Contents

The following Help Topics are available:

For Help on Help, Press F1

Introduction

Welcome to Chart FX !!!, the most advanced graphics library for Borland Delphi.

The electronics manual has been designed to allow you to read as much as or as little as you need. Every section provides logical separation of the information you'll need to get the most from Chart FX. A new type of helpful reference has been added to this help file; this is:

Icon Convention: which allows you to easily detect what type of reference or property you are accesing.

If you are a new Chart FX programmer we suggest the following steps to start using this library:

1. Consult the **technical introduction** section to see how to integrate Chart FX to your development tool, and the files provided to achieve this task.

2. Carefully read the first two topics of the **programmer's guide** found in this help file. This will tell you how to create your first chart and pass data to it.

3. Continue reading the **programmer's guide** topics to get used to Chart FX properties and events.

After you know how to create and pass data to a chart, you can access quick and easy the properties provided in this help file to change (modify) the aspects of the chart according to your needs.

The chapters contained in the present help file are resumed in the following way:

Programmer's Guide: Getting started, a simple and concise way of starting using and mastering Chart FX, with description of the properties and events used to build the examples.

Properties & Methods Reference: A complete description of all the properties provided by the library, with examples that show how to use them in the majors development tools.

Events Reference: All events generated by the library are explained in detail here.

Conventions: This chapter explains the conventions used in this help, that includes text, icons and variable names used.

Constants: An alphabetical index of the constants declared by the library.

About Software FX, Inc: Important information of how to contact Software FX.

Technical Introduction

Based on the true fact that the end users wants control over the chart, we have included in Chart FX several tools that allow them to modify virtually any aspects of the chart, including: colors, patterns, gallery types, rotation, fonts, titles, grids and even the plotted data. Therefore, we suggest you create these tools and give the end users full access to them. This will dramatically reduce your programming efforts and your end users will appreciate the freedom given to manipulate the charts. You will find documentation of these tools in the users's guide of the printed manual.

Chart FX includes the following files

CFX32.OCX (OLE Control)
CFX32.LIC (License file)
CFXOCX.PAS (Include File. Contains all the constants needed to work with Chart FX)

Upgrade Information:

This CHART FX version is 2.0. You will find an Upgrade coupon inside your DELPHI package that entitles you to upgrade to Chart FX 3.0 for a very good price.

You can Contact Software FX, Inc. at **1-800-FXCHART** for more information on Chart FX 3.0.

Properties & Methods Reference. Chart FX Component

About	PaletteBar	SerLeg	Value	Adm
IniValue	Pattern	ShowDialog	VertGridGap	Decimals
ItemColor	PatternBar	ShowStatus	View3D	Angles3D
DecimalsNum	ItemWidth	PixFactor	Stacked	ViewRot3D
AutoIncrement	Edit	ItemStyle	PointType	WallWidth
BarHorzGap	KeyLeg	StatusText	BGap	FixedBkColor
KeySer	Style	XLegType	BkColor	FixedColor
ReturnValue	Tag	XValue	Chart3D	FixedGap
Legend	RGap	TopGap	YLeg	ChartStatus
FixedStyle	LegendWidth	RGB2DBk	ThisBkColor	ChartType
FixedWidth	LeftGap	RGB3DBK	ThisColor	FixLeg
LineBkColor	RGBBarHorz	ThisPoint	Color	Fonts
LineColor	RGBBk	ThisSerie	Const	LineStyle
RGBFont	ThisValue	ConstType	Grid	LineWidth
Title	hCtlWnd	NSeries	hFont	NValues
Scheme	ToolBar	CustTool	hText	Type

Methods

CloseData	CopyData	CopyBitmap
DblClk	ExportFile	ImportFile
OpenDataEx	Language	Paint
PrintIt	ReadTemplate	RigClk
Scroll	SetStatusItem	SetStripe
WriteTemplate		

Events Reference

ChangeColor
ChangeFont
ChangePalette
ChangePattern
ChangePattPal
ChangeString
ChangeType
ChangeValue
Destroy
GetLegend
GotFocus
LButtonDblClk
LostFocus
Menu
RButtonDown
ReadFile
ReadTemplate
UserScroll

Conventions

Constants and Structures

Constants and Structures	Used in
Chart Types	Type Property
Chart Status Constants	ChartStatus Property
Chart Styles	Style Property
chart_OpenData Constants	Index in OpenDataEx Method
Adm Constants	Index in Adm Property
Click Styles	Index in DblClk and RigClk methods
Color Schemes	Scheme Property
Decimal Items	Index in DecimalsNum Property
Fonts List	Font Property
Font Types	Index in Font Property
Grid Styles	Grid Property
High-Low constants	ThisSerie Property
Line Styles	LineStyle and FixedStyle Properties
Point Types	PointType Property
Stack Styles	Stacked Property
Status Items Styles	Stacked Property
Titles Styles	dwStyle field in CHART_STITEM struct
Tools Constants	CustTool Property

Software FX, Inc.
P.O. BOX 810893
Boca Raton FL 33481-0893

Voice: (407) 998-2377
Fax: (407) 998-2383

If you are a registered user of Chart FX and have any question about this or any of our products, please call Borland Technical support line.

Icons Conventions

Icon	Used when property or function ...
	Changes the appearance of the chart
	Works only in 3D mode
	Is only supported by horizontal bar charts
	Interact with the clipboard
	Change how colors are displayed
	Interact with files
	Interact with printer
	Modify the chart response
	Works with status bars
	Change the structure of the chart
	Is text related
	Is only supported by scatter charts
	Can be changed by end user interface (Menu, Dialog, ToolBar, etc.)

Property Scope

	The property is supported.
	The property is not supported.
	The property shows a dialog at design time.

Sample Code

Borland Delphi Sample Code.

Text Conventions

Typeface	Description
Bold	**Constants and functions defined by CHART FX**
`Blue courier`	`Borland Delphi Sample code`
Bold Italic	***Important Note***
[Brackets]	Optional code
BLDP	Borland Delphi

Variable names Conventions

Prefix	Type	Size	Description
b	**BOOL**	16 bits	Boolean (TRUE - FALSE)
n	**INT**	16 bits	Integer
w	**WORD**	16 bits	Unsigned Integer
l	**LONG**	32 bits	Long
dw	**DWORD**	32 bits	Unsigned Long
hwnd	**HWND**	16 bits	Window Handle
h	**HANDLE**	16 bits	Generic Handle
lp	**LONG POINTER**	32 bits	Pointer
s	**LPSTR**	32 bits	String Pointer

8. International Support

In order to support all the possible foreign languages, we provide a special directory called INTSUP. In this directory CHART FX install all the necessary files to make a DLL (Dynamic Link Library) with the resources that the library needs.

These files are:

File	Use	Proposed changes
IDMCHART.H	Header file	***Do not change this file !***
LANG20.DEF	Definition file	Change the LIBRARY topic to the name you will use for
LANG20.C	C Code	***Do not change this file !***
LANG20.H	Header file	***Do not change this file !***
LANG20.RC	Resource file	Change all the resources (Dialogs, Menu and String Table) to the language you need to support.

We also supply two makefiles to make even easier the process of making your own resources.

File	Development Tool
makefile	Borland C Compiler.

About Property. Chart FX Component 

Description

This property shows the usual "About" dialog box with version information.

Design Time 

RunTime 

Adm Property. Chart FX Component 

Description

A float property (Single Type in Borland Delphi) that sets administration values of the chart, the index supplied specify the related value.

Borland Delphi

```
[Tform.] Chart1.Adm[Index] := setting;
```

Remarks

The indexes that can be used with this property and the meaning of these settings are:

Index	What the value represents
CSA_MIN	Minimum value used in Y-axis
CSA_MAX	Maximum value used in Y-axis
CSA_GAP	Gap used in Y-axis. This value is calculated automatically each time you create a chart. If you want your Y-axis to put thick at a set interval, you would specify it with this index.
CSA_SCALE	Used scale. This constant is very useful when the values used in Y-axis are too big (i.e. 10.000.000,oo) in this case you can use a 1.000.000 scale and the values on the Y-axis will be divided by this scale.
CSA_YLEGGAP	Equivalent units of a y legend text.
CSA_PIXXVALUE	Equivalent unit representation of Y-axis in pixels.
CSA_XMIN	Minimum value used in X-axis (Scatter)
CSA_XMAX	Maximum value used in X-axis (Scatter)
CSA_XGAP	Gap used in X-axis (Scatter)

Data Type

Float (Single)

Design Time 

RunTime 

Angles3D Property. Chart FX Component 


Description

A Long value that sets or returns the angles used to draw the chart in 3DView mode. The X-angle is in the low word and the Y-angle is in the high word. The default value is 0,0.

Borland Delphi

```
[Tform.] Chart1.Angles3D [ := setting& ];
```

Remarks

This property is only used when drawing the chart in 3DView mode (when View3D Property is set to TRUE).

This function is provided to the end user through the rotation dialog.

At design time the programmer can use the ViewRot3D Property to set the initial values for these angles.

Data Type

Long

Design Time 

RunTime 

AutoIncrement Property. Chart FX Component

Description

A boolean property that sets the autoincrement mode, when this value is TRUE, the library will increment ThisPoint and ThisSerie each time you set one of the following properties: ThisBkColor, ThisColor, ThisValue

Borland Delphi

```
[Tform.] Chart1.AutoIncrement [ := setting% ];
```

Remarks

This property resets the ThisPoint and ThisSerie properties to 0 each time the maximum value is reached.

The library will try to increment ThisSerie first, if it is not possible (the maximum value was reached) it will increment ThisPoint (only if needed).

Data Type

Boolean

Design Time 

RunTime 

BarHorzGap Property. Chart FX Component

Description

An integer value (16 bits) that sets or returns the width or gap of the legend (Text) for a horizontal bar chart. The default value is 50.

Borland Delphi

```
[Tform.] Chart1.BarHorzGap [ := setting% ];
```

Remarks

This property affects only horizontal bar charts and is measured in device units (Pixels).

Data Type

Integer

Design Time 

RunTime 

BottomGap Property. Chart FX Component

Description

An integer value (16 bits) that sets or returns the gap between the bottom border of the chart and the border of the Chart FX Component. This gap is used to draw titles.The default value is 40.

Borland Delphi

```
[Tform.] Chart1.BottomGap [ := setting% ];
```

Remarks

This value is measured in device units (Pixels). The end user can modify this distance manually from the chart window by pointing the mouse near each border and dragging it to the desired position. to stop the user from changing this distance please refer to Style Property with **CS_RESIZEABLE** code.

Data Type

Integer

Design Time 

RunTime 

BkColor Property. Chart FX Component 


Description

A color (Long) value that sets or returns the background color that will be used to paint the markers corresponding to the series supplied as the index of the property.

Borland Delphi

```
[Tform.] Chart1.BkColor(Index) [ := setting& ];
```

Remarks

Before using this property the **COD_COLORS** communication channel must be opened with the OpenDataEx Method.

Data Type

Color (Long)

Design Time 

RunTime 

Chart3D Property; Chart FX Component 

Description

A boolean (Integer) value (16 bits) that sets or returns TRUE if the chart is in 3D and FALSE if it is in 2D.

Borland Delphi

```
[Tform.] Chart1.Chart3D [ := setting% ];
```

Remarks

This function is provided to the end user through the menu.

Data Type

Integer

Design Time 

RunTime 

ChartStatus Property; Chart FX Component 

Description

This property returns a long value that specifies what kind of changes the user has made.

Borland Delphi

```
value% := [Tform.] Chart1.ChartStatus;
```

Remarks

See Chart Status Constants for returned values.

Data Type

Long

Design Time 

RunTime 

ChartType Property; Chart FX Component 


Description

An Integer value (16 bits) that sets or returns the gallery type of the chart. (BAR, AREA, etc.). This property is a subset of the Type Property.

Borland Delphi

```
[Tform.] Chart1.ChartType [ := setting% ];
```

Remarks

This function is provided to the end user through the menu.

Design Time 

RunTime 

CloseData Method. Chart FX Component

Description

By calling this method you close the communications channel opened with the **OpenDataEx Method**, It's extremely important that you close all the opened channels.

The value assigned to this property is not used (must be set to zero).

Borland Delphi

```
[Tform.] Chart1.CloseData(nChannel);
```

Remarks

See OpenDataEx Method explanation.

Design Time 

RunTime 

Color Property. Chart FX Component 


Description

A color (Long) value that sets or returns the color that will be used to paint the markers corresponding to the series supplied as the index of the property.

Borland Delphi

```
[Tform.] Chart1.Color(Index) [ := setting& ];
```

Remarks

Before using this property the **COD_COLORS** communication channel must be opened with the OpenDataEx Method.

Data Type

Color (Long)

Design Time 

RunTime 

Const Property. Chart FX Component 

Description

A float property (Single Type in Borland Delphi) that sets or returns the value of a constant in the chart, the number of the constant to set (get) is represented by the index supplied.

