General

This bundle of classes named *MAAPlot* is a plotting component for Lazarus/fpc.

What it is - would it help you ?

MAAPlot is a set of classes written in Lazarus / fpc which can be used in your own Lazarus / fpc application.
Its purpose is plotting data having in mind scientific data. The initial purpose of this component is plotting measurement data or calculated data for visulaization.
Three main applications where considered as follows:

- Standard 2D data plotting, used for measurement data consisting of single points
  Example: A multimeter continuously delivers measurement values, i.e. 1 per second
- Fast 2D data plotting, used for spectrum analyzer or oscilloscope display
  Example: repeated FFTs from sampled soundcard data
- Waterfall plotting of 2D data, used for spectrum analyzer waterfall display
  Example: repeated FFTs from sampled soundcard data

For the standard mode several hundreds of datapoints can be plotted and autoscaled per second. For the fast modes a single core 1.3GHz processor will easily plot realtime data from 96kHz, 2 channel soundcard sampling for example.
Markers are provided like in a spectrum analyzer
Mouse control is provided for Zoom and Pan and a AdHoc marker
Furthermore a unlimited number of dataseries can be plottet into one single plot (i.e. for display of measurement data of a 20 channel thermometer)

The component is NOT:

- A function plotter
  However you can calculate function values (i.e. $y = x^2$) in your application and then plot it.
- A 3D engine
  However a simple 3D plot is implemented as well as a 3D waterfall.
- A dll for external use
  You might build a dll from the source. However Pascal datatypes are used in many method calls making calls from other languages difficult.

Chapter 1

Reference for unit 'uPlotClass'

1.1 Used units

Table 1.1: Used units by unit 'uPlotClass'

ame	Page
Classes	N/A
Controls	N/A
dateutils	N/A
Dialogs	N/A
ExtCtrls	N/A
ExtDlgs	N/A
Forms	N/A
FPimage	N/A
Graphics	N/A
IntfGraphics	N/A
LCLType	N/A
math	N/A
Menus	N/A
System	N/A
SysUtils	N/A
Types	N/A

1.2 Constants, types and variables

1.2.1 Constants

`c_GLOBALMAX = 1e30`

maximum value the component will handle

`c_GLOBALMIN = 1e-30`

minimum value the component will handle

`c_INVALIDCOORDINATE = -16777216`

return value used to detect errors

```
c_MAININTERVAL_MAXDIGITS = 10
```

maximum digits displayed on axis ticks

1.2.2 Types

```
TAutoScaleMode = (asFirstManualThenFit,asFit,asFit125,asFitNext,
                  asFitNextMargined)
```

Controls how the data is fitted in the viewrange by autoscaling

Table 1.2: Enumeration values for type TAutoScaleMode

Value	Explanation
asFirstManualThenFit	deprecated, series can be autoscaled with "growonly" now
asFit	fits the valuerange of the data into the plotting area
asFit125	like fit but extents the view area to the next integer number 1,2 or 5 (like in a oscissolsc
asFitNext	like fit but extents the view area to the next integer number
asFitNextMargined	like fit but extends the view area by a certain percentage (compile-time constant in uPlo

```
TAxisOrientation = (aoHorizontal,aoVertical,aoVariable)
```

Table 1.3: Enumeration values for type TAxisOrientation

Value	Explanation
aoVariable	drawing angle can be set to any value between 0..360°
aoHorizontal	fixed drawing angle of 0°
aoVertical	fixed drawing angle of 90°

```
TColorPoint = packed record
  Pt : TPoint;
  Color : TColor;
end
```

```
THIDAction = (haNone,haZoom,haPan,haAdHocMarker)
```

Some modification to the plot controlled by keyboard/mouse which will be mapped to keyboard/mouse states (like Shift+left button) by HID handler

Table 1.4: Enumeration values for type THIDAction

Value	Explanation
haAdHocMarker	add a marker on the fly
haNone	do nothing
haPan	pan the plot
haZoom	zoom the plot

```
THIDMouseState = (mcNone,mcMLeft,mcShiftMLeft,mcCtrlMLeft,mcWheel,
                  mcShiftWheel)
```

Available keyboard/mouse states which can be mapped to a HIDAction

Table 1.5: Enumeration values for type THIDMouseState

Value	Explanation
mcCtrlMLeft	<CTRL>+<Left mouse button>
mcMLeft	<Left mouse button>
mcNone	<>
mcShiftMLeft	<Shift>+<Left mouse button>
mcShiftWheel	<Shift>+<mouse wheel>
mcWheel	<mouse wheel>

```
THIDMouseStates = Set of THIDMouseState
```

```
TSeriesType = (stBASE,stPLAIN,stXY,stXYZ,stSPECTRUM,stWF2D,stWF3D)
```

Used only with the templates un uPlotTemplates

Table 1.6: Enumeration values for type TSeriesType

Value	Explanation
stBASE	
stPLAIN	
stSPECTRUM	
stWF2D	
stWF3D	
stXY	
stXYZ	

```
TValueRange = packed record
  min : Extended;
  max : Extended;
end
```

```
TZoomRecord = record
  dbZooming : Boolean;
  PlotRectIndex : Integer;
  dwOldRect : TRect;
  dwNewRect : TRect;
end
```

Used for the HID handler

1.3 EPlot

Can be used for improved error handling

1.4 TPlot

1.4.1 Description

Container class for all other elements
One or more plotrects - Two or more Axes - One or more Series

