

Chapter 1

Reference for unit 'uPlotSeries'

1.1 Used units

Table 1.1: Used units by unit 'uPlotSeries'

Name	Page
Classes	??
dateutils	??
ExtCtrls	??
FPImage	??
Graphics	??
GraphMath	??
GraphType	??
IntfGraphics	??
LCLType	??
math	??
System	??
sysutils	??
Types	??
uMAADataTypes	??
uPlotAxis	??
uPlotClass	??
uPlotDataTypes	??
uPlotInterpolator	??
uPlotUtils	??
uPlot__ResultWriter	??
useriesmarkers	??

1.2 Constants, types and variables

1.2.1 Constants

`c_FITMARGIN` = 0.04

`c_LegendSampleWidth` = 10

```
c_TraceMode_Names : Array[0..2] of string = ('ClearWrite', 'AverageN', 'MaxHold')
```

1.2.2 Types

```
TDrawTarget = packed record
  drwCanvas : TCanvas;
  drwBitMap : TBitmap;
  drwRect : TRect;
end
```

```
TTraceMode = (smClearWrite, smAveragingN, smMaxHold)
```

Table 1.2: Enumeration values for type TTraceMode

Value	Explanation
smAveragingN	display averaged N data
smClearWrite	display newest data
smMaxHold	display max value

1.3 TPlotSeries

1.3.1 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. Within this plot component, Series get the data to be plotted and perform the actual mapping to the screen. Therefore the data is not separated from the drawer in this component which means the "Series" is the datacontainer and the drawer. TPlotSeries is the fundamental class able to draw data 2D data onto the screen. However TPlotSeries does not store the data so any Redraw will be disappointing. Please use TXYPlotSeries or TXYspectrumPlotSeries for 2D plotting.

1.3.2 Method overview

Page	Property	Description
5	AddUnit	add a unit of measure to a axis
4	CheckRefresh	check whether the series needs a refresh
5	Clear	Clear all data
4	Create	
4	Destroy	
4	DoPlotToImage	trigger dataimage refresh
5	DrawLegendSample	draw a sample point
5	DrawXY	draw a datapoint onto the screen
3	GetAxesUsed	
5	GetLineEndpoint	Endpoint for LineStyle
3	GetUnitString	
3	PlotImage	The TImage component of the TPlot
3	Redraw	redraws the stored data
4	UpdateMarkers	update the markers

1.3.3 Property overview

Page	Property	Access	Description
6	AutoDraw	rw	unused
6	Caption	rw	Axis caption
6	ColoredAxis	rw	Color the values of a given axis
7	Interpolate	rw	Interpolate bewteen datapoints
7	Interpolator	r	Interpolator class
7	MarkerContainer	r	container for the markers
7	TransParency	rw	Transparency (opacity) for teh dataplot
7	Units	rw	Set all units
6	XAxis	rw	Index of the X axis
6	YAxis	rw	Index of the Y axis

1.3.4 TPlotSeries.Redraw

Synopsis: redraws the stored data

Declaration: procedure Redraw; Override

Visibility: protected

Description: not for simple TPlotSeries as nothing is stored. Otherwise the complete data stored in the Series is replotted. Note: When large Datasets are stored - for example 1000 lines of data each consisting of 2048 points - a Redraw might take some time.

1.3.5 TPlotSeries.PlotImage

Synopsis: The TImage component of the TPlot

Declaration: function PlotImage : TImage; Override

Visibility: protected

Description: The TImage component of the TPlot

1.3.6 TPlotSeries.GetAxesUsed

Declaration: function GetAxesUsed : TList; Override

Visibility: protected

Description:

1.3.7 TPlotSeries.GetUnitString

Synopsis:

Declaration: function GetUnitString(AAxisIndex: Integer) : ShortString; Override

Visibility: protected

Description:

1.3.8 TPlotSeries.Create

Synopsis:

Declaration: constructor Create(AOwnerPlot: TPlot); Override

Visibility: public

Description:

1.3.9 TPlotSeries.Destroy

Synopsis:

Declaration: destructor Destroy; Override

Visibility: public

Description:

1.3.10 TPlotSeries.CheckRefresh

Synopsis: check wether the series needs a refresh

Declaration: function CheckRefresh(var ANeedsRefresh: Boolean;
var ANeedsClear: Boolean) : Integer; Virtual

Visibility: public

Description: check wether the series needs a refresh

1.3.11 TPlotSeries.DoPlotToImage

Synopsis: trigger dataimage refresh

Declaration: procedure DoPlotToImage; Override

Visibility: public

Description: called from PlotRect before overlay of dataimage to the screen. Multiple Series in one PlotRect share the dataimage. If one or all Series need refresh, all Series plot to the dataimage before the plotrect draws it onto the screen.