Borland Delphi

```
[Tform.] Chart1.Const_(Index) [ := setting! ];
```

Remarks

Before using this property the **COD_CONSTANTS** communication channel must be opened with the OpenDataEx Method.

Data Type

Float (Single)

Design Time 

RunTime 

ConstType Property. Chart FX Component 

Description

An Integer value (16 bits) that sets or returns the type of the constant values and legends. This property affects how these elements are presented in a chart.

Borland Delphi

```
[Tform.] Chart1.ConstType(Index) [ := setting! ];
```

Remarks

This property affect constants setted with the Const Property.

The value must be a combination of the following flags:

CC_HIDETEXT	Hide the constants text.
CC_HIDE	Hide the constants.

Data Type

Integer

Design Time 

RunTime 

CopyBitmap Method; Chart FX Component 

Description

Allows the programmer to copy the chart to the clipboard (As a bitmap).

The value assigned to this property is not used (must be set to zero).

Borland Delphi

```
[Tform.] Chart1.CopyBitmap;
```

Remarks

This function is provided to the end user through the menu if permitted by the programmer by using the **CS_COPY** style.

CopyData Method; Chart FX Component 

Description

Allows the programmer to copy the charted data to the clipboard.

The value assigned to this property is not used (must be set to zero).

Borland Delphi

```
[Tform.] Chart1.CopyData;
```

Remarks

This function is provided to the end user through the menu if permitted by the programmer by using the **CS_COPY** style.

CustTool Property. Chart FX Component

Description

A Long value (32 bits) that sets or returns the visible buttons in the ToolBar.

Borland Delphi

```
[Tform.] Chart1.CustTool [ := setting& ];
```

Remarks

Setting of this property must contain a Bitwise **OR** of Tool Constants

Data Type

Long

Design Time 

RunTime 

DblClk Method. Chart FX Component 

Description

S the response of the library when the user makes a double click on a marker. The index will specify the type of response and the value must always be zero (0) except in the case of **CHART_MENUCLK** index where the value is the handle of the popup menu. The default value is **CHART_BALLOONCLK**.

Borland Delphi

```
[Tform.] Chart1.DblClk(nStyle,lExtra);
```

Remarks

See Click Styles for index supported values. Note that the library will **always** generate the DblClk event.

If you pass a menu to the library you must remember to destroy it when you are finished with it.

Decimals Property. Chart FX Component 

Description

An integer value (16 bits) that sets or returns the number of decimals used to show numbers in the chart. The default value is 2.

Borland Delphi

```
[Tform.] Chart1.Decimals [ := setting% ];
```

Remarks

This function is provided to the end user through the options dialog.

Data Type

Integer

Design Time 

RunTime 

DecimalsNum Property. Chart FX Component 

Description

An integer value (16 bits) that sets or returns the number of decimals used to show specific elements in the chart. The index specifies the item to change the number of decimals.

Borland Delphi

```
[Tform.] Chart1.DecimalsNum[Index] [ := setting% ];
```

Remarks

See Decimal Items for index supported values. This function is provided to the end user through the options dialog.

Data Type

Integer

Design Time 

RunTime 

Edit Property; Chart FX Component 

Description

This is a read-only property that holds a window handle when the user is trying to change a legend or value in the DataEditor.

Borland Delphi

```
hEdit := [Tform.] Chart1.Edit;
```

Remarks

This property must only be used in response of a notification of the ChangeString Event or the ChangeValue Event.

Data Type

Integer

Design Time 

RunTime 

ExportFile Method; Chart FX Component 

Description

Allows the programmer to create a file with all the attributes and data of the current chart.

Borland Delphi

```
[Tform.] Chart1.ExportFile(sFileName);
```

Remarks

sFileName must be a valid file name including path. This function is provided to the end user through the menu if permitted by the programmer by using the **CS_EXPORTFILE** style.

The file containing the saved chart is propietary and you can only display it by using the ImportFile Method or from the chart menu.

FixedBkColor Property. Chart FX Component 

Description

A Color (Long) value that sets or returns the background color used to paint the constant lines. The default value is white (RGB(255,255,255))

Borland Delphi

```
[Tform.] Chart1.FixedBkColor [ := setting& ];
```

Remarks

This property is used when FixedStyle Property is not equal to **CHART_SOLID**.

Data Type

Long

Design Time 

RunTime 

FixedColor Property. Chart FX Component 

Description

A Color (Long) value that sets or returns the foreground color used to paint the constant lines. The default value is red (RGB(255,0,0))

Borland Delphi

```
[Tform.] Chart1.FixedColor [ := setting& ];
```

Data Type

Long

Design Time 

RunTime 

FixedGap Property. Chart FX Component 

Description

An Integer value (16 bits) that sets or returns the minimum width of each unit in the X-axis where 0 means a default value choosen by the library. The default value is 0.

Borland Delphi

```
[Tform.] Chart1.FixedGap [ := setting% ];
```

Remarks

This property is measured in device units (Pixels).

Data Type

Integer

Design Time 

RunTime 

FixedStyle Property. Chart FX Component 

Description

An Integer value (16 bits) that sets or returns the style used to paint the constants lines. The default value is **CHART_DOT**.

Borland Delphi

```
[Tform.] Chart1.FixedStyle [ := setting% ];
```

Remarks

See Line Styles for supported values.

Data Type

Integer

Design Time 

RunTime 

FixedWidth Property. Chart FX Component 

Description

An Integer value (16 bits) that sets or returns the width used to paint the constants lines.The default value is 1.

Borland Delphi

```
[Tform.] Chart1.FixedWidth [ := setting% ];
```

Data Type

Integer

Design Time 

RunTime 

FixLeg Property; Chart FX Component 

Description

A string value that sets or returns the text of the Constant lines, the index supplied specifies the position of the legend.

Borland Delphi

```
[Tform.] Chart1.FixLeg[Index] [ := setting$ ];
```

Remarks

Normally you will supply as many legends as the number of constants of the chart. This is the same number that you supplied for the OpenDataEx Method with the **COD_CONSTANTS** code.

See also Const Property.

Data Type

String

Design Time 

RunTime 

Font Property; Chart FX Component 

Description

A Long value (32 bits) that sets the font used to draw different texts in a chart. The index of this property represents the text to change and the value represents the new font. The value must always be a combination (bitwise OR) of Font Styles.

Borland Delphi

```
[Tform.] Chart1.Fonts[Index] := setting&;
```

Remarks

This function is provided to the end user through the menu. See Font Types for index supported values. See Font Styles for property supported values.

You must take care of giving a TRUE-TYPE font when setting the left or right title.

Data Type

Long

Design Time 

RunTime 

Grid Property. Chart FX Component 

Description

An integer value (16 bits) that sets or returns the type of grid. The default value is **CHART_NONEGRID**.

Borland Delphi

```
[Tform.] Chart1.Grid [ := setting% ];
```

Remarks

This property does not affect Pie Charts. This function is provided to end user through the menu.

See Grid Styles for supported values.

Data Type

Integer

Design Time 

RunTime 

hCtlWnd Property. Chart FX Component

Description

This property returns the Window handle of the control, this value is exactly the same as the **hWnd** property and is provided for Gupta SQLWindows 4.1 compatibility.

Borland Delphi

```
hChartWnd := [Tform.] Chart1.hCtlWnd;
```

Data Type

Window handle (16 bits)

Design Time 

RunTime 

hFont Property; Chart FX Component 

Description

An Integer value (16 bits) that sets the font used to draw different text elements in a chart. The index of this property represents the text to change and the value represents the new font. The value must always be a valid font handle (HFONT)

Borland Delphi

```
[Tform.] Chart1.hFont[Index] := setting%;
```

Remarks

This function is provided to the end user through the menu. See Font Types for index supported values. You must take care of destroying the font passed to this property.

Data Type

Integer

Design Time 

RunTime 

HText Property; Chart FX Component 

Description

A string value that sets or returns the text that Chart FX will use in the dialog or balloon generated by the last double click or right click of the mouse. For example, when the user double clicks with the left mouse button a point in any of the series of the chart a default text containing the Series Name-Legend-point value is automatically generated, if you want to modify this default text, use this property in conjunction with the DblClk or RigClk events.

Borland Delphi

```
[Tform.] Chart1.HText [ := setting$ ];
```

Data Type

String

Design Time 

RunTime 

ImportFile Method; Chart FX Component 

Description

Allows the programmer to read a previously saved file with all the attributes and data into the current chart, this file must be created with the **ExportFile Method** or through the menu.

Borland Delphi

```
[Tform.] Chart1.ImportFile(sFileName);
```

Remarks

sFileName must be a valid file name including path. This function is provided to the end user through the menu if permitted by the programmer by using **CS_IMPORTFILE** style.

IniValue Property. Chart FX Component 

Description

A float property (Single Type in Borland Delphi) that sets or returns the initial value of a point in the chart, the point to set/get is represented by the index and the series is given by the current value of the ThisSerie Property.

Borland Delphi

```
[Tform.] Chart1.IniValue[Index] [ := setting! ];
```

Remarks

Before using this property the **COD_INIVALUES** communication channel must be opened with the OpenDataEx Method. The **IniValue** property is supported by bar charts only.

Data Type

Float (Single)

Design Time 

RunTime 

ItemColor Property. Chart FX Component 

Description

A Color (Long) value that sets or returns the foreground color used to paint the item specified in the index.

Borland Delphi

```
[Tform.] Chart1.ItemColor [ := setting& ];
```

Data Type

Long

Design Time 

RunTime 

ItemStyle Property. Chart FX Component 

Description

An integer value that sets or returns the style used to paint the item specified in the index.

Borland Delphi

```
[Tform.] Chart1.ItemStyle [ := setting& ];
```

Remarks

See Line Styles for supported values.

Data Type

Integer

Design Time 

RunTime 

ItemWidth Property. Chart FX Component 

Description

An integer value that sets or returns the width used to paint the item specified in the index.

Borland Delphi

```
[Tform.] Chart1.ItemWidth [ := setting& ];
```

Data Type

Integer

Design Time 

RunTime 

KeyLeg Property; Chart FX Component 

Description

A string value that sets or returns the text of the X Legend Keys, the index supplied specifies the position of the legend.

Borland Delphi

```
[Tform.] Chart1.KeyLeg[Index] [ := setting$];
```

Remarks

Normally you will supply as many legends as the number of points of the chart. The library will always draw this Key Legends assuming they are short enough.

Data Type

String

Design Time 

RunTime 

KeySer Property; Chart FX Component 

Description

A string value that sets or returns the text of the Series Legend Keys, the index supplied specifies the position of the legend.

Borland Delphi

```
[Tform.] Chart1.KeySer[Index] [ := setting$ ];
```

Remarks

Normally you will supply as many legends as the number of series of the chart. The library will always draw this Key Legends assuming they are short enough.

This property is supported by horizontal bar charts only.

Data Type

String

Design Time 

RunTime 

Language Method; Chart FX Component 

Description

This method is used to change the current language used by the library. This language is represented by a resource DLL that holds the dialogs,strings and menus needed by Chart FX.

Borland Delphi

```
[Tform.] Chart1.Language(sLangDLL);
```

Remarks

sLangDLL is the name of the resource DLL (including path). To support other languages, the package provides the necessary files (RC, DEF, DLG, C) to translate the information and build it to obtain a DLL (Dynamic Link Library) which is the kind of file that this message handles.

For more information on this topic please refer to **International Support** on previous sections of this manual

Loading this DLL follows the Windows standard search for this type of file, therefore you should place the new DLL containing the language support, in your PATH, WINDOWS or SYSTEM directory.

Legend Property; Chart FX Component 

Description

A string value that sets or returns the text of the X Legend, the index supplied specifies the position of the legend.

Borland Delphi

```
[Tform.] Chart1.Legend[Index] [ = setting$ ];
```

Remarks

Normally you will supply as many legends as the number of points of the chart.

Data Type

String

Design Time 

RunTime 

LegendWidth Property. Chart FX Component 

Description

An integer value (16 bits) that sets or returns the width of the legend window.The default value is 100.

Borland Delphi

```
[Tform.] Chart1.LegendWidth [ := setting% ];
```

Remarks

This value is measured in device units (Pixels). The end user can modify this distance manually from the chart window by pointing the mouse near the border of the legend and dragging it to the desired position. This property has no effect is the legend window is hidden.

Data Type

Integer

DesignTime 

RunTime 

LeftGap Property. Chart FX Component 

Description

An integer value (16 bits) that sets or returns the gap between the left border of the chart and the left border of the Chart FX Component. This gap is used to draw titles.The default value is 40.

Borland Delphi

```
[Tform.] Chart1.LeftGap [ := setting% ];
```

Remarks

This value is measured in device units (Pixels). The end user can modify this distance manually from the chart window by pointing the mouse near each border and dragging it to the desired position. to stop the user from changing this distance please refer to Style Property with **CS_RESIZEABLE** code.