1.4.2 Method overview

Page	Property	Description
8	AddPMContextItems	Build the context menu (popup menu)
9	AutoScalePlotRect	autoscale a plotrect
8	AutoScaleSeries	autoscale a series
8	CheckPMContextItems	Build the context menu (popup menu) - variable items
9	ClearAll	Frees everything except the TPLot, leaves an empty TPlot
7	Create	
7	Destroy	
7	DrawZoomRect	draw a rectangle
9	ExportToFile	export a bitmap
9	ForceRefreshFastPlotRects	call when update is needed in TimedRefresh mode
9	LockImage	
6	OnPaintImage	experimental
8	Pan	pan the plot with mouse
6	RegisterPlotObject	register a plot object
8	RemovePMContextItems	Clear the context menu (popup menu) - variable items
8	Repaint	Repaint the plot and all contained elements
7	ScrCoordToPlotRect	index of plotrect at the given screen coordinates
6	UnregisterPlotObject	
7	Zoom	Zoom the plotrect

1.4.3 Property overview

Page	Property	Access	Description
11	Axis	r	axis
10	AxisCount	r	number of axes within a plot
10	BackgroundColor	rw	background of the plot
12	ExportPrinterFriedlyColors	rw	more white, less black
12	ExportSize	rw	size in pixels for bitmap export
13	HIDHandler	r	HID handler class
9	MousePlotRectDown	r	Index of plotrect where mouse button was pressed
10	OnResize	rw	currently unused
10	PlotImage	r	TImage; everything is drawn on this TImage canvas
11	PlotRect	r	plotrect
11	PlotRectCount	r	number of plotrects within a plot
11	Series	r	series
11	SeriesCount	r	number of series wihtin a plot
12	SmartSizing	rw	not used
10	Style	rw	drawing style
12	TimedRefresh	rw	triggered refresh or immediate refresh

1.4.4 TPlot.RegisterPlotObject

Synopsis: register a plot object

Declaration: `function RegisterPlotObject(Obj: TObject) : Integer;  Virtual`

Visibility: protected

Description: PlotRects, Axes and Series need to be registered after Creation

1.4.5 TPlot.UnregisterPlotObject

Synopsis:

Declaration: `procedure UnregisterPlotObject(Obj: TObject);  Virtual`

Visibility: protected

Description:

1.4.6 TPlot.OnPaintImage

Synopsis: experimantal

Declaration: `procedure OnPaintImage(Sender: TObject)`

Visibility: protected

Description: We draw on the persistent bitmap of a TImage. This function was used to see if drawing during the OnPaint event is faster (it is not...)

1.4.7 TPlot.Create

Synopsis:

Declaration:
```
constructor Create(AOwner: TComponent);  Override
```

Visibility: public

Description:

1.4.8 TPlot.Destroy

Synopsis:

Declaration:
```
destructor Destroy;  Override
```

Visibility: public

Description:

1.4.9 TPlot.ScrCoordToPlotRect

Synopsis: index of plotrect at the given screen coordinates

Declaration:
```
function ScrCoordToPlotRect(X: Integer;Y: Integer;
                            out PlotRectIndex: Integer) : Integer
```

Visibility: public

Description: used for popup menu handling

1.4.10 TPlot.DrawZoomRect

Synopsis: draw a rectangle

Declaration:
```
procedure DrawZoomRect(AZoomInfo: TZoomRecord)
```

Visibility: public

Description: used in zoom mode with left mouse button. This mode can only zoom in.

1.4.11 TPlot.Zoom

Synopsis: Zoom the plotrect

Declaration:
```
procedure Zoom(AZoomInfo: TZoomRecord)
procedure Zoom(AZoomInfo: TZoomRecord;ACenterPoint: TPoint;
               AFactorX: Extended;AFactorY: Extended)
```

Visibility: public

Description: Zoom(AZoomInfo: TZoomRecord), based on a rectangle. Only zooms in as the rectangle is always smaller or equal than the data area.
Zoom(AZoomInfo: TZoomRecord; ACenterPoint: TPoint; AFactorX: Extended; AFactorY: Extended); Zooms in or out, based on the center coordinates and a factor in X and Y direction

1.4.12 TPlot.Pan

Synopsis: pan the plot

Declaration: `procedure Pan(AZoomInfo: TZoomRecord)`

Visibility: public

Description: pan the plot with mouse

1.4.13 TPlot.AddPMContextItems

Synopsis: Build the context menu (popup menu)

Declaration: `procedure AddPMContextItems(AMenu: TObject)`

Visibility: public

Description: Static menu entries common for all plotrects

1.4.14 TPlot.CheckPMContextItems

Synopsis: Build the context menu (popup menu)

Declaration: `procedure CheckPMContextItems(AMenu: TObject)`

Visibility: public

Description: Variable menu entries, based on the actual plotrect

1.4.15 TPlot.RemovePMContextItems

Synopsis:

Declaration: `procedure RemovePMContextItems(AMenu: TObject)`

Visibility: public

Description:

1.4.16 TPlot.Repaint

Synopsis: Repaint the plot and all contained elements

Declaration: `procedure Repaint;  Override`

Visibility: public

Description: Called from outside, i.e. after resizing

1.4.17 TPlot.AutoScaleSeries

Synopsis: autoscale a series

Declaration: `procedure AutoScaleSeries(ASeriesIndex: Integer;AGrowOnly: Boolean)`

Visibility: public

Description: If AGrowOnly=true we do not shrink. This is used for scaling more than one series wihtin a plotrect

1.4.18 TPlot.AutoScalePlotRect

Synopsis: autoscale a plotrect

Declaration: `procedure AutoScalePlotRect(APlotRectIndex: Integer)`

Visibility: public

Description: autoscale a plotrect

1.4.19 TPlot.LockImage

Synopsis:

Declaration: `procedure LockImage(ADoLock: Boolean)`

Visibility: public

Description:

1.4.20 TPlot.ExportToFile

Synopsis:

Declaration: `procedure ExportToFile(AFileName: TFilename)`

Visibility: public

Description:

1.4.21 TPlot.ClearAll

Synopsis:

Declaration: `procedure ClearAll`

Visibility: public

Description:

1.4.22 TPlot.ForceRefreshFastPlotRects

Synopsis: call when update is needed in TimedRefresh mode

Declaration: `procedure ForceRefreshFastPlotRects`

Visibility: public

Description: call when update is needed in TimedRefresh mode

1.4.23 TPlot.MousePlotRectDown

Synopsis: Index of plotrect where mouse button was pressed

Declaration: `Property MousePlotRectDown : Integer`

Visibility: protected

Access: Read

Description: Index of plotrect where mouse button was pressed

1.4.24 TPlot.PlotImage

Synopsis:

Declaration: `Property PlotImage : TImage`

Visibility: public

Access: Read

Description:

1.4.25 TPlot.OnResize

Synopsis: currently unused

Declaration: `Property OnResize : TNotifyEvent`

Visibility: public

Access: Read,Write

Description: deprecated ?

1.4.26 TPlot.BackgroundColor

Synopsis: background of the plot

Declaration: `Property BackgroundColor : TColor`

Visibility: public

Access: Read,Write

Description: background of the plot

1.4.27 TPlot.Style

Synopsis: drawing style

Declaration: `Property Style : TPlotStyleBase`

Visibility: public

Access: Read,Write

Description: deprecated, no function

1.4.28 TPlot.AxisCount

Synopsis: number of axes within a plot

Declaration: `Property AxisCount : Integer`

Visibility: public

Access: Read

Description: number of axes within a plot

1.4.29 TPlot.Axis

Synopsis: axis

Declaration: `Property Axis[Index: Integer]: TPlotAxisBase`

Visibility: public

Access: Read

Description: axis

1.4.30 TPlot.SeriesCount

Synopsis: number of series wihtin a plot

Declaration: `Property SeriesCount : Integer`

Visibility: public

Access: Read

Description: number of series wihtin a plot

1.4.31 TPlot.Series

Synopsis: series

Declaration: `Property Series[Index: Integer]: TPlotSeriesBase`

Visibility: public

Access: Read

Description: series

1.4.32 TPlot.PlotRectCount

Synopsis: number of plotrects within a plot

Declaration: `Property PlotRectCount : Integer`

Visibility: public

Access: Read

Description: number of plotrects within a plot

1.4.33 TPlot.PlotRect

Synopsis: plotrect

Declaration: `Property PlotRect[Index: Integer]: TBasePlotRect`

Visibility: public

Access: Read

Description: plotrect

1.4.34 TPlot.ExportSize

Synopsis: size in pixels for bitmap export

Declaration: `Property ExportSize : TSize`

Visibility: public

Access: Read,Write

Description: size in pixels for bitmap export

1.4.35 TPlot.ExportPrinterFriedlyColors

Synopsis: more white, less black

Declaration: `Property ExportPrinterFriedlyColors : Boolean`

Visibility: public

Access: Read,Write

Description: more white, less black

1.4.36 TPlot.TimedRefresh

Synopsis: triggered refresh or immediate refresh

Declaration: `Property TimedRefresh : Boolean`

Visibility: public

Access: Read,Write

Description: TimedRefresh = true means that new data added to a series is not immediately displayed but only when ForceRefreshFastPlotRects is called.
ForceRefreshFastPlotRects could be called by a timer i.e. 20 times a second.
If false, the screen is updated evereytime data is added to a series.
Note: When adding data to series very often (i.e. > 40 times a second), the GUI Thread might be blocked with screen updates. Use timed refresh instead to trigger the screen update no more than 40 times a second. Updating the data in the background is faster than updating the screen.

1.4.37 TPlot.SmartSizing

Synopsis: not used

Declaration: `Property SmartSizing : Boolean`

Visibility: public

Access: Read,Write

Description: deprecated / experimental. Not implemented. When resizing large datasets, Repaint is triggered on every mouse move of the containing control. This might block the GUI Thread. The idea was to suppress replotting of the data until the mousebutton is released.

1.4.38 TPlot.HIDHandler

Synopsis: HID handler class

Declaration: `Property HIDHandler : TPlotHIDHandlerBase`

Visibility: public

Access: Read

Description: The HID handler maps keyboard/mouse input to specific actions like Zoom, Pan and a AdHoc Marker.

1.5 TBasePlotRect

1.5.1 Description

A Plotrect is the container for axes and series belonging together. Usually a plotrect will hold 2 axes and a 2D series or 3 axes and a 3D series. One or more Plotrect are placed into a TPlot. The Plotrect does some size calculations to fit the dataplot into the desired area. Therefore the Plotrect can be regarded as the container class for the "plot" you get at the end with axes, captions and data plotted into. Note: More than one plotrect can be placed into a TPlot container class. The most common situation however would be to have one plotrect in one TPlot container.
Do not use the base class - please use the TPlotRect class. Check TPlotRect for further methods and properties

1.5.2 Method overview

Page	Property	Description
14	Create	
14	Destroy	
14	GetClientRect	Bounds of a logical rect used for calculations
14	GetFrameRect	Bounds of the PlotRect (i.e. the desired data plot)
14	PlotImage	TImage component of the owner TPlot
14	Redraw	Called when redrawing is needed