1.3.12 TPlotSeries.UpdateMarkers

Synopsis: update the markers

Declaration: procedure UpdateMarkers(AContainerIndex: Integer); Override

Visibility: public

Description: When Data has changed (i.e. when CheckRefresh = true) also the markers need to be updated (PeakLists etc.). Markerupdate is also neccessary when markers are added, removed or their properties are changed. Failure to call UpdateMarkers will result in erroneous marker drawing as these are based on older data.

1.3.13 TPlotSeries.GetLineEndpoint

Synopsis: Endpoint for LineStyle

Declaration: `function GetLineEndpoint(out ALineEndPoint: TPoint) : Integer; Virtual`

Visibility: public

Description: Since implementation of the Interpolator the endpoint for LineStyles can no longer be calculated on the fly. GetLineEndPoint delivers the endpoint for the last drawn datapoint.

1.3.14 TPlotSeries.DrawXY

Synopsis: draw a datapoint onto the screen

Declaration: `procedure DrawXY(X: Extended;Y: Extended); Virtual`

Visibility: public

Description: Draw only, no store

1.3.15 TPlotSeries.Clear

Synopsis: Clear all data

Declaration: `procedure Clear; Override`

Visibility: public

Description: Clear all data

1.3.16 TPlotSeries.AddUnit

Synopsis: add a unit of measure to a axis

Declaration: `procedure AddUnit(Aaxis: Char;AUnit: string)`

Visibility: public

Description: Axes must be given as 'x', 'y' or 'z', not the index.

1.3.17 TPlotSeries.DrawLegendSample

Synopsis: draw a sample point

Declaration: `function DrawLegendSample(X: Integer;Y: Integer;ACanvas: TCanvas;
ADrawVisible: Boolean) : Integer`

Visibility: public

Description: used by the LegendRect

1.3.18 TPlotSeries.XAxis

Synopsis: Index of the X axis

Declaration: Property `XAxis` : Integer

Visibility: public

Access: Read,Write

Description: Index of the X axis

1.3.19 TPlotSeries.YAxis

Synopsis: Index of the Y axis

Declaration: Property `YAxis` : Integer

Visibility: public

Access: Read,Write

Description: Index of the Y axis

1.3.20 TPlotSeries.AutoDraw

Synopsis: unused

Declaration: Property `AutoDraw` : Boolean

Visibility: public

Access: Read,Write

Description: unused

1.3.21 TPlotSeries.ColoredAxis

Synopsis: Color the values of a given axis

Declaration: Property `ColoredAxis` : Integer

Visibility: public

Access: Read,Write

Description: Color the values of a given axis

1.3.22 TPlotSeries.Caption

Synopsis: Axis caption

Declaration: Property `Caption` : string

Visibility: public

Access: Read,Write

Description: Axis caption

1.3.23 TPlotSeries.Units

Synopsis: Set all units

Declaration: Property Units : TStrings

Visibility: public

Access: Read,Write

Description: same purpose as AddUnit

1.3.24 TPlotSeries.Transparency

Synopsis: Transparency (opacity) for teh dataplot

Declaration: Property TransParency : Word

Visibility: public

Access: Read,Write

Description: any number between \$0 ... \$FFFF

1.3.25 TPlotSeries.MarkerContainer

Synopsis: container for the markers

Declaration: Property MarkerContainer : TMarkerContainer

Visibility: public

Access: Read

Description: A Series can have Markers (Seriesmarkers). All markers of a series are handeled by the Markercontainer.

1.3.26 TPlotSeries.Interpolate

Synopsis: Interpolate bewteen datapoints

Declaration: Property Interpolate : Boolean

Visibility: public

Access: Read,Write

Description: Connects subsequent datapoints with lines.

1.3.27 TPlotSeries.Interpolator

Synopsis: Interpolator class

Declaration: Property Interpolator : TInterpolator

Visibility: public

Access: Read

Description: Class doing the interpolation main routine: Interpolate(out AXYPixelLine: TXYPixelLine; Aline: TXYLine): Integer; Gets a dataline and delivers a pixelline

1.4 TXYPlotSeries

1.4.1 Description

The TXYPlotSeries is the standard 2D Series. It directly plots onto the PlotImage of the TPlot. It stores the data to be plotted.

