2.01)

VML,PDF,SVG – you could provoke some odd effects by setting an explicit y-range of (say) 0-10 then plotting a vector including a number like $-1.1E35$ in the series. We now limit the edge of the paper to around 10000pt or about 12ft square which means that the lines still spike off the visible region, but do not actually try to draw all the way to Mars.

EdgeAttributes – new property so we can set the colour and edge weight of piecharts. The line-style is accepted, but pies will ignore it until someone asks for a pie with dashed edges!

ValueStyle 'inside' – now works on piecharts. This writes the values in the sector (at the approximate centre of gravity) rather than appending them to the end of the X-labels. They are always written with 'transparent' text.



Y-captions – places top caption under top tickmark, unless we only have one! In this case it just uses the chart margins to find the best place. Only relevant to plots of unranked datasets like a scatter of 0 versus 0!

Updates to 19th November 2002 (Release 2.0k)

Tickmarks in VML – IE6 has a new rendering bug with non-parallel tickmarks. This badly affects Dial charts and also can affect Polar charts. Worked around by drawing each tick as a separate Polyline element in this case. Also, we now selectively enable anti-aliasing for angled lines, sectors of pies and circles, which makes these charts a lot smoother.

Nulls – these were correctly mapped to the current ‘missing value’ when they occurred in the argument to any plotting function, but were allowed to persist in properties such as ‘Versus’. Now any ‘array’ type property is screened for nulls in the same way.

Stepchart – was clearing the ‘Versus’ setting and NOT putting it back again. The error showed when you followed a Stepchart with a Linegraph and found that you had to reset ‘Versus’ to what it was before. Fixed.

Piechart – survives pies where the sum of the sectors is zero. This can happen either if the data is null, or there is one value (of zero) or if there are negative values so the sum comes to zero. None of these cases is valid, so we make an open red circle with a diagonal bar through it and give up.

RenderVML – was always forcing a 12pt margin above the chart. Now only does this if the *ShowPaper* flag is non-zero.

Gradients – the way all the built-in gradients work is to ‘lighten up’ the base colour by stirring in some white to make the target colour for the highlight. If you want to grade to an arbitrary colour, we now have the syntax:

```
grap1.DefineTile 77,"=34+Fuchsia"
```

This will use the pattern numbered 34 (numeric pattern ids only here) but will grade towards ‘Fuchsia’. Of course hex values are also allowed here as a way of specifying arbitrary colours. An alternative is to override the amount of white stirred in. The default is to grade to a target of the base colour + pattern-3 so ...

```
grap1.DefineTile 78,"=36+1"
```

... will give a much more intense ‘spotlight’ than normal here. This may get extended in the future to allow you to adjust the focus-position of radial gradients and so on, but the differences in the way gradients are specified for SVG, PDF and VML makes this a little tricky.

Polar charts – minor ticks were not being computed. These now behave as you would expect, so *XTickmarks=array(3 2)* will give ticks every 3 hrs when the X-style is ‘time’, with minor ticks on the hour. The default is to treat x-values as proportions here, so on a directional plot, you could use a setting like *XStyle="compass",XTickmarks=array(10 4)* to get ticks every 10 deg with 4 minor ticks in each interval. Y minor ticks are generated as for a conventional y-axis, but are always drawn centered across the axis.

Min-Max plots – two simple extensions. You can now use style *'terminated'* with this chart, which simply uses the current marker symbol at one or both ends of the line. Style *'drift'* limits the terminators to the 'far' end of the line and also works with the existing *'errorbar'* or *'arrow'* styles to make a directional line.

Updates to 3rd October 2002 (Release 2.0j)

Keys – should tolerate numeric values in the key 'text', as text which looks executable (such as a single key with the value "5") will get treated as a number and currently causes the key generation to fail. There is no reason in principle why you cannot pass an array of numbers as the key values here, so this will now work correctly too.

Missing Values – extended to allow a vector of values, any one of which will eliminate the matching point. This allows for the behaviour of many common data-loggers which may produce a variety of 'invalid' readings depending on the particular error-condition raised. Polar/Dial charts were not eliminating missing values at all, and Tower chart was confusing the colour cycle when there were missing towers. All are now correct.

Piecharts – were removing missing values, but should also mask the corresponding elements in *Explode* and *Radii* properties, as well as in the X-labels if these are set as spider tags. Fixed.

Barchart – was removing normal horizontal labels when style was 'inside' and the bar was too short to accommodate the text. Now the label skips on top of the bar (as for vertical labels) if this happens.

Locale – new property to allow a script to format dates and times in a non-default locale. This is required if you are running a web-server where pages are created in the language settings appropriate to the locale passed in the HTTP header. Normally, dates and times are formatted using the locale settings of the server, which is not correct in this case.