1.5.3 Property overview

Page	Property	Access	Description
15	AutoClientRect	rw	automatic size adjustment
15	AutoFrameRect	rw	automatic size adjustment
16	BottomLeft	r	BottomLeft point of Framerect
15	ClientRect	rw	Bounds of a logical rect used for calculations
15	DataRect	rw	Bounds of a logical rect used for LazIntfImage drawing
15	FrameRect	rw	Bounds of the plotrect within a TPlot
16	Heigth	r	Framerect height in pixels
16	OwnerPlot	rw	Ownerplot of the plotrect (TPlot class)
17	PlotRectIndex	r	Index of the plotrect
17	SeriesContainedIdx	r	List of series indices contained within the plotrect
16	TopLeft	r	TopLeft point of Framerect
17	Visible	rw	Visibility of the plotrect - unused
16	Width	r	Framerect width in pixels
17	Zooming	rw	Zooming mode active

1.5.4 TBasePlotRect.GetClientRect

Synopsis: Bounds of a logical rect used for calculations

Declaration: `function GetClientRect : TRect;  Virtual`

Visibility: protected

Description: Axis size calculations are based on the Clientrect (the name might be misleading. it is NOT the clientrect as defined in a TControl). The Clientrect is basically the Framerect shrunk by the label, axes captions, other text, legend and colorscale.

1.5.5 TBasePlotRect.GetFrameRect

Synopsis:

Declaration: `function GetFrameRect : TRect;  Virtual`

Visibility: protected

Description:

1.5.6 TBasePlotRect.Redraw

Synopsis: Called when redarwing is needed

Declaration: `procedure Redraw;  Virtual`

Visibility: protected

Description: Called when redrawing is needed

1.5.7 TBasePlotRect.Create

Synopsis:

Declaration: `constructor Create(OwnerPlot: TPlot);  Virtual`

Visibility: public

Description:

1.5.8 TBasePlotRect.Destroy

Synopsis:

Declaration: `destructor Destroy;  Override`

Visibility: public

Description:

1.5.9 TBasePlotRect.PlotImage

Synopsis: TImage component of the owner TPlot

Declaration: `function PlotImage : TImage`

Visibility: public

Description: TImage component of the owner TPLot

1.5.10 TBasePlotRect.ClientRect

Synopsis: Bounds of a logical rect used for calculations

Declaration: `Property ClientRect : TRect`

Visibility: public

Access: Read,Write

Description: Bounds of a logical rect used for calculations

1.5.11 TBasePlotRect.FrameRect

Synopsis: Bounds of the plotrect within a TPlot

Declaration: `Property FrameRect : TRect`

Visibility: public

Access: Read,Write

Description: Bounds of the plotrect within a TPlot

1.5.12 TBasePlotRect.DataRect

Synopsis: Bounds of a logical rect used for LazIntfImage drawing

Declaration: `Property DataRect : TRect`

Visibility: public

Access: Read,Write

Description: Bounds of a logical rect used for LazIntfImage drawing

1.5.13 TBasePlotRect.AutoClientRect

Synopsis: automatic size adjustment

Declaration: `Property AutoClientRect : Boolean`

Visibility: public

Access: Read,Write

Description: deprecated. AutoClientRect = false should not be used.

1.5.14 TBasePlotRect.AutoFrameRect

Synopsis: automatic size adjustment

Declaration: `Property AutoFrameRect : Boolean`

Visibility: public

Access: Read,Write

Description: AutoFrameRect = false should not be used. Tested only for AutoFrameRect = true

1.5.15 TBasePlotRect.Width

Synopsis:

Declaration: `Property Width : Integer`

Visibility: public

Access: Read

Description:

1.5.16 TBasePlotRect.Heigth

Synopsis: Framerect height in pixels

Declaration: `Property Heigth : Integer`

Visibility: public

Access: Read

Description: Framerect height in pixels

1.5.17 TBasePlotRect.TopLeft

Synopsis: TopLeft point of Framerect

Declaration: `Property TopLeft : TPoint`

Visibility: public

Access: Read

Description: deprecated.

1.5.18 TBasePlotRect.BottomLeft

Synopsis: TopLeft point of Framerect

Declaration: `Property BottomLeft : TPoint`

Visibility: public

Access: Read

Description: deprecated

1.5.19 TBasePlotRect.OwnerPlot

Synopsis: Ownerplot of the plotrect

Declaration: `Property OwnerPlot : TPlot`

Visibility: public

Access: Read,Write

Description: A TPlot

1.5.20 TBasePlotRect.Visible

Synopsis: Visibility of the plotrect

Declaration: `Property Visible : Boolean`

Visibility: public

Access: Read,Write

Description: deprecated. Will have no effect

1.5.21 TBasePlotRect.SeriesContainedIdx

Synopsis: Indices of series contained within the plotrect

Declaration: `Property SeriesContainedIdx : TFPList`

Visibility: public

Access: Read

Description: Receiver should free the resulting TFPList.

1.5.22 TBasePlotRect.PlotRectIndex

Synopsis: Index of the plotrect

Declaration: `Property PlotRectIndex : Integer`

Visibility: public

Access: Read

Description: probably not needed, given for historical purposes. Index of the plotrect within the TPlot container.