1.4.2 Method overview

Page	Property	Description
10	AddValue	Add a datapoint
10	CheckRefresh	check whether the series needs a refresh
10	Clear	Clear all data
9	Create	
9	Destroy	
10	ExportSeriesData	write the data to a file
9	GetAutoScaleRange	
8	GetNumStoredItems	
9	GetValueRange	
10	ImportSeriesData	read data from a file
9	IsInViewRange	X and Y value inside viewrange
8	Redraw	redraws the stored data
9	SetMaxStoredItems	

1.4.3 Property overview

Page	Property	Access	Description
11	MaxStoredItems	rw	maximum number of datapoints stored
11	NumStoredItems	r	actual number of datapoints stored

1.4.4 TXYPlotSeries.Redraw

Synopsis: redraws the stored data

Declaration: `procedure Redraw; Override`

Visibility: protected

Description: not for simple TPlotSeries as nothing is stored. Otherwise the complete data stored in the Series is replotted. Note: When large Datasets are stored - for example 1000 lines of data each consisting of 2048 points - a Redraw might take some time.

1.4.5 TXYPlotSeries.GetNumStoredItems

Synopsis:

Declaration: `function GetNumStoredItems : Integer`

Visibility: protected

Description:

1.4.6 TXYPlotSeries.SetMaxStoredItems

Synopsis:

Declaration: procedure SetMaxStoredItems(const AValue: Integer)

Visibility: protected

Description:

1.4.7 TXYPlotSeries.GetValueRange

Synopsis:

Declaration: function GetValueRange(AAxisIndex: Integer) : TValueRange; Override

Visibility: protected

Description:

1.4.8 TXYPlotSeries.GetAutoScaleRange

Synopsis:

Declaration: function GetAutoScaleRange(AAxisIndex: Integer) : TValueRange; Override

Visibility: protected

Description:

1.4.9 TXYPlotSeries.IsInViewRange

Synopsis: X and Y value inside viewrange

Declaration: function IsInViewRange(XValue: Extended; YValue: Extended) : Boolean

Visibility: protected

Description: Check whether a given X/ Y value is inside the viewrange.

1.4.10 TXYPlotSeries.Create

Synopsis:

Declaration: constructor Create(AOwnerPlot: TPlot); Override

Visibility: public

Description:

1.4.11 TXYPlotSeries.Destroy

Synopsis:

Declaration: destructor Destroy; Override

Visibility: public

Description:

1.4.12 TXYPlotSeries.CheckRefresh

Synopsis: check wether the series needs a refresh

Declaration: `function CheckRefresh(var ANeedsRefresh: Boolean;
var ANeedsClear: Boolean) : Integer; Override`

Visibility: public

Description: check wether the series needs a refresh

1.4.13 TXYPlotSeries.AddValue

Synopsis: Add a datapoint

Declaration: `procedure AddValue(X: Extended;Y: Extended)`

Visibility: public

Description: TXYPlotSeries take single datapoints with X and Y value

1.4.14 TXYPlotSeries.ExportSeriesData

Synopsis: write the data to a file

Declaration: `function ExportSeriesData(AFileName: TFilename) : Integer; Virtual`

Visibility: public

Description: write the data to a file

1.4.15 TXYPlotSeries.ImportSeriesData

Synopsis: read data from a file

Declaration: `function ImportSeriesData(AFileName: TFileName) : Integer; Virtual`

Visibility: public

Description: read data from a file

1.4.16 TXYPlotSeries.Clear

Synopsis: Clear all data

Declaration: `procedure Clear; Override`

Visibility: public

Description: Clear all data

1.4.17 TXYPlotSeries.MaxStoredItems

Synopsis: maximum number of datapoints stored

Declaration: Property MaxStoredItems : Integer

Visibility: public

Access: Read,Write

Description: maximum number of datapoints stored

1.4.18 TXYPlotSeries.NumStoredItems

Synopsis: actual number of datapoints stored

Declaration: Property NumStoredItems : Integer

Visibility: public

Access: Read

Description: actual number of datapoints stored

1.5 TXYspectrumPlotSeries

1.5.1 Description

This is the fast 2D PlotSeries. It was implemented to get a spectrumanalyser display (or oscilloscope etc.) able to plot many datapoints at once. The XY data is given as a vector. You can plot a arbitrary number of points at once, for example 2048 datapoints in one vector. You can expect to be able to plot more than 200kPts per second, i.e. 100 datasets with 2048 points each per second. Note: 100 screen updates per second might not be a good idea. If you have more than let's say 40 updates per second, please check the TimedRefresh property of the TPlot.