Data Type

Integer

Design Time 

RunTime 

LineBkColor Property. Chart FX Component 

Description

A Color (Long) value that sets or returns the background color used to paint the lines in a 2D line chart.The default value is white (RGB(255,255,255)

Borland Delphi

```
[Tform.] Chart1.LineBkColor [ := setting& ];
```

Remarks

This property is used when LineStyle Property is not equal to **CHART_SOLID**.

Data Type

Long

Design Time 

RunTime 

LineColor Property. Chart FX Component 

Description

A Color (Long) value that sets or returns the foreground color used to paint the lines in a 2D line chart.The default value is black (RGB(0,0,0))

Borland Delphi

```
[Tform.] Chart1.LineColor [ := setting& ];
```

Data Type

Long

Design Time 

RunTime 

LineStyle Property. Chart FX Component 

Description

An Integer value (16 bits) that sets or returns the style used to paint the lines in a 2D line chart.

Borland Delphi

```
[Tform.] Chart1.LineStyle [ := setting% ];
```

Remarks

See Line Styles for supported values.

Data Type

Integer

Design Time 

RunTime 

LineWidth Property. Chart FX Component 

Description

An Integer value (16 bits) that sets or returns the width used to paint the lines in a 2D line chart. The default value is 1.

Borland Delphi

```
[Tform.] Chart1.LineWidth [ := setting% ]
```

Data Type

Integer

Design Time 

RunTime 

NSeries Property. Chart FX Component 

Description

An integer value (16 bits) that sets the number of series the chart will have.The default value is 2.

Remarks

At runtime the programmer must use the OpenDataEx Method to set the number of series and points.

Data Type

Integer

Design Time 

RunTime 

NValues Property. Chart FX Component 

Description

An integer value (16 bits) that sets the number of points the chart will have.The default value is 4.

Remarks

At runtime the programmer must use the OpenDataEx Method to set the number of series and points.

Data Type

Integer

Design Time 

RunTime 

OpenDataEx Method. Chart FX Component 

Description

Opens a communication channel to send data to the chart object, the index represents the type of channel to be opened and the value represents the number of items.

Borland Delphi

```
[Tform.] Chart1.OpenDataEx(nChannel,n1,n2);
```

Remarks

Once the data is filled, you must use **CloseData Method** in order to close the communication channel. The indexes that can be used with this property and the meaning of these settings are:

nChannel	What the value represents	Related Property
COD_VALUES	MAKELONG(nSeries,nPoints)	Value
COD_CONSTANTS	nConstants	Const
COD_COLORS	nColors	Color
COD_STRIPES	nStripes	chart_SetStripe function
COD_INIVALUES	MAKELONG(nSeries,nPoints)	XValue
COD_XVALUES	MAKELONG(nSeries,nPoints)	IniValue
COD_STATUSITEMS	nStatusItems	chart_SetStatusItem function

PaletteBar Property; Chart FX Component 

Description

A boolean (Integer) value (16 bits) that determines whether the PaletteBar is visible or hidden.

Borland Delphi

```
[Tform.] Chart1.PaletteBar [ := setting% ];
```

Remarks

This function is provided to the end user through the menu.

Data Type

Integer

Design Time 

RunTime 

Pattern Property. Chart FX Component 


Description

An integer value (16 bits) that sets the pattern used to paint the serie supplied as an index of the property. This value must be less than 16 and represents the index of the pattern in the current PatternBar.

Borland Delphi

```
[Tform.] Chart1.Pattern[Index] = setting%;
```

Remarks

This property affects charts with pattern schemes only (See Scheme Property)

Data Type

Integer

Design Time 

RunTime 

PatternBar Property; Chart FX Component 

Description

A boolean (Integer) value (16 bits) that determines whether the PatternBar is visible or hidden.

Borland Delphi

```
[Tform.] Chart1.PatternBar [ := setting% ];
```

Remarks

This function is provided to the end user through the menu.

Data Type

Integer

Design Time 

RunTime 

PixFactor Property. Chart FX Component 

Description

An Integer value (16 bits) that sets or returns the equivalent unit representation of Y-axis in pixels. This means, that you can change the factor in pixels for each unit in the Y-axis, to accomplish vertical scroll bars in the chart window. The default value is 0.

Borland Delphi

```
[Tform.] Chart1.PixFactor [ := setting% ];
```

Remarks

This property is used in conjunction with the Adm Property (**CSA_PIXXVALUE** code) to modify the pixel factor of the unit in Y-axis.

A value of zero (0) means the library will choose an appropiate PixFactor for the chart to fit the current height.

Data Type

Integer

Design Time 

RunTime 

PointType Property. Chart FX Component 

Description

An integer value (16 bits) that sets or returns the type of point used to paint markers in line, mark, spline and scatter charts. The default value is **CHART_RECTMK**.

Borland Delphi

```
[Tform.] Chart1.PointType [ := setting% ];
```

Remarks

This function is provided to the end user through the options dialog.

See Point Types for supported values.

Data Type

Integer

Design Time 

RunTime 

PrintIt Method; Chart FX Component

Description

This method allows the programmer to print the chart.

The value assigned to this property is not used (must be set to zero).

Borland Delphi

```
[Tform.] Chart1.PrintIt;
```

Remarks

This function is provided to the end user through the menu if permitted by the programmer by using the **CS_PRINTABLE** style.

ReadTemplate Method; Chart FX Component 

Description

Allows the programmer to read a previously saved file with all the visual attributes (Colors, borders, etc.) into the current chart (without including data).

Borland Delphi

```
[Tform.] Chart1.ReadTemplate(sFileName);
```

Remarks

sFileName must be a valid file name including path. This function is provided to the end user through the menu if permitted by the programmer by using the **CS_TEMPLATE** style.

ReturnValue Property. Chart FX Component 

Description

This property sets the value that will be returned in the current event notification. ReturnValue is provided for Gupta SQLWindows 4.1 compatibility.

Borland Delphi

```
[Tform.] Chart1.ReturnValue := setting%;
```

Remarks

See Returning values for a detailed description of how to return values in other development tools.

Data Type

Integer

Design Time 

RunTime 

RightGap Property. Chart FX Component 

Description

An integer value (16 bits) that sets or returns the gap between the right border of the chart and the border of the Chart FX Component. This gap is used to draw titles.The default value is 40.

Borland Delphi

```
[Tform.] Chart1.RightGap [ := setting% ];
```

Remarks

This value is measured in device units (Pixels). The end user can modify this distance manually from the chart window by pointing the mouse near each border and dragging it to the desired position. to stop the user from changing this distance please refer to Style Property with **CS_RESIZEABLE** code.

Data Type

Integer

Design Time 

RunTime 

RGB2DBk Property; Chart FX Component 


Description

A Color (Long) value that sets or returns the color for the 2D charts background. The default value is light gray (RGB(192,192,192)).

Borland Delphi

```
[Tform.] Chart1.RGB2DBk [ := setting& ];
```

Remarks

This function is provided to the end user by allowing him to drag & drop a color from the PaletteBar to any part of this background.

Data Type

Long

Design Time 

RunTime 

RGB3DBK Property; Chart FX Component 



Description

A Color (Long) value that sets or returns the color for 3D charts background. The default value is white (RGB(255,255,255)).

Borland Delphi

```
[Tform.] Chart1.RGB3DBk [ := setting& ];
```

Remarks

This function is provided to the end user by allowing him to drag & drop a color from the PaletteBar to any part of this background. This property is used by 3D charts only.

Data Type

Long

Design Time 

RunTime 

RGBBarHorz Property; Chart FX Component 



Description

A Color (Long) value that sets or returns the color of the X legend background of a horizontal Bar Chart. The default value is cyan (RGB(0,255,255))

Borland Delphi

```
[Tform.] Chart1.RGBBarHorz [ := setting& ];
```

Remarks

This function is provided to the end user by allowing him to drag & drop a color from the PaletteBar to any part of the area of this Legend.

Data Type

Long

Design Time 

RunTime 

RGBBk Property; Chart FX Component 


Description

A Color (Long) value that sets or returns the color for the background sorrounding the chart. The default value is light gray (RGB(192,192,192))

Borland Delphi

```
[Tform.] Chart1.RGBBk [ := setting& ];
```

Remarks

This function is provided to the end user by allowing him to drag & drop a color from the PaletteBar to any part of this background.

Data Type

Long

Design Time 

RunTime 

RGBFont Property; Chart FX Component 


Description

A Color (Long) value that sets the color of the font used to draw different text elements in a chart. The index of this property represents the text to change and the value represents the new color.

Borland Delphi

```
[Tform.] Chart1.RGBFont[Index] := setting&;
```

Remarks

This function is provided to the end user through the menu. See Font Types for index supported values.

At design time the user can change the colors through the Chart FX Component properties.

Data Type

Long

Design Time 

RunTime 

RigClk Method. Chart FX Component 

Description

An Integer value (16 bits) that sets the response of the library when the user makes a right click on a marker. The index will specify the type of response and the value must always be zero (0) except in the case of **CHART_MENUCLK** index where the value is the handle of the popup menu. The default value is **CHART_BALLOONCLK**.

Borland Delphi

```
[Tform.] Chart1.RigClk(nStyle,lExtra);
```

Remarks

See Click Styles for index supported values. Note that the library will **always** generate the RigClk event.

If you pass a menu to the library you must remember to destroy it when you are finished with it.

Scheme Property. Chart FX Component 


Description

An integer value (16 bits) that sets or returns the scheme used to paint the markers. The default value is **CHART_CSSOLID**.

Borland Delphi

```
[Tform.] Chart1.Scheme [ := setting% ];
```

Remarks

This function is provided to end user through the options dialog.

See Color Schemes for supported values.

Data Type

Integer

Design Time 

RunTime 

Scroll Method; Chart FX Component 

Description

This method is used to modify the scroll position of the chart. The index specifies the scroll message (see wParam in WM_HSCROLL) and the value represents the scroll additional information (see lParam in WM_HSCROLL).

Borland Delphi

```
[Tform.] Chart1.Scroll(wScrollCode,lScrollParam);
```

Remarks

You can use this property to "syncronize" two charts in conjunction with the **UserScroll** Event

SerLeg Property; Chart FX Component 

Description

A string value that sets or returns the text of the Series, the index supplied specifies the position of the legend.

Borland Delphi

```
[Tform.] Chart1.SerLeg[Index] [ := setting$ ];
```

Remarks

Normally you will supply as many legends as the number of series of the chart.

Data Type

String

Design Time 

RunTime 

ShowDialog Method; Chart FX Component 

Description

This property is used to show any of the user interface dialogs provided by the library. The index value specifies the dialog to show.

Borland Delphi

```
[Tform.] Chart1.ShowDialog(nDialog,lExtra);
```

Remarks

nDialog	lExtra	Meaning
CDIALOG_EXPORTFILE	Not Used (*)	Export File Dialog
CDIALOG_IMPORFILE	Not Used (*)	Import File Dialog
CDIALOG_WRITETEMPLATE	Not Used (*)	Write Template Dialog
CDIALOG_READTEMPLATE	Not Used (*)	Read Template Dialog
CDIALOG_PAGESETUP	Not Used (*)	Page Setup Dialog
CDIALOG_ABOUT	Not Used (*)	About Dialog
CDIALOG_OPTIONS	Not Used (*)	Options Dialog
CDIALOG_EDITTITLES	Not Used (*)	Edit Titles Dialog
CDIALOG_FONTS	Font Types	Change Font Dialog
CDIALOG_ROTATE	Not Used (*)	Rotate Dialog

**** Not used values must be set to zero.***

ShowStatus Property; Chart FX Component 

Description

A boolean (Integer) value (16 bits) that determines whether the StatusBar is visible or hidden.

Borland Delphi

```
[Tform.] Chart1.ShowStatus := setting%;
```

Remarks

This function is provided to the end user through the menu if the statusBar has been previously created by the programmer. See also **chapter 5 of the programmer's guide**.

Data Type

Integer

Design Time 

RunTime 

Stacked Property. Chart FX Component 

Description

An integer value (16 bits) that sets or returns the type of stack used to draw area and bar charts. The default value is **CHART_NOSTACKED**.

Borland Delphi

```
[Tform.] Chart1.Stacked [ := setting% ];
```

Remarks

This property affects Bar and Area charts only. This function is provided to the end user through the options dialog.

See Stack Styles for supported values.

Data Type

Integer

Design Time 

RunTime 

StatusText Property; Chart FX Component 

Description

A string value that sets or returns the text of an existent status item. The index supplied represent the code (ID) of the item.

Borland Delphi

```
[Tform.] Chart1.StatusText[Index] [ := setting$ ];
```

Remarks

This property must be used only after you create the status bar with the SetStatusItem Method

Data Type

String

Design Time 

RunTime 

Style Property. Chart FX Component 

Description

A long value (32 bits) that sets or returns the style of the chart. This style refers to what the end user can do in the chart window, thus permitting to change from one type of chart to another, modify 3D View, Rotation, etc.

Borland Delphi

```
[Tform.] Chart1.Style [ := setting& ];
```

Remarks

Setting of this property must contain a bitwise **OR** of Chart Styles constants

Data Type

Long

Design Time 

RunTime 

Tag Property. Chart FX Component 

Description

Stores any extra data needed for your program. Unlike other properties, the value of the Tag property is not used by Borland Delphi; you can use this property to identify objects. The default value of this property is an empty string.

Borland Delphi

```
[Tform.] Chart1.Tag [ := setting% ];
```

Remarks

This property is not visible at design-time and cannot be saved.

Data Type

String

Design Time 

RunTime 

TopGap Property. Chart FX Component 

Description

An integer value (16 bits) that sets or returns the gap between the top border of the chart and the border of the Chart FX Component. This gap is used to draw titles.The default value is 40.

Borland Delphi

```
[Tform.] Chart1.TopGap [ := setting% ];
```

Remarks

This value is measured in device units (Pixels). The end user can modify this distance manually from the chart window by pointing the mouse near each border and dragging it to the desired position. to stop the user from changing this distance please refer to Style Property with **CS_RESIZEABLE** code.

Data Type

Integer

Design Time 

RunTime 

ThisBkColor Property. Chart FX Component 


Description

A color (Long) value that sets the background color that will be used to paint the markers corresponding to the series indicated by the ThisSerie Property.

Remarks

When "Each Bar" Type is set, this property uses the point indicated by the ThisPoint Property.

At runtime you can modify (get) the colors by using the BkColor Property

Data Type

Color (Long)

Design Time 

RunTime 

ThisColor Property. Chart FX Component 


Description

A color (Long) value that sets the color that will be used to paint the markers corresponding to the series indicated by the ThisSerie Property.

Remarks

When "Each Bar" Type is set, this property uses the point indicated by the ThisPoint Property.

At runtime you can modify (get) the colors by using the Color Property

Data Type

Color (Long)

Design Time 

RunTime 

ThisPoint Property; Chart FX Component 

Description

An Integer value (16 bits) that sets or returns the actual point.

Borland Delphi

```
[Tform.] Chart1.ThisPoint [ := setting% ];
```

Remarks

The properties that use **ThisPoint** are: ThisColor, ThisBkColor.

Data Type

Integer

Design Time 

RunTime 

ThisSerie Property; Chart FX Component 

Description

An Integer value (16 bits) that sets or returns the actual serie.

Borland Delphi

```
[Tform.] Chart1.ThisSerie [ := setting% ];
```

Remarks

The properties that use **ThisSerie** are: Value, XValue, IniValue, ThisColor, ThisBkColor.

For High-Low-Close and Open-High-Low-Close charts see HLC and OHCL constants.

Data Type

Integer

Design Time 

RunTime 

ThisValue Property. Chart FX Component 

Description

A float property (Single Type in Borland Delphi) that sets the value of a point in the chart, the point to set (get) is represented by the ThisPoint Property and the serie is given by the ThisSerie Property.

Borland Delphi

```
[Tform.] Chart1.ThisValue := setting!;
```

Remarks

Before using this property the **COD_VALUES** communication channel must be opened with the OpenDataEx Method.

This property works in conjunction with the AutoIncrement Property.

Data Type

Float (Single)

Design Time 

RunTime 

Title Property; Chart FX Component 

Description

A string value that sets or returns the titles of the chart. The type of title to set is specified through the index supplied.

Borland Delphi

```
[Tform.] Chart1.Title[Index] [ := setting$ ];
```

Remarks

This function is provided to the end user through the menu. See Title Types for index supported values.

Data Type

String

Design Time 

RunTime 

ToolBar Property; Chart FX Component 

Description

A boolean (Integer) value (16 bits) that determines whether the ToolBar is visible or hidden.

Borland Delphi

```
[Tform.] Chart1.ToolBar [ := setting% ];
```

Remarks

This function is provided to the end user through the menu.

Data Type

Integer

Design Time 

RunTime 

Type Property. Chart FX Component 

Description

A long value (32 bits) that sets or returns the type of the chart, this type includes gallery type as well as other visual elements in the chart window. The default value is **LINE | CT_SHOWPOINTS**.

Borland Delphi

```
[Tform.] Chart1.Type_ [ := setting& ];
```

Remarks

Setting of this property must contain a bitwise **OR** of Chart Types constants

Data Type

Long

Design Time 

RunTime 

Value Property. Chart FX Component 

Description

A float property (Single Type in Borland Delphi) that sets or returns the value of a point in the chart, the point to set (get) is represented by the index and the serie is given by the ThisSerie Property.

Borland Delphi

```
[Tform.] Chart1.Value[Index] [ := setting! ];
```

Remarks

Before using this property the **COD_VALUES** communication channel must be opened with the OpenDataEx Method.

Data Type

Float (Single)

Design Time 

RunTime 

View3D Property. Chart FX Component 


Description

A Boolean (Integer) value that sets or returns View mode of the chart in 3D. The default value is FALSE.

Borland Delphi

```
[Tform.] Chart1.View3D [ = setting% ]
```

Remarks

Removing the 3D View (FALSE) sets the axes at right angles independent of chart rotation. Setting the 3D View (TRUE) show axes in perspective.

This function is provided to the end user through the rotation dialog.

At design time the programmer must use the ViewRot3D Property

Data Type

Integer

Design Time 

RunTime 

VertGridGap Property. Chart FX Component 

Description

An Integer value (16 bits) that sets or returns the vertical grid gap. This is the distance (measured in X points) between 2 lines of the vertical grid

Borland Delphi

```
[Tform.] Chart1.VertGridGap [ := setting! ];
```

Remarks

The value must be greater than 0

Data Type

Integer

Design Time 

RunTime 

ViewRot3D Property. Chart FX Component 


Description

This property shows a dialog that allows the programmer to set/reset the 3D View and the angles used to draw the chart.

Remarks

At runtime the programmer must use the View3D and Angles3D properties

Design Time 

RunTime 

WallWidth Property. Chart FX Component 

Description

An integer value (16 bits) that sets or returns the wall's width of a 3D Chart. The default value is 8.

Borland Delphi

```
[Tform.] Chart1.WallWidth [ := setting% ];
```

Remarks

This property does not affect 2D Charts and is measured in device units (Pixels).

Data Type

Integer

Design Time 

RunTime 

WriteTemplate Method; Chart FX Component 

Description

Allows the programmer to create a template file with all the visual attributes (Colors, borders, etc) of the current chart without including data.

Borland Delphi

```
[Tform.] Chart1.WriteTemplate(sFileName);
```

Remarks

sFileName must be a valid file name including path. This function is provided to the end user through the menu if permitted by the programmer by using the **CS_TEMPLATE** style.

XLegType Property. Chart FX Component 

Description

An Integer value (16 bits) that sets or returns the type of the X Legend text. This property affects how this legend is presented in a chart.

Borland Delphi

```
[Tform.] Chart1.XLegType[Index] [ := setting! ];
```

Remarks

The value must be a combination of the following flags:

CL_NOTCLIPPED	Chart FX will not clip the X legends (Its programmers responsability to assure that the legends dont overlap each other).
CL_NOTCHANGECOLOR	Chart FX will not change the color of the legends that dont fit in the available space (the default behavior is to draw that legends in RED)
CL_HIDE	Chart FX will not draw X Legend.
CL_GETLEGEND	Chart FX will send an event every time it needs to draw a legend in the Y axis. You can use the **HTEXT Property** to get and set the string to draw. This event (**GetLegend**) will also be sent for the X axis in a scatter chart.

Data Type

Integer

Design Time 

RunTime 

XValue Property. Chart FX Component 

Description

A float property (Single Type in Borland Delphi) that sets or returns the X-axis value of a point in the chart, the point to set (get) is represented by the index and the serie is given by the ThisSerie Property.

Borland Delphi

```
[Tform.] Chart1.XValue[Index] [ := setting! ];
```

Remarks

Before using this property the **COD_XVALUES** communication channel must be opened with the OpenDataEx Method. This property is supported by scatter charts only.

Data Type

Float (Single)

Design Time 

RunTime 

YLeg Property; Chart FX Component 

Description

A string value that sets or returns the text of the Y Legend, the index supplied specifies the position of the legend.

Borland Delphi

```
[Tform.] Chart1.YLeg[Index] [ := setting$ ];
```

Remarks

This property mus be used in conjunction with Adm Property with **CSA_YLEGGAP** code. The value supplied with Adm property is the gap the library will use between the legends.

Data Type

String

Design Time 

RunTime 

SetStatusItem Method. Chart FX Component 

long SetStatusItem(int *nIndex,* **BOOL** *bHaveText,* **UINT** *ID,* **BOOL** *bFramed,* **int** *nWidth,* **int** *nMin,* **int** *nDesp,* **DWORD** *dwStyle***)**

SetStatusItem is a method that allows the programmer to pass items that will appear in the status bar (if created) of the window containing the chart. This function must be accessed after calling the **OpeDataEx method** with the COD_STATUSITEMS constant.

Parameter	Name	Description
int	nIndex	Index of the item
BOOL	bHaveText	Specifies if that item will contain text.
UINT	ID	ID number of the item
BOOL	bFramed	Specifies if that item will have a frame sorrounding it.
int	nWidth	Specifies the width in device units (pixels)
int	nMin	Specifies the minimum tracking width of the item in device units (pixels), in case the user resize the window containing the chart.
int	nDesp	Distance from the last item in device units (pixels)
DWORD	dwStyle	Specifies the style of the text in the item: **STITEM_SLEFT** = Left alignment **STITEM_SCENTER** = Centered **STITEM_SRIGHT** = Right alignment

Comments

This function has no effect without a chart_OpenData function call.

Although there is a way to create and change status bar items in the window containing the chart using chart_Send function and messages, this function is provided to support compatibility with other development tools.

See Also

OpenDataEx Method.

SetStripe Method. Chart FX Component 

long SetStripe(int *nIndex,* **double** *fBegin,* **double** *fEnd,* **DWORD** *dwColor***)**

SetStripe is a method that allows the programmer to pass two values in which a color frame is to be displayed in the background of the chart. This function is very useful when you want to denote a specific area in the chart. This function must be accessed after calling the chart_OpenData function with the COD_STRIPES constant.

Parameter	Name	Description
int	nIndex	Index of the stripe
double	fBegin	Numeric value corresponding to the beginning of the stripe
double	fEnd	Numeric value corresponding to the ending of the stripe
DWORD	dwColor	RGB color of the stripe

Comments

This function has no effect without a chart_OpenData function call.
This function can apply to any type of charts (except pie charts).

Sample

To set a blue stripe from 10 to 20 of a chart :

```
Chart1.OpenDataEx(COD_STRIPES,1,0);
Chart1.SetStripe(0,10,20,RGB(0,0,255);
Chart1.CloseData(COD_STRIPES);
```

See Also

OpenDataEx

Paint Method. Chart FX Component

long Paint(HWND *hChart,* **HDC** *hDC,* **int** *nLeft,* **int** *nTop,* **int** *nRight,* **int** *nBottom,* ***BOOL*** *bPrint,* ***LONG*** *lReserved***)**

Paint is a method that allows the programmer to draw a chart in any device context. This function is very useful when you want to print charts and others objects in the same page or more than one chart in a page.

Parameter	Name	Description
HDC	hDC	Device context where the chart is going to be drawn.
int	nLeft	x-coordinate of the upper-left corner of the bounding rectangle.
int	nTop	y-coordinate of the upper-left corner of the bounding rectangle.
int	nRight	x-coordinate of the bottom-right corner of the bounding rectangle.
int	nBottom	y-coordinate of the bottom-right corner of the bounding rectangle.
BOOL	bPrint	Indicates whether the chart is going to be shown as in the printer or as in the screen (in the printer the background color is not shown).
LONG	lReserved	Reserved. Must be set to 0.

Return Value

None.

Comments

The bounding rectangle must be in device units.

Sample

```
{ To print two charts into a 3000 x 3000 printer }
Chart1.Paint(Printer.Handle, 0, 0, 1500, 3000, 1, 0);
Chart1.Print(Printer.Handle, 1500, 0, 3000, 3000, 1, 0);
```

VBX Compatibility (DELPHI 1.0)

The Following Propeties have been changed to methods:

Old VBX Property	New OCX Method
CloseData	CloseData
CopyData	CopyData
CopyBitmap	CopyBitmap
DblClk	DblClk
ExportFile	ExportFile
ImportFile	ImportFile
OpenData	OpenDataEx
Language	Language
PrintIt	PrintIt
ReadTemplate	ReadTemplate
RigClk	RigClk
Scroll	Scroll
WriteTemplate	WriteTemplate

The Following Functions have been changed to methods:

Old VBX Function	New OCX Method
chart_SetStatusItem	SetStatusItem
chart_SetStripe	SetStripe
chart_Paint	Paint

ChangeColor Event. Chart FX Component

Description

This event occurs when the user changed a color (using the drag&drop technique).

Parameters

Name	Type	Description
nType	Integer	Color code (See remarks)
nIndex	Integer	Index of the changed color.

Remarks

nType Parameter	Description and nIndex Parameter
CCC_SERIE	Color of a series has changed, nIndex represents the series
CCC_SERIEBK	Back Color of a series has changed, nIndex represents the series
CCC_ONE	Color of markers has changed, nIndex ***Not used***.
CCC_ONEBK	Back Color of markers has changed, nIndex ***Not used***.
CCC_BARHORZ	Color the horizontal bar text has changed, nIndex ***Not used***.
CCC_BKGND	Background of the chart has changed, nIndex ***Not used***.
CCC_2DBK	Background of the 2D chart has changed, nIndex ***Not used***.
CCC_3DBK	Background of the 3D chart has changed, nIndex ***Not used***.

Return Value

The return value is not used (***Must NOT be set***)

ChangeFont Event. Chart FX Component

Description

This event occurs when the user changed a font.

Parameters

Name	Type	Description
nIndex	Integer	Index of the changed font.

Remarks

See Font Types for possible values of nIndex.

Return Value

The return value is not used (***Must NOT be set***)

ChangePalette Event. Chart FX Component

Description

This event occurs when the user changed a color in the PaletteBar.

Parameters

Name	Type	Description
nIndex	Integer	Index of the changed color.

Return Value

The return value is not used (***Must NOT be set***)

ChangePattern Event. Chart FX Component

Description

This event occurs when the user changed a pattern (using Drag&Drop feature).

Parameters

Name	Type	Description
nType	Integer	Pattern code (See remarks)
nIndex	Integer	Index of the changed pattern.

Remarks

nType Parameter	Description and nIndex Parameter
CCP_SERIE	Pattern of a series has changed, nIndex represents the series
CCP_ONE	Pattern of markers has changed, nIndex ***Not used***.

Return Value

The return value is not used (***Must NOT be set***)

ChangePattPal Event. Chart FX Component

Description

This event occurs when the user changed a pattern in the PatternBar.

Parameters

Name	Type	Description
nIndex	Integer	Index of the changed pattern.

Return Value

The return value is not used (***Must NOT be set***)

ChangeString Event. Chart FX Component

Description

This event occurs when the user is trying to change a string (legend) in the DataEditor.

Parameters

Name	Type	Description
nType	Integer	Legend code (See remarks)
nIndex	Integer	Index of the legend to is about to be changed.

Remarks

This event is only generated if the style of the chart contains the **CS_EDITABLE** flag.

nType Parameter	Description and nIndex Parameter
CCS_LEGEND	Changing X Legend, nIndex represents the point.
CCS_SERLEGEND	Changing Series Legend, nIndex represents the series.

Return Value

The return value is NULL (default) if Chart FX must accept the user changes.

See Returning values for a detailed description of how to return values in an event.

ChangeType Event. Chart FX Component

Description

This event occurs when the user is trying to change a value in the DataEditor.

Parameters

Name	Type	Description
nType	Integer	New Type.

Remarks

This event is generated when the gallery type of the chart is changed.

Return Value

The return value is not used (***Must NOT be set***)

ChangeValue Event. Chart FX Component

Description

This event occurs when the user is trying to change a value in the DataEditor.

Parameters

Name	Type	Description
dValue	Double	New value to be set.
nSerie	Integer	Clicked serie.
nPoint	Integer	Clicked point.

Remarks

This event is only generated if the style of the chart contains the **CS_EDITABLE** flag.

Return Value

The return value is NULL (default) if Chart FX must accept the user changes.

See Returning values for a detailed description of how to return values in an event.

Destroy Event. Chart FX Component

Description

This event occurs when the chart is about to be destroyed

Parameters

None.

Remarks

Please note that if the chart is being destroyed due to destruction of its parent, this event will not be generated, instead you must use the Destroy event of the parent window. i.e.

```
Borland Delphi:      OnDestroy event
```

Return Value

The return value is not used (***Must NOT be set***)

GetLegend Event. Chart FX Component

Description

This event occurs when Chart FX is about to draw a legend in the Y axis or the X axis of a scatter chart. In order to get/set the string to show you must use the **HText Property**.

Parameters

Name	Type	Description
bYText	Integer	TRUE if drawing the Y axis.

Remarks

This event is only generated if the **XLegType Property** contains the **CL_GETLEGEND** flag..

Return Value

If the return value is different from 0 Chart FX will draw the legend contained in the **Htext.**

See Returning values for a detailed description of how to return values in an event.

GotFocus Event. Chart FX Component

Description

This event occurs when the chart gains the focus

Parameters

None.

Return Value

The return value is not used (***Must NOT be set***)

LButtonDblClk Event. Chart FX Component

Description

This event occurs when the user makes a double click on any portion of the chart.

Parameters

Name	Type	Description
X	Integer	X screen coordinate.
Y	Integer	Y screen coordinate.
nSerie	Integer	Clicked serie.
nPoint	Integer	Clicked point.

Remarks

nSerie and nPoint are both set to **0xFFFF** if the user has not double clicked on any marker.

Return Value

The return value is NULL (default) if Chart FX must process this event according to the chart settings (See DblClk Method), if the return value is not null Chart FX does nothing.

See Returning values for a detailed description of how to return values in an event.

LostFocus Event. Chart FX Component

Description

This event occurs when the chart looses the focus

Parameters

None.

Return Value

The return value is not used (***Must NOT be set***)

Menu Event. Chart FX Component

Description

This event occurs when the user selects a menu item in a programmer defined popup menu displayed by Chart FX. This menus can be shown in response of a double click (See DblClk Method) or a right click of the mouse (See RigClk method).

Parameters

Name	Type	Description
wParam	Integer	ID of the menu item.
nSerie	Integer	Clicked serie.
nPoint	Integer	Clicked point.

Remarks

This event is generated only if the user makes a click over a marker.

Return Value

The return value is not used (***Must NOT be set***)

RButtonDown Event. Chart FX Component

Description

This event occurs when the user makes a right click on any portion of the chart.

Parameters

Name	Type	Description
X	Integer	X screen coordinate.
Y	Integer	Y screen coordinate.
nSerie	Integer	Clicked serie.
nPoint	Integer	Clicked point.

Remarks

nSerie and nPoint are both set to **-1** if the user has not right clicked on any marker.

Return Value

The return value is NULL (default) if Chart FX must process this event according to the chart settings (See RigClk Method), if the return value is not null Chart FX does nothing.

See Returning values for a detailed description of how to return values in an event.

ReadFile Event. Chart FX Component

Description

This event occurs when the user read a file (ImportFile menu option).

Parameters

None.

Return Value

The return value is not used (***Must NOT be set***)

ReadTemplate Event. Chart FX Component

Description

This event occurs when the user read a template (ReadTemplate menu option).

Parameters

None.

Return Value

The return value is not used (***Must NOT be set***)

UserScroll Event. Chart FX Component

Description

This event occurs when the user scroll through the chart.

Parameters

Name	Type	Description
wMsg	Integer	Scroll type (as wParam in WM_HSCROLL)
wParam	Integer	Info (as Loword(lParam) in WM_HSCROLL)

Remarks

This event can be used to "syncronize" two charts in conjunction with the **Scroll** Property.

Return Value

The return value is not used (***Must NOT be set***)

Returning values

In Chart FX there are special events that require the programmer to return different values, due to the lack of standard mechanisms in the VBX engine to return values in an event, we have implemented this in the following manner:

Borland Delphi
A ***"special"*** parameter called **nRes**, this parameter is initialized with NULL before calling the event handler, if you want to return something different you can freely assign a value to this parameter i.e:

```
Procedure Tform1.Chart1LButtonDblClk (Sender:TObject; VAR X,Y,nSerie,nPoint,nRet:
Integer);

       { put your code here }

       nRet := 1;
End;
```

Chart Types Table

LINE	**CT_3D**	**CT_SHOWZERO**	**CT_PIEVALUES**
BAR	**CT_HORZ**	**CT_EACHBAR**	
SPLINE	**CT_TOOL**	**CT_CLUSTER**	
MARK	**CT_PALETTE**	**CT_SHOWDATA**	
PIE	**CT_PATTERN**	**CT_DLGGRAY**	
AREA	**CT_MENU**	**CT_SCATTERLINE**	
PARETO	**CT_LEGEND**	**CT_COLORLINE**	
SCATTER	**CT_TOGETHER**	**CT_NOAREALINE**	
HILOW	**CT_POINTS**	**CT_NOBORDERS**	

Comments:

This first set of constants are used to define the type of chart you want to make or change (Type Property), you should specify only one of them, which means that you will have undesired results if you combine them in a bitwise **OR** (i.e **LINE | BAR**). These are the basic type of charts included in the package. Nevertheless, you can add special effects (i.e. 3D, Rotation, Grid, etc.) depending on the type of chart you are working with.

Constant	Description
LINE	Line Chart
BAR	Bar Chart (Including Horizontal, and stacked charts)
SPLINE	Curve-fitting Chart
MARK	Point Chart
PIE	Pie Chart
AREA	Area Chart (Including stacked charts)
PARETO	Pareto Chart (Statistical Chart. Special)
SCATTER	Scatter Chart
HILOW	Hi-Low Close Chart

The second set of constants are used to define several other aspects of the charts that can be very useful when you create the graph for the first time (instead of making several calls to other set functions). You combine them in a bitwise **OR** with the constants shown above. All of these are turned off by default, so you have to include them to activate these options.These are:

Constant	Description
CT_3D	To specify that the graph will be created or modified with 3D effect, if you include this constant the chart will be created as a 3D chart (if supported).
CT_HORZ	This constant works only with Bar Charts, and if it is included the bar chart will be created as a horizontal bar chart.
CT_TOOL	This constant specifies that the toolbar will be shown in the window containing the chart. This gives the end user access to the tools provided by the toolbar
CT_PALETTE	This constant will turn on the Palette Bar, providing the end user the ability to change the colors of several objects in the chart, such as: series, background, etc.
CT_PATTERN	This constant will turn on the Pattern Bar, providing the end user the ability to change patterns used in the series of the chart.
CT_MENU	This constant will turn on the menu of the chart, that provides the end user access to several options to modify the aspect of the chart.

CT_LEGEND	This constant will turn on the legend window in the chart.
CT_TOGETHER	This constant will cause a Bar Chart to display joined bars.
CT_POINTS	This constant will show the points on a Line or Spline Chart.
CT_SHOWZERO	This constant will cause a chart to set the starting point at zero. For example, if you have a bar chart with a minimum value of -50 and turn on this constant the starting point will be zero and you will have bars that go up or down, depending on their value.
CT_EACHBAR	This constant is used to specify that a chart with a single series will have distinct colors at each data marker. i.e. Each bar will have different colors.
CT_CLUSTER	This constant turn on the cluster options in which each data series is in its own row. To turn on this constant the CT_3D constant must be turned on.
CT_SHOWDATA	This constant turns on the Data Editor (When this options is enabled the chart will not be visible).
CT_DLGGRAY	This constant will cause the dialogs to be shown with a gray background, to provide support for applications that also use gray backgrounds. This keeps the graphics library consistent with the rest of the client application.
CT_SCATTERLINE	This constant specifies that a line must be drawn between the points of a Scatter chart.
CT_COLORLINE	This constant specifies that the lines of a 2D Line Chart must be drawn using colurs (the default behavior is to draw black lines)
CT_NOAREALINE	This constant specifies that the vertical lines of an Area Chart will not be drawn.
CT_NOBORDERS	This constant turns off the borders in bar charts.
CT_PIEVALUES	This constant specifies that the values must be painted in the pie chart (instead of painting the percentages).

Chart Status Constants

Flag	If the user has changed...
CHART_GSVALUES	Values
CHART_GSLEGENDS	X or Series Legend
CHART_GSCOLORS	Colors
CHART_GSPATTERNS	Patterns
CHART_GSPALETTE	Colors Palette
CHART_GSPATTPAL	Pattern Palette
CHART_GSREADTEMP	The user has read a template
CHART_GSREADFILE	The user has read a file
CHART_GSGALLERY	The gallery type
CHART_GSOPTIONS	The options dialog

Chart Styles Table

CS_CHLINE	**CS_SCROLLABLE**	**CS_EDITABLE**
CS_CHBAR	**CS_MINMAX**	**CS_FILEEXPORT**
CS_CHSPLINE	**CS_3D**	**CS_FILEIMPORT**
CS_CHMARK	**CS_HORZ**	**CS_TEMPLATE**
CS_CHPIE	**CS_TOGETHER**	**CS_PRINTABLE**
CS_CHAREA	**CS_STACKED**	**CS_3DVIEW**
CS_CHPARETO	**CS_SHOWPOINT**	**CS_RESIZEABLE**
CS_CHSCATTER	**CS_SCALE**	**CS_COPY**
CS_CHHILOW	**CS_TITLES**	**CS_ALL**
CS_CHDEFAULT	**CS_GRID**	

With these constants you can restrict the access to several functions provided to the end user by the menu or toolbar. Include the following constants in a bitwise OR, and you will activate that feature for the end user.

Constant	Description
CS_CHLINE	This will permit the end user to change to a Line chart, if another type of chart is being displayed. This feature can be accessed from the toolbar or from the menu.
CS_CHBAR	This will permit the end user to change to a Bar chart, if another type of chart is being displayed. This feature can be accessed from the toolbar or from the menu.
CS_CHSPLINE	This will permit the end user to change to a Curve-fitting chart, if another type of chart is being displayed. This feature can be accessed from the toolbar or from the menu.
CS_CHMARK	This will permit the end user to change to a Points chart, if another type of chart is being displayed. This feature can be accessed from the toolbar or from the menu.
CS_CHPIE	This will permit the end user to change to a Pie chart, if another type of chart is being displayed. This feature can be accessed from the toolbar or from the menu.
CS_CHAREA	This will permit the end user to change to an Area chart, if another type of hart is being displayed. This feature can be accessed from the toolbar or from the menu.
CS_CHPARETO	This will permit the end user to change to a Pareto chart (if supported), if another type of chart is being displayed. This feature can be accessed from the menu only
CS_CHSCATTER	This will permit the end user to change to a Scatter chart (if supported), if another type of chart is being displayed. This feature can be accessed from the menu only
CS_CHHILOW	This will permit the end user to change to a Hi-Low-Close chart (if supported), if another type of chart is being displayed. This feature can be accessed from the menu only.
CS_MINMAX	This will permit the end user to change the minimum or maximum value being used in the chart from the options dialog.
CS_3D	This will permit the end user to switch to 3D view any type of chart that is being displayed in 2D, this function can be accessed from the toolbar or from the menu.

CS_HORZ	This will permit the end user to change the aspect of a bar chart to be displayed in horizontal bars. Since this type of chart belongs to the family of standard bar chart, you can restrict the end user from changing to a horizontal bar chart. This option can be accessed from the toolbar or the options dialog.
CS_TOGETHER	By default all the bar charts are created with a gap between each bar. the user can remove this gap from the options dialog if you permit this feature with this constant.
CS_STACKED	This will permit the end user to change the aspect of a chart change to stack type. Since this type of chart belongs to the family of the standard bar (area) chart, you can restrict the user from changing to a stacked bar (area) chart. This option can be accessed from the options dialog
CS_SHOWPOINT	This will permit the end user to show or hide the points in a line, spline or similar charts. This function can be accessed from the options dialog.
CS_SCALE	This will permit the end user to change scale used on the Y-axis from the options dialog.
CS_TITLES	This will permit the end user to change the text being assigned to the different titles supported (Top, Bottom, Right or Left). This function is provided in the options dialog.
CS_FONTS	This will permit the end user to change the fonts used in any of the titles supported. This function is provided in the menu.
CS_EDITABLE	This will permit the end user to change the values actually being graphed. This function is provided in the Data Editor.
CS_FILEEXPORT	This will permit the end user to export (save) the current to a file. This function is provided in the toolbar or menu.
CS_FILEIMPORT	This will permit the end user to import (open) a previously saved chart. This function is provided in the toolbar or menu.
CS_SCROLLABLE	This will permit the end user to scroll if the current chart will not fit in the open window.
CS_PRINTABLE	This will permit the end user to print the contents of the chart window. This function is provided in the toolbar and in the menu.
CS_3DVIEW	This will permit the end user to modify the 3D View by accesing the 3D dialog where the user can rotate the view around the x or Y-axis.
CS_GRID	This will permit the end user to modify the actual grids (Vertical, Horizontal or None) being displayed in the chart. This function is provided in the toolbar and in the menu.
CS_RESIZEABLE	This will permit the end user to modify the internal borders of the chart. Note that if the chart is being displayed in a child window, the end user can resize the graph inside that window, but not the window itself.
CS_TEMPLATE	This will permit the end user to operate (Save or apply) templates to a chart. This function is provided in the menu.

CS_COPY	This will permit the end user to copy bitmap or data of actual chart to the clipboard, functionality provided in the toolbar and menu.
CS_ALL	This will permit the end user to access all the functions explained above.

chart_Get Constants

Constant	To get ...
CHART_GVALUES	plotted values
CHART_GXVALUES	X-axis values (in scatter charts only)
CHART_GINIVALUES	Initial values (in bar charts only)

chart_OpenData Constants

Constant	Open the communication channel to supply ...
COD_VALUES	Plotted values
COD_CONSTANTS	Constants
COD_COLORS	Colors
COD_STRIPES	Stripes
COD_INIVALUES	Initial values (for bar charts only)
COD_XVALUES	X-axis values (for scatter charts only)
COD_STATUSITEMS	StatusBar items

chart_SetAdm Constants

Constant	To change the ...
CSA_MIN	Minimum value used on the Y-axis
CSA_MAX	Maximum value used on the Y-axis
CSA_GAP	Y-axis values Gap.
CSA_SCALE	Used scale.
CSA_YLEYGAP	Equivalent units for a Y-axis legend text.
CSA_PIXXVALUE	Equivalent unit representation of Y-axis in pixels.
CSA_XMIN	Minimum value used on the X-axis (scatter).
CSA_XMAX	Maximum value used on the X-axis (scatter).

Click Styles

CHART_BALLOONCLK	Displays a Balloon Help
CHART_DIALOGCLK	Displays a predefined Dialog
CHART_NONECLK	Nothing
CHART_MENUCLK	Displays a Menu (This value is only supported at runtime)

Color Modes

CHART_FGROUND	To set foreground color
CHART_BGROUND	To set background color (used in pattern schemes)

Color Schemes

CHART_CSSOLID	Use solid colors (BkColor is not used)
CHART_CSBWPATTERN	Use BW Patterns (useful for monochrome devices)
CHART_CSPATTERN	Use color patterns

Decimal Items

Constant	Used to change number of decimals used in ...
CD_ALL	All the items.
CD_VALUES	Point Values (Ballon, Dialog and Data-Editor)
CD_YLEG	Y Legend Values.
CD_XLEG	X Legend Values.

Fonts List

CF_BOLD	**CF_FSCRIPT**	**CF_SCRIPT**
CF_ITALIC	**CF_FDECORATIVE**	**CF_SYMBOL**
CF_UNDERLINE	**CF_ARIAL**	**CF_TIMES**
CF_STRIKEOUT	**CF_COURIER**	**CF_TIMESNR**
CF_FDONTCARE	**CF_COURIERNEW**	**CF_WINGDINGS**
CF_FROMAN	**CF_HELVETICA**	
CF_FSWISS	**CF_MODERN**	
CF_FMODERN	**CF_ROMAN**	

/* Font Effects Supported*/

Constant	Description
CF_BOLD	Specifies whether the font is Bold
CF_ITALIC	Specifies whether the font is Italic
CF_UNDERLINE	Specifies whether the font is Underline
CF_STRIKEOUT	Specifies whether the font is Strikeout

/* Font Families Supported*/
Note: The font family is used in case the font specified is not found (or not installed) in Windows environment, by specifying this family Windows can create a similar font that you want to set.
To obtain more information about font families please refer to Windows SDK help file.

Constant	Description
CF_FDONTCARE	No family given
CF_FROMAN	Specifies whether the family is Roman
CF_FSWISS	Specifies whether the family is Swiss
CF_FMODERN	Specifies whether the family is Modern
CF_FSCRIPT	Specifies whether the family is Script
CF_FDECORATIVE	Specifies whether the family is Decorative

/* Font Typefaces Supported*/

Constant	Description
CF_ARIAL	Specifies whether the Typeface is Arial
CF_COURIER	Specifies whether the Typeface is Courier
CF_COURIERNEW	Specifies whether the Typeface is Courier New
CF_HELVETICA	Specifies whether the Typeface is Helvetica
CF_MODERN	Specifies whether the Typeface is Modern
CF_ROMAN	Specifies whether the Typeface is Roman
CF_SCRIPT	Specifies whether the Typeface is Script
CF_SYMBOL	Specifies whether the Typeface is Symbol
CF_TIMES	Specifies whether the Typeface is Times
CF_TIMESNEWR	Specifies whether the Typeface is Times New Roman
CF_WINGDINGS	Specifies whether the Typeface is WingDings

Font Types

CHART_LEFTFT	Left Title
CHART_RIGHTFT	Right Title
CHART_TOPFT	Top Title
CHART_BOTTOMFT	Bottom Title
CHART_XLEGFT	X Legend
CHART_YLEGFT	Y Legend
CHART_FIXEDFT	Constants
CHART_LEGENDFT	Legend

Grid Styles

CHART_NONEGRID	Remove both grids
CHART_HORZGRID	Set horizontal grid
CHART_VERTGRID	Set vertical grid
CHART_BOTHGRID	Set both grids

High-Low constants

Constant	Description
HLC_HIGH	High Series for a High-Low-Close chart
HLC_LOW	Low Series for a High-Low-Close chart
HLC_CLOSE	Close Series for a High-Low-Close chart
OHLC_OPEN	Open Series for an Open-High-Low-Close chart
OHLC_HIGH	High Series for an Open-High-Low-Close chart
OHLC_LOW	Low Series for an Open-High-Low-Close chart
OHLC_CLOSE	Close Series for an Open-High-Low-Close chart

Line Styles

Constant	Style Description
CHART_SOLID	Solid Pen
CHART_DASH	Dashed Pen
CHART_DOT	Dotted Pen
CHART_DASHDOT	Dash-Dotted Pen
CHART_DASHDOTDOT	Dash-Dot-Dotted Pen

Point Types

CHART_RECTMK	Displays a Rectangle
CHART_CIRCLEMK	Displays a Circle
CHART_TRIANGLEMK	Displays a Triangle
CHART_MARBLEMK	Displays a Marble
CHART_CUBEMK	Displays a Cube (in 2D displays a rectangle)
CHART_MANYMK	Displays different type of markers
CHART_NONEMK	No points

Stack Styles

CHART_NOSTACKED	Remove stacked option.
CHART_STACKED	Set stacked option
CHART_STACKED100	Set stacked 100% option

Status Items Styles

CHART_STLEFT	Left Alignment
CHART_STCENTER	Center
CHART_STRIGHT	Right Alignment

Titles Styles

CHART_LEFTTIT	Left Title
CHART_RIGHTTIT	Right Title
CHART_TOPTIT	Top Title
CHART_BOTTOMTIT	Bottom Title

Tools Constants

Flag	Grouped in
CST_IMPORT	CST_FILE, CST_FILEEDIT
CST_EXPORT	CST_FILE, CST_FILEEDIT
CST_COPYDATA	CST_COPY, CST_FILEEDIT
CST_COPYBITMAP	CST_COPY, CST_FILEEDIT
CST_PRINT	CST_FILEEDIT
CST_AREA	CST_GALLERY
CST_BAR	CST_GALLERY
CST_BARHORZ	CST_GALLERY
CST_LINE	CST_GALLERY
CST_MARK	CST_GALLERY
CST_PIE	CST_GALLERY
CST_SPLINE	CST_GALLERY
CST_PARETO	CST_GALLERYEXT
CST_SCATTER	CST_GALLERYEXT
CST_HILOW	CST_GALLERYEXT
CST_3D	CST_VIEW
CST_ROTATE	CST_VIEW
CST_CLUSTER	CST_VIEW
CST_LEGEND	CST_LEGGRID
CST_VGRID	CST_LEGGRID
CST_HGRID	CST_LEGGRID
CST_TITLES	CST_OTHER
CST_FONTS	CST_OTHER
CST_TOOLS	CST_OTHER
CST_OPTIONS	CST_OTHER

CST_SPACE1
CST_SPACE2
CST_SPACE3
CST_SPACE4

CHART_STITEM struct

```
typedef struct tagSTITEM {
	LPSTR sText;		// Initial Text, NULL means never will have text
	UINT wIdm;		// Text ID, needed to change text
	BOOL bFrame;		// draw frame surrounding
	int nWidth;		// Initial width
	int nMin;		// Minimum width
	int nDesp;		// Offset in pixels respect previous item
	DWORD dwStyle;		// style of text
} CHART_STITEM;
```

' Sample 1 Delphi copy

```
Procedure user_GraphSetData()
{This function set the data to display}

       Chart1.OpenDataEx(COD_VALUES,2,8);

       Chart1.ThisSerie := 0;
       Chart1.Value[0] := 30;
       Chart1.Value[1] := 20;
       Chart1.Value[2] := 40;
       Chart1.Value[3] := 60;
       Chart1.Value[4] := 50;
       Chart1.Value[5] := 15;
       Chart1.Value[6] := 24;
       Chart1.Value[7] := 35;

       Chart1.ThisSerie := 1;
       Chart1.Value[0] := 45;
       Chart1.Value[1] := 60;
       Chart1.Value[2] := 30;
       Chart1.Value[3] := 60;
       Chart1.Value[4] := 80;
       Chart1.Value[5] := 45;
       Chart1.Value[6] := 15;
       Chart1.Value[7] := 45;

       Chart1.CloseData(COD_VALUES);
End;
```

' Sample 2 Delphi copy

```
Procedure user_SetNewValues(d1:Double;d2:Double)
Var
	i:integer;
Begin

	{this function receives the arrays with the new values}

	Chart1.OpenDataEx(COD_VALUES,2,8);

	Chart1.ThisSerie := 0;
	for i:=0 To 7 do
	begin
		Chart1.Value[i] := d1[i];
	end;

	Chart1.ThisSerie := 1;
	for i:=0 To 7 do
	begin
		Chart1.Value[i] := d1[i];
	end;

	Chart1.CloseData(COD_VALUES);
End;

Procedure user_DoSomething()
Var
	i:integer;
	d1: Array [1..8] of Double;
	d2: Array [1..8] of Double;

Begin

	{the code above is only for demonstrating the user_SetNewValues }
	{function in a random way}

	Randomize;

	for i:=0 to 7 do
	begin
		d1[i] := Random(99);
		d2[i] := Random(99);
	end;

	Call user_SetNewValues(d1,d2);
End;
```

' Sample 3 Delphi copy

```
Procedure user_SetStrings()
begin
	{This function sets titles and legends}

	{Titles}
	Chart1.Title[CHART_LEFTTIT] := 'Sales';
	Chart1.Title[CHART_BOTTOMTIT] := 'Months';

	{Legends}
	Chart1.Legend[0] := 'Jan';
	Chart1.Legend[1] := 'Feb';
	Chart1.Legend[2] := 'Mar';
	Chart1.Legend[3] := 'Apr';
	Chart1.Legend[4] := 'May';
	Chart1.Legend[5] := 'Jun';
	Chart1.Legend[6] := 'Jul';
	Chart1.Legend[7] := 'Aug';

	Chart1.SerLeg[0] := 'Product A';
	Chart1.SerLeg[1] := 'Product B';
End;
```

' Sample 4 Delphi copy

```
Procedure user_SetNewApp()
Begin
       {this function changes the appearance of the graph}

       {to set 3D view}
       Chart1.View3D := TRUE;
       Chart1.Angles3D := MAKELONG(30,60);

       {to set the grid}
       Chart1.Grid := CHART_HORZGRID;

       {to set the back colors}
       Chart1.RGBBk := RGB(0,255,0);
       Chart1.RGB2DBk := RGB(200,20,90);
       Chart1.RGB3DBk := RGB(200,20,90);
End;
```

' Sample 5 Delphi copy

```
Procedure user_SetNewTools()
begin
	{this function changes the visual tools of the graph}

	Chart1.OpenData(COD_STATUSITEMS,4,0);

	Chart1.SetStatusItem(0, TRUE, 101, True, 100, 50, 4, CHART_STLEFT);
	Chart1.SetStatusItem(1, TRUE, 102, False, 80, 80, 5, CHART_STLEFT);
	Chart1.SetStatusItem(2, FALSE, 0, True, 40, 40, 10, 0);
	Chart1.SetStatusItem(3, FALSE, 103, True, 50, 30, 4, CHART_STLEFT);

	Chart1.CloseData(COD_STATUSITEMS);

	{ modify the items }
	Chart1.StatusText[101] := 'Text 1';
	Chart1.StatusText[102] := 'Text 2';
	Chart1.StatusText[103] := 'Text 3';

	{show the statusBar}
	Chart1.ShowStatus := TRUE;

	{ Show ToolBar and PaletteBar }
	lType := Chart1.Type_;
	Chart1.Type_ := lType Or CT_TOOL Or CT_PALETTE;
End;
```

' Sample 6 Delphi copy

```
      {This extract of code must be pasted in the lbuttondblclk event}
procedure TForm1.Chart1LButtonDblClk(Sender: TObject; X, Y, nSerie,
  nPoint: Integer, VAR nRes: Integer);
var
   sx,sy,sPoint,sSerie: String;
   dValue: Double;

begin
    Str(X, sX);
    Str(Y, sY);
    Str(nPoint, sPoint);
    Str(nSerie, sSerie);

    If (nSerie = -1) OR (nPoint = -1) Then
        MessageDlg('Empty DobleClick X:' + sX + ' Y:' + sY, mtWarning, [mbOK], 0)
    Else
    begin
        Chart1.ThisSerie := nSerie;
        dValue := Chart1.Value[nPoint];
        If (dValue < 25) Then
            nRes := 1
        Else
            If (dValue < 50) Then
                Chart1.HText := 'Value between 25 and 50'
    end
end;

end.
```

' Sample 7 Delphi copy

```
Function GetSubMenu(hMenu:HMENU; nPos:Integer): HMENU;far;external 'user.exe';

procedure TForm1.FormCreate(Sender: TObject);
begin
    Chart1.OpenDataEx(COD_VALUES,2, 8);

    Chart1.ThisSerie := 0;
    Chart1.Value[0] := 30;
    Chart1.Value[1] := 20;
    Chart1.Value[2] := 40;
    Chart1.Value[3] := 60;
    Chart1.Value[4] := 50;
    Chart1.Value[5] := 15;
    Chart1.Value[6] := 24;
    Chart1.Value[7] := 35;


    Chart1.ThisSerie := 1;

    Chart1.Value[0] := 45;

    Chart1.Value[1] := 60;

    Chart1.Value[2] := 30;

    Chart1.Value[3] := 60;

    Chart1.Value[4] := 80;

    Chart1.Value[5] := 45;

    Chart1.Value[6] := 15;

    Chart1.Value[7] := 45;


    Chart1.CloseData(COD_VALUES);

end;


procedure TForm1.Button1Click(Sender: TObject);

var

   menu: Tmenu;


begin

    { read template }

    Chart1.ReadTemplate('OLDUSER.CHT');


    Chart1.YLeg[0] := 'Really Bad';

    Chart1.YLeg[1] := 'Bad';

    Chart1.YLeg[2] := 'Regular';
```

```
Chart1.YLeg[3] := 'Good';
Chart1.YLeg[4] := 'Excellent';

{ Tell Chart FX how to use the legends (GAP) }
Chart1.Adm[CSA_YLEGGAP] := 20;

{ Set the legends }
Chart1.Legend[0] := 'Peter Gabriel';
Chart1.KeyLeg[0] := 'PG';
Chart1.Legend[1] := 'Phil Collins';
Chart1.KeyLeg[1] := 'PC';
Chart1.Legend[2] := 'Bruce Springsteen';
Chart1.KeyLeg[2] := 'BS';
Chart1.Legend[3] := 'Tracy Chapman';
Chart1.KeyLeg[3] := 'TC';
Chart1.Legend[4] := 'Big Mamma Thornton';
Chart1.KeyLeg[4] := 'BM';
Chart1.Legend[5] := 'Paul Simon';
Chart1.KeyLeg[5] := 'PS';
Chart1.Legend[6] := 'Louis Armstrong';
Chart1.KeyLeg[6] := 'LA';
Chart1.Legend[7] := 'John Lennon';
Chart1.KeyLeg[7] := 'JL';

{ Series Legends }
Chart1.SerLeg[0] := 'First Test';
Chart1.SerLeg[1] := 'Second Test';

Chart1.Type_ := Chart1.Type_ OR CT_LEGEND;

Chart1.LeftGap := 70;

Button1.Enabled := False;

{ RightClk Menu }
```

```
    menu := Form1.Menu;
    Chart1.RigClk(CHART_MENUCLK,CLong(GetSubMenu(menu.Handle,0)));

end;

procedure TForm1.FormDestroy(Sender: TObject);
begin
    Chart1.WriteTemplate('OLDUSER.CHT');
end;

procedure TForm1.Chart1Menu(Sender: TObject; var wParam, nSerie, nPoint,
  nRes: Integer);
var
   s1,s2,s3:string;
begin
     Str(wParam, s1);
     Str(nPoint, s2);
     Str(nSerie, s3);
     MessageDlg('Menu Option:'+s1+' nPoint:'+s2+' nSerie:'+s3, mtWarning, [mbOK], 0);
end;
```

Delphi Programmers Guide

Chapters:

Delphi: Creating a simple chart

This chapter provides an overview of how to start to use the CHART FX library in order to create a simple chart, and covers the following topics:

1.1 Creating a chart.
1.2. Setting the data to display

1.1 Creating a chart

You can create a chart in the same way you create any of the VBX objects in the development tool you're using:

Design Time: You just ***Draw*** the control and set the initial properties.

1.2 Setting the data to display

Once you have created the chart you need at least specify the data that you want to display, note that failing to do that will cause the library to show random values.

In order to specify the data to be shown, you must use the **Value** property, this is a single (float) property that must be used as an array property:

```
Chart1.Value[nPoint] := dValue!;
```

This property tells the library that *dValue* is the value of the point *nPoint* in the serie *"pointed"* by the **ThisSerie** Property.

```
{Serie 0 , Point 3 , Value 10.5 }
Chart1.ThisSerie := 0;
Chart1.Value[3] := 10.5;
```

Nevertheless before using that property you need to be sure that the communications channel to the library is properly open. This is done through the pair of methods **OpenDataEx** and **CloseData**.

Finally your code to set the data will look like this:

```
{Open the VALUES channel specifying 2 Series and 7 Points}
Chart1.OpenDataEx(COD_VALUES,2, 7);

{Code to set the data}
Chart1.ThisSerie := 0;
for i := 0 to 2 do
begin
   Chart1.Value[i] := 9;
end;
Chart1.ThisSerie := 1;
for i := 0 to 2 do
begin
   Chart1.Value[i] := 15;
end;

{Close the VALUES channel}
Chart1.CloseData(COD_VALUES);
```

Related Properties:
OpenDataEx, Value, CloseData.

Delphi: Changing the data of an existent chart

This chapter describes the process involved in changing the data displayed in a chart created previously.

Once the chart is created, you can change any of the values displayed using the same properties explained in the chapter 1 "Creating a simple chart": **OpenDataEx**, **Value**, **CloseData**.

Calling **OpenDataEx** with a new number of series and points will destroy existing data and prepare the communications channel to receive new data.

```
{Open the VALUES channel specifying 4 Series and 8 Points}
{This call would destroy existent data}

Chart1.OpenDataEx(COD_VALUES,4, 8);

{Code to set the data}
for i := 0 to 4 do
begin
   Chart1.ThisSerie := i;
   for j := 0 to 8 do
   begin
      Chart1.Value[j] := 12;
   end;
end;

{Close the VALUES channel}
Chart1.CloseData(COD_VALUES);
```

If you only want to change the values without changing the number of points or series, you can use the flag **COD_UNCHANGE**, which means that you will keep all the old data except for what you change with **Value** property.

```
{Open the VALUES channel and keep number of series and points}
Chart1.OpenDataEx(COD_VALUES,COD_UNCHANGE COD_UNCHANGE);

{Modify an arbitrary point}
Chart1.ThisSerie := 1;
Chart1.Value[4] := 10.5;

{Close the VALUES channel}
Chart1.CloseData(COD_VALUES);
```

Related Properties:
OpenData, Value, CloseData.

Delphi: Changing legends and titles

This chapter provides a brief introduction to the titles and legends that can be set in a chart and covers the following topics:

3.1 Changing titles in a chart.
3.2 Changing X legend and series legend in a chart.
3.3 Restricting the user from change titles.

3.1 Changing titles in a chart

To change the titles of a chart you can set the **TITLE** property with a title code in the index and the new string in the value of the property, as shown below:

```
{Sets or Changes the chart titles}

Chart1.Title[CHART_LEFTTIT] := 'Revenues';
Chart1.Title[CHART_BOTTOMTIT] := 'Months';
```

3.2 Changing X legend and series legend

To change the legends of a chart you use the **Legend** property for the X legend and **SerLeg** property for the series legend and the code required to change that legends will be:

```
{Sets or changes the X legend }
{this code assumes that hwndChart has 2 series and 4 points }

Chart1.Legend[0] := 'Jan';
Chart1.Legend[1] := 'Feb';
Chart1.Legend[2] := 'Mar';
Chart1.Legend[3] := 'Apr';

Chart1.SerLeg[0] := 'Sales';
Chart1.SerLeg[1] := 'Revenues';
```

3.3 Restricting the user from changing titles

Although the library provides a way for the user to change the titles that the programmer specifies, you can restrict this in two ways:

a. Creating the chart without the **CS_TITLES** style.
b. Using the **Style** Property.

Related Properties:
Title, Legend, SerLeg, Style, Chart Styles,

Delphi: Changing visual attributes of a chart

This chapter provides an overview of the properties that Chart FX provide to change the appearance (also called visual attributes) of a chart and covers the following topics:

4.1 Managing 2D, 3D view

CHART FX provides 3 modes to display graphics:

a. 2D mode
b. 3D "plain" mode
c. 3D (with rotation angles) or 3D view mode (Needs 3D mode)

The default mode when you create a chart is 2D mode , nevertheless you can override this default and set 3D "plain" mode at design and running time using the **Type** Property or the **Chart3D** Property as follows:

```
{to set 3D plain mode}
Chart1.Chart3D := TRUE;
```

In order to set 3D view mode, the chart has to be in 3D "plain" mode and also need the following calls:

```
{to set 3D view with 30 X degrees and 60 Y degrees}
Chart1.View3D := TRUE;
Chart1.Angles3D := MAKELONG(30,60);

{to remove 3D view}
Chart1.View3D := FALSE;
```

4.2 Setting color schemes, grid and background colors

In this topic we will see how to change some of the attributes that define the way the chart is displayed. All these attributes can be changed by the end user through the menu bar and related dialogs.

Color scheme: Describes the way that CHART FX "paints" the markers (bars, points, pie slices, etc.) and can be Solid (means solid colors), BW Patterns (black and white patterns) and Patterns (color patterns).

Grid: Describes the horizontal and vertical lines drawn in the chart.

Background colors: Colors used to draw the different backgrounds in the chart.

Lets see the properties that provides this interaction:

```
{to set the color scheme}
Chart1.Scheme := CHART_CSPATTERN;

{to set the grid}
Chart1.Grid := CHART_HORZGRID;
```

```
{to set the back colors}
Chart1.RGBBk := RGB(0,255,0);
Chart1.RGB2DBk := RGB(200,20,90);
Chart1.RGB3DBk := RGB(200,20,90);
```

4.3 Changing Fonts

Although CHART FX uses a default Arial 8 Pts font to display titles, legends, and other texts in the chart, the programmer can change any of the fonts used by the library with the **Font, hFont** and **RGBFont** properties. There are two ways the programmer can specify a font:

a. Restricted Mode: (Font Property)

In this mode the programmer passes to the library a bitwise OR of the flags that defines a font: a typeface (**CF_ARIAL, CF_TIMES**, etc.) , effects (**CF_BOLD, CF_ITALIC**, etc.) the familiy (**CF_FDONTCARE, CF_FSWISS**, etc.) and the size in points of the font.

b. Free Mode: (hFont Property)

In this mode the programmer passes to the library a handle to a previously created font (HFONT). Note that in this mode it is a programmer's responsability to create and destroy the font using Windows functions (CreateFont and DeleteObject)

i.e.

```
{Set a title so we can see it}
Chart1.Title[CHART_BOTTOMTIT] := 'Times Font';

{to change the bottom title font}
Chart1.Font[CHART_BOTTOMFT] := CF_TIMES Or CF_BOLD Or 12;

{to change the color use to draw that font}
{remember Windows will use the nearest solid color}
Chart1.RGBFont[CHART_BOTTOMFT] := RGB(255,0,0);
```

Related Properties:
Type, View3D, Scheme, Grid, RGBBk, RGB2DBk, RGB3DBk, Font, hFont, RGBFont.

Delphi: Creating tools and other visual elements

This chapter provides a brief description of the visual elements that CHART FX can show and the way that the programmer can do it and covers the following topics:

5.1 Showing and hiding the ToolBar, LegendBar, PaletteBar and PatternBar.
5.2 Creating a StatusBar.
5.3 Showing and modifying a StatusBar.

5.1 Showing and hiding the ToolBar, LegendBar, PaletteBar and PatternBar.

CHART FX gives to the end user 4 ToolWindows that offer an easy and intuitive way of changing some of the characteristics of the chart:

Window	Flag	Used to ...
ToolBar	**CT_TOOL**	Change 3D view, gallery type, grids and so on.
LegendBar	**CT_LEGEND**	Show X and series legend.
PaletteBar	**CT_PALETTE**	Change colors using the drag & drop technique
PatternBar	**CT_PATTERN**	Change patterns using the drag & drop technique

From the programmers point of view the work this tools provide is completely transparent and can be customized in two ways:

At design time:
Setting the appropiate flags in the **Type** Propery.

At runtime:
Using the **Type_** Property accordingly.

```
Chart1.Type_ := BAR Or CT_TOOL Or CT_LEGEND;
```

5.2 Creating a StatusBar

CHART FX has built-in code to create and draw StatusBars, this kind of window is widely used in many comercial applications to inform the end user of the status of the program and relevant information.
There are two ways of creating a status bar and which one you choose depend basically on the development tool you are using.

Microsoft Visual Basic

If the development tool you are using does not provide user-defined structures (or casting) the easiest way to create a status bar is using **OpenDataEx** and **CloseData** properties in conjunction with the **chart_SetStatusItem** function

```
chart_SetStatusItem(HWND hwndChart,int nItem,BOOL bText, UINT wIdm,...)
```

The parameters of the **chart_SetStatusItem** are the window handle of the chart, the number of items to set and the rest of the parameters (excepting bText) are the same as explained in the **CHART_STITEM** structure. The only limitation is that you cannot initialize text items.

An example code to create a StatusBar would be:

```
{Open the communications channel}
Chart1.OpenDataEx(COD_STATUSITEMS,4,0);
```

```
{Set the items}
hWnd = Chart1.handle;
Chart1.SetStatusItem(0,TRUE,IDM_TEXT1,TRUE,100,50,4,CHART_STLEFT);
Chart1.SetStatusItem(1,TRUE,IDM_TEXT2,TRUE,80,80,5,CHART_STCENTER);
Chart1.SetStatusItem(2,FALSE,NULL,TRUE,40,40,10,NULL);
Chart1.SetStatusItem(3,TRUE,IDM_TEXT3,TRUE,50,30,2,CHART_STRIGHT);

{Close the items channel}
Chart1.CloseData(COD_STATUSITEMS);
```

5.3 Showing and modifying a StatusBar

The code needed to show/hide a StatusBar is:

```
{TRUE means Show, FALSE means HIDE}
Chart1.ShowStatus := TRUE;
```

The code needed to modify a StatusBar text item is:

```
{wIdm is the UINT code you assign to the item at creation time}
Chart1.StatusText[wIdm] := 'New Text';
```

Related Properties:
OpenDataEx, CloseData, ShowStatus, CHART_STITEM, chart_SetStatusItem.

Delphi: Handling notification messages 

This chapter explains what kind of events a chart generates, and how to handle this messages. This is covered by the following topics:

6.1 What an event is

Events are the standard way controls inform parent windows of changes and related information, this is the way all controls work in VBX environments and is also how CHART FX works.

The events a chart can generate are:

Event	Explanation
*LButtonDblClk	The user has double clicked any portion of the chart.
*RButtonDown	The user has right clicked any portion of the chart.
*ChangeString	The user wants to change a string
*ChangeValue	The user wants to change a value
Destroy	The chart is going to be destroyed.
ReadFile	A file has been read.
ReadTemplate	A template has been read.
ChangeColor	The user changed a color
ChangePattern	The user changed a pattern
ChangeFont	The user changed a font
ChangePalette	The user changed a color in the PaletteBar
ChangePattPal	The user changed a pattern in the PatternEditor
Menu	The user select a programmer defined menu item.

* means that by handling this event, the programmer can alter the default processing.

6.2 How to handle these events

To handle any of these events you must trap them (that means telling the environment that your program wants to be notified), this action depends on the development tool you're using i.e. in MS Visual Basic you double click on the object and select the desired event.

LButtonDblClk: **RButtonDown:**	
Standard Processing:	Shows a menu,dialog,balloon or none of the above. This can be controlled through the **DblClk** (**RigClk**) method(s).
Handling:	If the program returns a not NULL value the library does not show anything.
Parameters:	x, y, nSerie and nPoint
Related Properties:	If the setting is a dialog or balloon the program can change the text using the **HText** property.

ChangeString:	
Standard Processing:	Accept the user's changes.
Handling:	If the program returns a not NULL value the library does not change anything.

Parameters: nType = **CCS_LEGEND** or **CCS_SERLEGEND**
nIndex

Related Properties: The program can obtain/change the new value using the **Edit** Property (that returns a handle to an edit control) and the standard windows functions: SetWindowText and GetWindowText.

ChangeValue:

Standard Processing: Accept the user's changes.

Handling: If the program returns a not NULL value the library does not change anything.

Parameters: dValue, nSerie and nPoint.

Related Properties: The program can obtain/change the new value using the **Edit** Property (that returns a handle to a edit control) and the standard windows functions: SetWindowText and GetWindowText.

ChangeColor:

Parameters: nType = **CCC_SERIE**, **CCC_SERIEBK, CCC_ONE, CCC_ONEBK, CCC_BARHORZ, CCC_BKGND, CCC_2DBK,** or **CCC_3DBK.**
nIndex.

ChangePattern:

Parameters: nType = **CCP_SERIE** or **CCP_ONE.**
nIndex.

ChangeFont:
ChangePalette:
ChangePattPal:

Parameters: nIndex.

Menu:

Parameters: wParam, nPoint and nSerie

Delphi: Advanced techniques

This chapter presents some interesting techniques you can use in your charts, assuming you are familiar with CHART FX and some windows programming techniques and covers the following topics:

7.1 Setting and handling a menu for the right click of the mouse.
7.2 Modifying the Y legend.
7.3 Setting the chart so you can always read the X legend.
7.4 Saving user preferences in a template.

7.1 Setting and handling a menu for the right click of the mouse.

If you need to provide a menu for the right click of the mouse you have to keepo four things in mind:

a. How to tell CHART FX to use a menu

```
Chart1.RigClk(CHART_MENUCLK, CLong(hMenu));
```

b. How to handle messages generated by the menu.

CHART FX will send you a Menu Event.

c. When to create and destroy the menu:

Note that you can not destroy the menu once you use **RigClk (DblClk) method** because CHART FX use the handle you provided, so we recommend that you create the menu when your application start and destroy the menu when your application finish.
If the chart is a child window you can create the menu in the Load event of the parent window and destroy it in the Unload event.

d. How to create and destroy the menu:

CHART FX will use the TrackPopupMenu function so your menu will have to be a Popup menu, the windows functions that you need are CreatePopupMenu or GetSubMenu and DestroyMenu.

7.2 Modifying the Y Legend

If the values you are charting are "enumerated" (what matters is not the numerical value but the meaning of the value) i.e.:

The calification in a test can be interpreted as in the following table:

0 - 20	Really Bad
20 - 40	Bad
40 - 60	Regular
60 - 80	Good
80 - 100	Excellent

CHART FX offers an easy way to show that "meanings" of the Y values and that is what we called Y Legends. Let's show how can you chart the example shown above.

```
{Set legends}
Chart1.YLeg[0] := 'Really Bad';
```

```
Chart1.YLeg[1] := 'Bad';
Chart1.YLeg[2] := 'Regular';
Chart1.YLeg[3] := 'Good';
Chart1.YLeg[4] := 'Excellent';

{Tell CHART FX how use the legends (GAP)}
Chart1.Adm[CSA_YLEGGAP] := 20;
```

7.3 Setting the chart so you can always read X Legends

If the X Legends are large you have two ways of doing this:

a. Use Key Legends:

Key Legends are legends that "must" be shorter than normal legends and tell the user what he is charting, when the user double click on the values, he will see the normal legends.

In order to use key legends you need to set the **KeyLeg** proerty in the same way (and maybe loop) you set the **Legend** Property**.**

```
Chart1.KeyLeg[i] := 'MyKey';
```

*b. Use **FixedGap** Property*

This code will fix the minimum size (pixels) that a marker (bar, point, etc) will occupy so the the legend always fit in that space.

```
{60 pixels}
Chart1.FixedGap := 60;
```

7.4 Saving user preferences in a template

If your application wants the chart to "remember" the last used configuration (colors, patterns, 3D view , etc.) you can use the **ReadTemplate** and **WriteTemplate** properties in conjuction with the **Destroy** event in the following manner:

```
{Initialization time (may be Form1.Load)}
Chart1.ReadTemplate('USERLAST.CHT');

{Closing time (Destroy Event or Form1.Unload)}
Chart1.WriteTemplate('USERLAST.CHT');
```

Related Properties:

RigClk, YLeg, KeyLeg, FixedGap, ReadTemplate, WriteTemplate.