Locale (again) – was not handling unicode values quite correctly, so the string for 'AM' in (say) Korean was not coming through to SVG as `오전` which should now be OK.

Unicode characters – it is no longer necessary to 'escape' semi-colons in heading text if you require Unicode characters expressed in the entity format such as "Hello `–` World". The first semi-colon following any occurrence of "`&#`" in the text is always passed through to the final output. See above for the reason!

Updates to 16th September 2002 (Release 2.0i)

Response surface – adds in the 'Contour' option to come in line with Cloud. The 'density' is ignored here, as you effectively forced it to be 1:1 by passing the entire grid of data as the original argument.

Value tags – multiline labels were not being split out for the ‘vertical’ style of labels. This now works as you would expect – the labels are horizontally centred in barcharts, but for scatter plots the default is to write the lines to the left of the point, leaving the last line in the same place as if it were a single-line label. Use style ‘centre’ to have the labels centred above the data points.

Updates to 30th August 2002 (Release 2.0h)

Postscript symbols – the ‘WING’ font now maps out automatically to /ZapfDingbats when the output is rendered to PDF or EPS. Many of the characters are in different places, but this is relatively easy to code around when you know the target format.

Barchart – a single bar with a scalar numeric label broke the labelling code. Also in Linegraph, but less likely to happen here. Fixed.

Updates to 18th August 2002 (Release 2.0g)

X-labels – if a multi-line set of labels was too cramped, only the first line was being thinned to avoid collisions. Fixed.

Tower charts – multi-line X and Y labels now work as you would expect. Use the line-break character or pass an array of arrays of strings to generate multi-line labels here.

Notes – chart notes tolerate numeric values, as in:

```
grap1.Note "4321",12,14
```

This was causing an error, as any executable string is checked for and automatically executed, as it could be a variable set up with *SetValue*. This seems too dangerous, and of little benefit, so we no longer do it.

Scatter – when a ‘where’ property has been used to compress the data, the associated ‘values’ (also hints, tips, hrefs) were not being masked to match, and so were out of line with the plotted points. Fixed.

Early warning of change – we are thinking of using patterns 16-39 to be various canned effects like gradient fills and transparency settings. This means that anyone using *DefineTile* should make sure that the tile id is >40 as lower numbers may produce the appropriate canned effect, rather than the specified bitmap!

Value tags – text tags are now formatted by the *ValueFormat* setting in a similar way to numbers. Typical would be ‘XXXXXXX’ which simply restricts the length – longer items are trimmed to one less than the length and an ellipsis character (...) is added to the end. In addition, insertions are supported in the usual way, so a picture like ‘SKU:XX/XXXX’ would format codes such as ‘HK1324’ nicely.

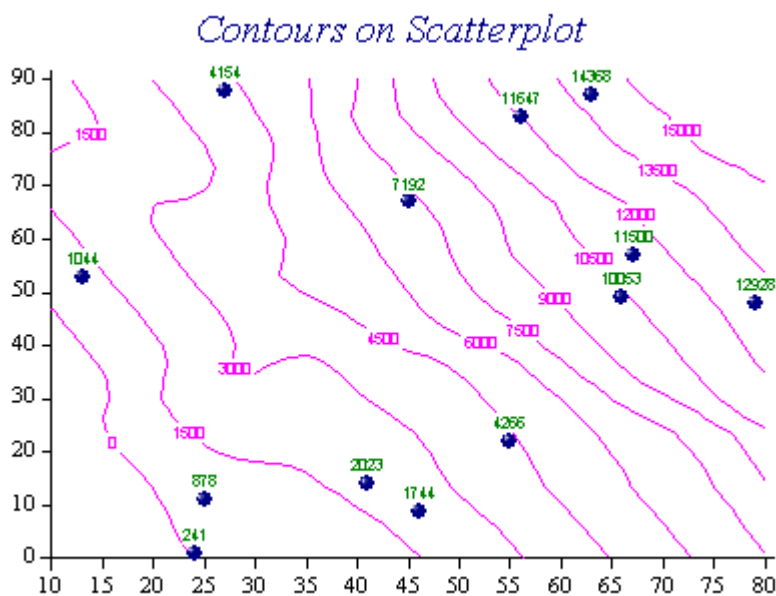
Notes – the y-position was not being negated for inverted y-axis, so notes were misplaced vertically in this case. Fixed.

Updates to 5th July 2002 (Release 2.0h)

Bitmap, Metafile – accept whatever we give them now, as IE can view most image types, including Bitmap and Metafile. PDF can only accept bitmap here, so error-traps and quits if the filetype cannot be processed.

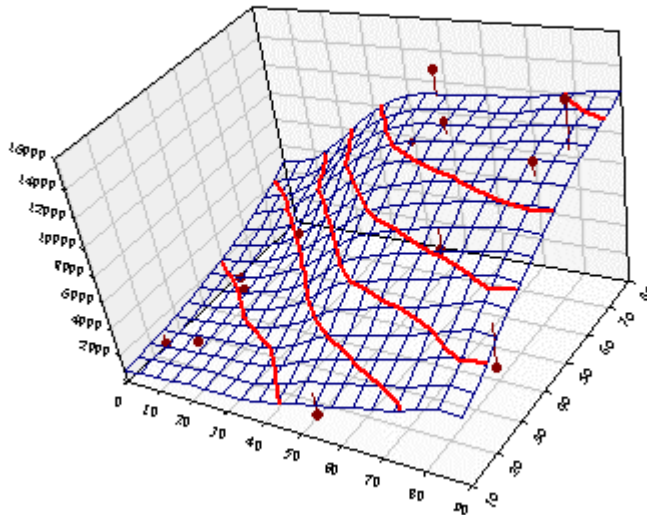
Tiled surfaces (SVG) – if the tile pattern is user-defined it fills and edges the tile as separate operations. This allows you to make a mesh surface with semi-transparent tiles, but have the lines drawn normally.

Scatter – if ‘values’ have been set for all datapoints (as numbers) this will now notice the ‘contour’ style and the ‘flex’ and ‘ZTick’ settings to draw contour lines based on the values. This allows a standard 2D scatterplot to show a map of values in a 3rd dimension.



Cloud – implementation of fitted response surface using weighted exponential and modelfit styles ('trend' and 'model'). Adds a *Density* property to control the number of mesh lines drawn and a *Contour* style to get contour lines drawn at Z-tickmarks. Note that the 'risers' style will draw the risers from the fitted surface if there is one. You may also use style *Projected* to get the contours drawn on the xy-plane. New property *ContourAttributes* "Red,solid,1.2" controls the appearance of the contour lines.

Trend Surface with Contours



Linegraph – style 'risers' brought into line with the 3D charts. It now plots the risers from the regression line to emphasise the residuals if style 'modelfit' has been used.

Linegraph – adds new properties *Fitted* and *Residuals* which are read-only. These are used to store the fitted values and residuals after a modelfit, so that you can do some further processing or plotting without having to work these out again from the coefficients. Note that missing data values will be reflected as missing fitted values or residuals.

Step – takes note of *BarEdge* property (default 0.4pt) which allows you to set this to zero to get unedged, filled regions. This is a useful alternative to *XZones* where the beginning and end of each zone is data-driven, rather than fixed at design time.

Updates to 14th June 2002 (Release 2.0g)

LineGraph – setting *xstyle* 'log' did not correctly log the data when the style 'xyplot' was used, rather than setting the 'versus' property. Fixed.

Stepchart – sets the 'crisp-edges' flag to ensure edges of step-charts are not anti-aliased by SVG renderers.

Keys or Legends – new 'autofit' style which steps down the font-size until all the key text fits into the allowed frame. Gives up and just overflows when the text reaches 3pt

which is as small as we can read it. This works with all the other key-styles, e.g. reversed, vertical.

Wrapped notes – the recent change to metrics calculations could occasionally put the wrapped notes algorithm into an infinite loop. Fixed.

JavaScript – always protects the script as CDATA just to be on the safe side. We can accidentally make ill-formed XML otherwise. If the script does not contain any '<' characters it is taken as a filename and a link is made to it, rather than embedding the entire content.

Surface shading – was always setting 'crisp' edges. Now takes note of the setting appropriate to the chart type so filled polar charts have anti-aliased edges.

SVG circles, ticks, lines, markers – all make a small compensation for the reduced line-weight when anti-aliased lines are used. This is particularly obvious in the grid-lines on polar charts which almost vanish when anti-aliased. The intention is that you see no change of weight when you set the SVG viewer to 'higher quality' off using the right-mouse menu.

Updates to 6th June 2002 (Release 2.0f)

Linegraph – adds new 'segmented' style which draws a series of separated line-segments where there are missing values in the data. A segment will only get drawn if there are at least 2 points in each block, after the data has been partitioned on the missing values for each set of y-values given.

PDF metrics – was using a version of the metrics table with AR and HE fonts both included, but the font table only had AR, so we picked up incorrect metrics for Courier. Error showed up on barcharts with vertical value labels set in any of the Courier fonts when the labels were 'inside' the bars, as they were under-measured and stuck out of the tops! Fixed.

Updates to May 14th (Release 2.0e)

Already fixed

PNG – adding 'effects' to more line types (SVG only) was not always dummied correctly and could crash the PNG output. Fixed (and uploaded 2nd May).

PDF – adding the option to have round-cornered boxes was not handled correctly in all cases, so some examples could crash the PDF rendering. Also fixed and uploaded 2nd May.