1.5.2 Method overview

Page	Property	Description
13	AddLine	
13	CheckRefresh	
13	Clear	
12	Create	
12	Destroy	
14	DoPlotToImage	
12	DrawLine	
13	ExportSeriesData	
13	GetLineEndpoint	
12	GetValueRange	
13	ImportSeriesData	
12	Redraw	

1.5.3 Property overview

Page	Property	Access	Description
14	ActualLine	rw	
14	AverageN	rw	set N for averaging
14	DisplayedLine	r	unused
15	StoredLined	r	unused
14	TraceMode	rw	Trace mode like in a spectrum analyzer

1.5.4 TXYspectrumPlotSeries.GetValueRange

Synopsis:

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

Visibility: `protected`

Description:

1.5.5 TXYspectrumPlotSeries.DrawLine

Synopsis:

Declaration: `procedure DrawLine(ALine: TXYLine); Virtual`

Visibility: `protected`

Description:

1.5.6 TXYspectrumPlotSeries.Redraw

Synopsis:

Declaration: `procedure Redraw; Override`

Visibility: `protected`

Description:

1.5.7 TXYspectrumPlotSeries.Create

Synopsis:

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

Visibility: `public`

Description:

1.5.8 TXYspectrumPlotSeries.Destroy

Synopsis:

Declaration: `destructor Destroy; Override`

Visibility: `public`

Description:

1.5.9 TXYSpectrumPlotSeries.CheckRefresh

Synopsis:

Declaration: `function CheckRefresh(var ANeedsRefresh: Boolean;
var ANeedsClear: Boolean) : Integer; Override`

Visibility: public

Description:

1.5.10 TXYSpectrumPlotSeries.GetLineEndpoint

Synopsis:

Declaration: `function GetLineEndpoint(out ALineEndPoint: TPoint) : Integer; Override`

Visibility: public

Description:

1.5.11 TXYSpectrumPlotSeries.ExportSeriesData

Synopsis:

Declaration: `function ExportSeriesData(AFileName: TFilename) : Integer; Override`

Visibility: public

Description:

1.5.12 TXYSpectrumPlotSeries.ImportSeriesData

Synopsis:

Declaration: `function ImportSeriesData(AFileName: TFileName) : Integer; Override`

Visibility: public

Description:

1.5.13 TXYSpectrumPlotSeries.Clear

Synopsis:

Declaration: `procedure Clear; Override`

Visibility: public

Description:

1.5.14 TXYSpectrumPlotSeries.AddLine

Synopsis:

Declaration: `procedure AddLine(ALine: TXYLine)`

Visibility: public

Description:

1.5.15 TXYSpectrumPlotSeries.DoPlotToImage

Synopsis:

Declaration: procedure DoPlotToImage; Override

Visibility: public

Description:

1.5.16 TXYSpectrumPlotSeries.ActualLine

Synopsis:

Declaration: Property ActualLine : Int64

Visibility: protected

Access: Read,Write

Description:

1.5.17 TXYSpectrumPlotSeries.TraceMode

Synopsis: Trace mode like in a spectrum analyzer

Declaration: Property TraceMode : TTraceMode

Visibility: public

Access: Read,Write

Description: TTraceMode = (smClearWrite, smAveragingN, smMaxHold) ClearWrite draws one line at a time (always the newest) AveragingN draws the average out of N lines MaxHold draws the maximum out of all lines

1.5.18 TXYSpectrumPlotSeries.AverageN

Synopsis: set N for averaging

Declaration: Property AverageN : Integer

Visibility: public

Access: Read,Write

Description: set N for averaging

1.5.19 TXYSpectrumPlotSeries.DisplayedLine

Synopsis: unused

Declaration: Property DisplayedLine : TXYLine

Visibility: public

Access: Read

Description: unused

1.5.20 TXYspectrumPlotSeries.StoredLined

Synopsis: unused

Declaration: Property StoredLined[AIndex: Integer]: TXYLine

Visibility: public

Access: Read

Description: unused

1.6 TXYWF3DPlotSeries

1.6.1 Description

Draws a waterfall display out of 2D data. Basically the same as the 2D waterfall. However the Y axis is also drawn as usual. The plot is shifted in direction of the Z axis. For the 3D waterfall the Z axis is not vertical but has an arbitrary angle.