1.5.23 TBasePlotRect.Zooming

Synopsis: Zooming mode active

Declaration: `Property Zooming : Boolean`

Visibility: public

Access: Read,Write

Description: Set to true when zooming is active. Used to draw a zoomrect (when left mouse button zoom is active). TODO: property will be deprecated in the future as we will check HIDhandler for this

1.6 THelperFormsBase

1.6.1 Description

Forms for popup menu

1.6.2 Method overview

Page	Property	Description
18	Create	
18	Destroy	

1.6.3 Property overview

Page	Property	Access	Description
18	OwnerPlot	rw	

1.6.4 THelperFormsBase.Create

Synopsis:

Declaration: `constructor Create(AOwner: TComponent);  Override`

Visibility: public

Description:

1.6.5 THelperFormsBase.Destroy

Synopsis:

Declaration: `destructor Destroy;  Override`

Visibility: public

Description:

1.6.6 THelperFormsBase.OwnerPlot

Synopsis:

Declaration: `Property OwnerPlot : TPlot`

Visibility: public

Access: Read,Write

Description:

1.7 TPlotAxisBase

1.7.1 Description

A axis is the representation of a data coordinate (i.e. X, Y or Z values). Axes have labels and units of measure.
The axis is used for datapoint calculations (i.e. mapping of a specific value to the screen). Scaling is linear or logarithmic to a arbitrary log base.
Although cartesian axes usually are rectangular, drawing can be done with any arbitrary angle.
Important: all datapoint calculations are done based on the position of the "axis" TPlotAxis

wether it is visible or not. Axes drawn for indication only are called CloneAxis and are defined in the next class.

Do not use the base class - please use the TPlotAxis class. Check TPlotAxis for further functions and properties

1.7.2 Method overview

Page	Property	Description
20	CheckSize	Delivers size used for drawing
20	Create	
20	Destroy	
19	GetPixelsPerValue	
19	GetViewRange	
20	PlotImage	
20	Redraw	
19	SetViewRange	

1.7.3 Property overview

Page	Property	Access	Description
21	Orientation	rw	vertical or horizontal
21	OwnerPlot	rw	Ownerplot
21	OwnerPlotRect	rw	Containing plotrect
22	PixelsPerValue	r	screen pixels per axis value
21	Style	rw	Plotstyle used for the axis
21	ViewRange	rw	viewport of the data
22	Visible	rw	visibility of axis

1.7.4 TPlotAxisBase.GetViewRange

Synopsis:

Declaration: `function GetViewRange : TValueRange;  Virtual`

Visibility: protected

Description:

1.7.5 TPlotAxisBase.SetViewRange

Synopsis:

Declaration: `procedure SetViewRange(AValue: TValueRange);  Virtual`

Visibility: protected

Description:

1.7.6 TPlotAxisBase.GetPixelsPerValue

Synopsis:

Declaration: `function GetPixelsPerValue : Extended;  Virtual`

Visibility: protected

Description:

1.7.7 TPlotAxisBase.Redraw

Synopsis:

Declaration: `function Redraw(ADrawVisible: Boolean) : TRect;  Virtual`

Visibility: protected

Description:

1.7.8 TPlotAxisBase.PlotImage

Synopsis:

Declaration: `function PlotImage : TImage`

Visibility: protected

Description:

1.7.9 TPlotAxisBase.Create

Synopsis:

Declaration: `constructor Create(OwnerPlot: TPlot);  Virtual`

Visibility: public

Description:

1.7.10 TPlotAxisBase.Destroy

Synopsis:

Declaration: `destructor Destroy;  Override`

Visibility: public

Description:

1.7.11 TPlotAxisBase.CheckSize

Synopsis: Delivers size used for drawing

Declaration: `function CheckSize(out ANetAxisRect: TRect) : TRect;  Virtual`

Visibility: public

Description: Used by the plotrect for sizing during redraw

1.7.12 TPlotAxisBase.OwnerPlot

Synopsis: Ownerplot

Declaration: `Property OwnerPlot : TPlot`

Visibility: public

Access: Read,Write

Description: TPLot

1.7.13 TPlotAxisBase.OwnerPlotRect

Synopsis: Containing plotrect

Declaration: `Property OwnerPlotRect : TBasePlotRect`

Visibility: public

Access: Read,Write

Description: One axis has exactly one OwnerPlotRect

1.7.14 TPlotAxisBase.Style

Synopsis: Plotstyle used for the axis

Declaration: `Property Style : TPlotStyleBase`

Visibility: public

Access: Read,Write

Description: Mostly for text size, ticklines and colors

1.7.15 TPlotAxisBase.Orientation

Synopsis: vertical or horizontal

Declaration: `Property Orientation : TAxisOrientation`

Visibility: public

Access: Read,Write

Description: Historic properties. Initially we thought of a typical cartesian coordinate system with one axis vertical and one horizontal. Proeprty still works but is replaced by aoVariable together with a given drawangle.

1.7.16 TPlotAxisBase.ViewRange

Synopsis: viewport of the data

Declaration: `Property ViewRange : TValueRange`

Visibility: public

Access: Read,Write

Description: axis scaling from min value to max value.

1.7.17 TPlotAxisBase.PixelsPerValue

Synopsis: screen pixels per axis value

Declaration: `Property PixelsPerValue : Extended`

Visibility: public

Access: Read

Description: Internal use.

1.7.18 TPlotAxisBase.Visible

Synopsis: visibility of axis

Declaration: `Property Visible : Boolean`

Visibility: public

Access: Read,Write

Description: The Axis is always used for calculations wether visible or not. In case you want to display a axis for indication only at other coordinates, use the Cloneaxis functionality of the TPLotAxis class. Typical use of the visible property is in 3D coordinate systems where the actual axis is at the rear of the 3D plot and should not be drawn with labels attached.

1.8 TPlotHIDHandlerBase

1.8.1 Description

Maps some specific keyboard/mouse input (i.e. shift +left mouse button or mouse wheel) to a plot related function like zoom, pan..