New updates

SVG – adds `pointer-events="visible"` to all bars where there is an href set. This ensures we get the pointing hand even for unfilled bars. The Stepchart uses unfilled, unedged bars to get jumps to work where the style is 'norisers' so this is essential here.

Scatterplot – was not eliminating markers for missing values when these were user-defined with text as the marker symbol. The result was a small heap of tiny characters at the chart origin! Fixed.

Character markers – the `#nnn` coding style only worked for chars above 128. This is silly, as you might well want to give the decimal value for a low-end WingDing or other symbol. Fixed.

Character markers – one nasty ‘gotcha’ was defining a marker as ‘arb,12,-’ as we attempted to execute the ‘-’ thinking it might be a number. Fixed.

Character markers in SVG and VML – defined markers with any decimal value are now OK, and are simply passed out in `&#nnnn;` format. This allows any Unicode char to be used as a marker.

DefineFont – tolerates a list of font names (maybe quoted with single quotes) which is then correctly passed through to VML or SVG.

Piechart – spider tags now use two ‘polyText’ blocks rather than being drawn one at a time. Makes for neater SVG and PDF output as we can group the alignment and font settings.

Updates to April 23rd (Release 2.0d)

Barchart – updated to take note of current ‘effect’ as well as defined tile. This allows us to do interesting things in SVG, as it can define a 3D filter or an animation, so the bars can ‘grow’ to full size over time etc.

Text elements – heading, footnote, captions and labels all now take notice of any defined effects via a style setting such as `HeadingStyle="effect:12"` which allows any CSS ‘extras’ to be easily inserted into the saved VML or SVG file, and also makes it possible to add animations such as fade-in headings very easily to SVG output.

Axis styles – XYZ axes now notice style *nolabels* which allows axes to get tick marks and gridlines but prevents labels being drawn. Useful if you have a vertical or horizontal trellis where you probably only want the bottom or leftmost chart-axis labelled, but ticks on all of them.

Response – style ‘risers’ now works as it should. May be used with *nolines* and *nomarkers* to give a ‘bristle’ plot which can be a good way to view a dense matrix of Z values for a rectangular xy grid. Could be achieved before with the cloud chart, but this is much easier to work with if you have a rectangular array of Z values.

LineGraph – notices style ‘crisp’ which kills SVG anti-aliasing for the line. The apparent effect is to embolden the line somewhat when the line-weight is low (around 1-pixel).

Max-min – uses filled rectangles if the ‘nib’ is very broad. Otherwise Windows draws the ‘lines’ with round ends which can bulge out under the axis and collide with the value

tags and generally make a mess. This has no effect on SVG or PDF output which drew it correctly already!

Font metrics – now uses accurate metrics of all defined fonts, having cached the 14 ‘base’ fonts on installation. This supersedes the approximate width calculations which were ‘lean and mean’ but could result in very long key text occasionally colliding with the next key marker, particularly in Finland with text having lots of Scandics! Adds about 8K to the memory overhead, but will become increasingly useful as more things (like notes) start to do automatic wrapping.

We also handle **Unicode strings** like “非常” rather better now when they occur in key text. The approach is to replace the Unicode string with ‘00\\’ (where ‘\’ is your chosen Newline) so all Chinese characters get a standard width equal to 2 digits in the chosen font. Maybe one day we will need to do this completely correctly, but this is fast and good enough for most purposes.

Updates to April 4th 2002 (Release 2.0c)

Max-Min – a GraPL user has long-term min/max averages to plot as a backcloth to a temperature chart. This can be done with a filled lineplot but much better to do it with min-max which already has the data shaped up right. Use style ‘filled’ to fill the region between the two sets of y-coords. Generally these will not cross over, but the chart will work OK if the lines do cross (e.g. imports/exports showing the ‘trade gap’). For now only one shading is allowed – maybe in the future we will allow 2 settings for ‘a above b’ and ‘b above a’ respectively.

Max-Min – broke on style ‘Redraw’. Fixed.

Barchart – Horizontal bars were throwing out the X-caption twice! Fixed.

Notes – note-box redone to use new *Flowbox* macro which allows round corners and does the shadows as part of the macro, so we save some code here. Notes now have style option ‘Rounded’ and there will shortly be some new properties to give full control over the corner radius and the shadow opacity and offset. Keys and the chart frame can have round corners in the same way.

X-labels – was storing an incorrect value for the space to allow for labels with intercept axes, resulting in a misplaced x-caption later on. Fixed.

SetSeparators – override the Windows default settings for thousands and decimal separators. If only one char is given, it replaces the decimal separator only.

Updates to March 13th 2002 (Release 2.0b)

Keys – allows more length for line-keys when the line-weight is heavy. This gives enough space to show all reasonable dash patterns as the dash cycle is increased in proportion to the line-weight.

Ylabels – was checking for the presence of ‘;’ in labels before splitting them at newlines – should be looking for your chosen Newline character! Also Keys should use your Newline as the column-break marker as well as the line-break. Fixed.

Note,Text,Draw – these were changed in V2 to initialise the frame if they were the first function called. Too many people have charts which rely on the old (not well documented) behaviour, so backed out. To get the initialisation you must call Null as before. However they do take note of the current frame origin (usually 0,0) and assume all co-ords are given relative to it.

RenderSVG(opt) – opt=3 is now taken to be ‘fit to browser’ regardless of window shape, whereas 2 preserves aspect as before.

Boxplot – adjusts whiskers correctly for nib-weight. Also the length of the T-pieces you get with the ‘Errorbars’ style here.

Autorange – was checking for xstyle ‘DATE’ and ensuring no ticks were made below 1-per-day (correct) but when style is ‘DATE,TIME’ it should not apply this rule as the data may well be part-day. Fixed.

Picture format – was not always handling the decimal separator correctly when run in European locales. Fixed.

>>13/03/2002

Updates to February 2002 (Version-2)

Linegraph – data blocking removed. Printers have enough memory now and this was becoming a nuisance when applying scripts to lines and markers. For VML and SVG and PDF it is much better to have a single ‘object’ per line as the attributes can be efficiently grouped here.

Trace chart – was fooled if categories were set up and the dataset was ‘square’ with the same number of observations as species types. Fixed.

JavaSet – order of arguments changed so the ‘normal case’ has the arguments at the left of the list – now [;prop value row col] so row and col can be elided easily when this is called from VB.

Keys – honours ‘marker scale’ when this has a scalar value. The idea of scaled markers was to allow ‘incidence plots’ with markers scaled to the data value, but of course you can use it simply to beef up ALL the markers, in which case it should show up in the key.

Linegraph – 3D effect works with markers as well as lines. Also we allocate extra space in the key for scaled markers.

Max-Min plot – supports ‘values’ style. Value tags are written at both ends of the line, unless one end is all zero. Arrowheads are now omitted from the zero ends if the chart is one-sided.

Trend and modelfit – died if all data was ‘missing values’. Fixed.

Axis captions – attend captions were not placed quite correctly when charts became small (trellis) and the axes had arrowheads. Fixed.

Value tips – automatically generated value tips (value-style ‘tip’) did not get highbars changed to standard -ve signs. Fixed.

Linegraph – hints/tips were misplaced when a plot with markers also had a trendline or modelfit. Fixed.

Authoranging – attempts a better default decision when you have a time-based axis and there is only one datapoint! Checks the style (time,date etc) and sets a notional minimum value a few hours/days back. This is still a little arbitrary, but is better than the current behaviour of forcing zero, which for a date axis means 31/12/1899.

Picture formatting – two changes, one which is NOT backward compatible. This now treats the chars ‘.’, ‘,’ as ‘local decimal’ and ‘local thous’ characters so you must ALWAYS code in American (as in VB) here. When the chart runs, it will use the appropriate separator characters from the current locale instead of the chars in the picture. This allows you to make a chart definition which can be shipped to a different locale and run correctly. The second change is to allow a leading ‘(’ to indicate that negatives should be parenthesised – do NOT include a trailing ‘)’ as this is just treated as an ordinary insertion character.

Dial – new chart type. Emulates a dashboard dial with one or more pointer(s) and one or more warning zones. Adds new properties ‘XZ,YZ,IZ,ZA’ to set up zones – these are supported by the normal XY plots (e.g. for control charts) as expected. Set up the attributes with ‘ZA’ (colour,patt,edge) if needed – the defaults are Red,solid,none – and add zones as required. You can set matching vectors of X and Y zones to make rectangular areas.

Notes – a new property ‘NoteNudge’ which is in points, and allows you to place a note using chart co-ordinates (which baselines it at the xy given), and then nudge it an absolute distance, say to get the text descenders clear of a gridline, or to offset from a data-point by an amount related to the fontsize.

Value tags – supports multi-line value tags, split with ‘;’ as usual. Called with an extra parameter as these may need to be pushed up (the common case), vertically centred (HBars and Horiz VLines) or pushed down (bottom tag on vertical VLine is the only case so far).

SetNewline – does what it says on the tin! One special case is ‘\n’ which is the only 2-char newline marker accepted.

Bookmark – allows any chart in a multi-page document to be ‘bookmarked’ so that it generates an appropriate entry in the PDF outline tree.

Notes – addition of a 4th (optional) argument giving the maximum width allowed. Text wider than this is wrapped into the width. Soft-hyphens are allowed (~ is the default character used, set with `SoftHyphen="@"`).

Text – when called with a new chart (or new frame) this now throws out the frame (and any headings/footers) automatically for you. This means that you no longer need to remember to add Null before writing the text.

Xbar – should be able to make bars of differing widths, so the ‘Barwidth’ setting can be a vector (reshaped as required) rather than a single number. Also applies to Gantt charts..

Note – if called before any chart has been drawn, it now initialises the frame, margins and so on. Eliminates the need to call Null if all you want is a page with some notes.

***** *BACKED OUT IN LATEST RELEASE* *****

Authoranging – two small fixes to date axes. Setting a ‘monthly’ style on data like (1,2,3) should show Jan-1900. Probably a user error, but should not crash! Fixed. Also style ‘date’ will not generate tick- intervals of <1 now as the ‘day’ is the minimum sane unit here.

Updates to December 2001

Linegraph – drawing order changed so labels can draw on top of axis. Supports new label styles ‘middle’ (applies to X labels) and ‘centre’ (for y-labels) which draw the labels over the axes.

Axes – arrowheads were not drawn on secondary Y-axes.

Authoranging – no longer rounds down to the nearest whole tick interval for axis style ‘date’. This allows a FOOTSIE plot to start on a Monday with a tick interval of 5, and so put the ticks every Monday.

X-labels – if the Xlabels are set to an integer vector (which are actually coded dates) this now notices the ‘date’ style and shows them as dates. Similarly with times. Saves some formatting effort.

Value-style ‘hints’ or ‘tips’ – also implemented for Piecharts. Sorry pies, forgot all about you last time! That should cover all possibilities here.

Trace charts – Kite charts and similar trace styles now use the ‘category’ variable to allow groups of data to be coloured/shaded differently. The example Kite script now has two sorts of trees and Iceland2 has temperature probes above and below ground level as solid and dashed lines. The key is made from the unique categories in the usual way.

Keys – was allowing space below the X-caption even when the X-style put the caption 'at end' so it takes no space! Fixed.

Null chart – should take note of trellis-style 'skip' to throw a new cell! Fixed.

Gantt chart – an explicit setting of 'Barwidth' was only being honoured for the first block of bars, in this case 36. The setting then reverted to the default of 1.5 times the value-font size. Also the 'barwidth' was effectively being doubled as it was used both top and bottom of the centre-line. Fixed.

Value tags – now clears the value tags once they have been applied to a chart. We are unlikely to use the same tags twice and it is not obvious how to use 'novalues' to turn them off.

Include – new function to permit inclusion of arbitrary text into output stream. Currently only useful for SVG where it allows filters and gradients to be applied as 'defined tiles' having been set up by the appropriate XML strings beforehand.

Hints – hints can now be set for an entire array by leaving out the row number. To hint the line only in VML, set the hint on (row=0,col=n) which also works for JavaScript handlers, for example to have a line light up on mouseover.

Axes – internal change to use 'polyTick' which has the effect of turning off anti-aliasing when output as SVG. Probably also in VML when we need to keep IE6 under control!

Linegraph – adds a new option to have a 3D effect applied to any of the lines in a data series. Currently this is relevant to SVG only but other formats such as PDF may one day support it. New property 'Effects' sets the effect cycle and DefineEffect associates an SVG filter with a numeric id. To make the effects available to the SVG output, call Include "textfile" to pass the SVG code verbatim into the saved output.

Draw – optional left argument switches this to use chart co-ordinates. Also will take vectors of numeric vectors as an alternative to matrices on the right. Make life a lot easier for VBScript where arrays of arrays are the natural way to do it!

Autoranging – for annual data the X-range should be interpreted as years. If only a single number is given, this sets the lower end of the range as there is no natural 'zero' point here.

Axis labels – accept 'TimePicture' property to set the time format for axes with style 'time'. Similarly 'datepicture' sets the date picture, e.g. 'ddd, dd-MMM-yy' would be typical. See the Windows documentation for details. Also you can give dates in text format for range and datum settings for date-formatted axes e.g. 'xdatum = "21/9/89,21/12/89"'. New xstyle 'monthly' puts ticks at month boundaries and defaults the data picture to 'MMM'.

RenderEPS – pL macro amended to work correctly with null polylines. The current version crashes *Acrobat Distiller* if the array of points to be drawn is empty.

Axis captions – captions 'atend' could clash with arrowheads when the axis line-weight was very heavy (say 4 pt). Fixed.

Cloud charts – markers are scaled correctly for the projection, which slightly improves the 3D illusion here.

Missing data – tolerates 'null' as missing value. This is really only relevant for server applications, where database queries may well return 'null' for some elements and the nulls should always be treated as missing values.

Linegraph – allows the trend line or modelfit to be drawn on top of the data (the default is to draw the data last) with style 'ontop'.

Updates to September 2001

Trace – should do YCaption after YLabels to allow for label length. Fixed.

X-labels – correctly stores low-point of label set when labels are angled. The X-caption now adjusts correctly for angled label.

Hints – was getting confused by the combination of data hints and a data window. The index values for the hints should be the index in the original series. Same for tips and hrefs. Fixed.

Piecharts – algorithm to avoid clashing spider labels makes no sense for the 'rose diagram' style where not all the sectors have the same radius. Bypassed unless the pie is uniform.

Polar charts – additional style 'centre' which marks the centre with a small dot after all the data has been drawn. This improves shaded charts which hide the axes and are hard to read accurately otherwise.

Trend lines – restricts generated trend line so that it stays inside the span of the y-axes. Particularly important for VML charts as otherwise the chart can get displaced from its labels due to the use of CSS2 positioning for the text.

Y-labels – extends support to multiline labels using ';' as the separator in the normal way.

Headings etc – anywhere the ';' is used to split lines it can now be escaped with a preceding '\' as in ch.Set 'Footer' 'This\; that'. See below also.

Keys – multiline text now supported in the key text. Split with ';' as usual. This only makes a lot of sense in a vertical key but will work as expected in the horizontal key style.

Xcaption – may be split with ';' as for headings and other things. Ycaption – as above, but only for style 'atend'

Barchart values – two new options which can be combined as required. These are only relevant to stacked charts – setting `vstyle 'sector'` shows the values of each sector (rather than the cumulative total) and setting `vstyle 'middle'` is a third option to the set `'ontop|insidemiddle'` which writes the labels vertically centered, rather than towards the top of each sector. These would probably be used in combination. The `'sector'` style also applies to surface charts. Both `'sector'` and `'middle'` apply as you would expect to horizontal barcharts.

Barcharts – X-labels with parentheses were getting 'escaped' twice so the parens appeared in the output with a leading backslash. Barcharts only! Fixed.

Axis styles – style `'time,date'` now decodes a floating-point time and date vector to `"DD/MM/YY HH:MM"` in the appropriate local style. Note that tickmarks are given in hours here, as for `'time'` used alone.

Secondary Y-axes. Some tidying up of old code. Right-aligned secondary axis can now be placed anywhere you like with the `'XInt'` property. This gives the interesting new option of 'back-to-back' axes in the middle of the chart. As a consequence `ch.Ysec` now resets `'XInt'` and also `'YCap'` which it should have done before. If the secondary axis is drawn beyond the end of the X-axis then ALL the tick marks on it are labelled.

Frequency plot – was forcing the Y-caption to "Frequency" always. Now only sets this (also the gap and group-gap) if these properties have not been previously changed by the calling function.

X,Y,Value labels – if you set a numeric vector here as the labels it now has the appropriate 'picture' format applied to it. Previously it was always adjusted to 3 significant figures and just formatted. The default is now a simple primitive format, so existing charts which (probably unknowingly) relied on the adjustment to 3 sig figs could start showing more decimal places. Add an explicit formatter to fix the problem.

Picture format – new character `'~'` is used to suppress unwanted digits. An obvious use of this is to show years as '98' '99' '00' '01' with a format string `'~~00'` which will accept, then bury, the century part.

XStyle 'Annual' – when plotting vs a date vector this puts ticks every year at 1-Jan (so the ticks are very slightly irregular). The `XTick` property is interpreted as the number of years between ticks in this case.

JavaSet – this can be used to hide arbitrary information (e.g. product codes) in each VML or SVG element so an `onClick` handler can easily use this for database lookup.

New values styles 'Hints' and 'Tips' added. Rather than writing the values as text, these cause the data values to be written as hints or tips so that they show up in the viewer status line, or in VML output as status- line hints or real pop-up tips as required. A lot simpler than setting the `'hint'` or `'tip'` separately if all you want is the numbers on the points.

Barcharts – horizontal bars were not quite a mirror of vertical bars when it came to setting tick-intervals and labels. Fixed so both behave exactly the same.

Updates to July 2001

Polar charts – setting style 'NOAX' turns off both axes here. Setting style 'arrows' draws risers and arrows even if 'risers' has not been set.

Trace charts – now shows correct (filled box) key type for trace charts with style 'surface' or 'kite'. Also clips 'inner' ticks at the end of the real y-axis if this does not extend far enough.

Tower charts – gap between towers restricted to >0

Pie charts – pie explosion limited to range 0-100%

Axes – two small cosmetic changes. New style 'clipped' can be applied to x-axis or y-axis to prevent the axis line drawing beyond the final tick. Also if you set the 'arrows' style the axes stick a little further out which makes a better visual effect as it stops the arrowhead clashing with the top/right tickmark.

Trace charts – were ranging the inner axis incorrectly with some datasets. Fixed.

Linegraph – user-set Y-range did not work correctly with 'Inverted' style of Y-axis (data plotted with the lower values at the top). Fixed.

Auto-ranging – improved handling of X-axis with style 'time'. The auto-generated tick interval is now a sensible number of minutes and you can give the X-tick value in hours rather than decimal days.

Lineplot – takes note of new keystyle 'inline' which places the key text at the end of each line. This only makes sense for time-series where the lines are reasonably well-separated, but is more readable than a conventional key in this case. Also allows 'arrow' style on lines now – adds an arrowhead to the last line segment. Adds new style 'indexed' which re-bases all series to begin at 100. Also rather basic support for 'Log' style axes. Needs more work for log axes which span less than a power of 10. Also formats degrees:minutes nicely with 'latlong' style (same code).

Keys – little change from the outside, but a thorough rewrite inside to use control-structures and generally remove 10 years of accumulated gunk. It now boxes multi-line and vertical keys accurately, and has one small extra tweak – you can put ';' at the front of a key element to force a column-break in a key with style 'Vertical'. Placement of keys positioned using chart co-ordinates is a little more accurate. Default placement for 3D charts corrected (was unfixed ~ 6 months ago).

Aggregation – scatter plots failed when you used the same variable in both the 'GroupBy' and 'vs' properties. Now it detects this and uses the result of the grouping automatically as the 'vs' variable.

Notes – was boxing centred and right-aligned notes incorrectly. Fixed.

Linegraph – honours ‘ontop’ style for values. This can be used to write the values on top of the large circular markers, as per the ‘epidemic’ charts in the recent Guardian newspaper. Use a bold, white Arial font! Also, 0 was always being blanked – use value format ## from now on if you want to hide zero labels!

Keys – simple request from Finland to be able to omit the markers from the ‘line&marker key’ on request. Using ‘plain’ as a keystyle now does this. The default is to key with lines and markers as before. Also keystyle ‘reversed’ works on horizontal as well as vertical keys now.

Trendlines – style ‘growth’ does a weighted logfit to give an exponential growth curve.

Linegraph – reverses the order of drawing ticks and labels. By doing the labels first we thin the ticks when we have been forced to thin the labels. Looks much better if you force XT=1 and allow clashing labels to be thinned automatically. Still not recommended though!

Authoranging – new axis style ‘Exact’ stops this rounding up to the next whole tick interval. Minor ticks (if required) are continued up to the end of the axis, so a few minor ticks may appear beyond the last major tick. Most useful for charts with surface fill, where it looks ugly when the surface ends some way short of the last tick on the axis.

Keys, notes and chart box now support ‘shadowed’ style which adds a simple offset grey box underneath. The line-weight of the border increases the shadow offset marginally.

Axis captions – default position allows a little more space to left/right of the labels – mainly so VML looks neater.

Keys – more space at right-hand side for boxed key. Again for VML where the text positioning is not quite as exact as it should be.

Trace charts – were forcing a zero baseline when you asked for surface shading. Now only does this for Kite charts where it is essential. Y-axis slipped by 0.5 (rather than 0.4) to balance trace better (it is midlined).

Towers – labels were not being correctly generated when Group Into was set. They had slipped one tickmark early – fixed. NB When Categorise Into is set tower charts should really set up the bin names as the y-labels. Currently they get the bins set as the key – should be an alternative.

X-labels – label positioning corrected for labels rotated by 270 degrees which is unusual but a reasonable alternative to 90 degrees for vertical labels. Rotations like ~ 30 are OK too now.

Footnotes – multiline footnotes now work as you would expect.

Keys – kstyle 'nokey' not being honoured – fixed.

Piecharts – allow for piecharts starting at 3 o'clock with the 'horizontal' style. Raises some new issues with colliding labels as the small sectors are now likely to pile up here rather than at 12:00 which is easier. Also allows exact placement of the pie with (Piecentre=200,80) – this gives the centre of the pie in points relative to the frame origin.

grapl.Text – low-level text output using the same approach as grapl.Draw. A very thin cover so much faster than Note for bulk text, such as tabular summaries of data. All coordinates are in points relative to the origin of the current frame (lower left corner + margins). Uses the current Note Font for the text style and accepts a 3-element vector (vtv) xx yy. Note style is use to set the alignment and 'underlined' is also honoured here.

Stepcharts – style 'floating' added (suitable for interest-rate graphs). This just draws the stepped line rather than filling down to the X-axis. Also the stepchart no longer assumes 'forcezero' on the Y-axis as there was never any good reason for this.

Axes – either axis can now be made invisible independently with the new 'invisible' style. This does not draw tickmarks or labels but does write the axis labels and gridlines.

XBar – zero values in the data caused value labels to slip. Fixed.