1.6.2 Method overview

Page	Property	Description
15	Create	
16	Destroy	
16	DoPlotToImage	
15	DrawLine	
15	Redraw	

1.6.3 TXYWF3DPlotSeries.DrawLine

Synopsis:

Declaration: procedure DrawLine(ALine: TXYLine); Override

Visibility: protected

Description:

1.6.4 TXYWF3DPlotSeries.Redraw

Synopsis:

Declaration: procedure Redraw; Override

Visibility: protected

Description:

1.6.5 TXYWF3DPlotSeries.Create

Synopsis:

Declaration: constructor Create(AOwnerPlot: TPlot); Override

Visibility: public

Description:

1.6.6 TXYWF3DPlotSeries.Destroy

Synopsis:

Declaration: destructor Destroy; Override

Visibility: public

Description:

1.6.7 TXYWF3DPlotSeries.DoPlotToImage

Synopsis:

Declaration: procedure DoPlotToImage; Override

Visibility: public

Description:

1.7 TXYWFPlotSeries

1.7.1 Description

Draws a waterfall display out of 2D data

1.7.2 Method overview

Page	Property	Description
18	AddLine	Add a dataline
18	Clear	
17	Create	
17	Destroy	
18	DoPlotToImage	
16	DrawLine	
18	ExportSeriesData	
17	GetAxesUsed	
17	GetValueRange	
18	ImportSeriesData	
19	MarkNext	next dataline is marked
17	Redraw	

1.7.3 Property overview

Page	Property	Access	Description
19	ActualBaseLine	rw	internal use only
19	LineWidth	rw	Linewidth
19	TimePerLine	rw	Time per dataline
19	YThreshold	rw	Threshold

1.7.4 TXYWFPlotSeries.DrawLine

Synopsis:

Declaration: procedure DrawLine(ALine: TXYLine); Virtual

Visibility: protected

Description:

1.7.5 TXYWFPlotSeries.Redraw

Synopsis:

Declaration: procedure Redraw; Override

Visibility: protected

Description:

1.7.6 TXYWFPlotSeries.GetValueRange

Synopsis:

Declaration: function GetValueRange(AAxisIndex: Integer) : TValueRange; Override

Visibility: protected

Description:

1.7.7 TXYWFPlotSeries.GetAxesUsed

Synopsis:

Declaration: function GetAxesUsed : TList; Override

Visibility: protected

Description:

1.7.8 TXYWFPlotSeries.Create

Synopsis:

Declaration: constructor Create(AOwnerPlot: TPlot); Override

Visibility: public

Description:

1.7.9 TXYWFPlotSeries.Destroy

Synopsis:

Declaration: destructor Destroy; Override

Visibility: public

Description:

1.7.10 TXYWFPlotSeries.DoPlotToImage

Synopsis:

Declaration: procedure DoPlotToImage; Override

Visibility: public

Description:

1.7.11 TXYWFPlotSeries.ExportSeriesData

Synopsis:

Declaration: function ExportSeriesData(AFileName: TFilename) : Integer; Override

Visibility: public

Description:

1.7.12 TXYWFPlotSeries.ImportSeriesData

Synopsis:

Declaration: function ImportSeriesData(AFileName: TFileName) : Integer; Override

Visibility: public

Description:

1.7.13 TXYWFPlotSeries.Clear

Synopsis:

Declaration: procedure Clear; Override

Visibility: public

Description:

1.7.14 TXYWFPlotSeries.AddLine

Synopsis: Add a dataline

Declaration: procedure AddLine(ALine: TXYLine)

Visibility: public

Description: Data is give as a complete vector of XY data. Length is arbitrary. The Waterfall series is basically a 3D series but gets 2D data only. X data is drawn as usual. Y data is drawn as a colored point only. Z data is not given. The first line of data is plottet to the Z=0 coordinate. When subsequent datalines are given, the plot is shifted along the Z coordinate and the new dataline is plottet again to the Z== coordinate. So the Z coordinate is 'faked' - usually it is time. In order to have correct Z axis scaling TimePerLine needs to be set and the datalines need to represent equidistant data. If dataline given are NOT equidistant, the Z axis will still be drawn equidistant leading to a wrong Z axis scaling. Full handlig of the Z coordinate is planned but not implemented in the present version for speed issues.

1.7.15 TXYWFPlotSeries.MarkNext

Synopsis: next dataline is marked

Declaration: procedure MarkNext

Visibility: public

Description: merely a debug function.

1.7.16 TXYWFPlotSeries.LineWidth

Synopsis: Linewidth

Declaration: Property LineWidth : Extended

Visibility: public

Access: Read,Write

Description: Width of each dataline drawn. Smoothest display is achieved with integer values (i.e. 1 to 5). However shifting can be done with non integer values. When LineWidth is set, TimePerLine is adjusted accordingly.

1.7.17 TXYWFPlotSeries.TimePerLine

Synopsis: Time per dataline

Declaration: Property TimePerLine : Extended

Visibility: public

Access: Read,Write

Description: Distance between consecutive datalines. As mostly this would be in time units, the properties name is TimePerLine. WhenTimePerLine is set, LineWidth is adjusted accordingly.

1.7.18 TXYWFPlotSeries.ActualBaseLine

Synopsis: internal use only

Declaration: Property ActualBaseLine : Cardinal

Visibility: public

Access: Read,Write

Description: currently unused ? Used during Redraw to limit the number of lines to be redrawn to the minimum required (i.e. when 5000 lines are stored and the screen height is only 1000 lines, only 20% need to be redrawn).

1.7.19 TXYWFPlotSeries.YThreshold

Synopsis: Threshold

Declaration: Property YThreshold : Extended

Visibility: public

Access: Read,Write

Description: Values greater than the Threshold are plotted. Values below the threshold are transparent.

1.8 TXYZPlotSeries

1.8.1 Description

3D PlotSeries

1.8.2 Method overview

Page	Property	Description
21	AddValue	Add a datapoint
22	Clear	Clear all data
21	Create	
21	Destroy	
20	DrawXY	
21	GetAxesUsed	
21	GetUnitString	
20	GetValueRange	Valuerange of stored data
21	IsInViewRange	
20	Redraw	

1.8.3 Property overview

Page	Property	Access	Description
22	ZAxis	rw	Index of the Z axis

1.8.4 TXYZPlotSeries.DrawXY

Synopsis:

Declaration: procedure DrawXY(X: Extended;Y: Extended;Z: Extended); Reintroduce

Visibility: protected

Description:

1.8.5 TXYZPlotSeries.Redraw

Synopsis:

Declaration: procedure Redraw; Override

Visibility: protected

Description:

1.8.6 TXYZPlotSeries.GetValueRange

Synopsis: Valuerange of stored data

Declaration: function GetValueRange(AAxisIndex: Integer) : TValueRange; Override

Visibility: protected

Description: Returns min and max value of the data for a given axis index

1.8.7 TXYZPlotSeries.GetAxisUsed

Synopsis:

Declaration: function GetAxesUsed : TList; Override

Visibility: protected

Description:

1.8.8 TXYZPlotSeries.GetUnitString

Synopsis:

Declaration: function GetUnitString(AAxisIndex: Integer) : ShortString; Override

Visibility: protected

Description:

1.8.9 TXYZPlotSeries.IsInViewRange

Synopsis:

Declaration: function IsInViewRange(XValue: Extended; YValue: Extended;
ZValue: Extended) : Boolean

Visibility: protected

Description:

1.8.10 TXYZPlotSeries.Create

Synopsis:

Declaration: constructor Create(AOwnerPlot: TPlot); Override

Visibility: public

Description:

1.8.11 TXYZPlotSeries.Destroy

Synopsis:

Declaration: destructor Destroy; Override

Visibility: public

Description:

1.8.12 TXYZPlotSeries.AddValue

Synopsis: Add a datapoint

Declaration: procedure AddValue(X: Extended; Y: Extended; Z: Extended)

Visibility: public

Description: TXYZPlotSeries take single datapoints with X, Y and Z value

1.8.13 TXYZPlotSeries.Clear

Synopsis: Clear all data

Declaration: procedure `Clear`; `Override`

Visibility: `public`

Description: Clear all data

1.8.14 TXYZPlotSeries.ZAxis

Synopsis: Index of the Z axis

Declaration: Property `ZAxis` : `Integer`

Visibility: `public`

Access: `Read,Write`

Description: Index of the Z axis