1.8.2 Method overview

Page	Property	Description
23	Create	
23	Destroy	
23	GetHIDActionState	
23	GetHIDActionStatesAvail	
23	SetHIDAction	

1.8.3 Property overview

Page	Property	Access	Description
24	OnMouseDown	r	
24	OnMouseMove	r	
24	OnMouseUp	r	
24	OnMouseWheel	r	
24	OwnerPlot	r	
25	ZoomInfo	r	

1.8.4 TPlotHIDHandlerBase.Create

Synopsis:

Declaration:
```
constructor Create(AOwnerPlot: TPlot);  Virtual
```

Visibility: public

Description:

1.8.5 TPlotHIDHandlerBase.Destroy

Synopsis:

Declaration:
```
destructor Destroy;  Override
```

Visibility: public

Description:

1.8.6 TPlotHIDHandlerBase.GetHIDActionStatesAvail

Synopsis:

Declaration:
```
procedure GetHIDActionStatesAvail(AHIDAction: THIDAction;
                                  out AHIDMouseStates: THIDMouseStates)
                                  ;  Virtual;  Abstract
```

Visibility: public

Description:

1.8.7 TPlotHIDHandlerBase.SetHIDAction

Synopsis:

Declaration:
```
function SetHIDAction(AHIDAction: THIDAction;
                      AHIDMouseState: THIDMouseState) : Integer;  Virtual
                      ;  Abstract
```

Visibility: public

Description:

1.8.8 TPlotHIDHandlerBase.GetHIDActionState

Synopsis:

Declaration:
```
procedure GetHIDActionState(AHIDAction: THIDAction;
                            out AHIDMouseState: THIDMouseState);  Virtual
                            ;  Abstract
```

Visibility: public

Description:

1.8.9 TPlotHIDHandlerBase.OwnerPlot

Synopsis:

Declaration: `Property OwnerPlot : TPlot`

Visibility: public

Access: Read

Description:

1.8.10 TPlotHIDHandlerBase.OnMouseDown

Synopsis:

Declaration: `Property OnMouseDown : TMouseEvent`

Visibility: public

Access: Read

Description:

1.8.11 TPlotHIDHandlerBase.OnMouseUp

Synopsis:

Declaration: `Property OnMouseUp : TMouseEvent`

Visibility: public

Access: Read

Description:

1.8.12 TPlotHIDHandlerBase.OnMouseMove

Synopsis:

Declaration: `Property OnMouseMove : TMouseMoveEvent`

Visibility: public

Access: Read

Description:

1.8.13 TPlotHIDHandlerBase.OnMouseWheel

Synopsis:

Declaration: `Property OnMouseWheel : TMouseWheelEvent`

Visibility: public

Access: Read

Description:

1.8.14 TPlotHIDHandlerBase.ZoomInfo

Synopsis:

Declaration: `Property ZoomInfo : TZoomRecord`

Visibility: public

Access: Read

Description:

1.9 TPlotMenuHelpers

Methods called from the TPlot to invoke some actions like bitmap export or show a helper form (i.e. for manual scaling)

1.9.1 Method overview

Page	Property	Description
25	Create	
25	Destroy	
27	DoExportImportData	
27	DoMenuPlotRectStyleChoose	
26	DoMenuPRAutoScale	
27	DoMenuSeriesColorChoose	
27	DoMenuSeriesMarkersChoose	
27	DoMenuSeriesScale	
27	DoMenuSeriesStyleChoose	
26	EvalMenuAxesOpts	
26	EvalMenuPlotOpts	Handling of popup menu input
26	EvalMenuPlotRectOpts	
26	EvalMenuSeriesOpts	

1.9.2 Property overview

Page	Property	Access	Description
28	OwnerPlot	r	

1.9.3 TPlotMenuHelpers.Create

Synopsis:

Declaration: `constructor Create(AOwnerPlot: TPlot);  Virtual`

Visibility: public

Description:

1.9.4 TPlotMenuHelpers.Destroy

Synopsis:

Declaration: `destructor Destroy;  Override`

Visibility: public

Description:

1.9.5 TPlotMenuHelpers.EvalMenuPlotOpts

Synopsis: Handling of popup menu input

Declaration: `procedure EvalMenuPlotOpts(Sender: TObject)`

Visibility: public

Description: Handling of popup menu input

1.9.6 TPlotMenuHelpers.EvalMenuSeriesOpts

Synopsis:

Declaration: `procedure EvalMenuSeriesOpts(Sender: TObject)`

Visibility: public

Description:

1.9.7 TPlotMenuHelpers.EvalMenuPlotRectOpts

Synopsis:

Declaration: `procedure EvalMenuPlotRectOpts(Sender: TObject)`

Visibility: public

Description:

1.9.8 TPlotMenuHelpers.EvalMenuAxesOpts

Synopsis:

Declaration: `procedure EvalMenuAxesOpts(Sender: TObject)`

Visibility: public

Description:

1.9.9 TPlotMenuHelpers.DoMenuPRAutoScale

Synopsis:

Declaration: `procedure DoMenuPRAutoScale(Sender: TObject)`

Visibility: public

Description:

1.9.10 TPlotMenuHelpers.DoMenuSeriesScale

Synopsis:

Declaration: `procedure DoMenuSeriesScale(ASeriesIndex: Integer)`

Visibility: public

Description:

1.9.11 TPlotMenuHelpers.DoMenuSeriesColorChoose

Synopsis:

Declaration: `procedure DoMenuSeriesColorChoose(ASeriesIndex: Integer)`

Visibility: public

Description:

1.9.12 TPlotMenuHelpers.DoMenuSeriesStyleChoose

Synopsis:

Declaration: `procedure DoMenuSeriesStyleChoose(ASeriesIndex: Integer)`

Visibility: public

Description:

1.9.13 TPlotMenuHelpers.DoMenuSeriesMarkersChoose

Synopsis:

Declaration: `procedure DoMenuSeriesMarkersChoose(ASeriesIndex: Integer)`

Visibility: public

Description:

1.9.14 TPlotMenuHelpers.DoMenuPlotRectStyleChoose

Synopsis:

Declaration: `procedure DoMenuPlotRectStyleChoose(APlotRectIndex: Integer)`

Visibility: public

Description:

1.9.15 TPlotMenuHelpers.DoExportImportData

Synopsis:

Declaration: `procedure DoExportImportData(AImport: Boolean;ASeriesIndex: Integer)`

Visibility: public

Description:

1.9.16 TPlotMenuHelpers.OwnerPlot

Synopsis:

Declaration: `Property OwnerPlot : TPlot`

Visibility: public

Access: Read

Description:

1.10 TPlotSeriesBase

1.10.1 Description

A Series (TPlotSeries) gets the data to be plotted (i.e. from the main application). Series (as implemented so far) have X and Y axis or X,Y and Z axis.
The data is mapped to the screen for display by use of the axes defined.

Do not use the base class - please use the TPlotSeries class. Check different series in uPlotSeries for further functions and properties.

1.10.2 Method overview

Page	Property	Description
31	Clear	delete any stored data
30	Create	
30	Destroy	
30	DoPlotToImage	called when a update is requested
30	DrawPoint	
30	DrawSamplePoint	draw a sample for the legend
29	GetAutoScaleRange	suggested view range for a axis
29	GetAxesUsed	
29	GetUnitString	
29	GetValueRange	
29	PlotImage	the TImage of the TPlot
29	Redraw	
31	UpdateMarkers	called for marker update

1.10.3 Property overview

Page	Property	Access	Description
34	AutoScaleMode	rw	control behaviour of autoscaling
31	AutoScaleRange	r	min and max value for suggested axis-scaling
31	AxesUsed	r	List of axes indices used by the series
33	IsFastSeries	r	standard or fast series
32	OwnerAxis	rw	
32	OwnerPlot	rw	
33	SeriesIndex	r	index of the series
33	SeriesType	r	type of the series
32	Style	rw	style for drawing
32	UnitString	r	Units represented by a specific axis
31	ValueRange	r	min and max values of the data stored
32	Visible	rw	Visibility of the series

1.10.4 TPlotSeriesBase.Redraw

Synopsis:

Declaration: `procedure Redraw;  Virtual`

Visibility: protected

Description:

1.10.5 TPlotSeriesBase.PlotImage

Synopsis:

Declaration: `function PlotImage : TImage;  Virtual`

Visibility: protected

Description:

1.10.6 TPlotSeriesBase.GetAxesUsed

Synopsis:

Declaration: `function GetAxesUsed : TList;  Virtual`

Visibility: protected

Description:

1.10.7 TPlotSeriesBase.GetValueRange

Synopsis:

Declaration: `function GetValueRange(AAxisIndex: Integer) : TValueRange;  Virtual`

Visibility: protected

Description:

1.10.8 TPlotSeriesBase.GetAutoScaleRange

Synopsis:

Declaration: `function GetAutoScaleRange(AAxisIndex: Integer) : TValueRange;  Virtual`

Visibility: protected

Description:

1.10.9 TPlotSeriesBase.GetUnitString

Synopsis:

Declaration: `function GetUnitString(AAxisIndex: Integer) : ShortString;  Virtual`

Visibility: protected

Description:

1.10.10 TPlotSeriesBase.DrawPoint

Synopsis:

Declaration:
```
procedure DrawPoint(Pt: TPoint;Canvas: TCanvas);  Overload
procedure DrawPoint(Pt: TPoint;Canvas: TCanvas;AColor: TColor)
                    ;  Overload
procedure DrawPoint(Pt: TPoint;ADataImage: TLazIntfImage);  Overload
procedure DrawPoint(Pt: TPoint;ADataImage: TLazIntfImage;AColor: TColor)
                    ;  Overload
procedure DrawPoint(Pt: TPoint;ADataImage: TLazIntfImage;
                    AFPColor: TFPColor;AAlphaBlend: Boolean;
                    AAlphaMergeOnly: Boolean);  Overload
```

Visibility: protected

Description:

1.10.11 TPlotSeriesBase.DrawSamplePoint

Synopsis:

Declaration:
```
procedure DrawSamplePoint(Pt: TPoint;Canvas: TCanvas;BeginNew: Boolean)
```

Visibility: protected

Description:

1.10.12 TPlotSeriesBase.Create

Synopsis:

Declaration:
```
constructor Create(OwnerPlot: TPlot);  Virtual
```

Visibility: public

Description:

1.10.13 TPlotSeriesBase.Destroy

Synopsis:

Declaration:
```
destructor Destroy;  Override
```

Visibility: public

Description:

1.10.14 TPlotSeriesBase.DoPlotToImage

Synopsis: called when a update is requested

Declaration:
```
procedure DoPlotToImage;  Virtual;  Abstract
```

Visibility: public

Description: Usually called by the PlotRect redraw routine

1.10.15 TPlotSeriesBase.UpdateMarkers

Synopsis: called for marker update

Declaration: `procedure UpdateMarkers(AContainerIndex: Integer); Virtual; Abstract`

Visibility: public

Description: Markes keep track of a peak list and in general refer to the data stored within the series. UpdateMarkers updates the markers for the new data.

1.10.16 TPlotSeriesBase.Clear

Synopsis: delete any stored data

Declaration: `procedure Clear; Virtual; Abstract`

Visibility: public

Description: delete any stored data

1.10.17 TPlotSeriesBase.AxesUsed

Synopsis: List of axes indices used by the series

Declaration: `Property AxesUsed : TList`

Visibility: public

Access: Read

Description: Indices of the axes used by this series (i.e. X axis and Y axis)

1.10.18 TPlotSeriesBase.ValueRange

Synopsis: min and max values of the data stored

Declaration: `Property ValueRange[AAxisIndex: Integer]: TValueRange`

Visibility: public

Access: Read

Description: min and max values of the data stored

1.10.19 TPlotSeriesBase.AutoScaleRange

Synopsis: min and max value for suggested axis-scaling

Declaration: `Property AutoScaleRange[AAxisIndex: Integer]: TValueRange`

Visibility: public

Access: Read

Description: suggested scaling of a given axis. Depending on autscalemode property

1.10.20 TPlotSeriesBase.OwnerPlot

Synopsis:

Declaration: `Property OwnerPlot : TPlot`

Visibility: public

Access: Read,Write

Description:

1.10.21 TPlotSeriesBase.OwnerAxis

Synopsis:

Declaration: `Property OwnerAxis : Integer`

Visibility: public

Access: Read,Write

Description:

1.10.22 TPlotSeriesBase.Style

Synopsis: style for drawing

Declaration: `Property Style : TPlotStyleBase`

Visibility: public

Access: Read,Write

Description: use a TSeriesStyle for this

1.10.23 TPlotSeriesBase.Visible

Synopsis: Visibility of the series

Declaration: `Property Visible : Boolean`

Visibility: public

Access: Read,Write

1.10.24 TPlotSeriesBase.UnitString

Synopsis: Units represented by a specific axis

Declaration: `Property UnitString[AAxisIndex: Integer]: ShortString`

Visibility: public

Access: Read

Description: Series have units (like meters, seconds, volts).

1.10.25 TPlotSeriesBase.SeriesIndex

Synopsis: index of the series

Declaration: `Property SeriesIndex : Integer`

Visibility: public

Access: Read

Description: index of the series

1.10.26 TPlotSeriesBase.IsFastSeries

Synopsis: standard or fast series

Declaration: `Property IsFastSeries : Boolean`

Visibility: public

Access: Read

Description: The historic mode of operation is drawing points directly onto the canvas of the TImage (PlotImage). Series working that way are TPlotSeries, TXYPlotSeries and TXYZPlotSeries. For those historic series, 'IsFastSeries' is false. The newer concept is drawing on a memory bitmap (TLazIntfImage) and overlay it to the screen during TPLotRect.Redraw.
This concept is significantly faster, therefore 'IsFastSeries' = true for these series. Note that Redrawing significantly differs between fast and standard series. Fast and standard series cannot be mixed in one plotrect.

1.10.27 TPlotSeriesBase.SeriesType

Synopsis: type of the series

Declaration: `Property SeriesType : TSeriesType`

Visibility: public

Access: Read

Description: currently implemented types:

- stBASE: do not use
- stPLAIN: X/Y data, no storage
- stXY: X/Y data, with storage
- stXYZ: X/Y/Z data, with storage
- stSPECTRUM: X/Y data, with storage, indirect plotting to memory bitmap, data given as complete row
- stWF2D: X/Y/Z data, with storage, indirect plotting to memory bitmap, data given as complete row; Y coordinate is invisble but mapped to a color. "Waterfall display"
- stWF3D: X/Y/Z data, with storage, indirect plotting to memory bitmap, data given as complete row Y coordinate is mapped to a color. "Waterfall display"

1.10.28 TPlotSeriesBase.AutoScaleMode

Synopsis: control behaviour of autoscaling

Declaration: `Property AutoScaleMode : TAutoScaleMode`

Visibility: public

Access: Read,Write

Description: for possible modes, see type declaration of TAutoScaleMode

1.11 TPlotStyleBase

1.11.1 Description

Drawing of points, circles etc. is implemented in a descendant of TPlotStyleBase. Please use a TPlotStyle or descendant, do not use the base style.

1.11.2 Method overview

Page	Property	Description
35	Create	
35	Destroy	
34	DrawPoint	Draw a point to the canvas
34	DrawSamplePoint	

1.11.3 TPlotStyleBase.DrawPoint

Synopsis: Draw a point to the canvas

Declaration:
```
procedure DrawPoint(Pt: TPoint;Canvas: TCanvas);  Virtual;  Overload
procedure DrawPoint(Pt: TPoint;Canvas: TCanvas;AColor: TColor);  Virtual
                    ;  Overload
procedure DrawPoint(Pt: TPoint;ADataImage: TLazIntfImage);  Virtual
                    ;  Overload
procedure DrawPoint(Pt: TPoint;ADataImage: TLazIntfImage;AColor: TColor)
                    ;  Virtual;  Overload
procedure DrawPoint(Pt: TPoint;ADataImage: TLazIntfImage;
                    AFPColor: TFPColor;AAlphaBlend: Boolean;
                    AAlphaMergeOnly: Boolean);  Virtual;  Overload
```

Visibility: protected

Description: Draw a point to the canvas

1.11.4 TPlotStyleBase.DrawSamplePoint

Synopsis:

Declaration:
```
procedure DrawSamplePoint(Pt: TPoint;Canvas: TCanvas;BeginNew: Boolean)
                          ;  Virtual;  Overload
```

Visibility: protected

Description:

1.11.5 TPlotStyleBase.Create

Synopsis:

Declaration: `constructor Create; Virtual`

Visibility: public

Description:

1.11.6 TPlotStyleBase.Destroy

Synopsis:

Declaration: `destructor Destroy; Override`

Visibility: public

Description: