Welcome to First Impression

Welcome to the **Borland Edition of First Impression** from Visual Components. Visual Components develops and markets a full range of tools for the component-based developer. Our product portfolio includes best of class tools for data analysis, charting, rich text and spell checking. We offer a royalty-free runtime license for all of our OCX products.

The **Professional Edition of First Impression** supports features not found in this version of First Impression. Attempting to use these features through the user interface, or by using OCX properties and methods will result in an error.

For additional information regarding the **Professional Edition of First Impression**, including Upgrade information, contact Visual Component Sales at 800-884-8665.

Other Ways of Contacting Visual Components:

- **By telephone.** You can contact our sales staff at (800) 884-8665 on weekdays between 8:30 a.m. and 5:30 p.m., central time.
- **By FAX.** You can contact us by FAX at (913)599-6597.
- **On the Internet.** Contact us at:

  World Wide Web - http://www.visualcomp.com

  Electronic Mail - sales@visualcomp.com
- **Via BBS.** You can contact us through our bulletin board service at (913) 599-6713.
- **Via CompuServe.** You can contact us through CompuServe - 74774,443.

  Visual Components also maintains a section in the MS Windows Components A+ Forum on CompuServe. These sections are used for peer to peer support and the distribution of example projects, maintenance releases, etc. To reach the Visual Components section, type:

  GO VISTOOLS

  When communicating with Visual Components via the CompuServe forums, include our account number with all messages. This assures that your message receives prompt attention.
- **By mail.** Address your correspondence to:

  Visual Components, Inc.

  15721 College Blvd.

  Lenexa, Kansas 66219

  or

  Visual Components Europe

  Lenexa House

  11 Eldon Way

  Paddock Wood, Kent

  England TN12 6BE

  Tel: +44 1892 834343

  Fax: +44 1892 835843

BBS: +44 1892 835579

Reaching Our Technical Support Department

You can receive support directly from Visual Components Technical Support engineers by purchasing a support plan.

- You can call the support line directly at (913) 599-6500 and pay $10 per call.
- You can call (800) 884-8665 to purchase an annual support contract for $249 per developer per year. This Gold Support plan gives you access to a number of on-line information sources as well as the following:

1. Unlimited technical support calls.
2. When calling technical support, your call is placed in a priority queue for faster service.
3. Expedited open case resolution. For calls that cannot be resolved immediately, an action plan is developed within 24 hours of the customer's inquiry, and the customer is updated every 48 hours on continuing open cases.
4. Automatic beta program enrollment.

About Help

This reference guide help presents alphabetical references of First Impression's OCX objects, methods, events, and properties. Refer to the First Impression OCX User's Guide for background information about using First Impression and task-oriented discussions of First Impression features.

Documentation Conventions

Throughout this documentation, a set of typographic conventions are used to define elements and references to First Impression items. Recognizing these conventions will assist your comprehension of this documentation.

Convention example	Description
AxisSelected, AllowSelections, Select,	Names of events, properties, and methods, are in proper case and bold font.
To install First Impression:	A series of numbered instructions are preceded by an introductory line. The introductory line begins with an arrowhead.
1.Type a:\setup.	Numbered instructions provide step-by-step directions for performing tasks. The instructions should be performed in the order they are presented.
	In numbered steps, items you are to enter are shown in Letter Gothic font.
chart plot	In general sections, italic text is used for the first occurrence of a new term.
thicknessratio	In reference sections, italic text indicates variable or argument information you must supply.
[axisid]	In reference sections, italic text surrounded by square brackets indicates optional arguments.
{TRUE|FALSE}	In reference sections, text surrounded by braces indicates you must make a choice among the items inside the braces. Choices are separated by vertical bars.
VtChart1.AllowDithering	Letter Gothic font is used for all code examples.
VCFI32.OCX	File names are presented in upper case text.
VtChart1.RowCount 'number of rows	In code examples, an apostrophe precedes a comment.

New Features in First Impression Professional Edition

- **OCX Support.** First Impression functionality is revealed through OCX objects, properties and methods.
- **Real-Time Charts**. First Impression now supports real-time charts.
- **Elevation Charts**. 2D Contour and 3D Surface charts have been added to the First Impression family of charts.
- **3D Versions of Charts**. First Impression now includes 3D XYZ and 3D Scatter charts.
- **DataGrid Editor**. Users can now directly edit and input data for their First Impression charts.
- **Chart Wizard**. First Impression now includes a chart wizard to step users through building or modifying a chart.
- **Extended Datapoint Markers**. First Impression now offers a 3D ball as a marker type.
- **Customizable Mousepointer.** You can now control the shape and behavior of the mouse pointer when is over the First Impression control.
- **Enhanced Color Management** . First Impression now provides enhanced color displays for 8-bit color monitors and monitors set to 256 color mode.

For additional information regarding the **Professional Edition of First Impression**, including Upgrade information, contact Visual Component Sales at 800-884-8665.

Adding the OCX to Your Application

The process you use to add an OCX to your application varies slightly from one development environment to another. In most cases it consists of:

- Adding the OCX control to your project.
- Selecting the control's tool from the tool bar and drawing the control on a form or in a window.

Consult your development environment documentation for specific steps to add a control to your application.

Basic Concepts

First Impression is a Custom Control (OCX). It can be accessed directly by environments that support OCX containers.

- In design mode, randomly generated data is provided to allow you to design a default format for the chart control. You can override this data by inserting specific values in the data grid.
- In run mode, data can be assigned to the data grid associated with the chart. This data is charted using the default format you created during design time. You can also allow additional formatting of the chart during run time.
- The chart format can be modified from code or from the built-in user interface.
- Charts can be saved with a form or in a separate file.
- Any formatting changes you make to a chart using the First Impression interface are only saved if you are in design mode. Any changes made through the interface in run mode are not saved with the chart.

Distributing First Impression Applications

Please read the license agreement that was shipped with this package. You are bound by the licensing agreement contained in that document.

Redistributing Files

You can use all the files accompanying this product for development of an application. You can redistribute the run time version of the software according to the terms of the license agreement.

The following list of files are important. Read the paragraph below to learn what each file is used for.

	32-bit Files
1	VCFI32.OCX
	VCFI32WZ.DLL
2	OC30.DLL
	MFCANS32.DLL
	MSVCRT20.DLL
3	KERNEL32.DLL
	USER32.DLL
	GDI32.DLL
	COMDLG32.DLL

This first group of files are the First Impression redistributable files. These files need to be in the Windows System directory, on the computers Path, or for Windows NT and Windows 95, in the directory specified by your application's Per Application Path key in the Registry Database.

The second group of files are Microsoft redistributable files necessary to for this OCX to operate. Your programming environment should have installed and registered these files. If they were not, or they were an older version, the First Impression installer updated and registered them. These files were then copied to ReDist16 and ReDist32 in the installation directory, depending on which installation options were chosen. You may run the installer again and install only these files.

The third group of files should be present on any system running windows. You should not remove or update these files. They are included only to form a complete list of files needed to use this OCX.

Note TrueType fonts must be installed on the system running First Impression in order for First Impression to operate.

Understanding Objects

First Impression's OCX programming interface is constructed from a collection of defined objects. This makes it easier for you to control your applications by manipulating these objects in procedures. This chapter introduces objects and explains how to use them to simplify the things you want to do in First Impression.

Each time you create a First Impression OCX control in your application, you create a chart object. The chart is an object that you use to graphically display data.

Since a chart is an object, it has characteristics called properties that control the appearance or behavior of the chart. **ShowLegend** is a property of the **VtChart** object that determines whether or not a legend appears on the chart. Properties also allow you to change the attributes of the chart object. For example, you can set properties to hide the chart legend or to determine the number of rows or columns in a datagrid.

You affect an object by changing its properties. When you set or return a property's value it changes the object it belongs to.

Related Topics:

Using Properties

Property Data Types

Setting Properties

Returning Property Values

Properties that take an index

Default Properties

Understanding Object Methods

Using Objects in Code

The Object Model

Object Collections

Using Multiple Objects in Code

General Purpose Objects

Using Properties

With First Impression properties, you can perform a variety of tasks, such as changing the color of a chart backdrop, or the size of the legend text. Many properties allow you to perform complex tasks with very little coding.

When you use properties in your code, you can either set, or change the value of the property, or return the property's current value. Most properties are read-write. This means you can set and return them. However, some properties are read-only. This means you can return their current value, but you cannot set the property to change the value.

When you refer to a property, you must list the object name first, followed by a period, and then the name of the property. In the following example, the **AllowSeriesSelection** property of the object called **VtChart** is set to True.

```
VtChart1.AllowSeriesSelection = True
```

Related Topics:

Understanding Objects

Property Data Types

Setting Properties

Returning Property Values

Properties that take an index

Default Properties

Understanding Object Methods

Using Objects in Code

The Object Model

Object Collections

Using Multiple Objects in Code

General Purpose Objects

Property Data Types

Before setting or returning the value of a property, you must know its data type. Generally, a property's value can be a numeric value, a character string, or a boolean (True|False) value. Some objects even contain other objects. This occurs when an object has a property whose data type is **object**. Object variables are stored as addresses that refer to any actual objects within an application. To determine the correct data type for a particular property, consult the Object API Reference.

Related Topics:

Understanding Objects

Using Properties

Setting Properties

Returning Property Values

Properties that take an index

Default Properties

Understanding Object Methods

Using Objects in Code

The Object Model

Object Collections

Using Multiple Objects in Code

General Purpose Objects

Setting Properties

When you set a property, you assign it a new value. To accomplish this you use the equal sign to set the property equal to an expression that describes the new value. In the following example, the **ColumnCount** property is used to set the number of columns in the datagrid to 5.

```
VtChart1.ColumnCount = 5
```

The following example shows the code required to read an existing chart file from disk using the **ReadFromFile** property**.**

```
VtChart1.ReadFromFile = "c:\charts\examples\elevation.vtc"
```

Related Topics:

Understanding Objects

Using Properties

Property Data Types

Returning Property Values

Properties that take an index

Default Properties

Understanding Object Methods

Using Objects in Code

The Object Model

Object Collections

Using Multiple Objects in Code

General Purpose Objects

Returning Property Values

Return a property value to determine the condition of an object before your procedure performs additional actions.

In the following example, the code returns the status of the datapoint lines on a chart. By returning the true status, the user can find out if the lines are set to appear on the chart before performing additional actions.

```
if VtChart1.ShowLegend = True then
With VtChart1.Legend.VtFont
.Name = “Arial”
.Size = 14
End With
End If
```

Related Topics:

Understanding Objects

Using Properties

Property Data Types

Setting Properties

Properties that take an index

Default Properties

Understanding Object Methods

Using Objects in Code

The Object Model

Object Collections

Using Multiple Objects in Code

General Purpose Objects

Properties that Take an Index

Some properties take an index to identify the specific entity for which they set or return a value. To set a property for a single object in an object collection, you must supply an index to indicate which item in the collection you wish to action. For instance, to set the sides property for a Series Collection use the following code:

```
‘Set all the bars in a bar chart to 7 sides
For Each seriesItem in VtChart1.Plot.SeriesCollection
seriesItem.Bar.sides = 7
Next
```

The same thing could be accomplished through the following code:

```
Dim sc As Object
Dim ii as Integer
Set sc = VtChart1.Plot.SeriesCollection
For ii = 1 to sc.Count
sc(ii).series.Bar.sides = 7
Next
```

Related Topics:

Understanding Objects

Using Properties

Property Data Types

Setting Properties

Returning Property Values

Default Properties

Understanding Object Methods

Using Objects in Code

The Object Model

Object Collections

Using Multiple Objects in Code

General Purpose Objects

Default Properties

Default properties are used if no property is specified in code. The **Text** property is the default property for the **Title** object, and is assumed to be the current property, if no property is specifically identified in code. The following example shows how this might work:

```
‘ Set the title, font size and orientation of the chart title text.
Dim chtitle as object
Set chtitle = VtChart1.Title
chtitle = “Precipitation”
chtitle.TextLayout.Orientation = 1
Set titlefont = chtitle.VtFont
titlefont.Size = 14
```

In this case, two variables were set and used within the code example. *chtitle* was declared as a variable for VtChart1.Title, and *titlefont* was declared as a variable for chtitle.VtFont. The use of variables greatly saves time, and speeds the code-writing process.

In this example, the Text property was not specifically indicated, but because it is the **default** property for the object, changes were made to the chart title text.

Related Topics:

Understanding Objects

Using Properties

Property Data Types

Setting Properties

Returning Property Values

Properties that take an index

Understanding Object Methods

Using Objects in Code

The Object Model

Object Collections

Using Multiple Objects in Code

General Purpose Objects

Understanding Object Methods

Objects have methods which perform actions. **PrintChart** is a method of the **VtChart** object that can be used to send a copy of the chart to the printer.

Using Methods

There are two types of methods, those that take arguments, and those that do not. Methods that take no arguments initiate an action and return no value. For example, you can use the **ActivateSelectionDialog** method to display the dialog box associated with the currently selected chart part**.**

```
VtChart1.ActivateSelectionDialog
```

If a method does take arguments, you must be aware of whether the method returns a value. In Visual Basic, if the method does not return a value, or you don't wish to save the returned value, the method arguments appear without parentheses.

```
VtChart1.InsertColumns 4,2
```

If you do wish to save the value returned by a method, then the arguments appear with parentheses around them. The keyword *Call* can be used.

```
return value = VtChart1.InsertColumns (4,2)
```

There are times when methods change the value of a property. For example, when you use the **DeleteColumns** method to delete columns from the datagrid, the value of the **ColumnCount** property is changed to reflect the number of columns.

Related Topics:

Understanding Objects

Using Properties

Property Data Types

Setting Properties

Returning Property Values

Properties that take an index

Default Properties

Using Objects in Code

The Object Model

Object Collections

Using Multiple Objects in Code

General Purpose Objects

Using Objects in Code

When writing your First Impression application, determine which objects you need to accomplish a task, and then use those objects. Typically, the code you write accomplishes one or more of the following:

- Set a value of one of the object's properties in order to change to state of the object.
- Return the value of one of the object's properties in order to determine the state of the object.
- Perform a task through the object by using one of its methods.

The following two examples demonstrate how you can determine the state of an object or perform a task.

The **ColumnCount** property might be used in code to inform you of the number of datagrid columns.

```
'Return the number of columns in the chart datagrid.
Dim count as Integer
count = VtChart1.ColumnCount
msgbox count
```

The **EditPaste** method might be used in code to paste the chart to the clipboard.

```
'Paste the chart to the clipboard in WMF format.
VtChart1.EditPaste
```

Related Topics:

Understanding Objects

Using Properties

Property Data Types

Setting Properties

Returning Property Values

Properties that take an index

Default Properties

Understanding Object Methods

The Object Model

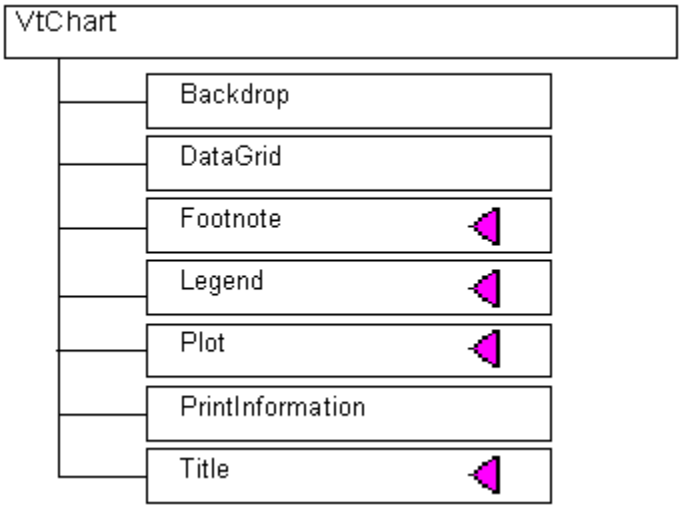
Object Collections

Using Multiple Objects in Code

General Purpose Objects

The Object Model

First Impression has one main object -- the VtChart object, which is made up of seven other objects. The following diagram illustrates the VtChart object and its subobjects.



Before you can use the VtChart object effectively, it is important to understand its relationships to other objects. You can navigate through the entire object hierarchy by clicking on a **pink arrow**. The pink arrow indicates that you can expand the object to view its subobjects. By clicking on a **pink arrow**, you can explore the object infrastructure and learn more about individual objects and object collections. To return to the parent object, click on the **blue arrow**.

VtChart Code Sample

Related Topics:

Understanding Objects
Property Data Types
Returning Property Values
Default Properties
Using Objects in Code
Using Multiple Objects in Code
Using Properties
Setting Properties
Properties that take an index
Understanding Object Methods
Object Collections
General Purpose Objects

VtChart Code Sample

```
VtChart1.Backdrop
VtChart1.DataGrid
VtChart1.Footnote
VtChart1.Legend
VtChart1.Plot
VtChart1.PrintInformation
VtChart1.Title
```

Object Collections

In First Impression, a collection is a set of related objects. For example, the **Series Collection** contains all the series in a given chart. Each object in a collection is an element of that collection.

Because collections are objects, themselves, they have their own properties and methods. You can use these properties and methods to control individual elements in the collection, as well as all objects in the collection.

Collection objects are accessed using the plural name of the object. For example, the **DataPoint** object belongs to the **DataPoints Collection**. The exception to this rule is **SeriesCollection.** Since Series is already plural, the compound name **SeriesCollection** refers to a collection of Series objects.

To access a single object in a collection, you must supply an index. For instance, to set the number of bar *sides* to 12 for the third **series** in a **collection,** use the following code:

```
VtChart1.Plot.SeriesCollection.Item(3).Bar.sides = 12
```

The *Item* index indicates which series to set the property for.

An object's position in the collection can change whenever a change occurs in the collection. Therefore, the position of any specific object in the collection is unpredictable. This unpredictability distinquishes a collection from an array. The following code example adds three LightSource items to the LightSources Collection and then deletes all of the LightSources.

```
Dim xx As Object
Set xx = VtChart1.Plot.light.LightSources

xx.Add 0, 0, 1, 0.21
xx.Add -3, 0, 2, 0.22
xx.Add 3, 0, 2, 0.23

For n = 1 To xx.Count
xx.Remove (1)
Next
```

Related Topics:

Understanding Objects

Using Properties

Property Data Types

Setting Properties

Returning Property Values

Properties that take an index

Default Properties

Understanding Object Methods

Using Objects in Code

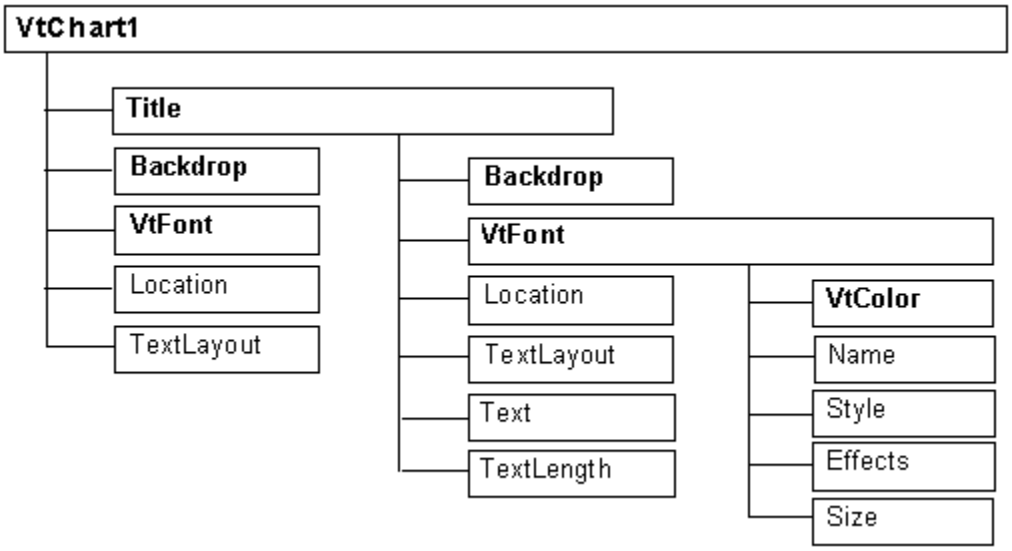
The Object Model

Using Multiple Objects in Code

General Purpose Objects

Using Multiple Objects in Code

You can fully appreciate First Impression's object model when you begin to write code. The hierarchical structure of the model makes it necessary to locate the specific object you wish to change by mapping through its parent objects.



For Example, the **VtFont** object is a subobject of the **Title** object which is a subobject of the **VtChart** object. If you want to change the font on the chart title, you actually specify the change to the font object's **Name** property. Without simplifying the task, you could specify all 3 objects and the Name property as follows:

```
'Set the chart title font to Helvetica.
Set VtChart1.Title.VtFont.Name = "Helvetica"
```

By setting the name property for the VtFont object, you are ultimately changing the original chart object.

Saving Time with Variables

You can save yourself a great deal of typing by declaring objects in your code. The following example demonstrates how you would change the chart title font. You simplify the task by declaring a variable, *TitleFont* and then assign that variable to the object.

```
'Set the chart title font to Helvetica.
Dim TitleFont as object
Set TitleFont = VtChart1.Title.VtFont
TitleFont.Name = "Helvetica"
TitleFont.Size = "14"
```

To further simplify the task, the following code uses the WITH statement:

```
'Set the chart title font and size to Helvetica 14 point using the WITH
statement
With VtChart1.Title.VtFont
.Name = "Helvetica"
.Size = 14
End With
```

The best way to take advantage of time-saving variables is to set multiple properties for the same object in a single statement. This is implied in the above example by setting the font name and size. Theoretically, you could use the variable *TitleFont*, and set all the **VtFont** properties for the chart title in one fell swoop.

Related Topics:

Understanding Objects

Using Properties

Property Data Types

Setting Properties

Returning Property Values

Properties that take an index

Default Properties

Understanding Object Methods

Using Objects in Code

The Object Model

Object Collections

General Purpose Objects

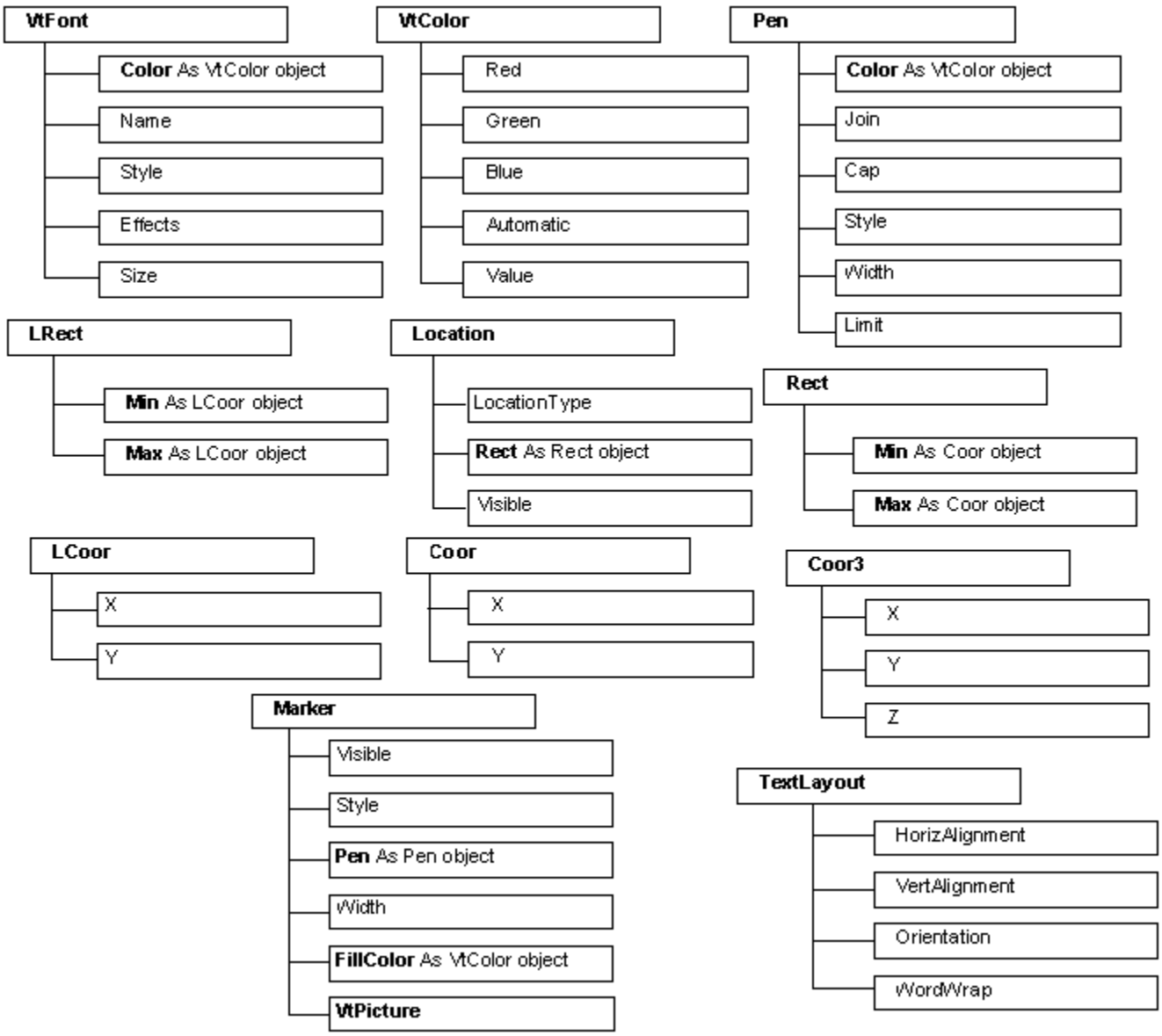
General Purpose Objects

Some objects are used for general purposes, and can be applied to many different objects and are not limited to a single relationship.

The **VtColor object** is a good example of a general purpose object. It exists as a subobject to many other objects and is not limited to a single relationship within the object model. **FrameColor** and **FillColor** are two properties that refer to the **VtColor** object. These two properties have an *object* data type and therefore refer to the the **VtColor** object itself.

For a better understanding of this concept, refer to the Backdrop object diagram.

Below is a diagram of other general purpose objects:



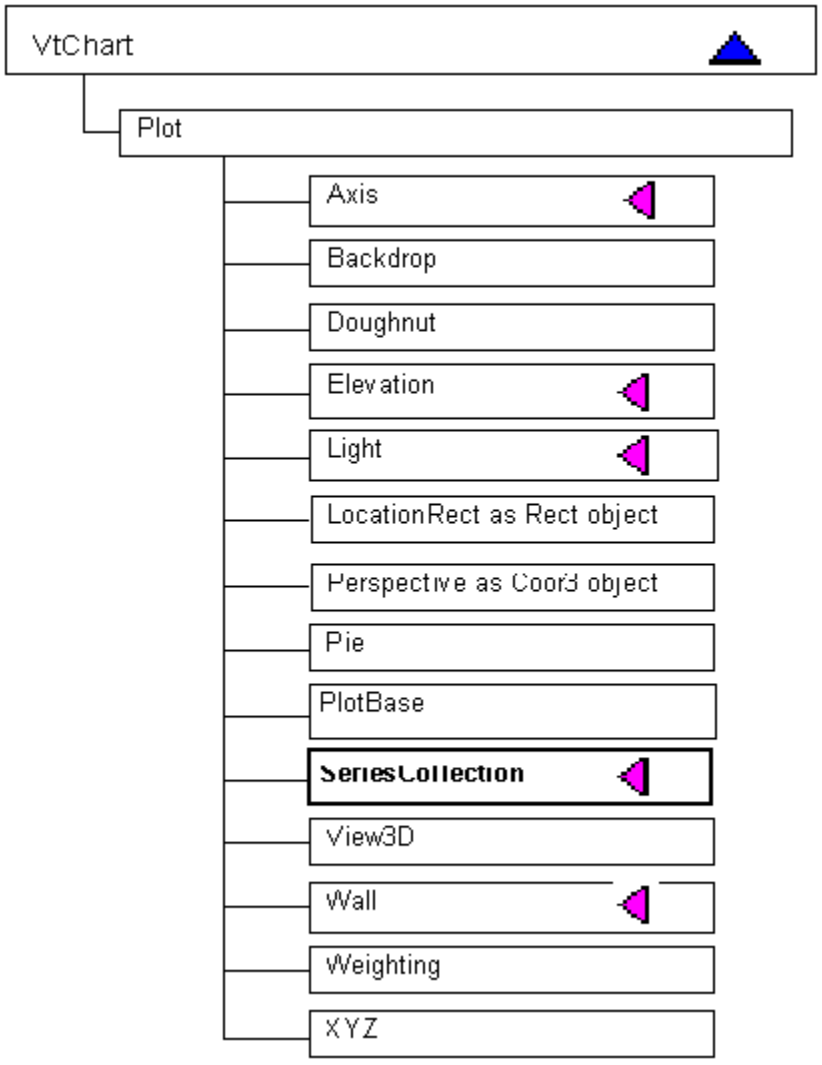
Related Topics:

Understanding Objects

Using Properties

Property Data Types

The Plot Object

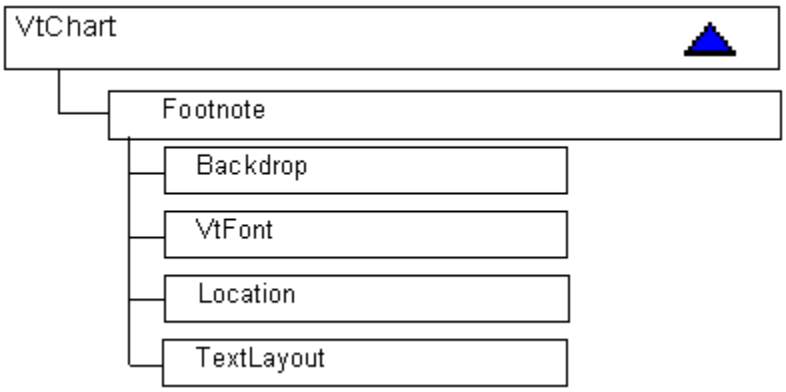


Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.
Plot Object Code Sample

Plot Object Code Sample

```
VtChart1.Plot.Axis
VtChart1.Plot.Backdrop
VtChart1.Plot.Doughnut
VtChart1.Plot.Elevation
VtChart1.Plot.Light
VtChart1.Plot.LocationRect
VtChart1.Plot.Perspective
VtChart1.Plot.Pie
VtChart1.Plot.PlotBase
VtChart1.Plot.SeriesCollection
VtChart1.Plot.View3D
VtChart1.Plot.Wall
VtChart1.Plot.Weighting
VtChart1.Plot.XYZ
```

The Footnote Object

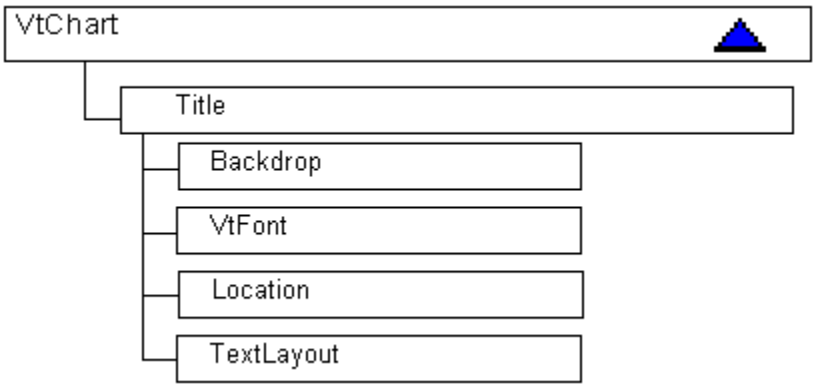


Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.
Footnote Code Sample

Footnote Code Sample

```
VtChart1.Footnote.Backdrop
VtChart1.Footnote.VtFont
VtChart1.Footnote.Location
VtChart1.Footnote.TextLayout
```

The Title Object

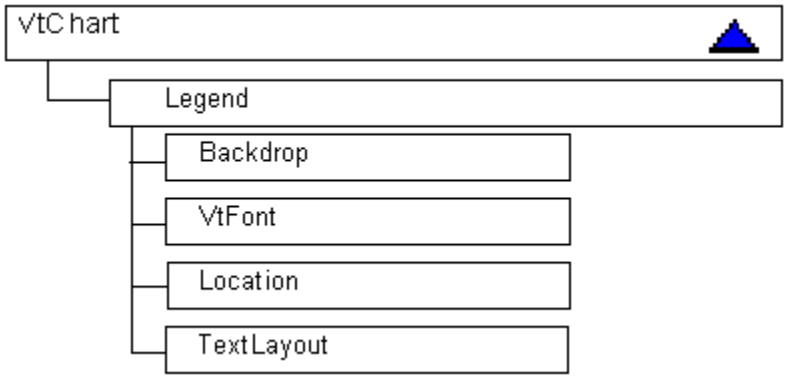


Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.
Title Code Sample

Title Code Sample

```
VtChart1.Title.Backdrop
VtChart1.Title.VtFont
VtChart1.Title.Location
VtChart1.Title.TextLayout
```

The Legend Object

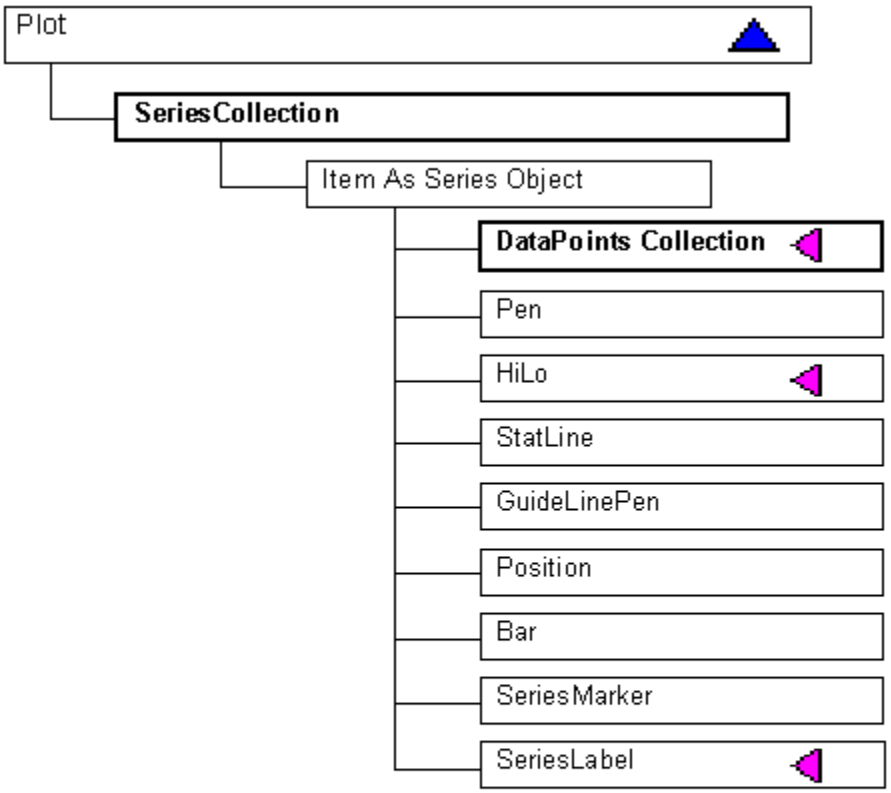


Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.
Legend Code Sample

Legend Code Sample

```
VtChart1.Legend.Backdrop
VtChart1.Legend.VtFont
VtChart1.Legend.Location
VtChart1.Legend.TextLayout
```

The SeriesCollection Collection



The SeriesCollection is a collection of Series objects.

Item refers to an individual Series within the collection.

Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.

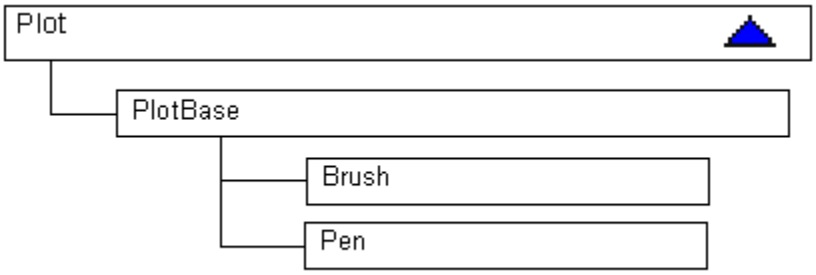
SeriesCollection Code Sample

SeriesCollection Code Sample

```
VtChart1.Plot.SeriesCollection.Item(2).DataPoints
VtChart1.Plot.SeriesCollection.Item(2).Pen
VtChart1.Plot.SeriesCollection.Item(2).HiLo
VtChart1.Plot.SeriesCollection.Item(2).Statline
VtChart1.Plot.SeriesCollection.Item(2).GuideLinePen
VtChart1.Plot.SeriesCollection.Item(2).DataPointLabel
VtChart1.Plot.SeriesCollection.Item(2).Position
VtChart1.Plot.SeriesCollection.Item(2).Bar
VtChart1.Plot.SeriesCollection.Item(2).SeriesMarker
VtChart1.Plot.SeriesCollection.Item(2).SeriesLabel
```

Note: (2) refers to an index representing the second series in the collection. You should substitute an appropriate index in your code to access the correct series.

The PlotBase Object

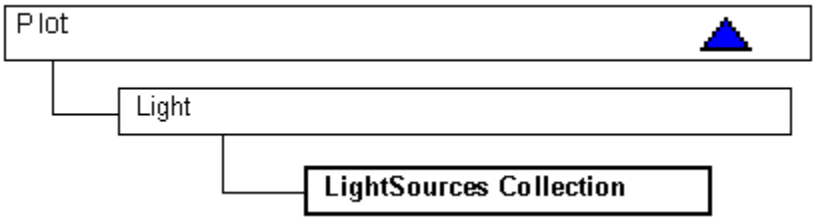


Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.
PlotBase Code Sample

PlotBase Code Sample

```
VtChart1.Plot.PlotBase.Brush
VtChart1.Plot.PlotBase.Pen
```

The Light Object

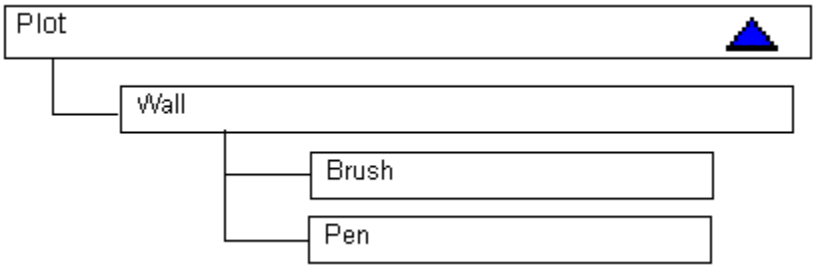


Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.
Light Code Sample

Light Code Sample

```
VtChart1.Plot.Light.LightSources
```

The Wall Object

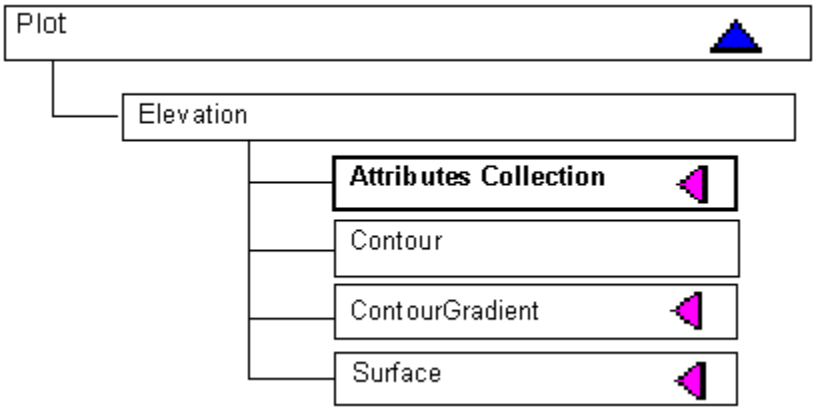


Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.
Wall Code Sample

Wall Code Sample

```
VtChart1.Plot.Wall.Brush
VtChart1.Plot.Wall.Pen
```

The Elevation Object

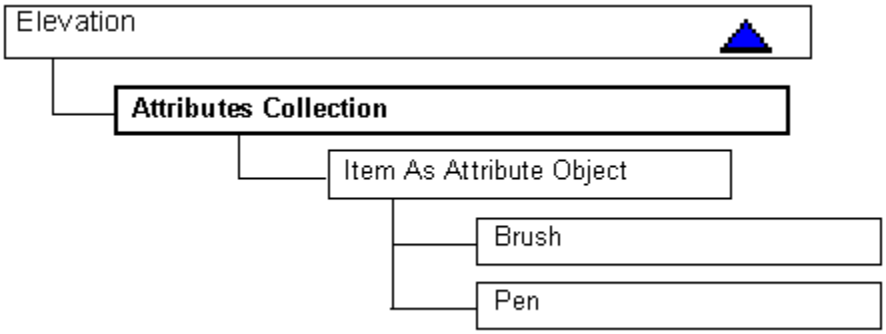


Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.
Elevation Code Sample

Elevation Code Sample

```
VtChart1.Plot.Elevation.Attributes
VtChart1.Plot.Elevation.Contour
VtChart1.Plot.Elevation.ContourGradient
VtChart1.Plot.Elevation.Surface
```

The Attributes Collection



The Attributes Collection is a collection of Attribute objects.

Item refers to an individual Attribute within the collection.

Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.

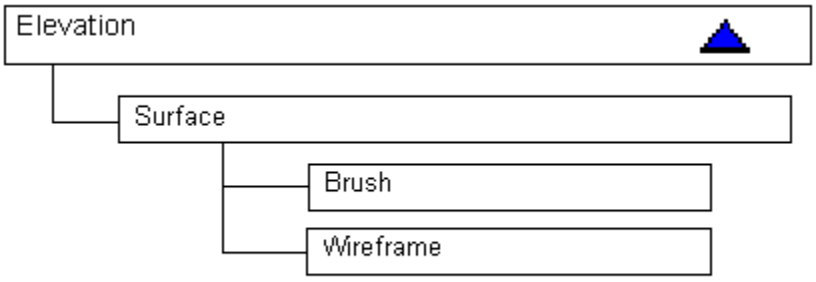
Attributes Collection Code Sample

Attributes Collection Code Sample

```
VtChart1.Plot.Elevation.Attributes.Item(2).Brush
VtChart1.Plot.Elevation.Attributes.Item(2).Pen
```

Note: (2) refers to an index representing the second attribute in the collection. You should substitute an appropriate index number in your code to access the correct attribute.

The Surface Object

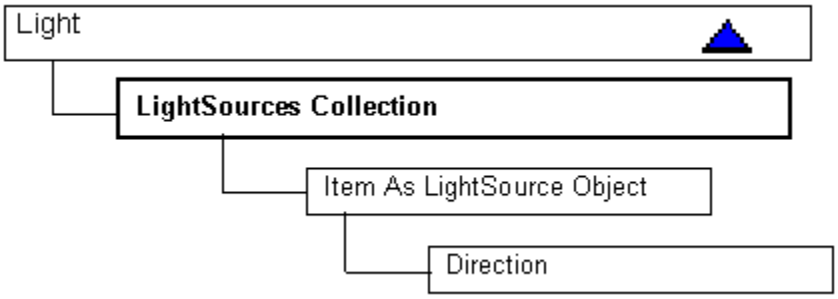


Surface Code Sample

Surface Code Sample

```
VtChart1.Plot.Elevation.Surface.Brush
VtChart1.Plot.Elevation.Surface.Wireframe
```

The LightSources Collection



The LightSources Collection is a collection of LightSource objects.

Item refers to an individual LightSource within the collection.

Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.

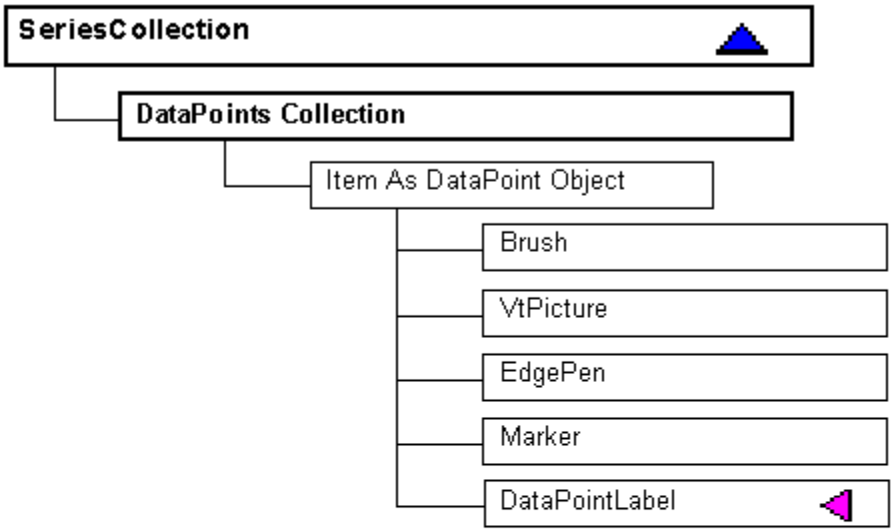
LightSources Code Sample

LightSources Code Sample

```
VtChart1.Plot.Light.LightSources.Item(2).Direction
```

Note: (2) refers to an index representing the second lightsource in the collection. You should substitute an appropriate index number in your code to access the correct lightsource.

The DataPoints Collection



The DataPoints Collection is a collection of DataPoint objects.

Item refers to an individual DataPoint within the collection.

Click on the blue arrow to view parent objects.

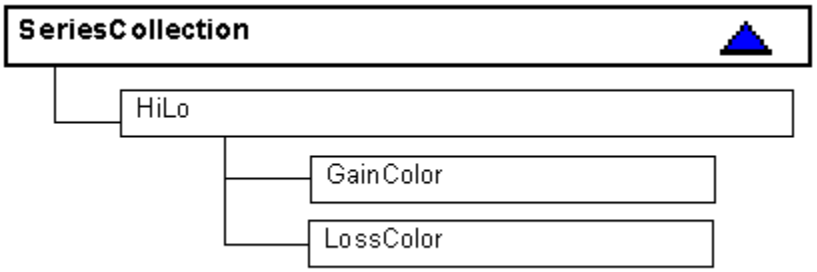
DataPoints Code Sample

DataPoints Code Sample

```
VtChart1.Plot.SeriesCollection.Item(2).DataPoints.Item(2).Brush
VtChart1.Plot.SeriesCollection.Item(2).DataPoints.Item(2).VtPicture
VtChart1.Plot.SeriesCollection.Item(2).DataPoints.Item(2).EdgePen
VtChart1.Plot.SeriesCollection.Item(2).DataPoints.Item(2).Marker
```

Note: (2) refers to an index representing the second series or data point in the collection. You should substitute an appropriate index number in your code to access the correct series or data point.

The HiLo Object



GainColor and **LossColor** are **VtColor** objects.

Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.

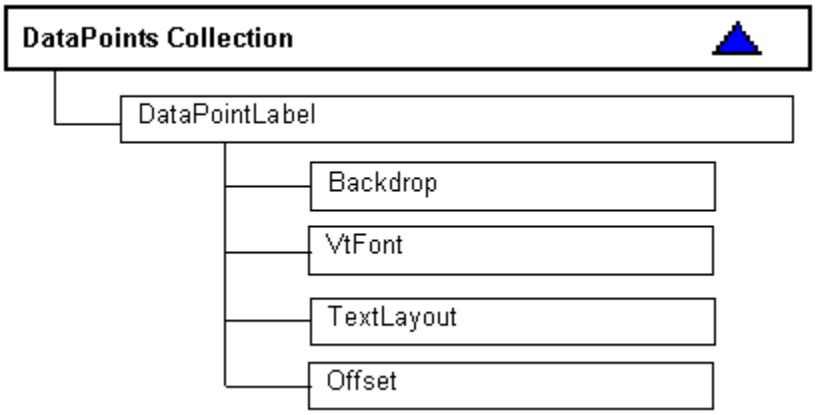
HiLo Code Sample

HiLo Code Sample

```
VtChart1.Plot.SeriesCollection.Item(2).HiLo.GainColor
VtChart1.Plot.SeriesCollection.Item(2).HiLo.LossColor
```

Note: (2) refers to an index representing the second series in the collection. You should substitute an appropriate index number in your code to access the correct series.

The DataPointLabel Object



Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.

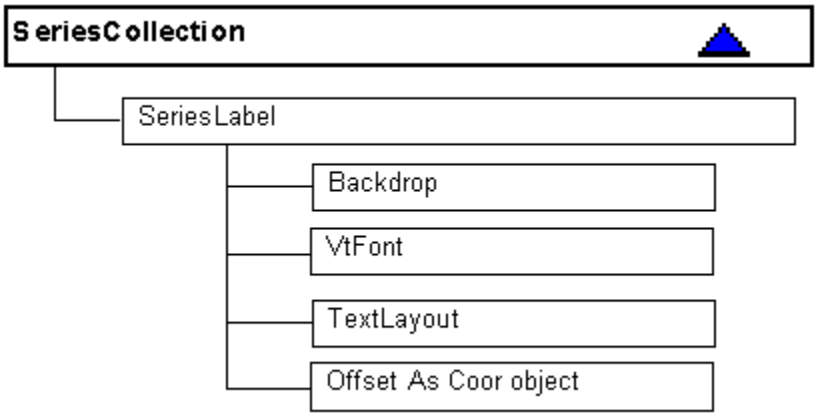
DataPointLabel Code Sample

DataPointLabel Code Sample

```
VtChart1.Plot.SeriesCollection.Item(2).DataPointLabel.Backdrop
VtChart1.Plot.SeriesCollection.Item(2).DataPointLabel.VtFont
VtChart1.Plot.SeriesCollection.Item(2).DataPointLabel.TextLayout
VtChart1.Plot.SeriesCollection.Item(2).DataPointLabel.Offset
```

Note: (2) refers to an index representing the second series in the collection. You should substitute an appropriate index number in your code to access the correct series.

The SeriesLabel Object



Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.

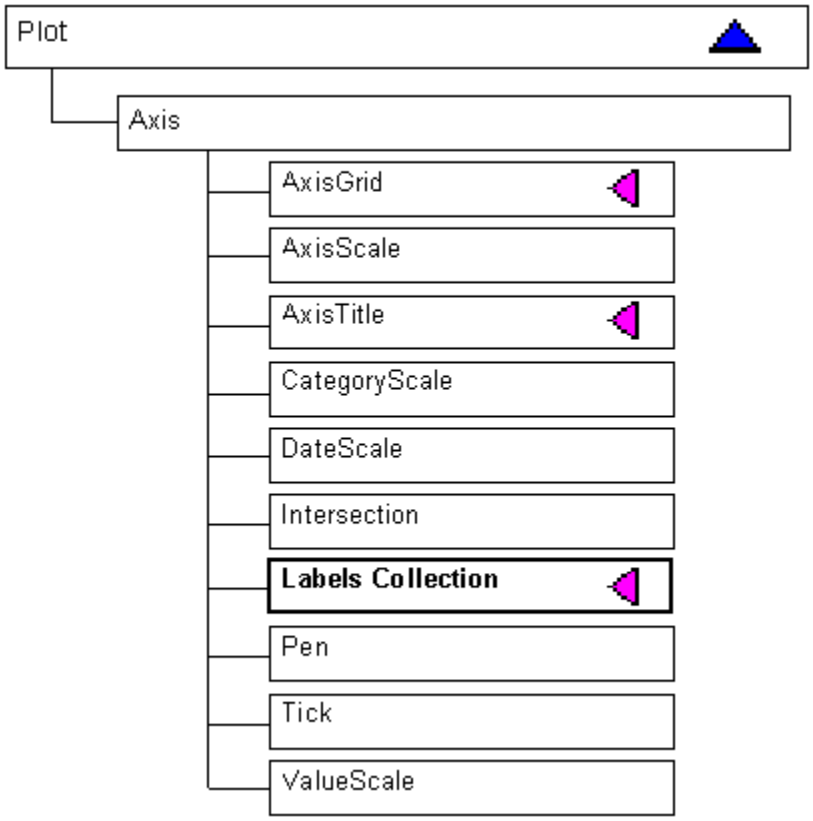
SeriesLabel Code Sample

SeriesLabel Code Sample

```
VtChart1.Plot.SeriesCollection.Item(2).SeriesLabel.Backdrop
VtChart1.Plot.SeriesCollection.Item(2).SeriesLabel.VtFont
VtChart1.Plot.SeriesCollection.Item(2).SeriesLabel.TextLayout
VtChart1.Plot.SeriesCollection.Item(2).SeriesLabel.Offset
```

Note: (2) refers to an index representing the second series in the collection. You should substitute an appropriate index number in your code to access the correct series.

The Axis Object



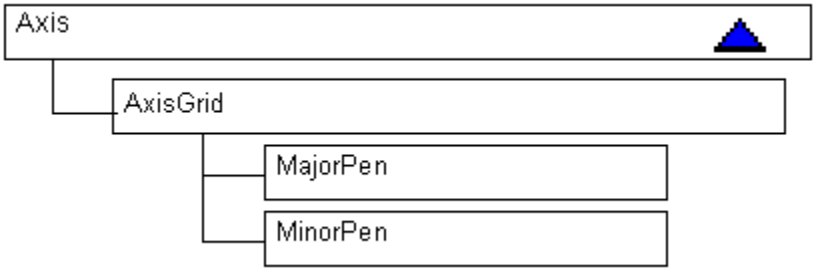
Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.

Axis Code Sample

Axis Code Sample

```
VtChart1.Plot.Axis.AxisGrid
VtChart1.Plot.Axis.AxisScale
VtChart1.Plot.Axis.AxisTitle
VtChart1.Plot.Axis.CategoryScale
VtChart1.Plot.Axis.DateScale
VtChart1.Plot.Axis.Intersection
VtChart1.Plot.Axis.Labels
VtChart1.Plot.Axis.Pen
VtChart1.Plot.Axis.Tick
VtChart1.Plot.Axis.ValueScale
```

The AxisGrid Object



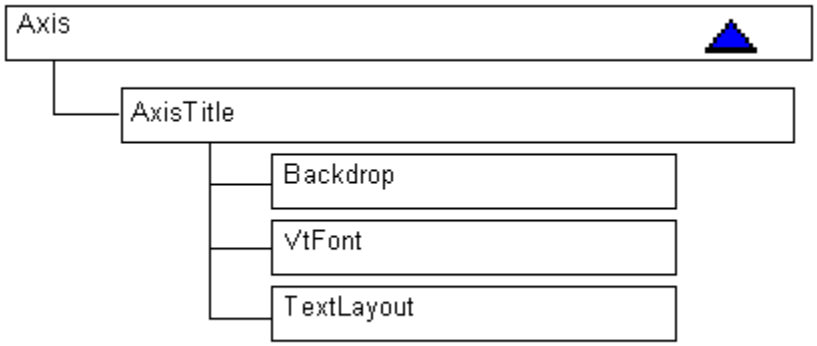
Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.

AxisGrid Code Sample

AxisGrid Code Sample

```
VtChart1.Plot.Axis.AxisGrid.MajorPen
VtChart1.Plot.Axis.AxisGrid.MinorPen
```

The AxisTitle Object



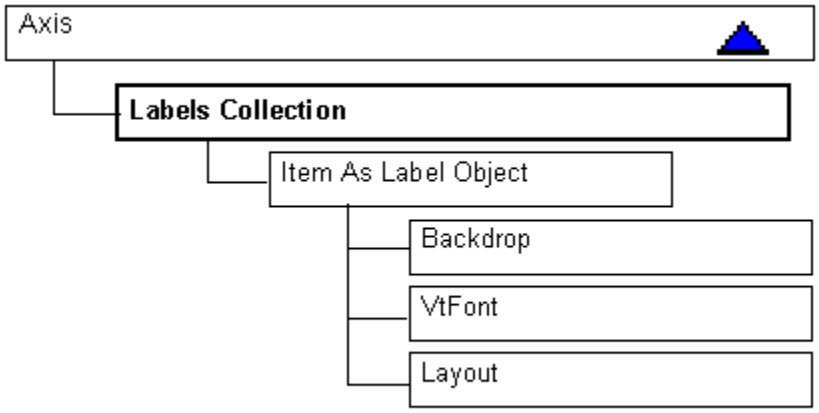
Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.

AxisTitle Code Sample

AxisTitle Code Sample

```
VtChart1.Plot.Axis.AxisTitle.Backdrop
VtChart1.Plot.Axis.AxisTitle.VtFont
VtChart1.Plot.Axis.AxisTitle.TextLayout
```

The Labels Collection



The Labels Collection is a collection of Label objects.

Item refers to an individual Label within the collection.

Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.

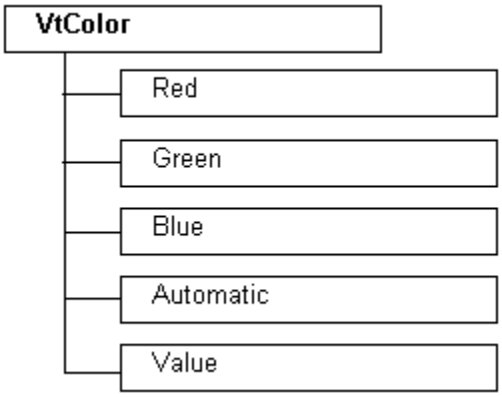
Labels Code Sample

Labels Code Sample

```
VtChart1.Plot.Axis.Labels.Item(2).Backdrop
VtChart1.Plot.Axis.Labels.Item(2) VtFont
VtChart1.Plot.Axis.Labels.Item(2).Layout
```

Note: (2) refers to an index representing the second label in the collection. You should substitute an appropriate index number in your code to access the correct label.

The VtColor Object



VtColor Code Sample

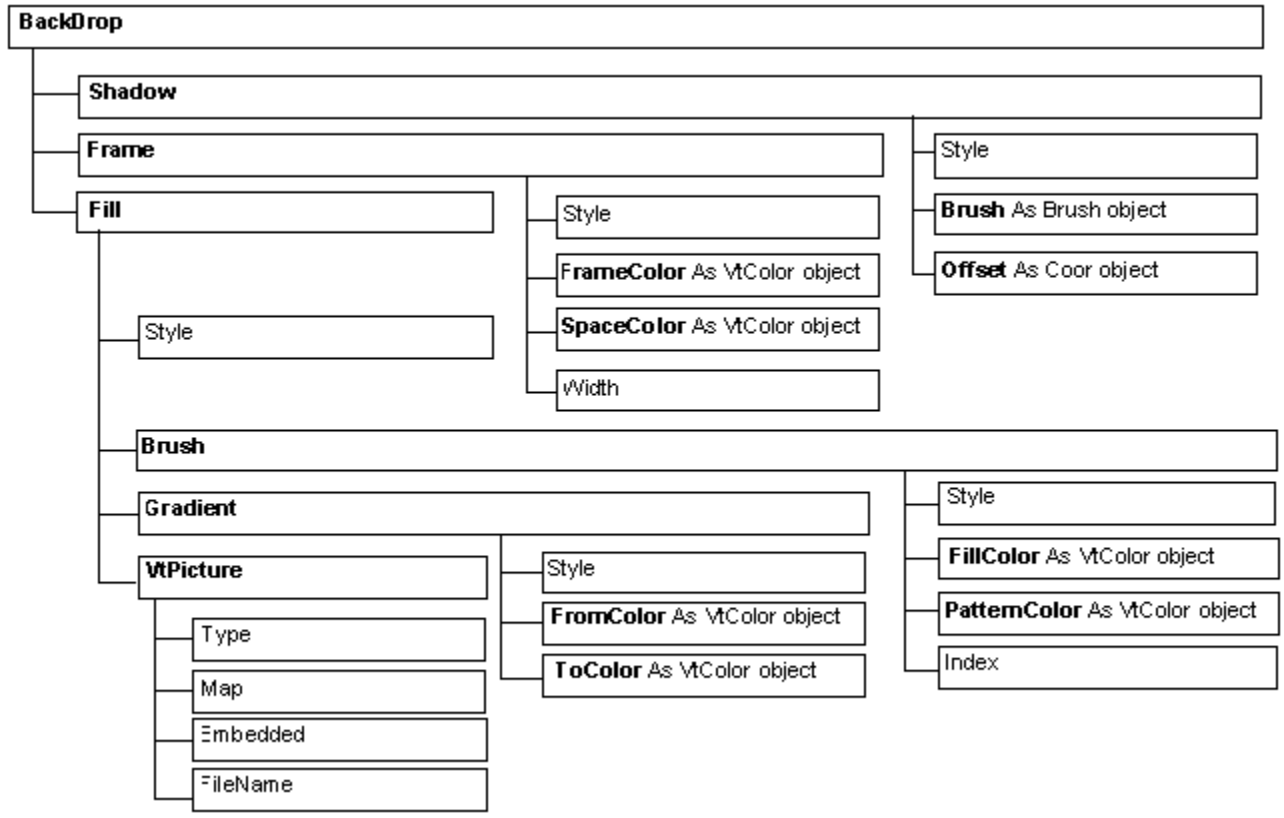
VtColor Code Sample

```
VtChart1.Plot.PlotBase.Brush.Pen.Color
```

Note: Color is a Pen property of type VtColor object. VtColor is a general purpose object, and therefore, applies to many other objects. The Pen object is just one example of an object that uses the VtColor object.

The Backdrop Object

The Backdrop object contains several examples of VtColor objects.



Backdrop Code Sample

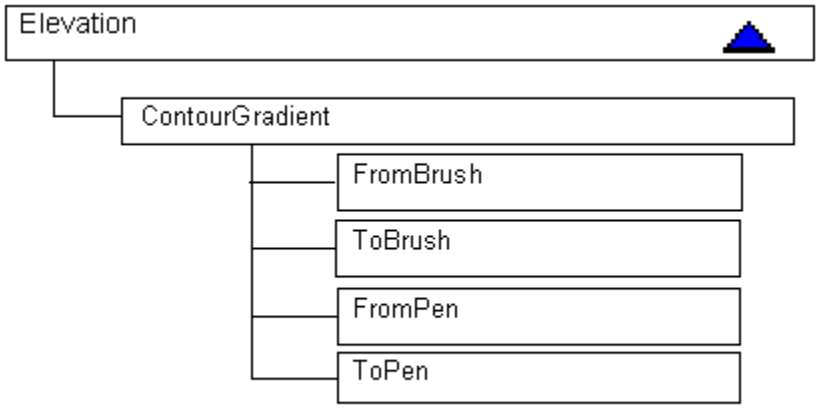
The following example demonstrates how you would change the frame color on a chart backdrop. By declaring **BackdropFrameColor** as an object, you no longer have to type **VtChart1.Backdrop.Frame.FrameColor** every time you reference the frame color.

```
'Set the chart backdrop frame color to Red.
Dim BackdropFrameColor as object
Set BackdropFrameColor = VtChart1.Backdrop.Frame.FrameColor
BackdropFrameColor = RGB (255,0,0)
```

Backdrop Code Sample

```
VtChart1.Backdrop.Shadow
VtChart1.Backdrop.Frame
VtChart1.Backdrop.Fill
```

The ContourGradient Object



FromBrush, ToBrush, FromPen and **ToPen** are **VtColor** objects.
Click on the pink arrow to view subobjects. Click on the blue arrow to view parent objects.
ContourGradient Code Sample

ContourGradient Code Sample

```
VtChart1.Plot.Elevation.ContourGradient.FromBrush
VtChart1.Plot.Elevation.ContourGradient.ToBrush
VtChart1.Plot.Elevation.ContourGradient.FromPen
VtChart1.Plot.Elevation.ContourGradient.ToPen
```

Chart Data Grids

A data grid is associated with each chart. This data grid holds the values that are charted, as well as the text that is used to display series labels and category labels on the chart. You can think of a chart data grid as an array or table containing the chart information.

	Model 421	Model 532	Model 629
East	4	40	62
Central	22	18	62
West	40	10	39

Related Topics:

Simple Data Grids
More Complex Data Grids
Identifying Data Grid Elements
Multiple Level Labels
Creating the Data Grid

Manipulating the Data Grid Using Properties

Deleting Data Grid Elements

Inserting Data Grid Elements

Changing Data Grid Values or Text

Linking to a Formula One Spreadsheet

Data Grid Requirements of Various Chart Types

Simple Data Grids

Most data can be simply arranged in series of rows and columns. In this format, each value in the chart can be identified by one column label and one row label. For example, in the following illustration, the value 4 can be identified by the row label East and column label Model 421.

	Model 421	Model 532	Model 629
East	4	40	62
Central	22	18	62
West	40	10	39

Click on the columns, rows and labels of the data grid to learn about the information they contain.
In general, data grid columns are represented as series on a chart, and data grid rows are represented as data points within a chart series. However, in First Impression, you can specify if you want data grid rows to contain chart series data.
The organization of information in the data grid varies for each chart type. Specific information about how the data grid should be organized for different chart types is presented at the end of this help document.

Related Topics:

Chart Data Grids

More Complex Data Grids

Identifying Data Grid Elements

Multiple Level Labels

Creating the Data Grid

Manipulating the Data Grid Using Properties

Deleting Data Grid Elements

Inserting Data Grid Elements

Changing Data Grid Values or Text

Linking to a Formula One Spreadsheet

Data Grid Requirements of Various Chart Types

More Complex Data Grids

First Impression can also chart more complex data. In the simple data grid discussed previously, each value is identified by a column label and a row label. However, many times it is useful to view chart data that belongs to more than one category. You can accomplish this by adding levels of labels representing the new categories in the data grid.

The following illustration expands on the simple data grid discussed earlier in this chapter. One new level of column labels and one new level of row labels are added to the original data grid. In addition, three new rows of data have been added to the data grid.

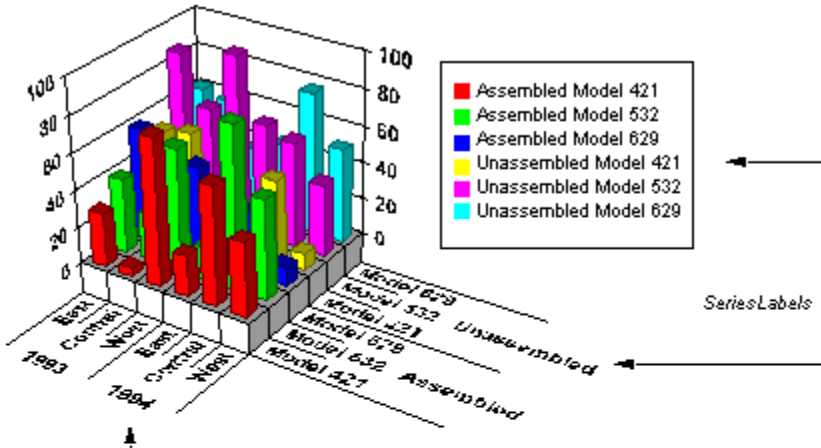
Notice that the value 4, previously identified by the row label East and the column label Model 421 can now be identified by the row labels 1994 and East, and by the column labels Assembled and Model 421. This provides more specific information about the data being charted.

This new level of labels identifies the year the sale took place.

		Assembled			Unassembled		
		Model 421	Model 532	Model 629	Model 421	Model 532	Model 629
1993	East	29	60	88	40	52	62
	Central	79	49	95	66	29	37
	West	64	35	57	88	44	77
1994	East	4	40	62	25	55	59
	Central	22	18	62	18	12	44
	West	40	10	39	54	9	51

This new level of labels identifies whether the models were assembled before they were sold.

Following is an illustration of a 3D bar chart created using this data grid.



		Assembled			Unassembled		
		Model 421	Model 532	Model 629	Model 421	Model 532	Model 629
1993	East	29	60	88	40	52	62
	Central	79	49	95	66	29	37
	West	64	35	57	88	44	77
1994	East	4	40	62	25	55	59
	Central	22	18	62	18	12	44
	West	40	10	39	54	9	51

Related Topics:

Chart Data Grids

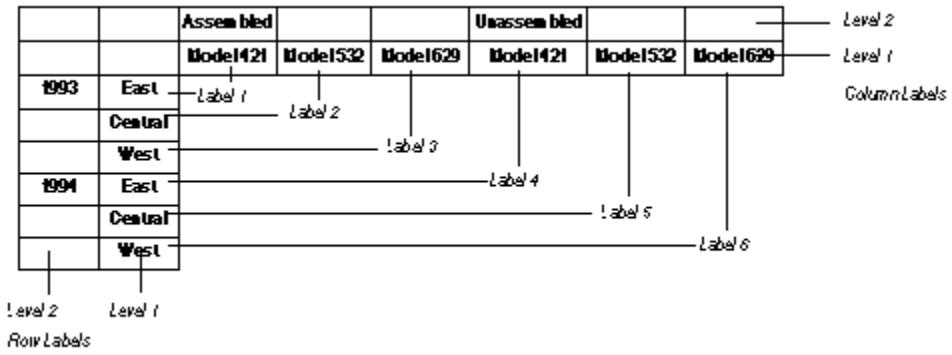
Identifying Data Grid Elements

There are two types of information stored in the data grid: data and labels. Columns of data are numbered from left to right, beginning with 1. Rows of data are numbered from top to bottom, beginning with 1.

		Assembled			Unassembled		
		Model421	Model532	Model629	Model421	Model532	Model629
1993	East	29	60	88	40	52	62
	Central	79	49	95	66	29	37
	West	64	35	57	88	44	77
1994	East	4	40	62	25	55	59
	Central	22	18	62	18	12	44
	West	40	10	39	54	9	51
		Column 1	Column 2	Column 3	Column 4	Column 5	Column 6

Row 1 Row 2 Row 3 Row 4 Row 5 Row 6

Note For information about identifying columns in data grids used to draw more complex chart types, refer to the section titled Special Chart Types later in this chapter.
Column labels identify the data columns. They are numbered from left to right within each level. Multiple levels of column labels are numbered from bottom to top, beginning with 1. Row labels identify the data rows. They are numbered from top to bottom, beginning with 1. Multiple levels of row labels are numbered from right to left beginning with 1.



Related Topics:

Chart Data Grids

Simple Data Grids

More Complex Data Grids

Multiple Level Labels

Creating the Data Grid

Manipulating the Data Grid Using Properties

Deleting Data Grid Elements

Inserting Data Grid Elements

Changing Data Grid Values or Text

Linking to a Formula One Spreadsheet

Data Grid Requirements of Various Chart Types

Multiple Level Labels

In order to accommodate the need for multiple levels of labels, First Impression must understand the relationship between labels at each level. The following illustration shows how First Impression interprets two levels of labels.

		1993				1994	
		Q1	Q2	Q3	Q4	Q1	Q2
US	Eastern Region	29	60	88	40	52	62
	Central Region	79	49	95	66	29	37
	Western Region	64	35	57	88	44	77
Europe	Northern Region	4	40	62	25	55	59
	Southern Region	22	18	62	18	12	44

This data grid reflects the way most people arrange multiple levels of labels in a spreadsheet or table.

		1993	1993	1993	1993	1994	1994
		Q1	Q2	Q3	Q4	Q1	Q2
US	Eastern Region	29	60	88	40	52	62
US	Central Region	79	49	95	66	29	37
US	Western Region	64	35	57	88	44	77
Europe	Northern Region	4	40	62	25	55	59
Europe	Southern Region	22	18	62	18	12	44

This data grid illustrates how First Impression interprets those labels.

In order to properly match lower and higher level labels, First Impression analyses the text in a label and compares it to the text in the next label. If the text is the same, or a null string, First Impression assumes the lower level label is still associated with the same higher level label. It continues associating lower level labels as long as it encounters the same text or a null string. When it encounters a new string, it stops associating lower level labels with the old string and begins associating them with the new string. The following illustration shows an example of this process.

		1993	*null string*	*null string*	*null string*	1994	*null string*
		Q1	Q2	Q3	Q4	Q1	Q2
US	Eastern Region	29	60	88	40	52	62
null string	Central Region	79	49	95	66	29	37
null string	Western Region	64	35	57	88	44	77
Europe	Northern Region	4	40	62	25	55	59
null string	Southern Region	22	18	62	18	12	44

These null strings tell First Impression that the labels Q2, Q3, and Q4 are associated with 1993.

Once First Impression encounters the string 1994, it stops associating 1993 with columns and begins associating 1994 with any null strings it finds until the next string is encountered.

> **Important** When comparing the contents of each column label, First Impression does differentiate between null strings and a string of spaces. If it encounters a string of spaces, it assumes it is a new label. Information on how to ensure a column label contains a null string is provided in the Using Properties to Manipulate the Data Grid section later in this chapter.

Related Topics:

Chart Data Grids

Simple Data Grids

More Complex Data Grids

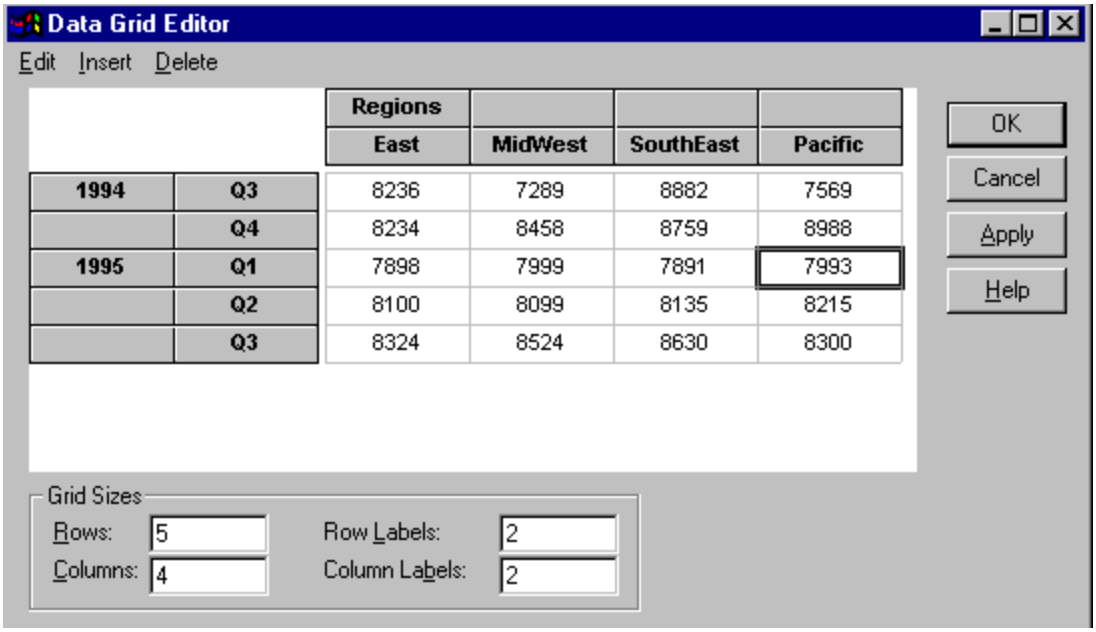
Identifying Data Grid Elements

Creating the Data Grid

The data grid is created when you draw the chart control on a form. First Impression uses the settings for the **RowCount, ColumnCount, RowLabelCount,** and **ColumnLabelCount** to determine the initial dimensions of the data grid. When it is first created, the data grid is filled with random data. You can use this random data to format your chart, or you can modify that data to represent specific data values.

The Data Grid Editor

Users can easily modify the data in the Data Grid without setting property values by using the Data Grid Editor. For more information about the Data Grid Editor, see Chart Basics in the User's Guide.



Related Topics:

Chart Data Grids
Simple Data Grids
More Complex Data Grids
Identifying Data Grid Elements
Multiple Level Labels
Manipulating the Data Grid Using Properties
Deleting Data Grid Elements
Inserting Data Grid Elements
Changing Data Grid Values or Text
Linking to a Formula One Spreadsheet
Data Grid Requirements of Various Chart Types

Manipulating the Data Grid Using Properties

First Impression provides a number of properties that allow you to resize the data grid, or modify a value, column label, or row label. Changing any of these properties causes the chart to redraw and reflect the changes in the data grid.

- Use the **ColumnCount** and **RowCount** properties to determine the number of data columns and data rows in the data grid. Rows and columns containing labels are not included in the ColumnCount and RowCount. If the new column or row count is greater or smaller than the current count, columns are added to or deleted from the right and rows are added to or deleted from the bottom of the data grid. Any data in the deleted columns or rows is lost.
- Use the **ColumnLabelCount** property to set the number of levels of column labels on the data grid. Use the **RowLabelCount** property to set the number of levels of row labels on the data grid. If the new count is greater than the current count, new levels of labels are added to the top or left side of the data grid. If the new count is smaller than the current count, levels of labels are deleted from the top or left side of the data grid. Any labels in deleted levels are lost.
- Use the **Column** and **Row** properties to identify a specific point in the data grid. Data columns are numbered from left to right beginning with 1. Rows in the data grid are numbered from top to bottom beginning with 1. Any labels in the data grid are not counted.

	Model 421	Model 532	Model 629
East	4	40	62
Central	22	18	62
West	40	10	39

Column = 3
Row = 1

Important In most chart types, each column in the data grid represents a series on the chart. However, First Impression supports a number of special chart types that require two or more columns to chart a series. Specific information about the data grid requirements of these special chart types is provided later in this chapter.

- Use the **ColumnLabelIndex** and **RowLabelIndex** properties to identify a specific row or column of labels. For example, if you want to change a label in the second level of column labels, you must set ColumnLabelIndex to 2. Levels of column labels are numbered from bottom to top, beginning at 1. Levels of row labels are numbered from right to left, beginning at 1.
- Use the **ColumnLabel** and **RowLabel** properties to change the label identifying a row or column. ColumnLabel holds the label text on the column identified by the Column property and in the level identified by the ColumnLabelIndex property. RowLabel holds the text associated with the row identified by the Row property and in the level identified by the RowLabelIndex property.

Note If you are removing an existing label from a column or row, highlight it in the Properties Window and press Delete. This tells First Impression to place a null string in the label. If you remove the existing label by highlighting it and pressing the space bar, you are actually replacing the label with spaces instead of a null string.

- Use the **Data** property to modify an individual value in the data grid. Use Column and Row properties to identify the data you wish to change before using the Data property which assigns the new data value. If you assign a value with the Data property, the RandomFill property is automatically set to FALSE to prevent you from losing the data you entered.
- Use **RandomFill** to generate random data for prototyping a chart.
- Use the **SetSize** method to change all the dimensions of the data grid at once. You can use this method to set the column count, row count, column label count, and row label count in one operation. This saves you from having to use four separate properties to set each parameter individually.

Related Topics:

Chart Data Grids

Simple Data Grids

More Complex Data Grids

Identifying Data Grid Elements

Multiple Level Labels

Creating the Data Grid

Deleting Data Grid Elements

Inserting Data Grid Elements

Changing Data Grid Values or Text

Linking to a Formula One Spreadsheet

Data Grid Requirements of Various Chart Types

Deleting Data Grid Elements

The methods and properties discussed under Manipulating and Resizing the Data Grid can be used to delete columns, rows, or levels of labels from the "end" of each dimension. However, there are times when you need to delete elements in a specific location within the data grid. For each method, you must identify the first element to delete by number. You must also specify the total number of elements to delete.

- Use **DeleteColumns** to delete one or more specific columns from the data grid. Columns are deleted from the first column you specify to the right.
- Use **DeleteRows** to delete one or more specific rows from the data grid. Rows are deleted from the first row you specify down.

Note Both **DeleteColumns** and **DeleteRows** delete cells and their labels.

- Use **DeleteColumnLabels** to delete one or more specific levels of column labels from the data grid. Levels of column labels are deleted from the first level you specify up.
- Use **DeleteRowLabels** to delete one or more specific levels of row labels from the data grid. Levels of row labels are deleted from the first level you specify to the left.

Related Topics:

Chart Data Grids

Simple Data Grids

More Complex Data Grids

Identifying Data Grid Elements

Multiple Level Labels

Creating the Data Grid

Manipulating the Data Grid Using Properties

Inserting Data Grid Elements

Changing Data Grid Values or Text

Linking to a Formula One Spreadsheet

Data Grid Requirements of Various Chart Types

Inserting Data Grid Elements

You can also use methods to insert new elements at specific locations in the data grid.

Important When new labels are added to the data grid, either by adding a new level of labels or adding rows or columns to the data grid, they contain a null string. When new columns or rows of data are added, their status depends on the current setting of the RandomFill property. If RandomFill is set to False, the columns and rows contain a null value. If RandomFill is set to True, the columns and rows are filled with random data.

- Use **InsertColumns** to add one or more data columns to a specific location in the data grid. The new data columns are added to the left of the column you specify. You must also specify how many new columns to add.
- Use **InsertRows** to add one or more data rows to a specific location in the data grid. The new data rows are added above the row you specify. You must also specify how many new rows to add.
- Use **InsertColumnLabels** to add one or more levels of column labels at a specific location in the data grid. The new levels are added beneath the level you specify. You must also specify how many new levels to add.
- Use **InsertRowLabels** to add one or more levels of row labels at a specific location in the data grid. The new levels are added to the right of the level you specify. You must also specify how many new levels to add.

Related Topics:

Chart Data Grids

Simple Data Grids

More Complex Data Grids

Identifying Data Grid Elements

Multiple Level Labels

Creating the Data Grid

Manipulating the Data Grid Using Properties

Deleting Data Grid Elements

Changing Data Grid Values or Text

Linking to a Formula One Spreadsheet

Data Grid Requirements of Various Chart Types

Special Chart Types

Changing Data Grid Values or Text

You can also change the data or labels in a data grid. When you change a data value or label, the data grid is updated internally and the chart is redrawn to reflect the new value.

- Use the **Data** property to return the value currently assigned to a specific point in the data grid or to set a new value for a specific point. You must identify the position by row and column number. You can also use this property to insert a null value in that position or return a null flag to determine if the data grid holds a null value in that position.
- Use **ColumnLabel** and **RowLabel** methods to return the text currently assigned to a specific position within the column or row label or to set a new label for a specific level on a specific row or column. You must identify the specific row or column which the label identifies and the specific level of label. To replace an existing label with a null string, you can enter zero as the label text argument.
- To import data from an array, use the **CopyDataFromArray** method. To import data into an array, use the **CopyDataToArray** method.

Related Topics:

Chart Data Grids

Simple Data Grids

More Complex Data Grids

Identifying Data Grid Elements

Multiple Level Labels

Creating the Data Grid

Manipulating the Data Grid Using Properties

Deleting Data Grid Elements

Inserting Data Grid Elements

Linking to a Formula One Spreadsheet

Data Grid Requirements of Various Chart Types

Special Chart Types

Linking to a Formula One Spreadsheet

You can create a link between a First Impression chart control and a Formula One spreadsheet control. Once the link has been activated, the data from the spreadsheet is imported into the data grid. As long as the link remains active, the spreadsheet notifies the chart control if any charted data changes. The chart automatically redraws to reflect the change.

To organize your spreadsheet data appropriately, you must understand the data grid information presented earlier in this chapter and the data grid requirements of each chart type.

Setting Up the Link

First Impression provides three properties to set up the link between the spreadsheet control and the chart control.

- Use the **SsLinkBook** property to identify the spreadsheet.
- Use the **SsLinkRange** property to identify the range within the spreadsheet control that contains the data you want to chart.
- Use the **SsLinkMode** property to specify whether or not the link is active and how First Impression interprets the spreadsheet data.

Identifying the Spreadsheet

You must identify the Formula One spreadsheet you want to use as the data source for the chart. Identify the spreadsheet by table name and not the handle of the spreadsheet control. You can determine the table name using the Formula One TableName property.

Specifying a data range

You must also identify the range within the spreadsheet that holds the data you want to chart. This range can be identified by name or coordinates, although the use of named ranges is recommended.

There are also a number of other rules that apply when specifying a data range:

- A defined name used as the source range for the link must contain absolute references, not relative references.
- The source range cannot contain external sheet references.
- First Impression does not interpret "backward" ranges. For example, enter the range $A$4:$D$19, not $D$19:$A$4.

Controlling the Link Behavior

There are three different modes you can use to link to a spreadsheet control: off, on and autoparse.

- **Off**. The link is not active. Any changes in the spreadsheet are not automatically updated in the chart.
- **On**. The link is active and changes in the spreadsheet are automatically displayed in the chart. In this mode, First Impression makes no attempt to determine if the data in the specified range is text labels or data. You must first identify how many levels of row and column labels, and how many rows and columns of data your data range contains. Enter these values in the ColumnCount, RowCount, ColumnLabelCount, and RowLabelCount properties to set the dimensions of the data grid. If these dimensions do not match the data in the spreadsheet range, you may get unexpected results.
- **AutoParse**. The link is active and changes in the spreadsheet are automatically displayed in the chart. In this mode, First Impression analyses the data in the specified spreadsheet range and attempts to determine the number of levels of column and row labels. It also tries to determine how many data columns and data rows are in the specified range. First Impression then adjusts the settings of the **ColumnCount, RowCount, ColumnLabelCount**, and **RowLabelCount** properties to match the dimensions of the data imported from the spreadsheet.

Note: AutoParse mode can be very useful when linking to simple ranges of data that are organized in exactly the way First Impression expects to find the data. However, using On mode and setting the data grid dimensions yourself is recommended in most cases that involve more complex data or unusual data arrangements. For example, if your data does not contain column labels, AutoParse may become confused and you may not get the chart you expect.

Related Topics:

Chart Data Grids

Simple Data Grids

More Complex Data Grids

Identifying Data Grid Elements

Multiple Level Labels

Creating the Data Grid

Manipulating the Data Grid Using Properties

Deleting Data Grid Elements

Inserting Data Grid Elements

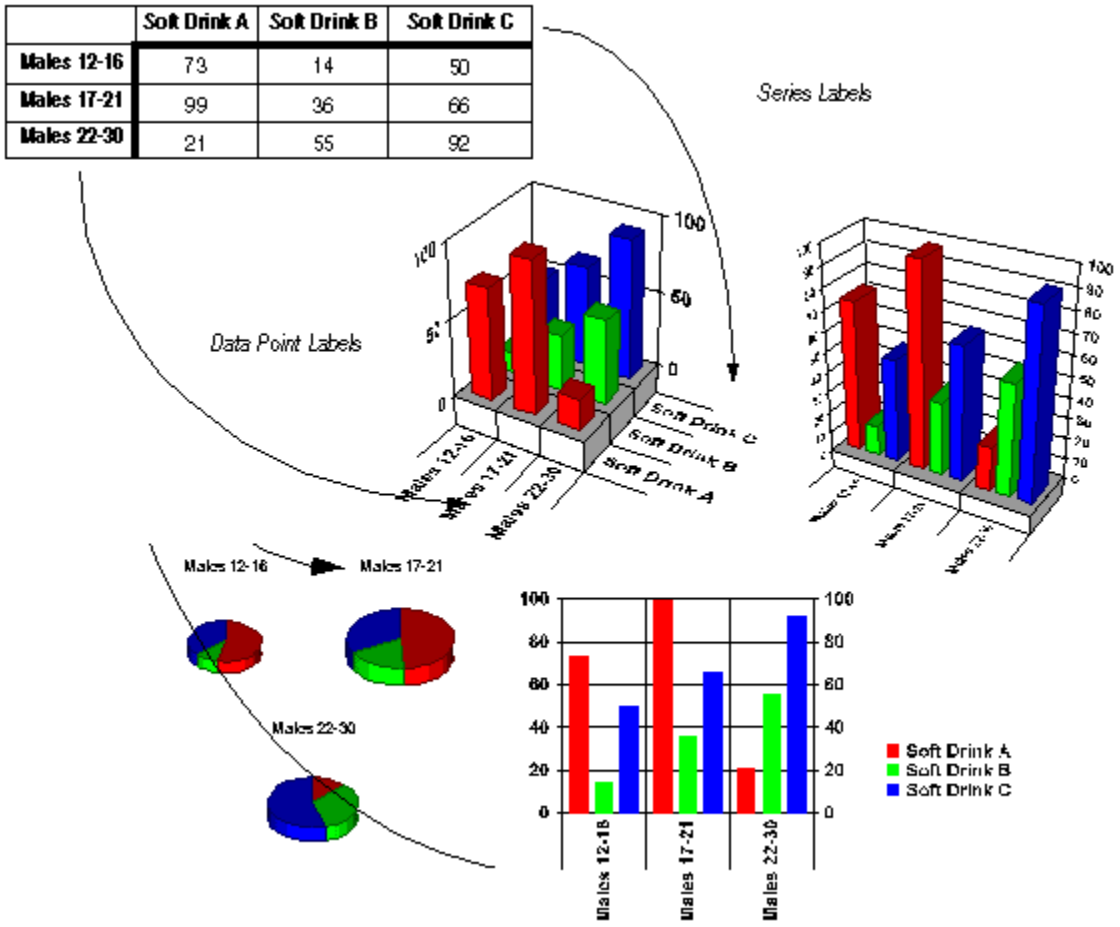
Changing Data Grid Values or Text

Data Grid Requirements of Various Chart Types

Data Grid Requirements of Various Chart Types

Most of the common chart types such as area, line, bar, pie, doughnut, and radar charts expect the same organization of information in the data grid. The column labels are used to identify chart series. The row labels are used to identify categories in the chart. Each column of data in the data grid displays one series on the chart. Each row value in the column is an individual data point in that series.

The following illustration shows a sample data grid and several charts drawn based on that data.



Related Topics:

Chart Data Grids

Simple Data Grids

More Complex Data Grids

Identifying Data Grid Elements

Multiple Level Labels

Creating the Data Grid

Manipulating the Data Grid Using Properties

Deleting Data Grid Elements

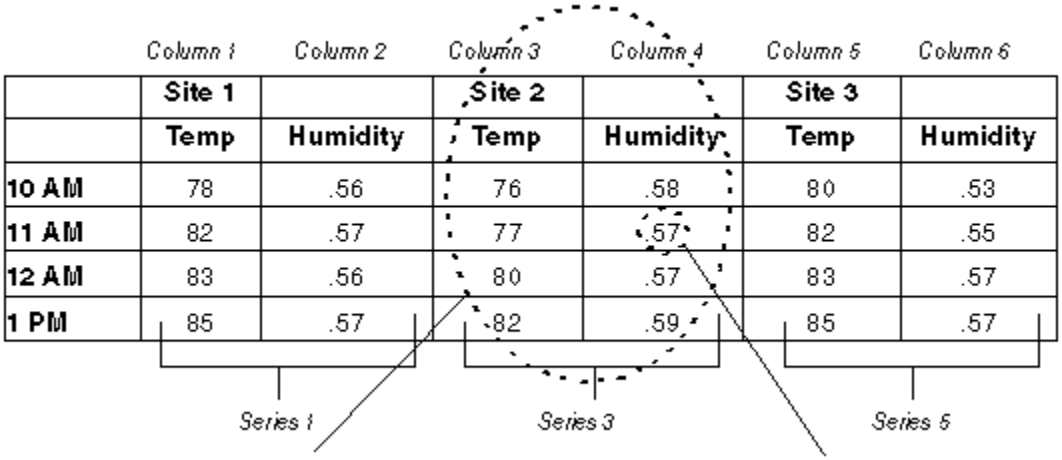
Inserting Data Grid Elements

Changing Data Grid Values or Text

Special Chart Types

First Impression supports a number of chart types that require multiple columns of data to draw a series. For each of these charts, a series is drawn when there are enough columns of data available. Any extra columns of data are ignored. For example, on a polar chart, a series requires two columns of data; if there are three columns of data in the data grid, the third column is ignored. The column label on the first column in each series is used to identify the series in the legend. Additionally, any formatting applied to the first column is used to display the series on the chart. Any formatting applied to the second column is ignored.

When columns in the data grid are combined to create a series, the column identification numbers do not change. If you need to refer to a series in code, refer to it by the number of the first column in the series. For example, the data grid in the following illustration contains six columns and is used to draw an XY chart. Since it requires two columns of data for each series in the chart, the six columns results in a chart with three series. These would be identified as series 1, series 3, or series 5.



To turn on the display of guidelines in this series, refer to series three in the following code:

```
Vtchart1.plot.seriescollection.item(3).
showguideline (vtchaxisidx) = True
```

To change the value in this position in the data grid refer to row 2, column 4 in the following code:

```
With Vtchart1
.column = 4
.row = 2
.data = 58
endwith.
```

The specific organization required for the data grid is discussed under each chart type.

Related Topics:

XY Charts

Polar Charts

Bubble Charts

Basic Hi-Lo Charts

Hi-Lo-Close Charts

Open Hi-Lo-Close Charts

Gantt Charts

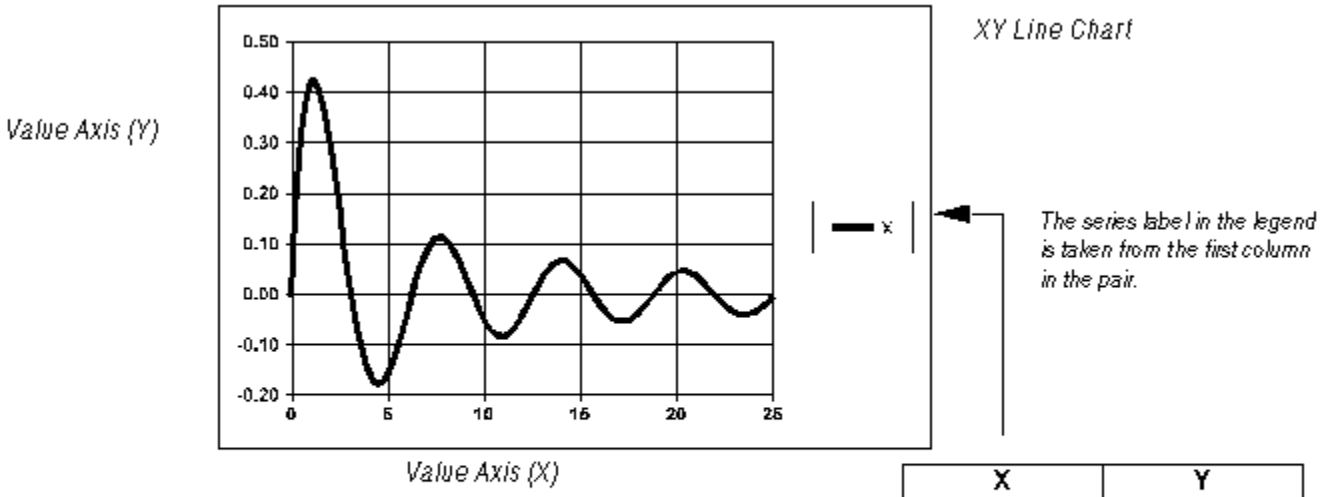
Elevation Charts

3D XYZ Charts

3D Scatter Charts

XY Charts

An XY chart plots two pairs of numbers as x and y coordinates. Each series on the chart requires two columns of data in the data grid. The first column holds the x coordinate and the second column the y coordinate. The partial data grid in the following illustration shows the organization required for an XY chart.



X	Y
0.1963495408	0.163071339
0.3926990817	0.274778261
0.5890486225	0.349624439
0.7853981634	0.39604991

Related Topics:

Special Chart Types

Polar Charts

Bubble Charts

Basic Hi-Lo Charts

Hi-Lo-Close Charts

Open Hi-Lo-Close Charts

Gantt Charts

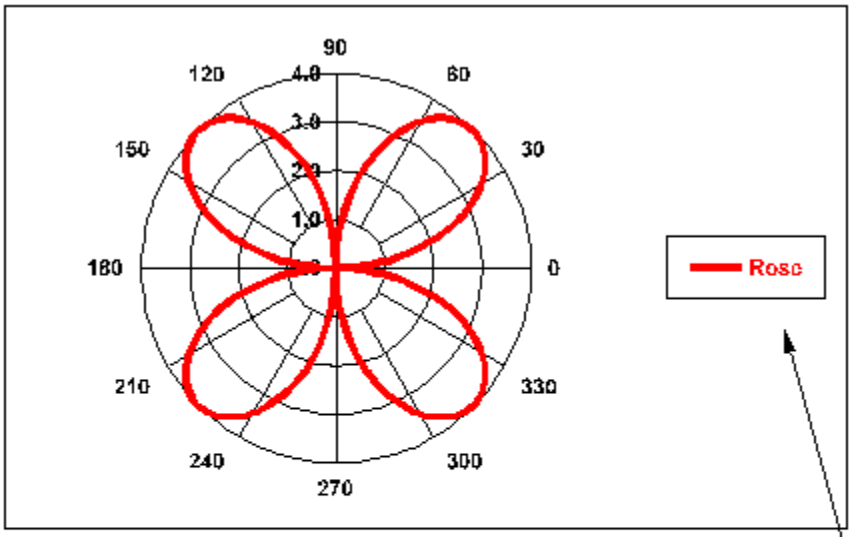
Elevation Charts

3D XYZ Charts

3D Scatter Charts

Polar Charts

Polar charts require two columns in the data grid for each chart series: the first data column holds the distance from the center of the chart (the Radius) and the second column holds the angle from the zero (0) position of the chart (the Angle.) The column label on the first column in each series is used to identify the series in the legend. Additionally, any formatting applied to the first column is used to display the series on the chart. Any formatting applied to the second column is ignored. The partial data grid in the following illustration shows the organization required for a polar chart.



Polar Chart

The series label in the legend is taken from the first column in the pair.

(Angle)

Rose	Radius
.39207	2.8125
.078036	5.625
1.16114	8.4315
1.53073	11.25

Related Topics:

Special Chart Types

XY Charts

Bubble Charts

Basic Hi-Lo Charts

Hi-Lo-Close Charts

Open Hi-Lo-Close Charts

Gantt Charts

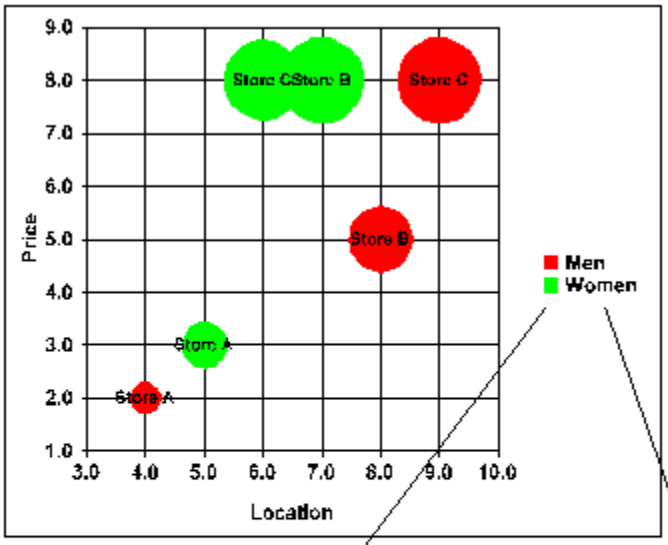
Elevation Charts

3D XYZ Charts

3D Scatter Charts

Bubble Charts

Each series displayed on a bubble chart requires three columns of data in the data grid. The first column holds the X coordinate, the second column holds the Y coordinate, and the third column determines the relative bubble size. The column label on the first column in each series is used to identify the series in the legend. Additionally, any formatting applied to the first column is used to display the series on the chart. Any formatting applied to the second and third columns are ignored.



Since this chart type requires three columns of data, every first column label of three is used to identify the series in the legend.

In this illustration the category names are being displayed as data point labels.

	(X)	(Y)	(Size)	(X)	(Y)	(Size)
	Men	Prices	Score	Women	Prices	Score
Store A	4	2	6	5	3	8
Store B	8	5	13	7	8	15
Store C	9	8	17	6	8	14

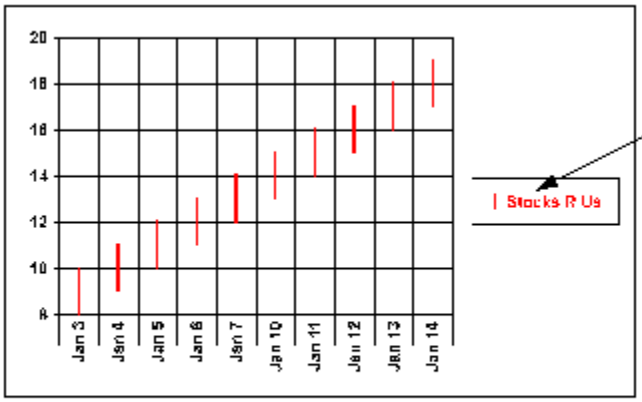
Related Topics:

Special Chart Types
XY Charts
Polar Charts
Basic Hi-Lo Charts
Hi-Lo-Close Charts
Open Hi-Lo-Close Charts
Gantt Charts
Elevation Charts
3D XYZ Charts
3D Scatter Charts

Basic Hi-Lo Charts

The data in the data grid must be organized appropriately for each type of hi-lo chart supported by first impression.

The basic hi-lo chart requires two columns for each data series: the first column is used for the high point, and the second is used for the low point.



Hi-Lo Chart

(High)

	Stocks R Us	Low
Jan 3	10	8
Jan 4	11	9
Jan 5	12	10
Jan 6	13	11
Jan 7	14	12
Jan 10	15	13
Jan 11	16	14
Jan 12	17	15
Jan 13	18	16
Jan 14	19	17

Related Topics:

Special Chart Types
XY Charts
Polar Charts
Bubble Charts
Hi-Lo-Close Charts
Open Hi-Lo-Close Charts
Gantt Charts

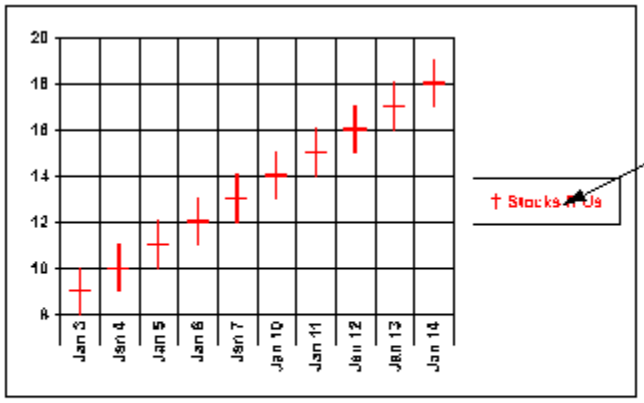
Elevation Charts

3D XYZ Charts

3D Scatter Charts

Hi-Lo-Close Charts

A hi-lo-close chart requires three columns for each data series: the first column is used for the high point, the second for the low point, and the third for the closing point.



Hi-Lo-Close Chart

(High)

	Stocks R Us	Low	Close
Jan 3	10	8	9
Jan 4	11	9	10
Jan 5	12	10	11
Jan 6	13	11	12
Jan 7	14	12	13
Jan 10	15	13	14
Jan 11	16	14	15
Jan 12	17	15	16
Jan 13	18	16	17
Jan 14	19	17	18

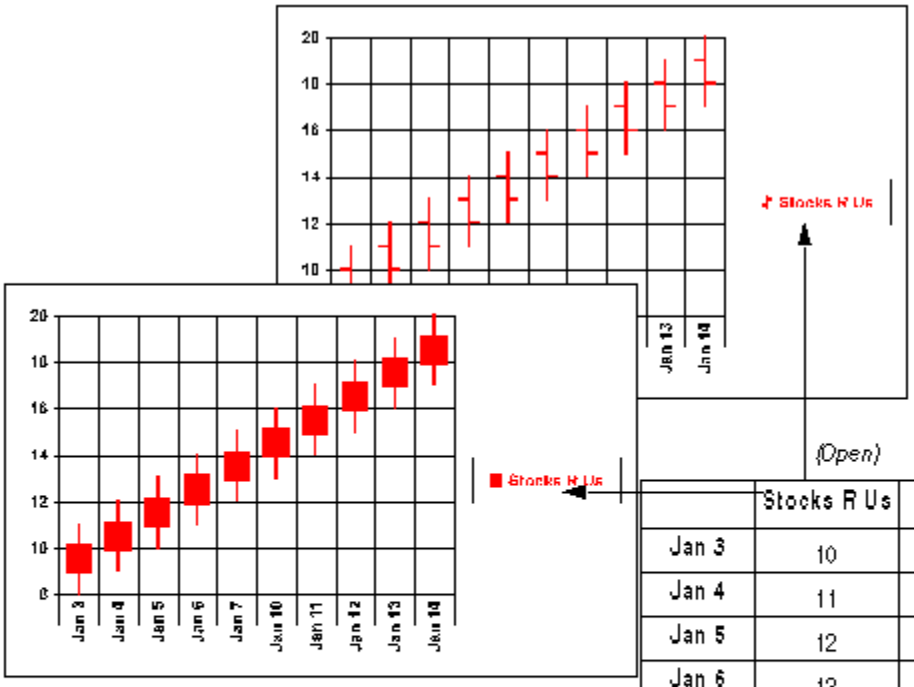
Related Topics:

Special Chart Types
XY Charts
Polar Charts
Bubble Charts
Basic Hi-Lo Charts
Open Hi-Lo-Close Charts
Gantt Charts
Elevation Charts
3D XYZ Charts

3D Scatter Charts

Open Hi-Lo-Close Charts

An open-hi-lo-close chart requires four columns for each data series: the first column is used for the open point, the second for the high point, the third for the low point, and the fourth for the closing point.



Open Hi-Lo-Close Chart

Open Hi-Lo-Close Bar Chart

(Open)

	Stocks R Us	High	Low	Close
Jan 3	10	11	8	9
Jan 4	11	12	9	10
Jan 5	12	13	10	11
Jan 6	13	14	11	12
Jan 7	14	15	12	13
Jan 10	15	16	13	14
Jan 11	16	17	14	15
Jan 12	17	18	15	16
Jan 13	18	19	16	17
Jan 14	19	20	17	18

Related Topics:

Special Chart Types

XY Charts

Polar Charts

Bubble Charts

Basic Hi-Lo Charts

Hi-Lo-Close Charts

Gantt Charts

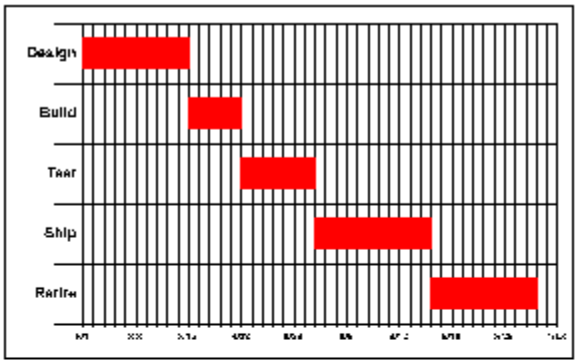
Elevation Charts

3D XYZ Charts

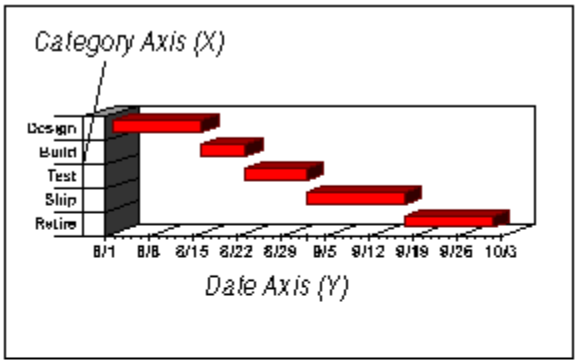
3D Scatter Charts

Gantt Charts

Use a gantt chart to display range bars which indicate the relative time required for various activities within a project. This chart type requires two columns of data for each series. The first column is the start time and the second column is the duration. The column label on the first column in each series is used to identify the series in the legend. Additionally, any formatting applied to the first column is used to display the series on the chart. Any formatting applied to the second column is ignored.



2D Gantt Chart



3D Gantt Chart

	Start Date	Duration
Design	34547	14
Build	34561	16
Test	34567	10
Ship	34577	15
Retire	34592	14

The start date is a number calculated as the number of days since January 1, 1900. If you are linking to a Formula One spreadsheet, you can enter a date in a more understandable format such as 1-23-93. Since the spreadsheet stores that date as the number of days since January 1, 1900 First Impression can use the data to create a date axis for your chart. The duration is a number of days. Again, if you are using the Formula One spreadsheet to provide data for the chart, you can use an expression to subtract a start date from a finish date to determine the duration.

Related Topics:

Special Chart Types

XY Charts

Polar Charts

Bubble Charts

Basic Hi-Lo Charts

Hi-Lo-Close Charts

Open Hi-Lo-Close Charts

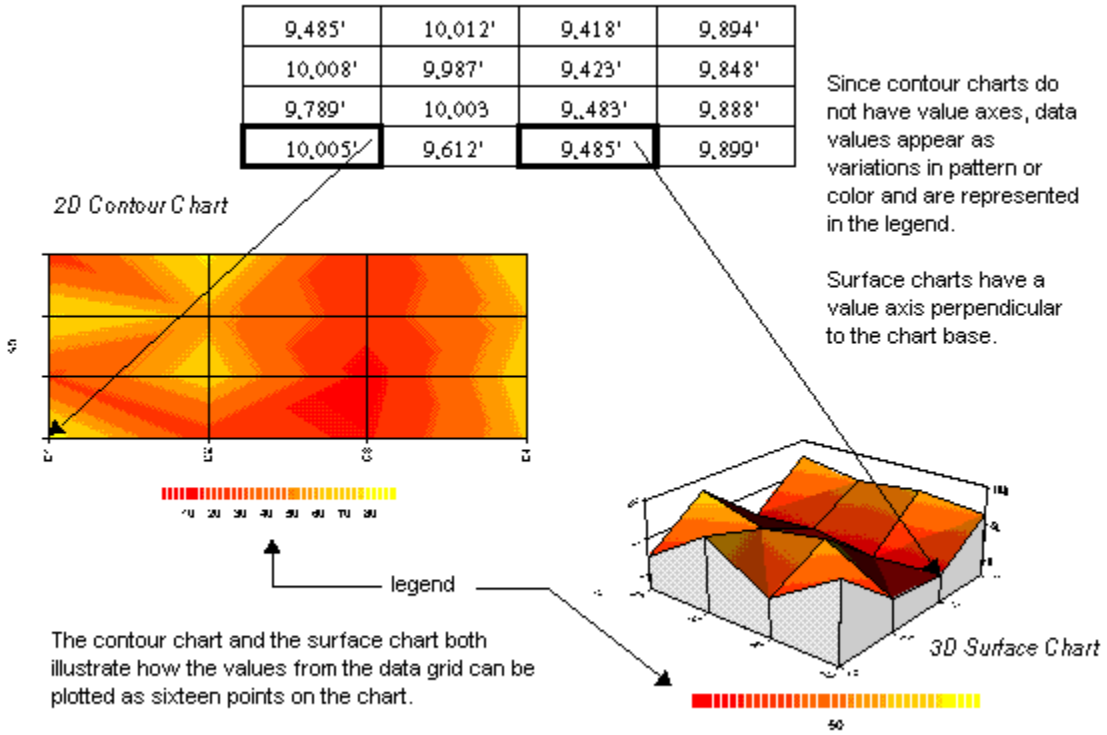
Elevation Charts

3D XYZ Charts

3D Scatter Charts

Elevation Charts

Use Elevation charts to display topographic data. There are two types of elevation charts: 2D contour charts and 3D surface charts. In contour charts, data values within a specific range appear on the chart in two-dimensional bands of the same color. In surface charts, data values within a specific range have the same elevation. Data values can also be illustrated with contour bands or lines of the same color projected on the three-dimensional surface model.



Related Topics:

Special Chart Types

XY Charts

Polar Charts

Bubble Charts

Basic Hi-Lo Charts

Hi-Lo-Close Charts

Open Hi-Lo-Close Charts

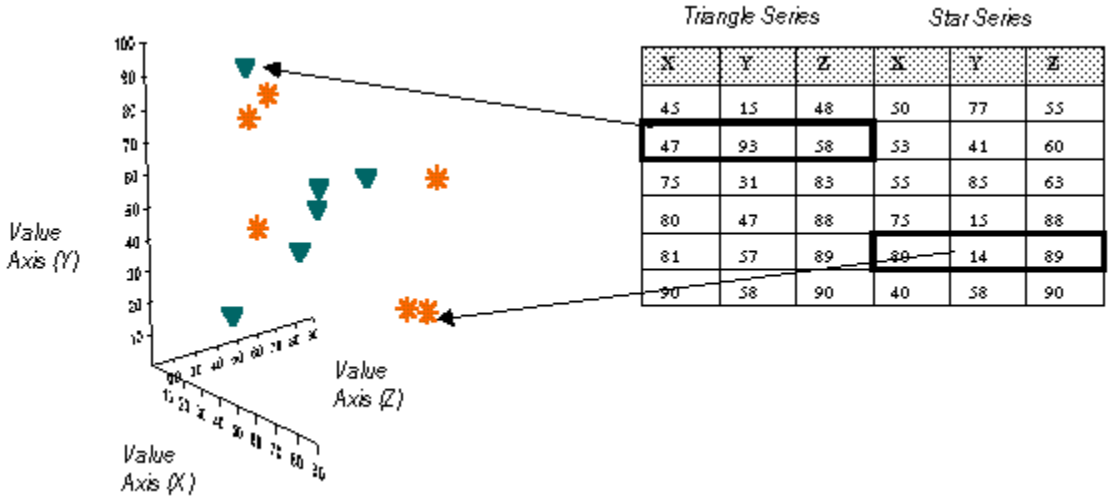
Gantt Charts

3D XYZ Charts

3D Scatter Charts

3D XYZ Charts

This chart displays variation and relationships over three sets of values plotted on a three-dimensional grid. Each series on the chart requires three columns of data in the data grid.



X	Y	Z	X	Y	Z
45	15	48	50	77	55
47	93	58	53	41	60
75	31	83	55	85	63
80	47	88	75	15	88
81	57	89	80	14	89
90	58	90	40	58	90

Related Topics:

Special Chart Types

XY Charts

Polar Charts

Bubble Charts

Basic Hi-Lo Charts

Hi-Lo-Close Charts

Open Hi-Lo-Close Charts

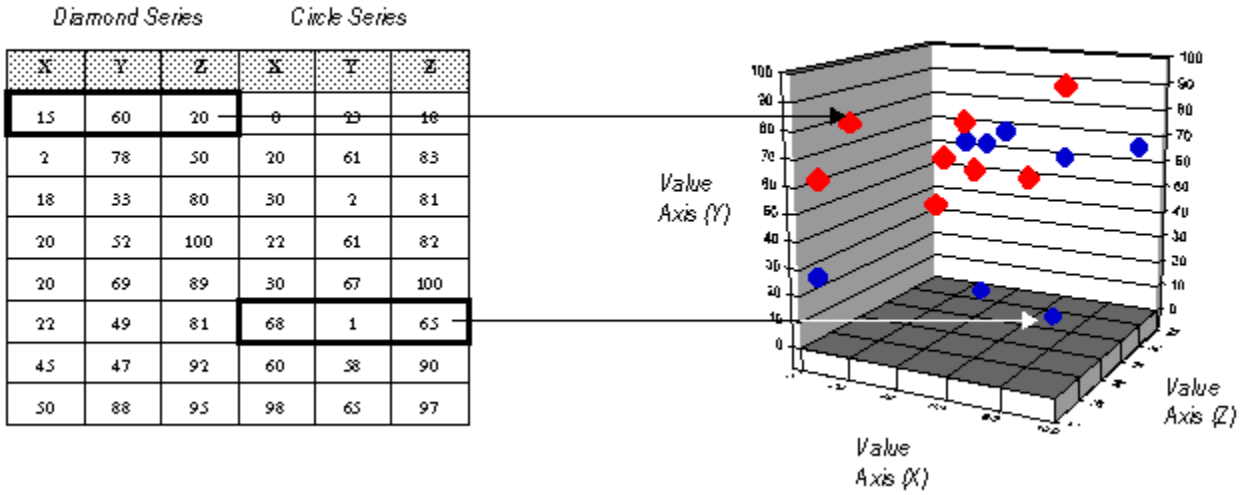
Gantt Charts

Elevation Charts

3D Scatter Charts

3D Scatter Charts

This chart is drawn as a three-dimensional grid with a base and walls. It plots data points created by the intersection of three different coordinate values. Each series on the chart requires three columns of data in the data grid.



Related Topics:

Special Chart Types
XY Charts
Polar Charts
Bubble Charts
Basic Hi-Lo Charts
Hi-Lo-Close Charts
Open Hi-Lo-Close Charts
Gantt Charts

Elevation Charts

3D XYZ Charts

Column Label Text

The text in these column labels is used as series labels to identify the series in a chart legend.

Row Label Text

The text in these row labels is used as data point labels on a chart.

Data Grid Columns

Data grid columns are represented as series on a chart.

Data Grid Rows

Data grid rows are represented as data points within a chart series.

Using Events

First Impression provides a set of events that allows you to track and monitor actions performed on a chart control by users of your application. Events allow you to respond to user's actions and control the operations of the chart control

One of the most common uses of events is to prevent the default behavior on an action. For example, you may want to prevent your end-user from being able to select certain chart elements. You might also want to prevent the display of First Impression dialog boxes when a user double clicks on items.

The following pages provide a complete alphabetical Event Reference.

ApplyChanges Event

Description

This event occurs when the user presses the Ok or Apply button in a tabbed dialog.

Syntax

Sub VtChart1_ApplyChanges ( )

AxisActivated Event

Description

This event occurs when the user double clicks a chart axis. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Sub VtChart1_AxisActivated (*axisId* As Integer, *axisIndex* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

axisId identifies a specific axis.

The following table describes the valid constants for this argument.

Constants	Description
VtChAxisIdX	Identifies the X axis
VtChAxisIdY	Identifies the Y axis
VtChAxisIdY2	Identifies the secondary Y axis
VtChAxisIdZ	Identifies the Z axis

axisIndex is reserved for future use. For this version of First Impression, 1 is the only valid value for this *argument*.

mouseFlag indicates a key is held down when the mouse button is clicked.

The following table lists the constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting an axis and displaying the Format Axis dialog box.

AxisLabelActivated Event

Description

This event occurs when the user double clicks an axis label. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Sub VtChart1_AxisLabelActivated (*axisId* As Integer, *axisIndex* As Integer, *labelSetIndex* As Integer, *labelIndex* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

axisId identifies a specific axis.

The following table describes the valid constants for this argument.

Constants	Description
VtChAxisIdX	Identifies the X axis
VtChAxisIdY	Identifies the Y axis
VtChAxisIdY2	Identifies the secondary Y axis
VtChAxisIdZ	Identifies the Z axis

axisIndex is reserved for future use. For this version of First Impression, 1 is the only valid value for this argument.

labelSetIndex identifies the level of labels you are double clicking on. Levels of labels are numbered from the axis out, beginning with 1.

labelIndex is currently unused.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the axis labels and displaying the Format Axis Label dialog box.

AxisLabelSelected Event

Description

This event occurs when the user clicks an axis label.

Syntax

> **Sub VtChart1_AxisLabelSelected** (*axisId* As Integer, *axisIndex* As Integer, *labelSetIndex* As Integer, *labelIndex* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

axisId identifies a specific axis.

The following table describes the valid constants for this argument.

Constants	Description
VtChAxisIdX	Identifies the X axis
VtChAxisIdY	Identifies the Y axis
VtChAxisIdY2	Identifies the secondary Y axis
VtChAxisIdZ	Identifies the Z axis

axisIndex is reserved for future use. For this version of First Impression, 1 is the only valid value for this argument.

labelSetIndex identifies the level of labels you are clicking on. Levels of labels are numbered from the axis out, beginning with 1.

labelIndex is currently unused.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the axis and displaying selection handles along the axis title.

AxisSelected Event

Description

This event occurs when the user clicks a chart axis.

Syntax

Sub VtChart1_AxisSelected (*axisId* As Integer, *axisIndex* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

axisId identifies a specific axis.

The following table describes the valid constants for this argument.

Constants	Description
VtChAxisIdX	Identifies the X axis
VtChAxisIdY	Identifies the Y axis
VtChAxisIdY2	Identifies the secondary Y axis
VtChAxisIdZ	Identifies the Z axis

axisIndex is reserved for future use. For this version of First Impression, 1 is the only valid value for this argument.

mouseFlag indicates a key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set cancel equal to True to prevent the user from selecting the axis.

AxisTitleActivated Event

Description

This event occurs when the user double clicks an axis title. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Sub VtChart1_AxisTitleActivated (*axisId* As Integer, *axisIndex* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

axisId identifies a specific axis.

The following table describes the valid constants for this argument.

Constants	Description
VtChAxisIdX	Identifies the X axis
VtChAxisIdY	Identifies the Y axis
VtChAxisIdY2	Identifies the secondary Y axis
VtChAxisIdZ	Identifies the Z axis

axisIndex is reserved for future use. For this version of First Impression, 1 is the only valid value for this argument.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the axis title and displaying the Format Axis Title dialog box.

AxisTitleSelected Event

Description

This event occurs when the user clicks an axis title.

Syntax

Sub VtChart1_AxisTitleSelected (*axisId* As Integer, *axisIndex* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

axisId identifies a specific axis.

The following table describes the valid constants for this argument.

Constants	Description
VtChAxisIdX	Identifies the X axis
VtChAxisIdY	Identifies the Y axis
VtChAxisIdY2	Identifies the secondary Y axis
VtChAxisIdZ	Identifies the Z axis

axisIndex is reserved for future use. For this version of First Impression, 1 is the only valid value for this argument.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the axis and displaying selection handles around the axis title.

ChartActivated Event

Description

This event occurs when the user double clicks the First Impression chart control, but not on a specific element in the chart. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Sub VtChart1_ChartActivated (*mouseFlag* As Integer, *cancel* As Integer)

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the chart and displaying the Format Chart dialog box.

ChartSelected Event

Description

This event occurs when the user clicks the First Impression chart control, but not on a specific element in the chart.

Syntax

Sub VtChart1_ChartSelected (*mouseFlag* As Integer, *cancel* As Integer)

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the chart control.

Click Event

Description

This event occurs when the user presses and then releases a mouse button over an object. This event is fired before the select event.

Syntax

Sub VtChart1_Click ( )

Remarks

For additional information, refer to the description of the Click event in the Microsoft Visual Basic Language Reference Manual.

DblClick Event

Description

This event occurs when a user presses and releases a mouse button and then presses and releases it again over an oject This event is fired before the activate event.

Syntax

Sub VtChart1_DblClick ( )

Remarks

For additional information, refer to the description of the DblClick event in the Microsoft Visual Basic Language Reference Manual.

DragDrop Event

Description

This event occurs when a drag-drop operation is complete.

Syntax

Sub VtChart1_DragDrop (*Source* As Control, *X* As Single, *Y* As Single)

Remarks

For additional information, refer to the description of the DragDrop event in the Microsoft Visual Basic Language Reference Manual.

DragOver Event

Description

This event occurs when a drag-drop operation is in process.

Syntax

Sub VtChart1_DragOver (*Source* As Control, *X* As Single, *Y* As Single, *State* As Integer)

Remarks

For additional information, refer to the description of the DragOver event in the Microsoft Visual Basic Language Reference Manual.

FootnoteActivated Event

Description

This event occurs when the user double clicks the chart footnote. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Sub VtChart1_FootnoteActivated (*mouseFlag* As Integer, *cancel* As Integer)

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the footnote and displaying the Format Footnote dialog box.

FootnoteSelected Event

Description

This event occurs when the user clicks the chart footnote.

Syntax

Sub VtChart1_FootnoteSelected (*mouseFlag* As Integer, *cancel* As Integer)

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the footnote.

GotFocus Event

Description

This event occurs when the First Impression window receives keyboard focus, either by clicking the object or changing the focus in code using the SetFocus method.

Syntax

Sub VtChart1_GotFocus ( )

Remarks

For additional information, refer to the description of the GotFocus event in the Microsoft Visual Basic Language Reference Manual.

KeyDown, KeyUp Events

Description

These events occur when the user presses (KeyDown) and releases (KeyUp) a key while the First Impression object has the focus.

Syntax

Sub VtChart1_KeyDown (*KeyCode* As Integer, *Shift* As Integer)

Sub VtChart1_KeyUp (*KeyCode* As Integer, Shift As Integer)

Remarks

For additional information, refer to the descriptions of the KeyDown and KeyUp events in the Microsoft Visual Basic Language Reference Manual.

KeyPress Event

Description

This event occurs when the user presses and releases an ANSI key.

Syntax

Sub VtChart1_KeyPress (*KeyAscii* As Integer)

Remarks

For additional information, refer to the description of the KeyPress event in the Microsoft Visual Basic Language Reference Manual.

LegendActivated Event

Description

This event occurs when the user double clicks the chart legend. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Sub VtChart1_LegendActivated (*mouseFlag* As Integer, *cancel* As Integer)

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the legend and displaying the Format Legend dialog box.

LegendSelected Event

Description

This event occurs when the user clicks the chart legend.

Syntax

Sub VtChart1_LegendSelected (mouseFlag As Integer, cancel As Integer)

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the legend.

LostFocus Event

Description

This event occurs when the First Impression window loses focus, either by clicking the object or changing the focus in code using the SetFocus method.

Syntax

Sub VtChart1_LostFocus ( )

Remarks

For additional information, refer to the description of the LostFocus event in the Microsoft Visual Basic Language Reference Manual.

MouseDown Event

Description

This event occurs when the user presses a mouse button.

Syntax

Sub VtChart1_MouseDown (*Button* As Integer, *Shift* As Integer, *X* As Single, *Y* As Single)

Remarks

For additional information, refer to the description of the MouseDown event in the Microsoft Visual Basic Language Reference Manual.

MouseMove Event

Description

This event occurs when the user moves the mouse.

Syntax

Sub VtChart1_MouseMove (*Button* As Integer, *Shift* As Integer, *X* As Single, *Y* As Single)

Remarks

For additional information, refer to the description of the MouseMove event in the Microsoft Visual Basic Language Reference Manual.

MouseUp Event

Description

This event occurs when the user releases a mouse button.

Syntax

Sub VtChart1_MouseUp (*Button* As Integer, *Shift* As Integer, *X* As Single, *Y* As Single)

Remarks

For additional information, refer to the description of the MouseUp event in the Microsoft Visual Basic Language Reference Manual.

PlotActivated Event

Description

This event occurs when the user double clicks the chart plot. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Sub VtChart1_PlotActivated (*mouseFlag* As Integer, *cancel* As Integer)

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the plot and displaying the Format Plot dialog box.

PlotSelected Event

Description

This event occurs when the user clicks the chart plot.

Syntax

Sub VtChart1_PlotSelected (*mouseFlag* As Integer, *cancel* As Integer)

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the plot.

PointActivated Event

Description

This event occurs when the user double clicks a data point. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

> **Sub VtChart1_PointActivated** (*series* As Integer, *dataPoint* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

series identifies the series containing the data point. Series are numbered in the order that their columns appear in the data grid, beginning with 1.

dataPoint identifies the data point's position in the series. Points are numbered in the order that their rows appear in the data grid, beginning with 1.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the data point and displaying the Format Data Point dialog box.

Remarks

The following example code shows how to use a PointActivated event to display information about a data point. When the user holds down the Shift key and double-clicks on a selected data point, a message box appears displaying the series number, the data point number, and the data point value.

```
Sub VtChart1_PointActivated (series As Integer, dataPoint As Integer,
mouseFlag As Integer, cancel As Integer)
Dim Flags As Integer
Dim NullFlag As Integer
Dim Cherror As Long
Dim TheData As Double

Flags = mouseFlag And VtChMouseFlagShiftKeyDown
If Flags Then
Form1.vtchart1.column = series
Form1.vtchart1.row = datapoint
TheData = Form1.vtchart1.data
MsgBox "Series:" & Str$(series) & Chr$(10) & " Point:" & Str$(dataPoint)
& Chr$(10) & " Data:" & Str$(TheData)
cancel = True
End If
End Sub
```

PointLabelActivated Event

Description

This event occurs when the user double clicks a data point label. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

> **Sub VtChart1_PointLabelActivated** (*series* As Integer, *dataPoint* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

series identifies the series containing the data point. Series are numbered in the order that their columns appear in the data grid, beginning with 1.

dataPoint identifies the data point's position in the series. Points are numbered in the order that their rows appear in the data grid, beginning with 1.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the data point label and displaying the Format Data Point Label dialog box.

PointLabelSelected Event

Description

This event occurs when the user clicks a data point label.

Syntax

> **VtChart1_PointLabelSelected** (*series* As Integer, *dataPoint* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

series identifies the series containing the data point. Series are numbered in the order that their columns appear in the data grid, beginning with 1.

dataPoint identifies the data point's position in the series. Points are numbered in the order that their rows appear in the data grid, beginning with 1.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the data point label.

PointSelected Event

Description

This event occurs when the user clicks a data point.

Syntax

> **Sub VtChart1_PointSelected** (*series* As Integer, *dataPoint* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

series identifies the series containing the data point. Series are numbered in the order that their columns appear in the data grid, beginning with 1.

dataPoint identifies the data point's position in the series. Points are numbered in the order that their rows appear in the data grid, beginning with 1.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the specified data point.

SeriesActivated Event

Description

This event occurs when the user double clicks a chart series. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Sub VtChart1_SeriesActivated (*series* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

series identifies the series. Series are numbered in the order that their columns appear in the data grid, beginning with 1.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the series and displaying the Format Series dialog box.

SeriesLabelActivated Event

Description

This event occurs when the user double clicks a series label. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Sub VtChart1_SeriesLabelActivated (*series* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

series identifies the series. Series are numbered in the order that their columns appear in the data grid, beginning with 1.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the series label and displaying the Format Series Label dialog box.

SeriesLabelSelected Event

Description

This event occurs when the user clicks a series label.

Syntax

Sub VtChart1_SeriesLabelSelected (*series* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

series identifies the series. Series are numbered in the order that their columns appear in the data grid, beginning with 1.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the series label.

SeriesSelected Event

Description

This event occurs when the user clicks a chart series.

Syntax

Sub VtChart1_SeriesSelected (*series* As Integer, *mouseFlag* As Integer, *cancel* As Integer)

series identifies the series. Series are numbered in the order that their columns appear in the data grid, beginning with 1.

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting all elements that make up the chart series.

TitleActivated Event

Description

This event occurs when the user double clicks the chart title. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Sub VtChart1_TitleActivated (*mouseFlag* As Integer, *cancel* As Integer)

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument.

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the title and displaying the Format Title dialog box.

TitleSelected Event

Description

This event occurs when the user clicks the chart title.

Syntax

Sub VtChart1_TitleSelected (*mouseFlag* As Integer, *cancel* As Integer)

mouseFlag indicates if another key is held down when the mouse button is clicked.

The following table lists the valid constant for this argument.

Constant	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.

Set *cancel* equal to True to prevent the user from selecting the chart title.

Attributes Collection

View Object Model Example Applies To

Description

A group of chart contours.

Syntax

Attributes

Property	Type	Description
Count	Long	Returns the number of contours in this collection. [*num =] Attributes.***Count*** This is a read-only property. **Part** / **Type** / **Description** *num* / Integer / The number of contours.
Item	Attribute object	A specific contour in this collection. *Attributes.***Item** (*attributeindex*) **Item** is the default property of the **Attributes Collection**. **Part** / **Type** / **Description** *attributeindex* / Integer / The position in the contour list. For more information, see **Item As Attribute Object**.

Method	Description
Add	Adds a contour to the buffered contour list. *Attributes.***Add** (*value*) **Part** / **Type** / **Description** *value* / Double / The contour value.
Remove	Deletes a particular contour from the buffered contour list. *Attributes.***Remove** (*index*) **Part** / **Type** / **Description** *index* / Integer / The position in the contour list.

Attributes Collection Applies To

Elevation Object

Item As Attribute Object

View Object Model **Example** **Applies To**

Description

A chart contour and its appearance.

Syntax

Item (*index*)

Property Quick Pick (click here for fast access to a property)

Brush Pen Text Value

Property	Type	Description
Brush	Brush object	Sets or returns the color and style used to display the chart contour as a band. *Item ( ).***Brush** *[= style]* **Part** — **Description** *style* — The brush style and color. For more information, see **Brush Object**
Pen	Pen object	Describes the color and style used to display the chart contour as a line. *Item ( ).***Pen** *[= style]* **Part** — **Description** *style* — The pen style and color. For more information, see Pen Object.
Text	String	The text used to identify the contour in the legend. *Item ( ).**Text** [= text]* **Part** — **Description** text — The text used to identify the contour.
Value	Double	Sets or returns the contour data value where the contour is displayed. *Item ( ).***Value** *[= num]* **Part** — **Description** *num* — The data value at the contour location.

Item As Attribute Object Applies To

Attributes Collection

Axis Object

View Object Model **Example** **Applies To**

Description

An axis on a chart.

Syntax

Axis (*axisID, index*)

Property Quick Pick (click here for fast access to a property)

AxisGrid	AxisScale	AxisTitle	CategoryScale	DateScale
Intersection	Labels	LabelLevelCount	Pen	Tick
ValueScale				

Part	Description
axisID	A **VtChAxisId** constant that identifies a specific axis. For more information, see **AxisId Constants**
index	Reserved for future use. Identifies the specific axis when there is more than one axis with the same axisID.

Property	Type	Description
AxisGrid	Object	The planar area surrounding a chart axis. *Axis.***AxisGrid** For more information, see **AxisGrid** object.
AxisScale	Object	The scale for a date axis. *Axis.***AxisScale** For more information, see **AxisScale** object.
AxisTitle	Object	An axis title on a chart. *Axis.***AxisTitle** For more information, see **AxisTitle** object.
CategoryScale	Object	The scale for a category axis. *Axis.***CategoryScale** For more information, see **CategoryScale** object.
DateScale	Object	The scale for a date axis. *Axis.***DateScale** For more information, see **DateScale** object.
Intersection	Object	The point at which an axis intersects an intersecting

		axis on a chart. *Axis.***Intersection** For more information, see **Intersection** object.
Labels	Collection	A group of chart axis labels. *Axis.***Labels** For more information, see LabelsCollection.
LabelLevelCount	Integer	Sets or returns the number of levels of labels for a given axis. *Axis.***LabelLevelCount** *[= count]* **Part** — **Description** *count* — The number of labels.
Pen	Pen object	Sets or returns the width and color used to draw an axis on a chart. A*xis.***Pen** For more information, see **Pen** object.
Tick	Object	A marker indicating a division along a chart axis. *Axis.***Tick** For more information, see **Tick** object.
ValueScale	Object	The scale for a value axis. *Axis.***ValueScale** For more information, see **ValueScale** object.

Axis Object Applies To

Plot Object

AxisGrid Object

View Object Model Example Applies To

Description

The planar area surrounding a chart axis.

Syntax

AxisGrid

Property	Type	Description
MajorPen	Pen object	Sets or returns the appearance of the major axis grid lines. *AxisGrid*.**MajorPen** For more information, see **Pen** object**.**
MinorPen	Pen object	Sets or returns the appearance of the minor axis grid lines. *AxisGrid*.**MinorPen** For more information, see **Pen** object.

AxisGrid Object Applies To

Axis Object

AxisScale Object

View Object Model Example Applies To

Description

Controls how chart values are plotted on an axis.

Syntax

AxisScale

Property Quick Pick (click here for fast access to a property)

Hide LogBase PercentBasis Type

Property	Type	Description
Hide	Boolean	Determines whether or not the axis is hidden. AxisScale.**Hide** [= *boolean*] If True, the axis scale, line, ticks and title are hidden. If False, the axis appears.
LogBase	Integer	Sets or returns the logarithm base used to plot chart values on a logarithmic axis. The axis type is controlled by the **Type** property**.** AxisScale.**LogBase** [= *base*] **Part** — **Description** *base* — The logarithm base. The default base is 10. The valid range is 2 to 100.
PercentBasis	String	Sets or returns the type of percentage used to plot chart values on a percent axis.The axis type is controlled by the **Type** property**.** AxisScale.**PercentBasis** [= *type*] **Part** — **Description** *type* — A **VtChPercentAxisBasis** constant used to describe the percentage used to plot percent axis values. For more information, see **PercentAxisBasis Constants.**
Type	Integer	Sets or returns the scale type of an axis. AxisScale.**Type** [= *type*] **Part** — **Description** *type* — A **VtChScaleType** constant describing the axis scale type. For more information, see **ScaleType Constants**.

AxisScale Object Applies To

Axis Object

AxisTitle Object

View Object Model Example Applies To

Description

An axis title on a chart.

Syntax

AxisTitle

Property Quick Pick (click here for fast access to a property)

Backdrop Text TextLayout TextLength Visible
VtFont

Property	Type	Description
Backdrop	Object	A shadow, pattern, or picture behind an axis title. *AxisTitle.***Backdrop** For more information, see **Backdrop** object**.**
Text	String	Sets or returns the text used to display an axis title. *AxisTitle.***Text** [= *text*] The **Text** property is the default property for the **AxisTitle** object**.** **Part** — **Description** *text* — The text used to display the axis title.
TextLayout	Object	Text positioning and orientation of an axis title. *AxisTitle.***TextLayout** For more information, see **TextLayout** object**.**
TextLength	Integer	Returns the length of the axis title. *AxisTitle.***TextLength** [= *size*] **Part** — **Description** size — The title length.
Visible	Boolean	Determines whether axis titles appear on the chart or not. *AxisTitle.***Visible** [= *boolean*] If True, axis titles appear on the chart. If False, axis titles do not appear.
VtFont	VtFont Object	The font used to display an axis title. *AxisTitle.***VtFont** For more information, see **VtFont** object**.**

AxisTitle Object Applies To

Axis Object

Backdrop Object

View Object Model Example View Structure Applies To

Description

A shadow, pattern, or picture behind an object.

Syntax

Backdrop

Property	Type	Description
Frame	Frame object	The appearance of the frame displayed behind an object. *Backdrop*.**Frame** For more information, see **Frame** object**.**
Fill	Fill object	The type and appearance of an object backdrop. *Backdrop*.**Fill** For more information, see **Fill**object.
Shadow	Shadow object	The appearance of a shadow displayed behind an object. *Backdrop*.**Shadow** For more information, see **Shadow** object**.**

Backdrop Object Applies To

Plot Object, DataPointLabel Object, SeriesLabel Object, Footnote Object, Title Object, Legend Object, VtChart Object, Item As Label Object, AxisTitle Object

Bar Object

View Object Model Example Applies To

Description

A bar on 3D bar charts.

Syntax

Bar

Property	Type	Description
Sides	Integer	Sets or returns the number of sides displayed for the bar. *Bar.***Sides** [= *num*] **Part** — **Description** *num* — The number of sides.
TopRatio	Single	Describes the percentage of the base size that is used to draw the top of the bar. *Bar.***TopRatio** [= *baspct*] **Part** — **Description** *baspct* — The base percentage.

Bar Object Applies To

Item As Series Object

Brush Object

View Object Model **Example** **Applies To**

Description

The fill type used to display a chart element.

Syntax

Brush

Property Quick Pick (click here for fast access to a property)

FillColor Index PatternColor Style

Property	Type	Description
FillColor	VtColor object	Sets or returns the brush fill color *Brush*.**FillColor** For more information, see **VtColor** object.
Index	Integer	Sets or returns the pattern or hatch used in the brush if *style* is set to ***VtBrushStylePattern*** or ***VtBrushStyleHatch.*** *Brush*.**Index** [= *num*] **Part** — **Description** *num* — A **VtBrushPattern** constant or **VtBrushHatch** constant describing the brush pattern. For more information, see BrushPatterns Constants. For more information, see **BrushHatches Constants** .
PatternColor	VtColor object	Sets or returns the brush pattern color. *Brush*.**PatternColor** For more information, see **VtColor** object.
Style	Integer	Sets or returns the brush style. *Brush*.**Style** [= *style*] **Part** — **Description** *style* — A **VtBrushStyle** constant describing the brush pattern. For more information, see BrushStyle Constants.

Brush Object Applies To

PlotBase Object, Item As DataPoint Object, Item As Attribute Object, Surface Object , Wall Object, Shadow Object, Fill Object

CategoryScale Object

View Object Model Example Applies To

Description

The scale for a category axis.

Syntax

CategoryScale

Property Quick Pick (click here for fast access to a property)

Auto DivisionsPerLabel DivisionsPerTick LabelTick

Property	Type	Description
Auto	Boolean	Indicates whether or not the axis is automatically scaled. *CategoryScale*.**Auto** [= *boolean*] If True, the axis is automatically scaled based on the data being charted. If False, the axis is not automatically scaled. Values in **DivisionsPerLabel** and **DivisionsPerTick** are used to determine the scale.
DivisionsPerLabel	Integer	Sets or returns the number of divisions to skip between labels. *CategoryScale*.**DivisionsPerLabel** [= *num*] If this property is set, then the **Auto** property is automatically set to **False**.

Part	Description
num	The number of divisions.

Property	Type	Description
DivisionsPerTick	Integer	Sets or returns the number of divisions to skip between tick marks. *CategoryScale*.**DivisionsPerTick** [= *num*] If this property is set, then the **Auto** property is automatically set to **False**.

Part	Description
num	The number of divisions.

Property	Type	Description
LabelTick	Boolean	Indicates whether category axis labels are centered on an axis tick mark. *CategoryScale*.**LabelTicks** [= *boolean*] If True, labels are centered on a tick mark. If False, labels are centered between two tick marks.

Category Scale Object Applies To

Axis Object

Contour Object

View Object Model Example Applies To

Description

A surface contour on a 3D elevation chart.

Syntax

Contour

Property	Type	Description
DisplayType	Integer	Sets or returns the type of contour displayed on a chart. *Contour*.**DisplayType** [= *type*]

Part	Description
type	A **VtChContourDisplayType** constant that describes the contour. For more information, see **ContourDisplayType Constants**.

Contour Object Applies To

Elevation Object

ContourGradient Object

View Object Model Example Applies To

Description

A gradient contour on a 3D elevation chart.

Syntax

ContourGradient

Property	Type	Description
FromBrushColor	VtColor object	Sets or returns the brush color for the beginning gradient color displayed on a chart. *ContourGradient.***FromBrush** [= *color*] **Part** — **Description** *color* — Describes the brush color. For more information, see **VtColor** object.
ToBrushColor	VtColor object	Sets or returns the brush color for the ending gradient color displayed on a chart. *ContourGradient.***ToBrush** [= *color*] **Part** — **Description** *color* — Describes the brush color. For more information, see **VtColor** object.
FromPenColor	VtColor object	Sets or returns the pen color for the beginning contour line on a chart. *ContourGradient.***FromPen** [= *color*] **Part** — **Description** *color* — Describes the pen color. For more information, see **VtColor** object
ToPenColor	VtColor object	Sets or returns the pen color for the ending contour line on a chart. *ContourGradient.***ToPen** [= *color*] **Part** — **Description** *color* — Describes the pen color. For more information, see **VtColor** object.

ContourGradient Object Applies To

Elevation Object

Coor Object

View Object Model Example View Structure Applies To

Description

The Coor object describes a floating x and y coordinate pair.

Syntax

Coor

Property	Type	Description
X	Single	Sets or returns the X value in a floating coordinate pair. *Coor*.**X** [= *x*]
Y	Single	Sets or returns the Y value in a floating coordinate pair. *Coor*.**Y** [= *y*]

Method	Description
Set	Sets the x and y coordinate values. *Coor*.**Set** (*x,y*)

Part	Type	Description
x,y	Single	The floating x and y coordinate pair.

Coor Object Applies To

SeriesLabel Object, DataPointLabel Object, Rect Object

Coor3 Object

View Object Model Example View Structure Applies To

Description

The Coor3 object describes a floating point x, y, and z coordinate.

Syntax

Coor3

Property	Type	Description
X	Single	Sets or returns the X value for a floating point coordinate. *Coor3.***X** [= *x*]
Y	Single	Sets or returns the Y value for a floating point coordinate. *Coor3.***Y** [= *y*]
Z	Single	Sets or returns the Z value for a floating point coordinate. Coor3.**Z** [= *z*]

Method	Description
Set	Sets the x, y and z coordinate values. *Coor3.***Set** (*x, y,z)*

Part	Type	Description
x, y, z	Single	The x,y, and z coordinate values for a floating point.

Coor3 Object Applies To

Plot Object

DataGrid Object

View Object Model Example Applies To

Description

A chart data grid.

Syntax

DataGrid

Property Quick Pick (click here for fast access to a property)

ColumnCount ColumnLabel ColumnLabelCount CompositeColumnLabel CompositeRowLabel
RowCount RowLabel RowLabelCount

Method Quick Pick (click here for fast access to a method)

DeleteColumns DeleteColumnLabels DeleteRows DeleteRowLabels GetData
InitializeLabels InsertColumns InsertColumnLabels InsertRows InsertRowLabels
MoveData RandomDataFill RandomFillColumns RandomFillRows SetData
SetSize

Property	Type	Description
ColumnCount	Integer	Sets or returns the number of columns in the current data grid. *DataGrid.***ColumnCount** [= *count*]

Part	Description
count	The number of columns.

Property	Type	Description
ColumnLabel	String	Sets or returns the label on a data column in the grid associated with a chart. *DataGrid.***ColumnLabel** (*column, labelIndex*) [= *text*]

Part	Type	Description
column	Integer	Identifies a specific data column. Columns are numbered from left to right beginning with 1. Any columns containing labels are not counted as data columns.
labelIndex	Integer	Identifies a specific label. If more than one level of column labels exist for the column, you must identify one of them. Column labels are numbered from bottom to top beginning at 1.
text	String	The column label text.

Property	Type	Description
ColumnLabelCount	Integer	Sets or returns the number of levels of column labels in the current data grid.

*DataGrid.***ColumnLabelCount** [= *count*]

Part	Description
count	The number of levels of column labels. Increasing the count adds levels to the grid. Decreasing the count deletes levels from the grid. Levels are added or deleted from the top of the grid.

CompositeColumnLabel String

Returns the multi-level label that identifies a column in the data grid.

[*label* =] *DataGrid.***CompositeColumnLabel** (*column*)

Part	Type	Description
column	Integer	Identifies a specific data column.
label	String	The data grid column label. Columns are numbered from left to right beginning with 1. Any columns containing labels are not counted as data columns. This is a read-only property.

CompositeRowLabel String

Returns the multi-level label that identifies a data grid row.

[*label* =]*DataGrid.***CompositeRowLabel** (*row*)

Part	Type	Description
row	Integer	Identifies a specific data row.
label	String	The data grid row label. Rows are numbered from top to bottom beginning with 1. Any rows containing labels are not counted as data rows. This is a read-only property.

RowCount Integer

Sets or returns the number of rows in the current data grid.

*DataGrid.***RowCount** [= *count*]

Part	Description
count	The number of rows.

RowLabel String

Sets or returns the row label in the current data grid.

*DataGrid.***RowLabel** (*row, labelIndex*) [= *text*]

Part	Type	Description
text	String	The text identifying the row label.
row	Integer	Specifies the row. Rows are numbered from top to bottom

			beginning at 1.
		labelIndex Integer	Specifies a specific level of row labels. Row Lables are numbered from left to right beginning at 1.

RowLabelCount — Integer

Sets or returns the number of levels of row labels in the current data grid.

*DataGrid.***RowLabelCount** [= *count*]

Part	Description
count	The number of levels of labels. Increasing the count adds levels to the grid. Decreasing the count deletes levels from the grid. Levels are added or deleted from the top of the grid.

Method	Description

DeleteColumns

Deletes columns of data and their associated labels from the data grid.

*DataGrid.***DeleteColumns** (*column, count)*

Part	Type	Description
column	Integer	Identifies a specific data column. Columns are numbered from left to right beginning with 1.
count	Integer	Specifies the number of columns you want to delete.

DeleteColumnLabels

Deletes levels of labels from the data columns in a data grid.

*DataGrid.***DeleteColumnLabels** (*labelIndex*, *count*)

Part	Type	Description
labelIndex	Integer	Identifies the number of the first level of labels you want to delete. Column label levels are numbered bottom to top, beginning with 1.
count	Integer	Specifies the number of label levels you want to delete. The number of columns being deleted is calculated from the column identified in *labelIndex* up.

DeleteRows

Deletes rows of data from the data grid.

*DataGrid.***DeleteRows** (*row, count)*

Part	Type	Description
row	Integer	Identifies a specific data row. Rows are numbered from top to bottom beginning with 1.
count	Integer	Specifies the number of rows you want to delete.

DeleteRowLabels

Deletes levels of labels and their associated labels from the data rows in a

data grid.

*DataGrid.***DeleteRowLabels** (*labelIndex, count*)

Part	Type	Description
labelIndex	Integer	Identifies the number of the first level of labels you want to delete. Row labels are numbered right to left, beginning with 1.
count	Integer	Specifies the number of label levels you want to delete. Row labels are deleted from the row identified by *labelIndex* to the left.

GetData

Returns the value currently stored in a specific data point in the data grid associated with a chart.

DataGrid.**GetData**

InitializeLabels

Assigns each label in the first level of data grid labels a unique identifier.

*DataGrid.***InitializeLabels**

InsertColumns

Adds one or more data columns to the data grid.

*DataGrid.***InsertColumns** (*column, count*)

Part	Type	Description
column	Integer	Identifies a specific data column. Columns are numbered from left to right beginning with 1.
count	Integer	Specifies the number of columns you want to insert.

InsertColumnLabels

Inserts levels of labels for the data columns in a data grid.

*DataGrid.***InsertColumnLabels** (*labelIndex, count*)

Part	Type	Description
labelIndex	Integer	Identifies the number of the first level of labels you want to insert. Column label levels are numbered bottom to top, beginning with 1.
count	Integer	Specifies the number of label levels you want to insert. The number of columns being inserted is calculated from the column identified in *labelIndex* up.

InsertRows

Adds one or more data rows to the data grid.

*DataGrid.***InsertRows** (*row, count*)

Part	Type	Description
row	Integer	Identifies a specific data row. Rows are numbered

		from top to bottom beginning with 1.
count	Integer	Specifies the number of rows you want to insert. Rows contain null data until you fill them with data.

InsertRowLabels

Inserts levels of labels from the data rows in a data grid.

*DataGrid.***InsertRowLabels** (*labelIndex, count*)

Part	Type	Description
labelIndex	Integer	Identifies the number of the first level of labels you want to insert. Row labels are numbered right to left, beginning with 1.
count	Integer	Specifies the number of label levels you want to insert. Row labels are inserted from the row identified by labelIndex to the left.

MoveData

Allows you to easily move a range of data within a data grid.

*DataGrid.***MoveData** (*top, left, bottom, right, overOffset, downOffset*)

Part	Type	Description
top	Integer	Identifies the first row in the range to move.
left	Integer	Identifies the first column in the range to move
bottom	Integer	Identifies the last row in the range to move
right	Integer	Identifies the last column in the range to move
overOffset	Integer	Identifies the horizontal direction data should be moved. A positive value moves data to the right, a negative value moves data to the left.
downOffset	Integer	Identifies the vertical direction data should be moved. A positive value moves data down, a negative value moves data up.

RandomDataFill

Fills the data grid associated with a specific chart with randomly generated data.

*DataGrid.***RandomDataFill**

RandomFillColumns

Fills a number of datagrid columns with random values.

*DataGrid.***RandomFillColumns** (*column, count*)

Part	Type	Description
column	Integer	Identifies the first column you wish to fill. Columns are numbered from left to right beginning with 1.
count	Integer	Specifies the number of columns you want to fill with random data.

RandomFillRows

Fills a number of datagrid rows with random values.

DataGrid.**RandomFillRows** (*row, count*)

Part	Type	Description
row	Integer	Identifies the first row you wish to fill. Rows are numbered from top to bottom beginning with 1.
count	Integer	Specifies the number of rows you want to fill with random data.

SetData

Sets the value for a specific data point in the data grid associated with a chart.

DataGrid.**SetData**

SetSize

Allows you to resize the number of data columns and rows, as well as the number of levels of column labels and row labels at one time.

DataGrid.**SetSize** (r*owLabelCount, columnLabelCount, dataRowCount, columnLabelCount*)

Part	Type	Description
rowLabelCount	Integer	Sets or returns the number of levels of row labels you want on the data grid.
columnLabelCount	Integer	Sets or returns the number of levels of column labels you want on the data grid.
dataRowCount	Integer	Sets or returns the number of data rows you want on the data grid.
dataColumnCount	Integer	Sets or returns the number of data columns you want on the data grid.

This method can be used in place of **RowCount, ColumnCount**, **RowLabelCount** and **ColumnLabelCount**.

If you reduce the size of the data grid, data in deleted rows or columns is destroyed.

DataGrid Object Applies To

VtChart Object

DataPoints Collection

View Object Model Example Applies To

Description

A group of chart datapoints.

Syntax

DataPoints

Property	Type	Description
Count	Long	Returns the number of data points in this collection. [*num =] DataPoints.***Count** This is a read-only property.

Part	Type	Description
num	Integer	The number of datapoints.

Property	Type	Description
Item	DataPoint object	A specific data point from this collection. *DataPoints*.**Item** (*datapoint*) **Item** is the default property of the DataPoints Collection.

Part	Type	Description
datapoint	Long	A datapoint.

DataPoints Collection Applies To

Item As Series Object

Item As DataPoint Object

View Object Model **Example** **Applies To**

Description

A data point on a chart.

Syntax

Item (*index*)

Property Quick Pick (click here for fast access to a property)

Brush DataPointLabel EdgePen Offset Marker

VtPicture

Method Quick Pick (click here for fast access to a method)

ResetCustom

Property	Type	Description
Brush	Brush object	The color and pattern used to display the data point. *Item ( )* .**Brush** For more information, see **Brush** object**.**
DataPointLabel	DataPointLabel object	A label on a datapoint. *Item ( ).* **DataPointLabel** For more information, see **DataPointLabel object**.
EdgePen	Pen object	The width and color used to draw the edge of a datapoint. *Item ( )* .**EdgePen** For more information, see **Pen** object**.**
Offset	Single	Sets or returns the distance that a data point is offset or pulled away. *Item ( ).***Offset** [= *offset*] **Part** — **Description** *offset* — The offset distance. Offset is measured in inches or centimeters depending upon your default Windows settings.
Marker	Marker object	The marker type used to draw a data point on a chart. *Item ( )* .**Marker** For more information, see **Marker object** If this property is set, then the **Auto** property of the **SeriesMarker** object is

		automatically set to False.
VtPicture	VtPicture object	The graphic that can be displayed as a datapoint. *Item ( )* .**VtPicture** For more information, see **VtPicture object**.

Method	Description
ResetCustom	Resets any custom attributes placed on a data point to the series default. *Item ( )*.**ResetCustom**

Item As DataPoint Object Applies To

DataPoints Collection

DataPointLabel Object

View Object Model **Example** **Applies To**

Description

The label for a data point on a chart.

Syntax

DataPointLabel

Property Quick Pick (click here for fast access to a property)

Backdrop	Component	Custom	LineStyle	LocationType
Offset	PercentFormat	Text	TextLayout	TextLength
ValueFormat	VtFont			

Method Quick Pick (click here for fast access to a method)

ResetCustomLabel	Select

Property	Type	Description
Backdrop	Backdrop object	A shadow, pattern or picture displayed behind a DataPointLabel. *DataPointLabel*.**Backdrop** For more information, see **Backdrop object**.
Component	Integer	Sets or returns the type of label to be used to identify the data point. *DataPointLabel*.**Component** [= *type*] **Part** — **Description** *type* — A **VtChLabelComponent** constant that identifies the label type. For more information, see **LabelComponent Constants**.
Custom	Boolean	Determines if custom text is used to label a data point. *DataPointLabel*.**Custom** [= *boolean*] If True, the label contains custom text. If False, the information specified by Components is used to label the data point.
LineStyle	Integer	Sets or returns the type of line used to connect a data point to a label. *DataPointLabel*.**LineStyle** [= *type*] **Part** — **Description** *type* — A **VtChLabelLineStyle** constant identifying the connecting line. For more information, see **LabelLineStyle Constants**.

LocationType | Integer

Sets or returns the standard position used to display the label.

DataPointLabel.**LocationType [=** *type***]**

Part	Description
type	A **VtChLabelLocationType** constant identifying label position. For more information, see **LabelLocationType Constants**.

Offset | Coor object

The distance that a data point label is offset or pulled away from one of the predefined (standard) label positions. The offset is added to the position calculated for the point based on the **LocationType** setting**.**

DataPointLabel.**Offset** [= *offset*]

For more information, see **Coor** object**.**

Part	Description
offset	The offset distance.

PercentFormat | String

Sets or returns a string that describes the format used to display the label as a percent. Use **Component** to change the label type.

DataPointLabel.**PercentFormat** [= *format*]

Part	Description
format	Describes the format used to display a the label as a percent.

The following table lists several

Examples of percentage format strings. The values listed at left are the valid formats.

	3	-3	.3
0%	300%	-300%	30%
0.0%	300.0%	-300.0%	30.0%
0.00%	300.00%	-300.00%	30.00%

Text | String

Sets or returns the text used to display a datapoint label.

DataPointLabel.**Text** [= *text*]

Text is the default property of the **DataPointLabel** object

Part	Description
text	Describes the datapoint label text.

TextLayout | Object

The position and orientation of data point label text.

*DataPointLabel.***TextLayout**

For more information, see **TextLayout** object**.**

TextLength — Integer

Returns the length of the data point label.

[*size* =] *DataPointLabel.***TextLength**

Part	Description
size	The length of the data point label text.

ValueFormat — String

Sets or returns the format used to display the label as a value. Use **Component** to change the label type.

*DataPointLabel.***ValueFormat** [= *format*]

Part	Description
format	Describes the format used to display the label as a value. For examples of valid format strings, see Value Format Strings.

VtFont — VtFont object

The font used to display data point label text.

*DataPointLabel.***VtFont**

For more information, see **VtFont** object**.**

Method	Description
ResetCustomLabel	Resets any custom attributes placed on a data point label to the series default *DataPointLabel.***ResetCustomLabel**
Select	Selects the specified chart datapoint label. *DataPointLabel.***Select**

Value Format Strings

The following table lists some examples of format strings that can be applied to values.

Examples	Format	3	-3	.3
All	General	3	-3	.3
Fixed	0	3	-3	0
	0.00	3.00	-3.00	0.30
	#,##0	3	-3	0
	#,##0.00	3.00	-3.00	0.30
	#,##0_);(#,##0)	3	(3)	0
	#,##0_);[RED](#,##0)	3	(3) in red	0
	#,##0.00_);(#,##0.00)	3.00	(3.00)	0.30
	#,##0.00_);[RED](#,##0.00)	3.00	(3.00) in red	0.30
Currency	$#,##0_);($#,##0)	$3	($3)	$0
	$#,##0_);[RED]($#,##0)	$3	($3) in red	$0
	$#,##0.00_);($#,##0.00)	$3.00	($3.00)	$0.30
	$#,##0.00_);[RED]$(#,##0.00)	$3.00	($3.00) in red	$0.30
Fraction	# ?/?	3	-3	2/7
	# ??/??	3	-3	3/10
Scientific	0.00E+00	3.00E+00	-3.00E+00	3.00E-01
	##0.0E+0	3.0E+0	-3.0E+0	3.0E-01

DataPointLabel Object Applies To

Item As DataPoint Object

DateScale Object

View Object Model Example Applies To

Description

The scale for a date axis.

Syntax

DateScale

Property Quick Pick (click here for fast access to a property)

Auto MajFreq MajInt Maximum Minimum
MinFreq MinInt SkipWeekend

Property	Type	Description
Auto	Boolean	Determines whether or not the date scale for a chart axis is automatically scaled. *DateScale*.**Auto** [= *boolean*] If True, the axis is automatically scaled based on the data being charted. If False, the axis is not automatically scaled. Values in **MajInt, MinInt, MajFreq** and **MinFreq** are used to determine the scale.
MajFreq	Integer	Sets or returns the number of major intervals that pass before a label is placed on the axis. *DateScale*.**MajFreq** [= *num*] If this property is set, then the **Auto** property is automatically set to **False**.

Part	Description
num	The number of intervals that pass before a label is placed on the axis. For Example, if the MajInt constant **VtChDateIntervalType** is set to monthly, but you only want every third month to display, you would set this argument to 3.

Property	Type	Description
MajInt	Integer	Sets or returns the type of interval used to display dates on an axis. Major grid lines are drawn at the major interval. *DateScale*.**MajInt** [= *type*] If this property is set, then the **Auto** property is automatically set to **False**.

Part	Description
type	A **VtChDateIntervalType** constant describing the interval used to display the dates on an axis where the major ticks and grid lines appear. For more information , see **DateIntervalType Constants**.

Property	Type	Description
Maximum	Double	Sets or returns the highest or ending value on the chart date axis.

DateScale.**Maximum** [= *value*]

If this property is set, then the **Auto** property is automatically set to **False**.

Part	Description
value	Ending date on the date axis.

Minimum	Double	Sets or returns the lowest or beginning value on the chart date axis. *DateScale*.**Minimum** [= *value*] If this property is set, then the **Auto** property is automatically set to **False**.

Part	Description
value	Beginning date on the date axis.

MinFreq	Integer	Sets or returns the number of intervals that pass before a label is placed on the axis. *DateScale*.**MinFreq** [= *num*] If this property is set, then the **Auto** property is automatically set to **False**.

Part	Description
num	The number of intervals that pass before a label is placed on the axis.

If the MinInt property constant **DateIntervalType Constants** is set to monthly, but you only want every third month to display, you would set this argument to 3.

MinInt	Integer	Sets or returns the type of interval used to display dates on an axis. Minor grid lines are drawn at the minor interval. *DateScale*.**MinInt** [= *type*] If this property is set, then the **Auto** property is automatically set to **False**.

Part	Description
type	A **VtChDateIntervalType** constant describing the interval used to display the dates on an axis where the minor ticks and grid lines appear. For more information, see **VtChDateIntervalType**.

SkipWeekend	Boolean	Determines whether or not dates that fall on the weekend are displayed on the chart. *DateScale*.**SkipWeekend** [= *boolean*] If True, weekend dates are not displayed on the chart. If False, weekend dates are displayed.

DateScale Object Applies To

Axis Object

Doughnut Object

View Object Model Example Applies To

Description

A doughnut chart type.

Syntax

Doughnut

Property	Type	Description
Sides	Integer	Sets or returns the number of sides displayed for a doughnut. *Doughnut*.**Sides** [= *num*] **Part** — **Description** *num* — The number of doughnut sides. Valid values are 1 to 360. If you use 1, First Impression determines howmany sides are needed to give the doughnut a round appearance based on the current division. Entering large numbers may impact performance.
InteriorRatio	Single	Sets or returns the ratio of the radius of the doughnut that is used to display the interior "hole" in the doughnut. *Doughnut*.**InteriorRatio** [= *ratio*] **Part** — **Description** *ratio* — The radius ratio. Valid values are .01 to 1.0.

Method	Description
Set	Sets the attributes for a doughnut chart. *Doughnut*.**Set**

Doughnut Object Applies To

Plot Object

Elevation Object

View Object Model **Example** **Applies To**

Description

An elevation chart.

Syntax

Elevation

Property Quick Pick (click here for fast access to a property)

Attributes AutoValues ColorType ColSmoothing Contour
ContourGradient RowSmoothing SeparateContourData Surface

Property	Type	Description
Attributes	Collection	A group of chart contour attributes. *Elevation.***Attribute**s For more information, see Attributes_Collection.
AutoValues	Boolean	Determines whether or not the values used to display chart contours are calculated automatically or if custom contours are displayed. *Elevation.***AutoValues** [= *values*] If **AutoValues** is True, each major axis division represents a separate contour. If False, any custom contours you have specified are unchanged.
ColorType	Integer	Sets or returns the color type used to display chart contours. *Elevation.***ColorType** [= *type*] **Part** / **Description** *type* — A **VtChContourColorType** constant describing the contour color. For more information, see **ContourColorType Constants**.
ColSmoothing	Integer	Sets or returns the smoothing factor to be applied to columns. *Elevation.***ColSmoothing** [= *factor*] **Part** / **Description** *factor* — The smoothing factor.
Contour	Object	A surface contour on a 3D elevation chart. *Elevation.***Contour** For more information, see **Contour *object*.**

ContourGradient	Object	A gradient contour or line on a 3D elevation chart. *Elevation.***ContourGradient** For more information, see **ContourGradient object**
RowSmoothing	Integer	Sets or returns to the smoothing factor to be applied to rows. *Elevation.***RowSmoothing** [= *factor*] **Part** **Description** *factor* The smoothing factor. The table below lists some Examples of valid smoothing factors.
SeparateContourData	Boolean	Specifies whether or not the data grid contains separate elevation and contouring data. *Elevation.***SeparateContourData** [= *boolean*] If True, the first half of the data (divided vertically between columns) contains elevation data and the second half contains contour data. If False, the entire data grid contains elevation data.
Surface	Object	A surface chart. *Elevation.****Surface*** For more information, see **Surface** object**.**

The table below lists some

Examples of valid smoothing factors.

Smoothing Factor	Description
0	The raw grid data is used with no smoothing.
1	This factor samples the spline data only at the original data grid locations.
2-32	A factor of 2 or more will break the row and column locations into the specified number of subdivisions. Since smoothing occurs for both rows and columns, a smoothing factor of 2 breaks a surface patch into 4 subpatches while a smoothing factor of 4 breaks a patch into 16 subpatches. Higher smoothing factors may slow drawing speed significantly.

Elevation Object Applies To

Plot Object

Fill Object

View Object Model Example Applies To

Description

The Fill object describes the type and appearance of an object's backdrop.

Syntax

Fill

Property	Type	Description
Brush	Brush object	The fill type used to draw an object. *Fill.***Brush** For more information, see **Brush** object**.**
Gradient	Gradient object	The type of gradient and color used to fill an object's backdrop. *Fill.***Gradient** For more information, see **Gradient** object**.** For more information, see **VtPicture** object**.**
Style	Integer	Sets or returns the type of fill. *Fill.***Style** [= *type]* **Part** — **Description** *type* — A **VtFillStyle** constant that describes the style of fill. A fill can have a brush, which is a solid color or patterned fill; or a gradient, which blends two colors to make up the fill. You can also use a graphic as a fill for some objects. The graphic can be placed on top of a brush or gradient fill. For more information, see **FillStyleConstants**.
VtPicture	VtPicture object	A graphic that can be displayed as an object backdrop. *Fill.***VtPicture** For more information, see **VtPicture object**.

Fill Object Applies To

Backdrop Object

Footnote Object

View Object Model Example Applies To

Description

Descriptive text that appears beneath a chart.

Syntax

Footnote

Property Quick Pick (click here for fast access to a property)

Backdrop Location Text TextLayout TextLength VtFont

Method Quick Pick (click here for fast access to a method)

Select

Property	Type	Description
Backdrop	Backdrop object	A shadow, pattern or picture displayed behind a chart footnote. *Footnote.***Backdrop** For more information, see **Backdrop** object**.**
Location	Location object	Sets or returns the current position of footnote text. *Footnote.***Location** For more information, see **Location** object**.**
Text	String	Sets or returns the text used to display the footnote. *Footnote.***Text** [= *text*] The **Text** property is the default property for the **Footnote** object. **Part** — **Description** text — The footnote text.
TextLayout	TextLayout object	Footnote text positioning and orientation. *Footnote.***TextLayout** For more information, see **TextLayout** object**.**
TextLength	Integer	Returns the length of the footnote text. [*size* =]*Footnote.***TextLength** **Part** — **Description** size — The length of the footnote text.

VtFont	VtFont object	The font used to display footnote text. *Footnote*.**VtFont** For more information, see **VtFont** object**.**

Method	Description
Select	Selects the specified chart footnote. *Footnote*.**Select**

Footnote Object Applies To

VtChart Object

Frame Object

View Object Model Example Applies To

Description

The Frame object holds information about the appearance of the frame around an object.

Syntax

Frame

Property	Type	Description
Style	Integer	Sets or returns the frame style. *Frame*.**Style** [= *type*] **Part** — **Description** *type* — A **VtFrameStyle** constant that describes the type of frame**.** For more information, see **FrameStyle Constants** .
Width	Single	Sets or returns the width of the frame, in points. *Frame*.**Width** [= *width*] **Part** — **Description** *width* — The width of the frame in points.
FrameColor	VtColor object	Sets or returns the frame color. *Frame*.**FrameColor** For more information, see **VtColor** object**.**
SpaceColor	VtColor object	Sets or returns the color of the space between frames. *Frame*.**SpaceColor** For more information, see **VtColor** object.

Frame Object Applies To

Backdrop Object

Gradient Object

View Object Model Example Applies To

Description

The Gradient object holds information about type of gradient used to fill an object. It also holds the colors used to make the gradient.

Syntax

Gradient

Property	Type	Description
FromColor	VtColor object	The color of the beginning gradient band. *Gradient.***FromColor** For more information, see **VtColor** object**.**
Style	Integer	Sets or returns the gradient style. *Gradient.***Style** [= *type*] **Part** / **Description** *type* — A **VtGradientStyle** constant that describes the type of gradient. For more information, see **GradientStyle Constants**.
ToColor	VtColor object	The color of the ending gradient band. *Gradient.***ToColor** For more information, see **VtColor** object.

Gradient Object Applies To

Fill Object

HiLo Object

View Object Model Example Applies To

Description

A hi-lo chart type.

Syntax

HiLo

Property	Type	Description
GainColor	VtColor object	The color used to indicate a gain in value for a series on a hi-lo chart. *HiLo*.**GainColor** For more information, see **VtColor** object**.**
LossColor	VtColor object	The color used to indicate a loss in value for a series on a hi-lo chart. *HiLo*.**LossColor** For more information, see **VtColor** object.

HiLo Object Applies To

Item As Series Object

Intersection Object

View Object Model Example Applies To

Description

The point at which an axis intersects an intersecting axis on a chart.

Syntax

Intersection

Property	Type	Description
Auto	Boolean	Determines whether or not the Intersection object uses the value of **Point** *t*o position the axis. *Intersection.***Auto** [= *boolean*] If True, the axis is positioned in its standard location. If False, the intersecting axis is positioned at the value indicated by **Point.**
AxisId	Integer	Returns a specific axis that intersects with the current axis. [*intersectId* =] *Intersection.***AxisId** **Part** — **Description** *intersectId* — Identifies the intersecting axis. This is a read-only property.
Index	Integer	Returns which Axis intersects another axis when there is more than one axis with the same id. [*index* =] *Intersection*.**Index** **Part** — **Description** *index* — The index of the intersecting axis. Currently, 1 is the only valid value for this argument. This is a read-only property.
LabelsInsidePlot	Boolean	Determines whether or not to leave the axis labels at the normal location or move them with the axis to the new intersection point. *Intersection*.**LabelsInsidePlot** [= *boolean*] If True, the axis labels remain at the normal location. If False, the labels move inside the plot to the new intersection point. If this property is set, then the **Auto** property is automatically set to **False**.
Point	Double	Sets or returns the point where the current axis intersects with another axis. *Intersection.***Point** [= *point*] If this property is set, then the **Auto** property is automatically set to **False**. **Part** — **Description** *point* — The point on an axis where the current axis intersects.

Intersection Object Applies To

Axis Object

Labels Collection

View Object Model Example Applies To

Description

A group of chart axis labels.

Syntax

Labels

Property	Type	Description
Count	Long	Returns a specific level of labels in this collection. [*num* =] *Labels*.**Count** This is a read-only property. **Part** / **Type** / **Description** *num* / Integer / The number of levels of labels in this collection.
Item	Label object	Returns a specific label in this collection. *Labels*.**Item** (*labelIndex*) **Item** is the default property of the **Labels Collection.** **Part** / **Type** / **Description** *labelIndex* / Integer / An axis label. This is a read-only property. For more information, see **Item As Label**.

Labels Collection Applies To

Axis Object

Item As Label Object

View Object Model **Example** **Applies To**

Description

A chart axis label.

Syntax

Item (*index*)

Property Quick Pick (click here for fast access to a property)

Auto Backdrop Format FormatLength Standing

TextLayout VtFont

Property	Type	Description
Auto	Boolean	Determines whether or not axis labels are automatically rotated to improve the chart layout. *Item ( )*.**Auto** [= *boolean*] If True, labels may be rotated. If False, the labels are not rotated. Long labels may not display properly.
Backdrop	Backdrop object	A shadow, pattern, or picture behind an axis label. *Item ( )*.**Backdrop** For more information, see **Backdrop object**.
Format	String	Sets or returns the characters that define the format used to display the axis label. *Item ( )*.**Format** [= *format*] Part: *format* — Description: Defines the format used to display the axis label. For more information, about defining label formats, see the following: Number Format String Examples Date and Time Format Examples Custom Format Examples.
FormatLength	String	Reurns the length of the format string. [*length* =] *Item ( )*.**FormatLength** Part: *length* — Description: The axis label text string length. This is a read-only property.
Standing	Boolean	Specifies whether axis labels are displayed laying down in the X or Z plane or rotated up on the text baseline to stand in the Y plane.

		Item ( ).**Standing** [= *boolean*] If True, axis labels are displayed rotated up on the text baseline to stand in the Y plane. If False, axis labels lay down in the X or Z plane.
TextLayout	TextLayout object	Sets or returns the positioning and orientation of axis label text. *Item ( )*.**TextLayout** For more information, see **TextLayout object**. If this property is set, then the **Auto** property is automatically set to **False**.
VtFont	VtFont object	The font used to display a chart axis label. *Item ( )*.**Font** For more information, see **VtFont object**.

Number Format String Examples

The following table lists the valid characters for the **Format** property and the results if the format is applied to a positive, negative and decimal number:

Category	Format	3	-3	.3
All	General	3	-3	.3
Fixed	0	3	-3	0
	0.00	3.00	-3.00	0.30
	#,##0	3	-3	0
	#,##0.00	3.00	-3.00	0.30
	#,##0_); (#,##0)	3	(3)	0
	#,##0_);[RED](#,##0)	3	(3) in red	0
	#,##0.00_);(#,##0.00)	3.00	(3.00)	0.30
	#,##0.00_);[RED](#,##0.00)	3.00	(3.00) in red	0.30
Currency	$#,##0_);($#,##0)	$3	($3)	$0
	$#,##0_);[RED]($#,##0)	$3	($3) in red	$0
	$#,##0.00_);($#,##0.00)	$3.00	($3.00)	$0.30
	$#,##0.00_);[RED]$(#,##0.00)	$3.00	($3.00) in red	$0.30
Percentage	0%	300%	-300%	30%
	0.0%	300.0%	-300.0%	30.0%
	0.00%	300.00%	-300.00%	30.00%
Fraction	# ?/?	3	-3	2/7
	# ??/??	3	-3	3/10
Scientific	0.00E+00	3.00E+00	-3.00E+00	3.00E-01
	##0.0E+0	3.0E+0	-3.0E+0	3.0E-01

Date and Time Format Examples

The following table lists the built-in date and time formats and the result of applying the format:

Category	Format	Result
Date	m/d/y	2/3/94
	d-mmm-yy	3-Feb-94
	d-mmm	3-Feb
	mmm-yy	Feb-94
	m/d/y h:mm	2/3/94 6:15
Time	h:mm AM/PM	3:29 PM
	h:mm:ss AM/PM	3:29:41 PM
	h:mm	15:29
	h:mm:ss	15:29:41
	mm:ss	29:41
	[h]:mm:ss	1:09:33

Custom Format Examples

You can also create a custom format. The following table lists the format symbols that can be used in a custom format string:

Format Symbol	Description
General	Displays the number in General format.
0	Digit placeholder. If the number contains fewer digits than the format contains placeholders, the number is padded with 0's. If there are more digits to the right of the decimal than there are placeholders, the decimal portion is rounded to the number of places specified by the placeholders. If there are more digits to the left of the decimal than there are placeholders, the extra digits are retained.
#	Digit placeholder. This placeholder functions the same as the 0 placeholder except the number is not padded with 0's if the number contains fewer digits than the format contains placeholders.
?	Digit placeholder. This placeholder functions the same as the 0 placeholder except that spaces are used to pad the digits.
. (period)	Decimal point. Determines how many digits (0's or #'s) are displayed on either side of the decimal point. If the format contains only #'s left of the decimal point, numbers less than 1 begin with a decimal point. If the format contains 0's left of the decimal point, numbers less than 1 begin with a 0 left of the decimal point.
%	Displays the number as a percentage. The number is multiplied by 100 and the % character is appended.
, (comma)	Thousands separator. If the format contains commas separated by #'s or 0's, the number is displayed with commas separating thousands. A comma following a placeholder scales the number by a thousand. For Example, the format 0, scales the number by 1000 (e.g., 10,000 would be displayed as 10).
E- E+ e- e+	Displays the number as scientific notation. If the format contains a scientific notation symbol to the left of a 0 or # placeholder, the number is displayed in scientific notation and an E or an e is added. The number of 0 and # placeholders to the right of the decimal determines the number of digits in the exponent. E- and e- place a minus sign by negative exponents. E+ and e+ place a minus sign by negative exponents and a plus sign by positive exponents.
$ - + / ( ) : space	Displays that character. To display a character other than those listed, precede the character with a back slash (\) or enclose the character in double quotation marks (" "). You can also use the slash (/) for fraction formats.
\	Displays the next character. The backslash is not displayed. You can also display a character or string of characters by surrounding the characters with double quotation marks (" ").
* (asterisk)	Repeats the next character until the width of the column is filled. You cannot have more than one asterisk in each format section.
[ ]	Displays hours greater than 24, or minutes or seconds greater than 60. Place the brackets around the leftmost part of the time code; for Example, [h]:mm:ss would allow the display of hours greater than 24.
"text"	Displays the text inside the quotation marks.
@	Text placeholder. Text replaces the @ format character.
m	Month number. Displays the month as digits without leading zeros (e.g., 1-

	12). Can also represent minutes when used with h or hh formats.
mm	Month number. Displays the month as digits with leading zeros (e.g., 01-12). Can also represent minutes when used with the h or hh formats.
mmm	Month abbreviation. Displays the month as an abbreviation (e.g., Jan-Dec).
mmmm	Month name. Displays the month as a full name (e.g., January-December).
d	Day number. Displays the day as digits with no leading zero (e.g., 1-9).
dd	Day number. Displays the day as digits with leading zeros (e.g., 01-31).
ddd	Day abbreviation. Displays the day as an abbreviation (e.g., Sun-Sat).
dddd	Day name. Displays the day as a full name (e.g., Sunday-Saturday).
yy	Year number. Displays the year as a two-digit number (e.g., 00-99).
yyyy	Year number. Displays the year as a four-digit number (e.g., 1900-2078).
h	Hour number. Displays the hour as a number without leading zeros (1-23). If the format contains one of the AM or PM formats, the hour is based on a 12-hour clock. Otherwise, it is based on a 24-hour clock.
hh	Hour number. Displays the hour as a number with leading zeros (01-23). If the format contains one of the AM or PM formats, the hour is based on a 12-hour clock. Otherwise, it is based on a 24-hour clock.
m	Minute number. Displays the minute as a number without leading zeros (0-59). The m format must appear immediately after the h or hh symbol. Otherwise, it is interpreted as a month number.
mm	Minute number. Displays the minute as a number with leading zeros (00-59). The mm format must appear immediately after the h or hh symbol. Otherwise, it is interpreted as a month number.
s	Second number. Displays the second as a number without leading zeros (0-59).
ss	Second number. Displays the second as a number with leading zeros (00-59).
AM/PM am/pm A/P a/p	12-hour time. Displays time using a 12-hour clock. Displays AM, am, A, or a for times between midnight and noon; displays PM, pm, P, or p for times from noon until midnight.
[BLACK]	Displays text in black.
[BLUE]	Displays text in blue.
[CYAN]	Displays text in cyan.
[GREEN]	Displays text in green.
[MAGENTA]	Displays text in magenta.
[RED]	Displays text in red.
[WHITE]	Displays text in white.
[YELLOW]	Displays text in yellow.
[COLOR n]	Displays text using the corresponding color in the color palette. n is an index to a color in the color palette.

Item As Label Object Applies To

Labels Collection

LCoor Object

View Object Model

Description

The LCoor object describes a long integer x and y coordinate pair.

Syntax

LCoor

Property	Type	Description
X	Long	Sets or returns a long integer X coordinate value. *LCoor.***X** [= *x*]
Y	Long	Sets or returns a long integer Y coordinate value. LCoor.**Y** [= *y*]

Method	Description
Set	Sets the X and Y coordinate value for the LCoor object. *LCoor.***Set** (*x,y*)

Part	Type	Description
x,y	Long	The x and y value for a long integer coordinate pair.

Legend Object

View Object Model Example Applies To

Description

The graphical key and accompanying text that describes the chart series.

Syntax

Legend

Property	Type	Description
Backdrop	Backdrop object	A shadow, pattern or picture displayed behind a chart legend. *Legend.***Backdrop** For more information, see **Backdrop** object**.**
Location	Location object	Sets or returns the current position of legend text. *Legend.***Location** For more information, see **Location** object**.**
TextLayout	TextLayout object	Legend text positioning and orientation. *Legend*.**TextLayout** For more information, see **TextLayout** object.
VtFont	VtFont object	The font used to display a chart legend. *Legend.***Font** For more information, see **VtFont** object**.**

Method	Description
Select	Selects the specified chart legend. *Legend*.**Select**

Legend Object Applies To

VtChart Object

Light Object

View Object Model Example Applies To

Description

The light source illuminating a 3D chart.

Syntax

Light

Property	Type	Description
AmbientIntensity	Single	Sets or returns the percentage of ambient light illuminating a 3D chart Valid values are 0.1 to 1.0 (100 %). If set to 100, all sides of the chart elements are fully illuminated no matter what light sources are turned on. If set at 0, there is no contribution from ambient light, only the sides of the chart elements facing active light sources are illuminated. *Light.***AmbientIntensity** [= *intensity*] **Part** — **Description** *intensity* — The chart light intensity.
EdgeIntensity	Single	Sets or returns the intensity of light used to draw the edges of objects in a 3D chart. Valid values are 0.1 (0) to 1.0 (100). An intensity of 0 draws the edgesas black lines and an intensity of 100 fully illuminates the edges using the element's pen color. *Light.***EdgeIntensity** [= *edgeint*] If this property is set, then the **EdgeVisible** property is automatically set to **True**. **Part** — **Description** *edgeint* — The edge light intensity.
EdgeVisible	Boolean	Determines whether or not edges are displayed on the elements in a 3D chart. *Light*.**EdgeVisible** [= *boolean*] If True, edges are visible. If False, edges are not displayed on elements in the 3D chart.
LightSources	Collection	A group of light sources.

Light Object Applies To

Plot Object

LightSources Collection

View Object Model Example Applies To

Description

A group of chart light sources.

Syntax

LightSources

Property	Type	Description
Count	Long	Returns the number of light sources in this collection. [*num* =] *LightSources*.**Count** This is a read-only property.

Part	Type	Description
num	Integer	The number of light sources.

Item	LightSource object	Sets or returns a specific light source from this collection. *LightSources*.**Item** (*lightindex*) **Item** is the default property of the **LightSources Collection.**

Part	Type	Description
lightindex	Integer	The position in the list of light sources. For more information, see **Item As LightSource Object**.

Method	Description
Add	Adds a light source to this collection. *LightSources*.**Add** (*x,y,z,Intensity)* Note If x,y and z are all set to zero, this will generate a VtChInvalidArgument error.

Part	Type	Description
x,y,z	Integer	The light source location.
Intensity	Single	The light source intensity.

Remove	Removes a light source from this collection. *LightSources*.**Remove** (*index*)

Part	Type	Description
index	Integer	A specific light source by position in the list of light sources..

LightSources Collection Applies To

Light Object

Item As LightSource Object

View Object Model Example Applies To

Description

The light source used to illuminate the elements in a 3D chart.

Syntax

Item (*LightIndex*)

Part	Description
LightIndex	A number identifying the light source.

Property	Type	Description
X	Single	The x coordinate for the light source location. *Item*.**x**
Y	Single	The y coordinate of the light source location *Item*.**y**
Z	Single	The z coordinate of the light source location. *Item*.**z**
Intensity	Single	Sets or returns the strength of the light coming from the light source. *Item*.**Intensity** [= *strength*] **Intensity** is the default property of the **LightSource** object.

Part	Type	Description
strength	Single	The light strength. If the intensity is set to 100, chart surfaces facing the light source are fully illuminated. If the light is set at 50, these surfaces receive 50 percent illumination from this light. Valid range is 0 to 100.

Method	Description
Set (x,y,z, Intensity)	Sets the X,Y, and Z coordinates and the intensity for the LightSource location. *Item*.**Set**

The following table lists the x,y,z coordinates of the nine predefined light positions that you can select using the 3D lighting tab of the Format Plot dialog box.

Position	X	Y	Z
Center	0	0	1
Left	-3	0	2
Right	3	0	2
High Center	0	-1	2
High Left	-3	-1	2

High Right	3	-1	2
Low Center	0	1	2
Low Left	-3	1	2
Low Right	3	1	2

Item As LightSource Object Applies To

LightSources Collection

Location Object

View Object Model Example Applies To

Description

The current position of object text.

Syntax

Location

Property	Type	Description
LocationType	Integer	Sets or returns the location of the text. *Location.***LocationType** [= *loc*] If this property is set, then the **Visible** property is automatically set to **True**. **Part** — **Description** *loc* — A **VtChLabelLocationType** constant describing the location of text. For more information, see **LabelLocationType Constants**.
Rect	Rect object	The location coordinates for the text. *Location.***Rect** This property is ignored unless **LocationType** is set to VtChLabelLocationTypeCustom. If this property is set, then the **LocationType** property is automatically set to **Custom**.
Visible	Boolean	Determines if text is visible. *Location*.**Visible** [= *boolean*] If True, the text is visible. If False, the text does not appear.

Location Object Applies To

Footnote Object, Title Object, Legend Object

LRect Object

View Object Model

Description

The LRect object defines a rectangle in long integer coordinates.

Syntax

LRect

Property	Type	Description
Max	Coor object	A long integer specifying the ending corner of the rectangle. *LRect*.**Max** For more information, see **Coor** object.
Min	Coor object	A long integer specifying the starting corner of the rectangle. *LRect*.**Min** For more information, see **Coor** object.

Marker Object

View Object Model Example View Structure Applies To

Description

A marker that identifies a data point on a chart.

Syntax

Marker

Property	Type	Description
FillColor	VtColor object	The color used to fill the marker. *Marker.**FillColor*** For more information, see **VtColor object.** If this property is set, then the **Visible** property is automatically set to **True**.
Pen	Pen object	The width used to draw the marker. *Marker.***Pen** For more information, see **Pen** object**.** If this property is set, then the **Visible** property is automatically set to **True**.
Size	Single	Size of the marker in points. *Marker.***Size** [= *size*] If this property is set, then the **Visible** property is automatically set to **True**. **Part** — **Description** size — The marker width.
Style	Integer	The marker style. *Marker.***Style** [= *type*] If this property is set, then the **Visible** property is automatically set to **True**. **Part** — **Description** *type* — A **VtMarkerStyle** constant that lists the marker type. For more information, see **MarkerStyle Constants**.
Visible	Boolean	Indicates whether markers are displayed. *Marker.***Visible** [= *boolean*] If True, markers are visible. If False, markers do not display.
VtPicture	VtPicture object	The graphic used to display the marker. *Marker.***VtPicture**

For more information, see **VtPicture** object**.**

If this property is set, then the **Visible** property is automatically set to **True**.

Marker Object Applies To

Item As DataPoint Object

Pen Object

View Object Model **Example** **Applies To**

Description

Describes the color and pattern of lines or edges on a chart.

Syntax

Pen

Property Quick Pick (click here for fast access to a property)

Cap Join Limit Style Width VtColor

Property	Type	Description
Cap	Integer	Sets or returns the how line ends are capped. *Pen.***Cap** [= *type*] **Part** / **Description** *type* — A **VtPenCap** constant that describes the line pen cap style. For more information, see **PenCap Constants**.
Join	Integer	Sets or returns how line segments are formed. *Pen.***Join** [= *type*] **Part** / **Description** *type* — A **VtPenJoin** constant that describes the style of pen join. For more information, see PenJoin Constants.
Limit	Single	Sets or returns the joint limit, in points, of the line. *Pen.***Limit** [= *joint*] **Part** / **Description** *joint* — A joint limit as a multiple of the line width. If two lines meet at a sharp angle, a mitered join results in a corner point that extends ar beyond the actual corner. If the distance from the inner join point to the outer join point exceeds the value in this variable, the join automatically changes to a bevel.
Style	Integer	Sets or returns the pen style used to draw the line. *Pen.***Style** [= *type*] **Part** / **Description** *type* — A **VtPenStyle** constant that describes the style of pen. For more information, see **PenStyle Constants**.

Width	Single	Sets or returns the pen width in points. *Pen.***Width** [= *width*] **Part** **Description** *width* The width of the line used to draw the pen.
VtColor	VtColor object	The pen color used to draw the line *Pen.***VtColor** For more information, see **VtColor** object**.**

Pen Object Applies To

Axis Object, Item As Series Object, Item As Attribute Object, PlotBase Object, Wall Object, Marker Object, Item As DataPoint Object, Axis Grid Object, Surface Object

Pie Object

View Object Model Example Applies To

Description

A pie chart.

Syntax

Pie

Property	Type	Description
ThicknessRatio	Single	Sets or returns the percentage of pie radius that is used to determine the height of a 3D pie. *Pie.***ThicknessRatio** [= *pctg*] **Part** — **Description** *pctg* — The pie radius percentage. The higher the percentage, the taller the pie. The maximum value is 1.0 (100%). The minimum value is 0.
TopRadiusRatio	Single	Sets or returns the percentage of a pie radius that is used to determine the size of the top of the 3D pie. *Pie.***TopRadiusRatio** [= *pctg*] **Part** — **Description** pctg — The pie radius percentage. A ratio of 1.0 (100) draws a cylinder. Values less than 100 result in a tapering of th top of the pie. 0 results in a cone.

Method	Description
Set	Provides a way to set the ThicknessRatio and TopRadiusRatio properties simultaneously for the Pie object. *Pie.***Set** (*thicknessratio, topradiusratio*) **Part** — **Description** *thicknessRatio* — The height of the 3D pie. *topRadiusRatio* — The size of the top of a 3D pie.

Pie Object Applies To

Plot Object

Plot Object

View Object Model **Example** **Applies To**

Description

The area upon which a chart is displayed.

Syntax

Plot

Property Quick Pick (click here for fast access to a property)

AngleUnit	AutoLayout	Axis	Backdrop	BarGap
Clockwise	DataSeriesInRows	DefaultPercentBasis	DepthtoHeightRatio	Doughnut
Elevation	Light	LocationRect	MaxBubbleToAxisRatio	Perspective
Pie	PlotBase	Projection	ScaleAngle	Series
Sort	StartingAngle	SubPlotLabelPosition	UniformAxis	View3D
Wall	WidthToHeightRatio	Weighting	xGap	XYZ
zGap				

Property	Type	Description
AngleUnit	Integer	Sets or returns the unit of measure used for all chart angles. *Plot.***AngleUnit** [= *unit*] **Part** — **Description** *unit* — A **VtAngleUnits c**onstant describing the unit of measure. The angles can be measured in degrees, radians, or grads. For more information, see **AngleUnits Constants**.
AutoLayout	Boolean	Determines whether or not the plot is in manual or automatic layout mode. *Plot.***AutoLayout** [= *boolean*] If True, the Plot object automatically determines the proper size and position of the plot based on the size and position of other elements. If False, the coordinates specified by **LocationRect** are used to position the plot.
Axis	Object	A chart axis. *Plot.***Axis** (*AxisId, index*) For more information, see **Axis** object**.** **Part** — **Description** *AxisId* — A **VtChAxisId** constant describing a specific axis. For more information, see **AxisId Constants** *index* — Identifies an axis when

there are more than one axis with the same AxisId.

Backdrop	Object	A shadow, pattern or picture displayed behind a chart plot. *Plot.***Backdrop** For more information, see **Backdrop** object**.**
BarGap	Single	Sets or returns the spacing of 2D bars or clustered 3D bars within a category. *Plot.***BarGap** [= *value*] **Part** — **Description** *value* — The bar spacing value. This is measured as a percentage of the bar width. A value of 0 results in the bars touching.
Clockwise	Boolean	Specifies whether or not pie, doughnut, polar and radar charts are drawn in a clockwise direction. *Plot.***Clockwise** [= *boolean*] If True, pie, doughnut, polar and radar charts are drawn in a clockwise direction. If False, the charts are drawn in a counterclockwise direction.
DataSeriesInRows	Boolean	Indicates whether series data is being read from a row rather than a column in a data grid. *Plot.***DataSeriesInRows** [= *boolean]* If True, series data is being read from a row in a data grid. If False, series data is being read from a column.
DefaultPercentBasis	Integer	Returns the default axis percentage basis for the chart. [*PercentBasis* =] *Plot*.**DefaultPercentBasis** For more information, see **PercentAxisBasis Constants**. This is a read-only property
DepthToHeightRatio	Single	Sets or returns the percentage of the chart height to be used as the chart depth. *Plot.***DepthToHeightRatio** [= *pctg*] **Part** — **Description** *pctg* — The chart height percentage.
Doughnut	Object	A doughnut chart. *Plot.***Doughnut** For more information, see **Coor** object**.**

Elevation	Object	An elevation chart. *Plot.***Elevation** For more information, see **Elevation** object**.**
Light	Object	The light source illuminating a chart. *Plot.***Light** For more information, see **Light** object**.**
LocationRect	Rect Object	The location of the chart plot using x and y coordinates. *Plot.***LocationRect** The values of this property are used to position the plot if **AutoLayout** is False**.** For more information, see **Rect** object**.** If this property is set, then the **AutoLayout** property is automatically set to **False**.
MaxBubbleToAxisRatio	Single	Sets or returns the percentage of the shortest chart axis that is used as the diameter of the largest bubble. All other bubbles are sized according to their relationship to the largest bubble. *Plot.***MaxBubbleToAxisRatio** [= *pctg]* **Part** — **Description** *pctg* — The chart axis percentage.
Perspective	Coor3 object	The position and distance (viewpoint) from which a 3D chart is viewed. *Plot.***Perspective** By default, 3D charts are viewed head-on at a distance of 200 percent (2.0) the depth of the chart. (Limited between 50-1000%) You can create dramatic effects or decrease distortion by moving the viewpoint vertically or horizontally. This point is represented by the z coordinate of the **Coor3** object**.** The x coordinate of the viewpoint is a position between 0 and 100% (1.0) with 0 representing the left edge of the chart and 100 representing the right edge of the chart. The y coordinate of the viewpoint is a position between 0 and 100% (1.0) with 0 representing the top edge of the chart and 100 representing the bottom edge of the chart. For more information, see **Coor3** object**.**
Pie	Object	A pie chart. *Plot.**Pie*** For more information, see **Pie** object**.**
PlotBase	Object	The area beneath a chart.

*Plot.***PlotBase**

For more information, see **PlotBase** object**.**

Projection	Integer	Sets or returns the type of projection used to display the chart. *Plot.***Projection** [= *type*] **Part** — **Description** *type* — A **VtProjectionType** constant used to describe the type of chart projection. For more information, see **ProjectionType Constants** .
ScaleAngle	Single	Sets or returns where you want to display the scale on a polar or radar chart. *Plot.***ScaleAngle** [= *angle*] **Part** — **Description** *angle* — The plot scale angle. Angles are measured in a counterclockwise direction with 0 starting at 3 o'clock. Angles can be measured in degrees, radians, or grads, depending on the current setting for *AngleUnits.* Valid values are -360 to 360 degrees.
Series	Object	A grouping of data points on a chart. *Plot.***Series** For more information, see **Item As Series Object.**
Sort	Integer	Sets or returns the type of sort order used in a pie or doughnut. *Plot.***Sort** [= *type*] **Part** — **Description** *type* — A **VtSortType c**onstant used to describe the plot sort order. For more information, see **SortType Constants.**
StartingAngle	Single	Sets or returns the position where you want to start drawing the pie, doughnut, polar or radar charts. *Plot.***StartingAngle** [= *angle*] **Part** — **Description** *angle* — This angle can be measured in degrees, radians, or grads, depending on the current *AngleUnits* setting. A value of 0 degrees indicates the 3 o'clock

		position. Setting the starting angle to 90 degrees moves the starting position to 12 o'clock if the *Clockwise* property is set to counterclockwise, or to 6 o'clock if it's set to clockwise. Valid values range from -360 to 360 degrees.
SubPlotLabelPosition	Integer	Sets or returns the position used to display a label on each pie or doughnut in a chart. *Plot.***SubPlotLabelPosition** [= *pos*] **Part** — **Description** *pos* — A **VtChSubPlotLabelLocationType** constant used to describe the position of the chart label. For more information, see **SubPlotLabelLocationType Constants**.
UniformAxis	Boolean	Specifies whether or not the unit scale for all value axes in a chart is uniform. *Plot.***UniformAxis** [= *boolean*] If True, the unit scale for all value axes is uniform. If False, the unit scale is not uniform. The unit scale is determined by the plot size and positioning set according to the **AutoLayout** or **LocationRect** property. If AutoLayout is True, the plot size and position are based on the size and position of other automatically laid out elements. If False, the coordinates specified by **LocationRect** are used to position the plot and determine the axes unit scale.
View3D	Object	The physical orientation of a 3D chart. *Plot.***View3D** For more information, see **View3D** object.
Wall	Object	A planar area depicting the Y axes on a 3D chart or the backdrop of a 2D chart. *Plot.***Wall** For more information, see **Wall** object.
WidthToHeightRatio	Single	Sets or returns the percentage of the chart height to be used as the chart width. *Plot.***WidthToHeightRatio** [= *pctg*] **Part** — **Description** *pctg* — The chart height percentage.
Weighting	Object	The size of a 2D or 3D pie or doughnut in relation to other pies or doughnuts in the same chart.

Plot.**Weighting**

For more information, see **Weighting** object.

xGap | Single | Sets or returns the spacing of bars between divisions on the X-axis. This space is measured as a percentage of the bar width.

Plot.**xGap** [= *spacing*]

Part	Description
spacing	The bar width percentage. A value of 0 results in the series of bars touching.

XYZ | Object | The point at which the axes of a 3D XYZ chart intersect.

Plot.**XYZ**

For more information see **XYZ** object.

zGap | Single | Sets or returns the spacing of 3D bars between divisions on the Z-axis. This space is measured as a percentage of the bar depth.

Plot.**zGap** [= *spacing*]

Part	Description
spacing	The bar depth percentage. A value of 0 results in the series of bars touching along the z axis.

Plot Object Applies To

VtChart Object

PlotBase Object

View Object Model Example Applies To

Description

The area beneath a chart.

Syntax

Base

Property	Type	Description
Brush	Brush object	The fill type used to display a chart plot base. *PlotBase*.**Brush** For more information, see **Brush** object**.**
BaseHeight	Single	Sets or returns the height of the 3D chart base in points. *PlotBase*.**BaseHeight** [= *height*] **Part** — **Description** *height* — The base height.
Pen	Pen object	Describes the color and pattern of lines or edges on a chart plotbase. *PlotBase*.**Pen** For more information, see **Pen** object.

PlotBase Object Applies To

Plot Object

Position Object

View Object Model Example Applies To

Description

The location where a chart series is drawn in relation to other series. If all series have the same order (position), then they are stacked.

Syntax

Position

Property	Type	Description
Excluded	Boolean	Determines whether or not a series is included on the chart. *Position.***Excluded** [= *boolean*] If True, the chart is drawn without including the series. If False, the series is included when the chart is drawn. As series may be included in a chart, but still not display because it is **Hidden.**
Hidden	Boolean	Determines whether or not a series is displayed on the chart. *Position.***Hidden** [= *boolean*] If True, the series does not display on the chart, however, any space allocated for the series still exists. If False, the series is displayed.
Order	Integer	Sets or returns the position of the series in the chart. If the position in order matches another series, the series are stacked. *Position.***Order** [= *order*]
StackOrder	Integer	Sets or returns in what position the current series is drawn if it is stacked with other series. *Position.***StackOrder** [= *position*]

Part	Description
order	The position order.

Part	Description
position	The order of the series if stacked with other series. Lower stack orders are on the bottom of the stack.

Position Object Applies To

Item As Series Object

PrintInformation Object

View Object Model **Example** **Applies To**

Description

The attributes describing how a chart should be printed.

Syntax

PrintInformation

Property Quick Pick (click here for fast access to a property)

BottomMargin CenterHorizontally CenterVertically LayoutForPrinter LeftMargin
Monochrome Orientation RightMargin ScaleType TopMargin

Method Quick Pick (click here for fast access to a method)

PrintCopies

Property	Type	Description
BottomMargin	Single	Sets or returns the amount of white space at the bottom edge of the paper. *PrintInformation.***BottomMargin** [= *space*] **Part** — **Description** *space* — The space at the bottom of the page. Margins are specified in inches or centimeters.
CenterHorizontally	Boolean	Determines whether or not the chart is centered horizontally on the page. *PrintInformation.***CenterHorizontally** [= *boolean*] If True, the chart is centered horizontally. If False, the chart is not centered horizontally.
CenterVertically	Boolean	Determines whether or not the chart is centered vertically on the page. *PrintInformation.***CenterVertically** [= *boolean*] If True, the chart is centered horizontally. If False, the chart is not centered vertically.
LayoutForPrinter	Boolean	Determines whether or not the chart is re-laid out to best fit onto the page. Results of any redrawing are only seen on the printed page. *PrintInformation.***LayoutForPrinter** [= *boolean*] If True, the chart is re-laid out. If False, the chart is not altered.
LeftMargin	Single	Sets or returns the amount of white space at the left edge of the paper. *PrintInformation.***LeftMargin** [= *space*] **Part** — **Description** *space* — The space at the left of the page. Margins are

specified in inches or centimeters.

Monochrome — Boolean

This property is not currently used.

Orientation — Integer

Sets or returns the physical layout of the chart on the printed page.

*PrintInformation.***Orientation** [= *layout*]

Part	Description
layout	A **VtPrintOrientation** constant describing how the chart is laid out on the printed page. For more information, see **PrintOrientation Constants**.

RightMargin — Single

Sets or returns the amount of white space at the right edge of the paper.

*PrintInformation.***RightMargin** [= *space*]

Part	Description
space	The space at the right of the page. Margins are specified in inches or centimeters.

ScaleType — Integer

Sets or returns how the chart is scaled to fit on the page.

*PrintInformation.***ScaleType** [= *type*]

Part	Description
type	A **VtPrintScaleType** constant describing how the chart is scaled. For more information, see **PrintScaleType Constants**.

TopMargin — Single

Sets or returns the amount of white space at the top edge of the paper.

*PrintInformation.***TopMargin** [= *space*]

Part	Description
space	The space at the top of the page. Margins are specified in inches or centimeters.

Method	Description

PrintCopies

Prints a number of copies of the chart.

PrintInformation.**PrintCopies** (*NumberCopies*)

Part	Type	Description
NumberCopies	Integer	The number of copies to be printed.

PrintInformation Object Applies To

VtChart Object

Rect Object

View Object Model **Example** **View Structure** **Applies To**

Description

The Rect object defines a coordinate location.

Syntax

Rect

Property	Type	Description
Min	Coor object	Sets or returns the starting corner of the rectangle. *Rect.***Min** For more information, see **Coor** object**.**
Max	Coor object	Sets or returns the ending corner of the rectangle. Rect.**Max** For more information, see **Coor** object.

Rect Object Applies To

Location Object, Plot Object

SeriesCollection Collection

View Object Model Example Applies To

Description

A group of chart series.

Syntax

SeriesCollection

Property	Type	Description
Count	Long	Returns the number of series in this collection. *[num =] SeriesCollection.***Count** This is a read-only property.

Part	Type	Description
num	Integer	The number of series.

Item	Series object	Sets or returns a specific series for this collection. *SeriesCollection.***Item** *(seriesindex)* **Item** is the default property of the **SeriesCollection**.

Part	Type	Description
seriesindex	Integer	A chart series.

SeriesCollection Collection Applies To

Plot Object

Item As Series Object

View Object Model **Example** **Applies To**

Description

A group of data points on a chart.

Syntax

Item (*seriesNum*)

Property Quick Pick (click here for fast access to a property)

Bar	DataPoints	GuidelinePen	HiLo	LegendText
Pen	Position	SecondaryAxis	SeriesLabel	SeriesMarker
SeriesType	ShowGuideLines	ShowLine	SmoothingFactor	SmoothingType
StatLine				

Method Quick Pick (click here for fast access to a method)

Select TypeByChartType

Part	Description
seriesNum	Identifies the series of the chart. Series are identified in the order of data grid columns, beginning with 1.

Property	Type	Description
Bar	Object	A bar on 3D bar charts. *Item ( ).***Bar** For more information, see **Bar** object.
DataPoints	Object	A datapoint on a chart. *Item ( ).***DataPoints**. For more information, see **DataPoints** Collection**.**
GuidelinePen	Pen object	Sets or returns the pattern of line and color used to display guide lines. *Item ( ).***GuidelinePen** For more information, see **Pen** object. Note If you set this property, then the **ShowGuideLines** property is automatically set to **True**.
HiLo	Object	A hi-lo chart type. *Item ( ).***HiLo**
LegendText	String	Sets or returns the text in the that identifies the series in the legend. *Item ( ).***LegendText** [= *text*]

Part	Description

		text The series legend text. The default Legend text is the sames as ColumnLabel.Text.
Pen	Pen object	Describes the pattern of line and color used to display series lines. *Item ( ).***Pen** For more information, see **Pen** object. If you set this property, then the **ShowLine** property is automatically set to **True**.
Position	Object	The location of a series in a chart. *Item ( ).***Position** For more information, see **Position** object.
SecondaryAxis	Boolean	Determines whether or not the series is charted on the secondary axis. *Item ( ).***SecondaryAxis** [= *boolean*] If True, the series is charted on the secondary axis. If False, the series is not charted on the secondary axis.
SeriesLabel	Object	A label on a chart series. *Item ( ).***SeriesLabel** For more information, see **SeriesLabel object** .
SeriesMarker	Object	A marker on a series data point. *Item ( ).***SeriesMarker** For more information, see **SeriesMarker** object.
SeriesType	Integer	Sets or returns the type used to display the current series *Item ( ).***SeriesType** [= *type*] **Part** — **Description** *type* — A **VtChSeriesType** constant used to describe the display series. For more information, see **SeriesType Constants**.
ShowGuideLines	Boolean	Determines whether or not the connecting datapoint lines on a chart are displayed for a series. *Item ( ).***ShowGuidelines** (*AxisId, index*) [= *boolean*] If True, series guidelines are displayed. If False, guidelines are not displayed. This property is automatically set to True if the Guideline Pen property is set. **Part** — **Description**

		AxisId A **VtChAxisId** constant describing the series axis you want to set this property for. For more information about **VtChAxisId**, refer to Appendix B, Constants.
ShowLine	Boolean	Determines whether or not the lines connecting data points on a chart are visible. *Item ( ).***ShowLine** [= *boolean*] If True, the lines connecting data points appear on the chart.If False, datapoint lines do not appear.
SmoothingFactor	Integer	Sets or returns the number of facets or points that are sampled between the chart data points to create the smoothing effect. *Item ( ).***SmoothingFactor** [= *num*] **Part** — **Description** num — The number of points between data points. The higher the number, the more smoothing occurs.
SmoothingType	Integer	Sets or returns the type of mathematical function used to smooth the series. *Item ( ).***SmoothingType** [= *type*] **Part** — **Description** *type* — A **VtSmoothingType** constant used to describe the type of function used to smooth the series. For more information, see **SmoothingType Constants**.
StatLine	Object	A statistics line on a chart. *Item ( ).***StatLine** For more information, see **StatLine** object.

Method	**Description**
Select	Selects the specified series on a chart. *Item ( ).***Select**
TypeByChartType	Returns the series type used to draw a series if the chart type is set to chType. This method allows you to get the series type information based on a specified chart type without actually setting the chart type. This method is equivalent to how the Type Tab of the Format Plot Dialog displays the appropriate series type based on the chart type you select. This is a read-only method. [*seriestype =] Item ( ).***TypeByChartType** (*chtype*) **Part** — **Type** — **Description**

seriestype	Integer	A **VtChSeriesType** constant that describes the returned type used to display a series. For more information, see **SeriesType Constants**.
chtype	Integer	A **VtChChartType** constant describing the chart type. For more information, see **ChartType Constants**.

Only certain series types are valid for each chart type. The following table lists the series types that are valid for each chart type.

Chart Type	Valid Series Types
2D Combination	VtChSeriesType2dBar, VtChSeriesType2dLine, VtChSeriesType2dArea, VtChSeriesType2dStep, VtChSeriesType2dBarHiLo
3D Combination	VtChSeriesType3dBar, VtChSeriesType3dLine, VtChSeriesType3dArea, VtChSeriesType3dStep,VtChSeriesType3dBarHiLo
Radar	VtChSeriesType2dRadarLine, VtChSeriesType2dRadarArea
Hi-Lo	VtChSeriesType2dHiLo, VtChSeriesType2dHLC,VtChSeriesType2dHLCRight, VtChSeriesType2dOHLC, VtChSeriesType2dOHLCBar, VtChSeriesType2dBar, VtChSeriesType2dLine, VtChSeriesType2dArea, VtChSeriesType2dStep, VtChSeriesType2dBarHiLo, VtChSeriesType2dDates
2D Bar	VtChSeriesType2dBar, VtChSeriesType2dBarHiLo
3D Bar	VtChSeriesType3dBar, VtChSeriesType3dBarHiLo
2D Horizontal Bar	VtChSeriesType2dHorizontalBar, VtChSeriesType2dHorizontalBarHiLo
3D Horizontal Bar	VtChSeriesType3dHorizontalBar, VtChSeriesType3dHorizontalBarHiLo
3D Clustered Bar	VtChSeriesType3dClusteredBar, VtChSeriesType3dClusteredBarHiLo
3D Scatter	VtChSeriesType 3dXYZ
Other	Series type is same as chart type.

Item As Series Object Applies To

SeriesCollection Collection

SeriesLabel Object

View Object Model **Example** **Applies To**

Description

The name or text that describes a series.

Syntax

SeriesLabel

Property Quick Pick (click here for fast access to a property)

Backdrop LineStyle LocationType Offset Text
TextLayout TextLength VtFont

Property	Type	Description
Backdrop	Backdrop object	A shadow, pattern or picture displayed behind a SeriesLabel. *SeriesLabel.***Backdrop** For more information, see **Backdrop** object**.**
LineStyle	Integer	Sets or returns the type of line used to connect a series to a label. *SeriesLabel.***LineStyle** [= *type*] **Part** / **Description** *type* — A **VtChLabelLineStyle** constant that describes the line connecting a series to a label. For more information, see .
LocationType	Integer	Sets or returns the standard position used to display the series label. *SeriesLabel.***LocationType** [= *type*] **Part** / **Description** *type* — A **VtChLabelLocationType** constant that describes the series label position. For more information, see LabelLocationType Constants.
Offset	Coor object	The distance, in the x and y direction, that the label is moved from one of the standard label positions. *SeriesLabel.***Offset** For more information, see **Coor** object**.**
Text	String	Sets or returns the text used to display the series label. *SeriesLabel.***Text** [= *text*] The **Text** property is the default property for the SeriesLabel object **Part** / **Description**

		text The series label text. The default series label text is the same as column label.
TextLayout	TextLayout object	The position and orientation of the series label text. *SeriesLabel.***TextLayout** For more information, see **TextLayout** object**.**
TextLength	Single	Returns the length of the series label text. [*length* =]*SeriesLabel.***TextLength** **Part** **Description** *length* The length of the series label text. This is a read-only property.
VtFont	VtFont object	The font used to display SeriesLabel text. *SeriesLabel.***VtFont** For more information, see **VtFont** object**.**

SeriesLabel Object Applies To

Item As Series Object

SeriesMarker Object

View Object Model Example Applies To

Description

Describes a marker that identifies a series data point on a chart.

Syntax

SeriesMarker

Property	Type	Description
Auto	Boolean	Determines if the **SeriesMarker** object assigns the next available marker to all data points in the series. *SeriesMarker.***Auto** [= *boolean*] If True, the **SeriesMarker** object assigns the marker. If False, you can assign a custom marker. This property is automatically set to **False** if the DataPoint.Item (*index*) **Marker** property is set.
Show	Boolean	Determines whether series markers are displayed on a chart *SeriesMarker*.**Show** [= *boolean*] If True, series markers are displayed on a chart. If False, series markers are not displayed.

SeriesMarker Object Applies To

Item As Series Object

Shadow Object

View Object Model Example Applies To

Description

The Shadow object holds information about the appearance of a shadow on a chart element.

Syntax

Shadow

Property	Type	Description
Brush	Brush object	The fill type used to display a shadow. *Shadow.***Brush** For more information, see **Brush** object**.**
Offset	Coor object	Sets or returns the location of the shadow described as an x and y coordinate pair. *Shadow.***Offset** For more information, see **Coor** object**.**
Style	Integer	Sets or returns the shadow style. *Shadow.***Style** [= *type*] **Part** — **Description** *type* — A **VtShadowStyle** constant used to describe the shadow type. For more information, see **ShadowStyle Constants**.

Shadow Object Applies To

Backdrop Object

StatLine Object

View Object Model Example Applies To

Description

Statistic lines displayed on a chart.

Syntax

StatLine

Property	Type	Description
Flags	Integer	Sets or returns which statistic lines are being displayed for a series. *StatLine.***Flags** [= *lines*] **Part** — **Description** *lines* — A **VtChStats** constant used to describe the stat line. If more than one statistics line is displayed, the constants are OR'ed together. For more information, see **StatsType Constants**
Style	Integer	Sets or returns the line type used to display the statistic line. *StatLine.***Style** (*type*)[= *style*] **Part** — **Description** *style* — A **VtPenStyle** constant used to describe the stat line style. For more information, see **PenStyle Constants**. *type* — The line type.
VtColor	VtColor object	Describes the color used to display stat lines. *StatLine.***VtColor** For more information, see **VtColor** object**.**
Width	Single	Sets or returns the width of stat line in points. *StatLine.***Width** [= *width*] **Part** — **Description** *width* — The thickness used to draw the statistics lines, in points.

StatLine Object Applies To

Item As Series Object

Surface Object

View Object Model **Example** **Applies To**

Description

A surface chart.

Syntax

Surface

Property Quick Pick (click here for fast access to a property)

Base Brush ColWireframe DisplayType Projection
RowWireframe WireframePen

Property	Type	Description
Base	Integer	Sets or returns how the base of a surface chart is represented. *Surface*.**Base** [= *type*] **Part** / **Description** *type* — A VtChSurfaceBaseType constant used to describe the surface chart base. For more information, see SurfaceBaseType Constants.
Brush	Brush object	Describes the color and style used to display the chart contour as a band. *Surface*.***Brush*** For more information, see **Brush** object**.**
ColWireframe	Integer	Sets or returns the column wireframe display status and appearance. *Surface*.**ColWireframe** [= *type*] **Part** / **Description** *type* — A **VtChSurfaceWireframeType** constant used to describe the column wireframe. For more information, see SurfaceWireframeType Constants.
DisplayType	Integer	Sets or returns how the chart surface itself is represented. *Surface*.**DisplayType** [= *type*] **Part** / **Description** *type* — A **VtChSurfaceDisplayType** constant used to describe the chart surface. For more information, see **SurfaceDisplayType Constants**.
Projection	Integer	Sets or returns the appearance of the planar contour chart projected above the surface.

Surface.**Projection** [= *type*]

Part	Description
type	A VtChSurfaceProjectionType constant used to describe the surface chart projection. For more information, see **SurfaceProjectionType Constants**.

RowWireframe — Integer

Sets or returns the row wireframe display status and appearance.

Surface.**RowWireframe** [= *type*]

Part	Description
type	A **VtChSurfaceWireframeType** constant used to describe the row wireframe. For more information, see **SurfaceWireframeType Constants**.

WireframePen — Pen object

The color and style used to display the chart contour as a wireframe.

Surface.**WireframePen**

For more information, see **Pen** object.

Surface Object Applies To

Elevation Object

TextLayout Object

View Object Model Example View Structure Applies To

Description

Text positioning and orientation.

Syntax

TextLayout

Property	Type	Description
WordWrap	Boolean	Determines whether text wraps or not. *TextLayout.***WordWrap** [= *boolean*] If True, the text wraps. If False, the text does not wrap.
HorzAlignment	Integer	Sets or returns the method of horizontal alignment of text. *TextLayout.***HorzAlignment** [= *type*] **Part** — **Description** *type* — A **VtHorizontalAlignment** constant used to describe the horizontal alignment method of text. For more information, see **HorizontalAlignment Constants**.
Orientation	Integer	Sets or returns the method of orientation for text. *TextLayout.***Orientation** [= *type*] **Part** — **Description** *type* — A **VtOrientation** constant used to describe the orientation method. For more information, see **Orientation Constants**.
VertAlignment	Integer	Sets or returns the method of vertical alignment of text. *TextLayout.***VertAlignment** [= *type*] **Part** — **Description** *type* — A **VtVerticalAlignment** constant used to describe the vertical alignment method of text. For more information, see **VerticalAlignment Constants**.

TextLayout Object Applies To

SeriesLabel Object, Footnote Object, Title Object, Legend Object, AxisTitle Object, DataPointLabel Object

Tick Object

View Object Model Example Applies To

Description

A marker indicating a division along a chart axis.

Syntax

Tick

Property	Type	Description
Length	Single	Sets or returns the length of axis tick marks, measured in points. *Tick*.**Length [=** *length***]** **Part** — **Description** *length* — The axis tick mark length.
Style	Integer	Sets or returns the position of the axis tick. *Tick*.**Style [=** *type***]** **Part** — **Description** *type* — A **VtChAxisTickStyle** constant used to describe the axis tick position. For more information, see **AxisTickStyle Constants**.

Tick Object Applies To

Axis Object

Title Object

View Object Model **Example** **Applies To**

Description

Text identifying the chart.

Syntax

Title

Property Quick Pick (click here for fast access to a property)

Backdrop Location Text TextLayout TextLength VtFont

Property	Type	Description
Backdrop	Backdrop object	A shadow, pattern or picture displayed behind a chart title. *Title*.**Backdrop** For more information, see **Backdrop** object**.**
Location	Location object	Sets or returns the current position of the chart title. *Title*.**Location** For more information, see **Location** object**.**
Text	String	Sets or returns the text used to display a chart title. *Title*.**Text** [= *text*] The **Text** property is the default property for the **Title** object. **Part** — **Description** *text* — The title text.
TextLayout	TextLayout object	The position and orientation of chart title text. [*size* =] *Title*.**TextLayout** For more information, see **TextLayout** object**.**
TextLength	Integer	Returns the length of chart title text. *Title*.**TextLength** [= *size*] **Part** — **Description** *size* — The title length. This is a read-only property.
VtFont	VtFont object	The font used to display a chart title.

Title.**VtFont**

For more information, see **VtFont** object**.**

Method	Description
Select	Selects the chart title. *Title*.**Select**

Title Object Applies To

VtChart Object

ValueScale Object

View Object Model Example Applies To

Description

Scale used to display a value axis.

Syntax

ValueScale

Property	Type	Description
Auto	Boolean	Determines whether or not automatic scaling is used to draw the value axis. *ValueScale*.**Auto** [= *boolean*] If True, the scale is automatically set based on the data being charted. If False, the values in **Minimum, Maximum, MajorDivisions** and **MinorDivisions** are used to scale the axis.
MajorDivision	Integer	Sets or returns the number of major divisions displayed on the axis. *ValueScale*.**MajorDivision** [= *num*] If this property is set, then the **Auto** property is automatically set to **False**. **Part** — **Description** *num* — Number of major divisions.
Maximum	Double	Sets or returns the highest or ending value on the chart value axis. *ValueScale*.**Maximum** [= *value*] If this property is set, then the **Auto** property is automatically set to **False**. **Note**: The maximum property should be set first, before the minimum property, to avoid a chart display error. **Part** — **Description** *value* — The highest axis value.
Minimum	Double	Sets or returns the lowest or beginning value on the chart value axis. *ValueScale*.**Minimum** [= *value*] If this property is set, then the **Auto** property is automatically set to **False**. **Note**: The maximum property should be set first, before the minimum property, to avoid a chart display error. **Part** — **Description** *value* — The lowest axis value.
MinorDivision	Integer	Sets or returns the number of minor divisions displayed on the axis. *ValueScale*.**MinorDivision** [= *num*]

If this property is set, then the **Auto** property is automatically set to **False**.

Part	Description
num	Number of minor divisions.

ValueScale Object Applies To

Axis Object

View3D Object

View Object Model Example Applies To

Description

The physical orientation of a 3D chart.

Syntax

View3D

Property	Type	Description
Elevation	Single	Describes the degree of elevation from which a 3D chart is viewed. *View3D.***Elevation** [= *degree*]

Part	Description
degree	The degree of elevation. Elevation can be any number from 0 to 90 degrees. If you set the elevation to 90 degrees, you look directly down onto the top of the chart. If you set the elevation to 0, you look directly at the side of the chart. The default elevation is 30 degrees. By default, degrees are used to measure elevation. However, these settings use the current settings for the **AngleUnits** property. The other options are: Grads (400 to a circle) and Radians (2 ' to a circle.)

Rotation	Single	Describes the degree of rotation from which a 3D chart is viewed. *View3D.***Rotation** [= *degree*]

Part	Description
degree	The degree of rotation. Rotation can range from 0 to 360 degrees. By default, degrees are used to measure rotation. However, these settings use the current settings for the AngleUnits property. The other options are: Grads (400 to a circle) and Radians (2 ' to a circle.)

Method	Description
Set	Sets the rotation and degree of elevation for a 3D chart. *View3D*.Set

View3D Object Applies To

Plot Object

VtChart Object

View Object Model Example

Description

A chart that graphically displays data.

Syntax

VtChart

Properties

ActiveSeriesCount	AllowDithering	AllowDynamicRotation	AllowSelections	AllowSeriesSelection
AllowUserChanges	AutoIncrement	Backdrop	Chart3d	ChartType
Column	ColumnCount	ColumnLabel	ColumnLabelCount	ColumnLabelIndex
Data	DataGrid	DoSetCursor	DrawMode	ErrorOffset
FileName	Footnote	FootnoteText	Handle	Legend
Picture	Plot	PrintInformation	RandomFill	Repaint
Row	RowCount	RowLabel	RowLabelCount	RowLabelIndex
SeriesColumn	SeriesType	ShowLegend	SsLinkMode	SsLinkRange
SsLinkBook	Stacking	TextLengthType	Title	TitleText
TwipsWidth	TwipsHeight			

Methods

AboutBox	ActivateFormatMenu	ActivateSelectionDialog	CopyDataFromArray	CopyDataToArray
Draw	EditCopy	EditChartData	EditPaste	GetBitmap
GetDllVersion	GetMetafile	GetSelectedPart	Layout	PrintChart
PrintSetupDialog	ReadFromFile	Refresh	ToDefaults	TwipsToChartPart
SelectPart	UseWizard	WritePictureToFile	WriteToFile	

ActiveSeriesCount Property

Description

Returns the number of series that appear on a chart based on the number of columns in the data grid and the type of chart being drawn.

Type

Integer

Syntax

[*count* =] *VtChart1*.**ActiveSeriesCount**

Part	Description
count	Number of series for the chart. This is a read-only property.

Applies To

VtChart Object

AllowDithering Property

Description

Sets or returns whether to disable color dithering for charts on 8-bit color monitors in order to enable use of First Impression's own color palette and enhance the chart display.

Type

Boolean

Syntax

VtChart1.**AllowDithering** [=*boolean*]

If True, color dithering is allowed. If False, First Impression's color palette is used for enhanced color matching and display.

Applies To

VtChart Object

AllowDynamicRotation Property

Description

Indicates whether users can interactively rotate 3D charts by holding down the control key to display the rotation cursor.

Type

Boolean

Syntax

VtChart1.**AllowDynamicRotation** [= *boolean*]

If True, the user can interactively rotate the chart with the cursor. If False, the user cannot interactively rotate the chart with the cursor. Users can, however, still rotate the chart through the dialog box options.

Applies To

VtChart Object

AllowSelections Property

Description

Indicates whether or not users can select chart objects.

Type

Boolean

Syntax

VtChart1.**AllowSelections** [= *boolean*]

If True, users can select chart objects. If False, users cannot select chart objects.

Applies To

VtChart Object

AllowSeriesSelection Property

Description

Indicates whether or not a series can be selected when a user clicks on an individual chart data point.

Type

Boolean

Syntax

*VtChart1.***AllowSeriesSelection** [= *boolean*]

If True, users can select a series by clicking a data point. If False, the user cannot select a series by clicking a data point, it selects a data point.

Applies To

VtChart Object

AllowUserChanges Property

Description

Indicates whether users can display the shortcut menu or double click on chart elements to display First Impression dialog boxes to format the current chart.

Type

Boolean

Syntax

VtChart1.**AllowUserChanges** [= *boolean*]

If True, the user can format the chart. If False, the user cannot format the chart.

Applies To

VtChart Object

AutoIncrement Property

Description

Allows the properties that set the current data point to be incremented during data entry without manually setting the Column and Row properties.

Type

Boolean

Syntax

VtChart1.**AutoIncrement** [= *boolean*]

If this property is set to True, as soon as you change the Data property, the Row property updates to the next row in the column. If you are at the end of a column, the Column property increments to the next column.

Applies To

VtChart Object

Backdrop Property

Description

A shadow, pattern, or picture behind a chart.

Type

Object

Syntax

VtChart1.**Backdrop**

For more information, see **Backdrop** object.

Applies To

VtChart Object

Chart3d Property

Description

Returns whether or not a chart is a 3D chart.

Type

Boolean

Syntax

[*boolean* =] *VtChart1*.**Chart3D**

If True, the chart is a 3D chart. If False, the chart is not a 3D chart.

This is a read-only property.

Applies To

VtChart Object

ChartType Property

Description

Sets or returns the chart type being used to display a chart.

Type

Integer

Syntax

*VtChart1.***ChartType** [= *type*]

Part	Description
type	A **VtChChartType** constant that describes the chart type. For more information, see **ChartType Constants**.

Applies To

VtChart Object

Column Property

Description

Sets or returns the current data column in the data grid.

Type

Integer

You must select a column before you can use other properties to change the column's corresponding chart series or any data point within the series.

Syntax

*VtChart1.***Column** [= *col*]

Part	Description
col	The current data column.

Applies To

VtChart Object

ColumnCount Property

Description

Sets or returns the number of columns in the current data grid.

Type

Integer

Syntax

*VtChart1.***ColumnCount** [= *count*]

Part	Description
count	The number of data columns.

Applies To

VtChart Object

ColumnLabel Property

Description

Sets or returns the label text associated with a column in the data grid.

Type

String

Syntax

*VtChart1.***ColumnLabel** [= *text*]

Part	Description
text	Sets or returns the label text associated with a column in the data grid.

This property sets the label for the column currently identified by the **Column** property.

Applies To

VtChart Object

ColumnLabelCount Property

Description

Sets or returns the number of levels of labels on the columns in the data grid.

Type

Integer

Syntax

*VtChart1.***ColumnLabelCount** [= *count*]

Part	Description
count	The number of column label levels. Set this property to add or delete levels of labels on data grid columns.

Column label levels are numbered from bottom to top, beginning at 1. Levels are added or subtracted from the top.

Applies To

VtChart Object

ColumnLabelIndex Property

Description

Integer

Type

Sets or returns a specific level of column labels.

Syntax

*VtChart1.***ColumnLabelIndex** [= *index*]

Part	Description
index	The column label level. To set a label on a column with more than one level of labels, or to return the current value for a label, you must first identify which level you want to affect. Column label levels are numbered from bottom to top, beginning at 1.

Applies To

VtChart Object

Data Property

Description

Inserts a value into the current data point in the data grid. If the current data point already contains a value, it is replaced by the new value. The chart is redrawn to reflect the new value for the current data point.

Type

Integer

Syntax

VtChart1.**Data** [= *value*]

Part	Description
value	The data point value.

Applies To

VtChart Object

DataGrid Property

Description

A chart data grid.

Type

Object

Syntax

VtChart1.**DataGrid**

For more information, see **DataGrid** object.

Applies To

VtChart Object

DoSetCursor Property

Description

Indicates whether or not the cursor can be set by the chart. The DoSetCursor property determines whether or not the application can control what the mouse pointer looks like.

Type

Boolean

Syntax

VtChart1.**DoSetCursor** [= *boolean*]

If True, the application can control the mouse pointer appearance. If False, the application cannot control the mouse pointer appearance.

Applies To

VtChart Object

DrawMode Property

Description

Sets or returns the method used to redraw the chart, and when the chart is redrawn.

Type

Integer

Syntax

VtChart1.**DrawMode** [= *type*]

Part	Description
type	A **VtChDrawMode** constant describing the redraw method. For more information, see .

ErrorOffset Property

Description

Sets or returns the adjustment to Visual Basic trappable error numbers returned by First Impression.

Type

Integer

Syntax

VtChart1.**ErrorOffset** [= *offset*]

Part	Description
value	The amount by which First Impression error numbers should by incremented.

Use this property only if the First Impression error range conflicts with values returned by the Visual Basic Err statement.

Applies To

VtChart Object

FileName Property

Description

Sets or returns the name by which a chart is loaded and saved.

Type

String

Syntax

*VtChart1.***Filename** [= *name*]

Part	Description
name	The file name.

Applies To

VtChart Object

Footnote Property

Description

Descriptive text that appears beneath a chart.

Type

Object

Syntax

VtChart1.**Footnote**

For more information, see **Footnote** object.

Applies To

VtChart Object

FootnoteText Property

Description

Sets or returns the text used as the footnote.

Type

String

Syntax

*VtChart1.***FootnoteText** [= *text*]

Part	Description
text	The footnote text. The same results can be achieved by using the text property of the the footnote object.

Applies To

VtChart Object

Handle Property

Description

Returns a unique number that can be used to reference the chart.

Type

Long

Syntax

[*num* =] *VtChart1.***Handle**

Part	Description
num	The unique number identifying a chart.

This is a read-only property.

Applies To

VtChart Object

Legend Property

Description

The chart key that identifies the chart series.

Type

Object

Syntax

VtChart1.**Legend**

For more information, see **Legend** Object

Applies To

VtChart Object

Plot Property

Description

The area upon which a chart is displayed.

Type

Object

Syntax

VtChart1.**Plot**

For more information, see **Plot** object.

Applies To

VtChart Object

PrintInformation Property

Description

The attributes describing how a chart should be printed.

Type

Object

Syntax

VtChart1.**PrintInformation**

For more information, see PrintInformation object.

Applies To

VtChart Object

RandomFill Property

Description

Indicates whether or not the data for a chart data grid was randomly generated.

Type

Boolean

Syntax

VtChart1.**RandomFill** [= *boolean*]

If True, random data was used to draw the chart. If False, no random data was generated. The user provided the data for the chart.

Applies To

VtChart Object

Repaint Property

Description

Determines if the First Impression control is repainted after a change is made to the chart.

Type

Boolean

Syntax

VtChart1.**Repaint** [= *boolean*]

Setting the Repaint property to False does not allow the First Impression control to repaint when a change is made to the chart. This is useful when several operations are performed on the chart and you do not want the chart to continually repaint during the process. Setting this property to True causes the control to refresh.

Applies To

VtChart Object

Row Property

Description

Sets or returns a specific row in the current column of a data grid.

Type

Integer

Syntax

*VtChart1.***Row** [= *num*]

Part	Description
num	A row number in the current column. Rows are numbered from top to bottom beginning with 1.

Applies To

VtChart Object

RowCount Property

Description

Sets or returns how many rows there are in each column of the grid.

Type

Integer

Syntax

*VtChart1.***RowCount** [= *count*]

Part	Description
count	The number of rows in a column.

Applies To

VtChart Object

RowLabel Property

Description

Sets or returns a data label that can be used to identify the current data point.

Type

String

Syntax

*VtChart1.***RowLabel** [= *text*]

Part	Description
text	The text for a row label. The label you specify sets the label for the data points identified by the Row property. This label appears along the category axis for most chart types and is used as the label for each individual pie in a pie chart. Label text may not be displayed if it is too long to fit on a chart.

Applies To

VtChart Object

RowLabelCount Property

Description

Sets or returns the number of levels of labels on the rows in the data grid.

Type

Integer

Syntax

*VtChart1.***RowLabelCount** [= *count*]

Part	Description
count	The number of label levels. Set this property to add or delete levels of labels from data grid rows. Row label levels are numbered from right to left, beginning at 1. Levels are added or subtracted from the left.

Applies To

VtChart Object

RowLabelIndex Property

Description

Selects a specific level of row labels.

Type

Integer

Syntax

*VtChart1.***RowLabelIndex** [= *index*]

Part	Description
index	A row label level. To set a label on a row with more than one level of labels, or to return the current value for a label, you must first identify which level you want to affect. Row label levels are numbered from right to left, beginning at 1.

Applies To

VtChart Object

SeriesColumn Property

Description

Sets or returns the column position for the current series data.

Type

Integer

Syntax

VtChart1.**SeriesColumn** [= *pos*]

Part	Description
pos	The position of the column containing the current series data. You can use this property to reorder series. If two series are assigned the same position, they are stacked.

Applies To

VtChart Object

SeriesType Property

Description

Sets or returns the type used to display the current series.

Type

Integer

Syntax

VtChart1.**SeriesType** [= *type*]

Part	Description
type	A **VtChSeriesType** constant describing the method used to display the series. For more information, see **SeriesType Constants**. You must select the series to change using the Column property before using the **SeriesType** property.

Applies To

VtChart Object

ShowLegend Property

Description

Indicates whether or not a legend is visible for a chart.

Type

Boolean

Syntax

VtChart1.**ShowLegend** [= *boolean*]

If True, the legend appears on the chart in the position indicated by the Location object. If False, the legend is not displayed on the chart. The default legend location is *right*.

Applies To

VtChart Object

SsLinkMode Property

Description

Sets or returns how the data in a Formula One spreadsheet is interpreted by First Impression.

Type

Integer

Syntax

*VtChart1.***SsLinkMode** [= *type*]

Part	Description
type	A **VtChSsLinkMode** constant describing the method used to interpret the spreadsheet data. For more information, see **SsLinkMode Constants**.

Applies To

VtChart Object

SsLinkRange Property

Description

Sets or returns the range of data within the spreadsheet that contains the source data for the chart.

Type

String

Syntax

*VtChart1.***SsLinkRange** [= *range]*

Part	Description
range	A string representing a range of data within the spreadsheet that holds the data you want to chart. If possible, use a named range as opposed to a row and column reference. If you enter an invalid named range, an error condition is returned. If you enter an invalid tablename or row column reference, no error condition is returned.

Applies To

VtChart Object

SsLinkBook Property

Description

Identifies a Formula One workbook to use as the data source for the chart.

Type

String

Syntax

VtChart1.**SsLinkBook** [= *book*]

Part	Description
book	A string identifying the Formula One workbook by the Formula One TableName property.

Applies To

VtChart Object

Stacking Property

Description

Sets whether or not all the series in the chart are stacked.

Type

Boolean

Syntax

VtChart1.**Stacking** [= *boolean*]

If True, all the chart series are stacked. If False, the chart series are not stacked.

This is a write-only property.

Applies To

VtChart Object

TextLengthType Property

Description

Sets or returns how text is drawn to optimize the appearance either on the screen or printed page.

Type

Integer

Syntax

*VtChart1.***TextLengthType** [= *type*]

Part	Description
type	A **VtTextLengthType** constant indicating the method used to draw text. For more information, see **TextLengthType Constants**.

Applies To

VtChart Object

Title Property

Description

Text identifying the chart.

Type

Object

Syntax

VtChart1.**Title**

For more information, see **Title** object.

Applies To

VtChart Object

TitleText Property

Description

Sets or returns the text displayed as the chart title.

Type

String

Syntax

*VtChart1.***TitleText** [= *text*]

Part	Description
text	The text used to display a chart title.

Note: VtChart1.TitleText provides a simple means to set or return the chart title. This property appears in the chart control Property dialog. This property is functionally identical to using the VtChart1.Title.Text property. The functionality has been duplicated because VtChart1.Title.Text does not appear as an option in the Property dialog.

Applies To

VtChart Object

TwipsWidth Property

Description

Returns the width (X value) of a chart in twips.

Type

Integer

Syntax

[*width* =] *VtChart1*.**TwipsWidth**

Part	Description
width	The width of the chart document measured in twips from the upper left corner of the chart control. This is a read-only property.

Applies To

VtChart Object

TwipsHeight Property

Description

Returns the height (Y value) of a chart in twips.

Type

Integer

Syntax

[*height* =] *VtChart1*.**TwipsHeight**

Part	Description
height	The height of the chart document measured in twips from the upper left corner of the chart control . This is a read-only property.

Applies To

VtChart Object

Picture Property

Description

Returns a handle that a Visual Basic picture control can use to request a picture of the current chart.

Type

Integer

Syntax

[*Picture* =] *VtChart1*.**Picture**

Part	Description
Picture	The handle to a picture of the chart.

Applies To

VtChart Object

AboutBox Method

Description

Displays the First Impression About box.

Syntax

VtChart1.**AboutBox**

Applies To

VtChart Object

ActivateFormatMenu Method

Description

Displays the floating format menu on the chart control.

Syntax

VtChart1.**ActivateFormatMenu** (*x* ,*y*)

Part	Type	Description
x,y	Long	Indicates the coordinates for the display location of the floating format menu. Enter the number of pixels from the top left corner of the control. Enter -1 and -1 to place the top left corner of menu at the center of the chart.

Applies To

VtChart Object

ActivateSelectionDialog Method

Description

Displays the dialog box associated with the currently selected chart part.

Syntax

VtChart1.**ActivateSelectionDialog**

Applies To

VtChart Object

CopyDataFromArray Method

Description

Copies data from an array into a data grid.

Syntax

VtChart1.**CopyFromArray** (*top, left, bottom, right, array*)

Part	Type	Description
top	Integer	Identifies the top edge of the area in the grid you want to fill with data.
left	Integer	Identifies the left edge of the area in the grid you want to fill with data.
bottom	Integer	Identifies the bottom edge of the area in the grid you want to fill with data.
right	Integer	Identifies the right edge of the area in the grid you want to fill with data.
array	Variant	Describes the array holding the data to be copied. This parameter should to be declared as some kind of 2D array. (e.g. array (5,5) as Double)

Applies To

VtChart Object

CopyDataToArray Method

Description

Copies data from a data grid into an array.

Syntax

VtChart1.CopyDataToArray(***top, left, bottom, right, array***)

Part	Type	Description
top	Integer	Identifies the top edge of the area in the grid you want to fill with data.
left	Integer	Identifies the left edge of the area in the grid you want to fill with data.
bottom	Integer	Identifies the bottom edge of the area in the grid you want to fill with data.
right	Integer	Identifies the right edge of the area in the grid you want to fill with data.
array	Variant	Describes the array that data is to be copied to. This parameter should to be declared as some kind of 2D array. (e.g. array (5,5) as Double)

Applies To

VtChart Object

Draw Method

Description

This method can be used to print a chart at a specific location on a page or to draw a scaled version of the chart to be displayed in a window.

Syntax

*VtChart1.***Draw** (*hDC, hDCType, Top, Left, Bottom,Right, Layout, Stretch*)

Part	Type	Description
hdc	Integer	A standard windows device context.
hdcType .	Integer	A **VtDcType** constant that identifies the type of context represented by hdc.

For more information, see DcType Constants.

Top	Long	The location that specifies in pixels where the top edge of the chart is to be drawn.
Left	Long	The location that specifies in pixels where the left edge of the chart is to be drawn.
Bottom	Long	The location that specifies in pixels where the bottom edge of the chart chart is to be drawn.
Right	Long	The location that specifies in pixels where the right edge of the chart is to be drawn.
Layout	Boolean	Controls how the chart is laid out on the device.

If Layout is True, the chart is laid out for the size of the rectangle. Labels and other chart elements may be moved or changed in order to optimize chart appearance. The result may not be WYSIWYG. If layout is False the output is WYSIWYG

Stretch	Boolean	Controls chart scaling.

Stretch is only valid if Layout is set to False. If Stretch is true, the chart is scaled up or down to fit the rectangle provided. If Stretch is False, the chart is printed at its actual size.

Applies To

VtChart Object

EditChartData Method

Description

Invokes the Data Grid Editor.

Syntax

VtChart1.**EditChartData**

Applies to

VtChart Object

EditCopy Method

Description

Copies the current chart to the clipboard in Windows Metafile format.

Syntax

VtChart1.**EditCopy**

Applies To

VtChart Object

EditPaste Method

Description

Pastes a Windows Metafile graphic from the clipboard into the current selection.

Syntax

VtChart1.**EditPaste**

Applies To

VtChart Object

GetBitmap Method

Description

Returns a handle to a copy of the current chart in bitmap format. Another control can use this handle to display the chart copy.

Syntax

[*handle* =] *VtChart1*.**GetBitmap** (*options*)

Part	Type	Description
options	Integer	Currently an unused argument.
handle	OLE_HANDLE	The handle to the chart. This is a read-only method.

Note: Some development environments do not support the *OLE_HANDLE* data type, if your environment does not support this type, use data type *LONG*.

Applies To

VtChart Object

GetDllVersion Method

Description

Identifies the version number of the First Impression DLL you are using.

Syntax

VtChart1.**GetDLLVersion** (*major, minor*)

Part	Type	Description
major	Integer	Specifies the major release number.
minor	Integer	Specifies the minor release number.

Applies To

VtChart Object

GetMetafile Method

Description

Returns a handle to a copy of the current chart in a metafile format. This handle can be used by another control to display the chart copy.

Syntax

VtChart1.**GetMetafile** (*handle, width, height, option*)

Part	Type	Description
width	Long	Returns the width of the chart control in twips
height	Long	Returns the height of the chart control in twips.
handle	OLE_HANDLE	The handle to the chart.
option	Integer	A **VtPictureOptions** constant that controls the type of metafile created by this method.

For more information, see **PictureOptions Constants.** .

This is a read-only method.

Note: Some development environments do not support the *OLE_HANDLE* data type, if your environment does not support this type, use data type *LONG*.

Applies To

VtChart Object

GetSelectedPart Method

Description

Identifies the currently selected chart element.

Syntax

VtChart1.**GetSelectedPart** (*part, index1, index2, index3, index4*)

Part	Type	Description
part	Integer	Specifies the chart part. Valid constants are **VtChPartType**.

For more information, see **PartType Constants**.

index1	Integer	If part refers to a series or a datapoint, this argument specifies which series.

Series are numbered in the order their corresponding columns appear in the data grid from left to right, beginning with 1.

If part refers to an axis or axis label, this argument identifies the axis type with a **VtChAxisId** constant.

For more information, see **AxisId Constants**.

index2	Integer	If part refers to a data point, this argument specifies which data point in the series identified by index1.
index3	Integer	If part refers to an axis label, this argument refers to the level of the label. Axis label levels are numbered from the axis out, beginning with 1.

If part is not an axis label, the argument is unused.

index4	Integer	This argument is unused at this time.

Applies To

VtChart Object

Layout Method

Description

Lays out a chart, forcing recalculation of automatic values.

Syntax

VtChart1.**Layout**

A chart is laid out the first time it is drawn. When any chart settings change, the chart is again laid out at the next draw. There are a number of settings the chart calculates, such as the axis minimum and maximum values, based on the chart type or some other setting. These values are not determined until the chart is laid out. If you attempt to "get" these automatic values before the chart is properly laid out, they will not reflect the new values.

Applies To

VtChart Object

PrintChart Method

Description

Sends the current chart to the current printer using the information set by the PrintInformation object.

Syntax

*VtChart1.***PrintChart**

Applies To

VtChart Object

PrintSetupDialog Method

Description

Displays the Print Setup dialog box.

Syntax

*VtChart1.***PrintSetupDialog**

Applies To

VtChart Object

ReadFromFile Method

Description

Opens an existing chart file.

Syntax

VtChart1.**ReadFromFile** (*filename*)

Part	Type	Description
filename	String	The name of the file to open

Applies To

VtChart Object

Refresh Method

Description

Repaints the chart.

Syntax

VtChart1.**Refresh**

Applies To

VtChart Object

ToDefaults Method

Description

Returns the chart to its initial settings.

Syntax

*VtChart1.***ToDefaults**

Applies To

VtChart Object

TwipsToChartPart Method

Description

Identifies a chart part by using the x and y set of coordinates on to identify that part.

Syntax

VtChart1.**TwipsToChartPart** (*xVal, yVal, part, index1, index2, index3, index4*)

Part	Type	Description
xVal,yVal	Long	The horizontal and vertical coordinates of the point.
part	Integer	A **VtChPartType** constant that identifies the chart part that is located at the xVal and yVal coordinates.

For more information, see **PartType Constants**.

index1	Integer	If *part* refers to a series or a datapoint, this argument specifies which series.

Series are numbered in the order their corresponding columns appear in the data grid from left to right, beginning with 1.

If *part* refers to an axis or axis label, this argument identifies the axis type using the **VtChAxisId** constant.

For more information, see **AxisId Constants**.

index2	Integer	If *part* refers to a data point, this argument specifies which data point in the series identified by index1.

Data points are numbered in the order their corresponding rows appear in the data grid from top to bottom, beginning with 1.

If *part* refers to an axis, axis title, or axis label, this argument refers to the axis index which is currently not used. In this case, the only valid value for this argument is 1.

index3	Integer	If *part* refers to an axis label, this argument refers to the level of the label. Axis label levels are numbered from the axis out, beginning with 1.

If *part* is not an axis label, the argument is unused.

index4	Integer	This argument is unused at this time.

Applies To

VtChart Object

SelectPart Method

Description

Selects the specified chart part.

Syntax

VtChart1.**SelectPart** (*part, index1, index2, index3, index4*)

Part	Type	Description
part	Integer	Specifies the chart part. Valid constants are **VtChPartType**.

For more information, see **PartType Constants**.

index1	Integer	If part refers to a series or a datapoint, this argument specifies which series.

Series are numbered in the order their corresponding columns appear in the data grid from left to right, beginning with 1.

If part refers to an axis or axis label, this argument identifies the axis type with a **VtChAxisId** constant.

For more information, see **AxisId Constants.**

index2	Integer	If part refers to a data point, this argument specifies which data point in the series identified by index1.

Data points are numbered in the order their corresponding rows appear in the data grid from top to bottom, beginning with 1.

If part refers to an axis, axis title, or axis label, this argument refers to the axis index which is currently not used. In this case, the only valid value for this argument is 1.

index3	Integer	If part refers to an axis label, this argument refers to the level of the label. Axis label levels are numbered from the axis out, beginning with 1.

If part is not an axis label, the argument is unused.

index4	Integer	This argument is unused at this time.

Applies To

VtChart Object

UseWizard Method

Description

Invokes the chart wizard.

Syntax

VtChart1.**UseWizard**

Applies To

VtChart Object

WritePictureToFile Method

Description

Saves a chart to a file in a supported graphic format.

Syntax

VtChart1.**WritePictureToFile** (*filename, pictureType, options*)

Part	Type	Description
filename	String	The name of the chart file.
pictureType	Integer	Specifies the format used to write the chart to the file. **VtPictureType** contains valid constants for this argument.

For more information, see **PictureType Constants**.

options	Integer	A **VtPictureOption** constant that controls the type of metafile created by this method.

For more information, see PictureOptions Constants.

Applies To

VtChart Object

WriteToFile Method

Description

Saves a chart to a file

Syntax

VtChart1.**WriteToFile** (*filename*)

Part	Type	Description
filename	String	The name of the chart file.

Applies To

VtChart Object

VtColor Object

View Object Model Example View Structure Applies To

Description

Describes a drawing color.

Syntax

VtColor

Property Quick Pick (click here for fast access to a property)

Automatic Blue Green Red Value

Property	Type	Description
Automatic	Boolean	Determines whether or not the color is calculated automatically. *VtColor.***Automatic** [= *boolean*] If True, color automatically picks up the brush color used on the chart series. If False, the color is determined based on the settings of **Value.** This is only used for edge pens and hi-lo gain and loss colors.
Blue	Integer	Sets or returns the blue component of the RGB value. *VtColor.***Green** [= *b*] **Part** / **Description**: b — The blue value.
Green	Integer	Sets or returns the green component of the RGB value. *VtColor.***Green** [= *g*] **Part** / **Description**: g — The green value.
Red	Integer	Sets or returns the red component of the RGB value. *VtColor.***Red** [= *r*] **Part** / **Description**: r — The red value.
Value	Integer	Sets or returns the RGB value of the VtColor object. *VtColor* [.*Value*] = **RGB** (*r,g,b*) **Value** is the default property of the VtColor object. so that the user can make an assignment w/o having to specify 'value'. **Part** / **Type** / **Description**: *RGB* — Integer — A return value representing an RGB color value.

r,g,b	Integer	The number in the range 0 to 255, inclusive, that represents the red, green or blue component of the color. Methods and properties that accept a color specification expect that specification to be a number representing an RGB color value. RGB specifies the relative intensity of red, green, and blue to cause a specific color to be displayed. The valid range for a normal RGB color is 0 to 16,777,215. The value for any argument to RGB that exceeds 255 is assumed to be 255.

The following table lists some standard colors and the red, green, and blue values they include:

Color	Red Value	Green Value	Blue Value
Black	0	0	0
Blue	0	0	255
Cyan	0	255	255
Green	0	255	0
Red	255	0	0
Magenta	255	0	255
Yellow	255	255	0
White	255	255	255

Method	Description
Set	Sets the red, green and blue values of the VtColor object. *VtColor.***Set** (*red,green,blue*)

Part	Type	Desription
red,green,blue	Integer	The values for the red, green and blue components of color.

Applies To

VtChart Object

VtColor Object Applies To

StatLine Object, VtFont Object, Pen Object Gradient Object, Marker Object, Frame Object, Brush Object, ContourGradient Object, HiLo Object

VtFont Object

View Object Model Example View Structure Applies To

Description

The font used to display chart text.

Syntax

VtFont

Property	Type	Description
Color	VtColor object	The color of the font used to display chart text. *VtFont.***Color** [= *text*] For more information, see **VtColor** object**.**
Effects	Integer	Sets or returns the font effects. *VtFont.***Effects** [= *effects*] **Part** — **Description** *effects* — A **VtFontEffect** constant describing the font effect. For more information, see **FontEffect Constants**.
Name	String	Sets or returns the name of the font. *VtFont.***Name** [= *text*] **Name** is the default property for the **VtFont** object. **Part** — **Description** *text* — The text containing the font name.
Size	Single	Sets or returns the size of the font in points. *VtFont.***Size** [= *size*] **Part** — **Description** *size* — The font size.
Style	Integer	Sets or returns the font style. *VtFont.***Style** [= *type*] **Part** — **Description** *type* — A **VtFontStyle** constant describing the style of font. For more information, see **FontStyle Constants** .

VtFont Object Applies To

Title Object, Footnote Object, Legend Object, SeriesLabel Object, DataPointLabel Object, Item As Label Object,Item as Series Object, Item as DataPoint Object, AxisTitle Object

VtPicture Object

View Object Model Example Applies To

Description

A graphic that can be displayed as part of a chart.

Syntax

Picture

Property	Type	Description
Embedded	Boolean	Specifies if a graphic file is actually part of the chart. *VtPicture*.**Embedded** [= *boolean*] If True, the graphic is saved with the chart. If False, the picture is not saved with the chart. To set up a reference to a separate graphics file, specify the path and filename in the **Filename** property**.**
Filename	String	Sets or returns the filename and path to a referenced graphic file. *VtPicture*.**Filename** [= *pathname*] The **Filename** property is the default property for the VtPicture object.

Part	Description
pathname	The name and path to a graphic file.

Property	Type	Description
Map	Integer	Sets or returns how the picture is displayed. *VtPicture*.**Map** [= *style*]

Part	Description
style	A **VtPictureMapType** that describes how the picture is displayed. For more information, see **PictureMapType Constants** .

Property	Type	Description
Type	Integer	Sets or returns the type of graphic file. *VtPicture*.**Type** [= *type*]

Part	Description
type	A **VtPictureType** constant that describes the type of graphic file. For more information, see **PictureType Constants.** .

VtPicture Object Applies To

Item As DataPoint Object, Marker Object, Fill Object

Wall Object

View Object Model Example Applies To

Description

A planar area depicting the Y axes on a 3D chart.

Syntax

Wall

Property	Type	Description
Brush	Brush object	The color and pattern used to display the wall area. *Wall.***Brush** For more information, see **Brush** object**.**
Pen	Pen object	The color and width of the wall edges. *Wall.***Pen** For more information, see **Pen** object**.**
Width	Single	Sets or returns the thickness of a plot wall in points *Wall.***Width** [= *thickness*]

Part	Description
thickness	The width of the plot wall.

Wall Object Applies To

Plot Object

Weighting Object

View Object Model Example Applies To

Description

The size of a 2D or 3D pie or doughnut in relation to other pies or doughnuts in the same chart.

Syntax

Weighting

Property	Type	Description
Basis	Integer	Sets or returns the type of weighting used to determine pie or doughnut size. *Weighting.***Basis** [= *type*] **Part** — **Description** *type* — A **VtChPieWeightBasis** constant that identifies the weighting type. For more information, see **PieWeightBasis Constants**.
Style	Integer	Sets or returns how the weighting factor is applied *Weighting.***Style** [= *method*] **Part** — **Description** *method* — A **VtChPieWeightStyle** constant that identifies the weighting factor method. For more information, see **PieWeightStyle Constants**.

Weighting Object Applies To

Plot Object

XYZ Object

View Object Model Example Applies To

Description

The point at which the axes of a 3D XYZ chart intersect.

Syntax

XYZ

Property	Type	Description
Automatic	Boolean	Sets or returns whether First Impression uses the xintersection value, yintersection value and zintersection value to position the intersection point *XYZ*.**Automatic** If True, First Impression automatically sets the X,Y and Z intersection points at zero intersection points (0,0,0). If True, then First Impression uses the x,y and z intersection values to position the axes intersections.
xIntersection	Double	Sets or returns the point on the X axis of a 3DXYZ chart where the Y and Z axes intersect. *XYZ*.**xIntersection** [= *point*] If this property is set, then the **Automatic** property is automatically set to **False**. **Part** — **Description** *point* — The X axis value.
yIntersection	Double	Sets or returns the point on theY axis of a 3DXYZ chart where the X and Z axes intersect. *XYZ*.**yIntersection** [= *point*] If this property is set, then the **Automatic** property is automatically set to **False**. **Part** — **Description** *point* — The Y axis value.
zIntersection	Double	Sets or returns the point on the Z axis of a 3DXYZ chart where the X andY axes intersect. *XYZ*.**zIntersection** [= *point*] If this property is set, then the **Automatic** property is automatically set to **False**. **Part** — **Description** *point* — The Z axis value.

XYZ Object Applies To

Plot Object

Attributes Collection Example

The following example sets the number of contour attributes for a 3D surface chart.

```
Private Sub Command1_Click()
Dim CAttribute As Object
Dim cnt1 As Integer, cnt2 As Integer

'' Change chart type to 3D Surface
VtChart1.ChartType = VtChChartType3dSurface
Set CAttribute = VtChart1.Plot.Elevation.Attributes

'' The current attributes count
cnt1 = CAttribute.Count
CAttribute.Add 45
'' The attributes count after add one attribute
cnt2 = CAttribute.Count
MsgBox "The number of contour attributes:" & Chr(10) & "Before adding: "
& cnt1 & " After adding: " & cnt2
End Sub
```

Axis Object Example

The following example reads the number of label levels present in the X Axis using the X Axis object.

```
Private Sub Command1_Click()
Dim XAxis As Object
Dim NumberOfLevels As Integer

' Read the number of label level present in the X Axis
Set XAxis = VtChart1.Plot.axis(VtChAxisIdX, 1)
NumberOfLevels = XAxis.LabelLevelCount
MsgBox "Number of Label Levels = " & Str(NumberOfLevels)
End Sub
```

AxisGrid Example

The following example changes the X AxisGrid line style to dashed.

```
Private Sub Command1_Click()

    ' Changes Grid line style to dashed
    With VtChart1.Plot.axis(VtChAxisIdX).AxisGrid
        .majorPen.style = VtPenStyleDashed
        .majorPen.VtColor.Set 255, 0, 0
    End With

End Sub
```

Axis Scale Example

The following example sets the x and y axes to percent scale for a 2D XY chart.

```
Private Sub Command1_Click()

    'Change both x and y axes to Percent scale for 2D XY chart
    Form1.VtChart1.chartType = VtChChartType2dXY
    For AxisId = VtChAxisIdX To VtChAxisIdY
        With Form1.VtChart1.plot.axis(AxisId).AxisScale
            .type = VtChScaleTypePercent
            .percentBasis = VtChPercentAxisBasisSumChart
        End With
    Next

End Sub
```

Axis Title Example

The following example makes the axis title visible for all axes of a 3D chart.

```
Private Sub Command1_Click()

    ' Makes Axis title visible for all axes of a 3D chart
    VtChart1.chartType = VtChChartType3dBar
    For axisId = VtChAxisIdX To VtChAxisIdZ
        With VtChart1.Plot.axis(axisId, 1).AxisTitle
            .visible = True
            Select Case axisId
                Case 0
                    .text = "X Axis Title"
                Case 1
                    .text = "Y Axis Title"
                Case 2
                    .text = "2nd Y Axis Title"
                Case 3
                    .text = "Z Axis Title"
            End Select
        End With
    Next

End Sub
```

Backdrop Example

The following example sets a horizontal gradient backdrop for a chart.

```
Private Sub Command1_Click()
    ' Sets Backdrop to Horizontal Gradient for a chart
    With VtChart1.backdrop.fill
        .style = VtFillStyleGradient
        .gradient.style = VtGradientStyleHorizontal
        .gradient.fromColor.Set 0, 0, 255   '  From Blue
        .gradient.toColor.Set 255, 0, 0     '  To Red
    End With

End Sub
```

Bar Example

The following example sets all series in the chart to display pyramids instead of bars using the sides property.

```
Private Sub Command1_Click()
    Dim series As Object

    ' Set Chart Type To 3D Bar
    VtChart1.chartType = VtChChartType3dBar

    ' Sets all Series to display pyramids instead of bars
    For Each series In VtChart1.Plot.SeriesCollection
        series.Bar.sides = 4
        series.Bar.topRatio = 0
    Next

End Sub
```

Brush Example

The following example sets a bold vertical line pattern for the chart backdrop using the Brush object.

```
Private Sub Command1_Click()

    ' Sets Backdrop to Fill - Brush Style
    VtChart1.backdrop.fill.style = VtFillStyleBrush

    ' Sets a pattern for the chart backdrop using the Brush object
    With VtChart1.backdrop.fill.Brush
        .style = VtBrushStylePattern
        .index = VtBrushPatternBoldVertical         ' Sets Pattern to
Bold Vertical lines
        .fillColor.Set 255, 0, 0     ' Fill Color = Red
        .patternColor.Set 0, 0, 255      ' Pattern  Color = Blue
    End With
End Sub
```

Category Scale Example

The following example sets the scaling attributes for a Category axis.

```
Private Sub Command1_Click()

    ' Sets scaling attributes for a Category axis
    VtChart1.chartType = VtChChartType2dLine

    With VtChart1.Plot.axis(VtChAxisIdX, 1).CategoryScale
        .auto = False                       ' Sets manual scaling
        .divisionsPerLabel = 2              ' Label appears every two
divisions
        .divisionsPerTick = 2               ' Ticks appear every two
divisions
        .LabelTick = True                   ' Labels are displayed on top
of Tick marks
    End With

End Sub
```

Contour Example

The following example changes the chart type to a 2D Contour chart and displays the contours as lines.

```
Private Sub Command1_Click()

    'Change Chart Type to 2D Contour
    VtChart1.chartType = VtChChartType2dContour
    ' Display Contours as Lines
    VtChart1.Plot.elevation.Contour.DisplayType =
VtChContourDisplayTypeCLines

End Sub
```

Coor 3 Example

The following example sets the chart perspective using the coordinate system.

```
Private Sub Command1_Click()
Dim ViewPoint  As Object

' Change the Chart Type to 3D Bar
VtChart1.chartType = VtChChartType3dBar
' Set Chart perspective
Set ViewPoint = VtChart1.plot.perspective
ViewPoint.x = 50
ViewPoint.y = 50
ViewPoint.z = 200
End Sub
```

DataGrid Example

The following example sets the chart parameters for a 3D Bar chart, fills the chart with random data and labels the data grid columns.

```
Private Sub Command1_Click()
    Dim rowLabelCount As Integer
    Dim columnLabelCount As Integer
    Dim rowCount As Integer
    Dim columnCount As Integer

    Set DataGrid = VtChart1.DataGrid
    VtChart1.chartType = VtChChartType3dBar

    With VtChart1.DataGrid
    '' Set Chart parameters using methods
        rowLabelCount = 2
        columnLabelCount = 2
        rowCount = 6
        columnCount = 6
        .SetSize rowLabelCount, columnLabelCount, rowCount, columnCount

        '' Random Fill the data
        .RandomDataFill

        ' Then  Assign Labels to second Level
        labelIndex = 2
        column = 1
        .ColumnLabel(column, labelIndex) = "Product 1"
        column = 4
        .ColumnLabel(column, labelIndex) = "Product 2"

        row = 1
        .RowLabel(row, labelIndex) = "1994"
        row = 4
        .RowLabel(row, labelIndex) = "1995"
    End With
End Sub
```

DataPoints Collection Example

The following example sets the markers for each datapoint in a chart.

```
Private Sub Command1_Click()
Dim DataPoint As Object
Dim Index As Integer

For Each DataPoint In VtChart1.plot.SeriesCollection.Item(1).DataPoints
' Set DataPoint marker visible
DataPoint.Marker.visible = True
DataPoint.Marker.width = 12
Next
End Sub
```

Data Point Example

The following example assigns a variable to a data point and sets the datapoint color and marker.

```
Private Sub Command1_Click()

    ' change the color and marker of First DataPoint in the First Series
    With VtChart1.Plot.SeriesCollection.Item(1).DataPoints.Item(1)
        ' Change Data Point color to blue
        .Brush.style = VtBrushStyleSolid
        .Brush.fillColor.Set 0, 255, 255      ' Set Color=Blue

        ' Set DataPoint marker visible
        .marker.visible = True
    End With

End Sub
```

DataPoint Label Example

The following example sets the label for the first datapoint in the first series of a chart.

```
Private Sub Command1_Click()

    ' Set data label to First DataPoint in the First Series
    With
VtChart1.Plot.SeriesCollection.Item(1).DataPoints.Item(1).DataPointLabel
        ' Display label of First DataPoint
        .locationType = VtChLabelLocationTypeAbovePoint
        .Component = VtChLabelComponentValue Or
VtChLabelComponentSeriesName Or VtChLabelComponentPointName
        .ValueFormat = "0.0"
        .lineStyle = VtChLabelLineStyleBent
        .backdrop.frame.style = VtFrameStyleSingleLine
    End With

End Sub
```

DateScale Example

The following example sets manual scaling parameters for monthly date axes.

```
Private Sub Command1_Click()

    ' Set Chart Type to 3d Gantt
    VtChart1.chartType = VtChChartType3dGantt
    With VtChart1.Plot.axis(VtChAxisIdY).DateScale
        ' Set Scaling for the Dates axis to manual
        .auto = False
        .majInt = VtChDateIntervalTypeMonthly
        .majFreq = 1                                    ' Show labels for
every month
        .MinInt = VtChDateIntervalTypeWeekly
        .minFreq = 2                                    ' Show lines
every 2 weeks

        ' Set Time Range for 6 months
        .minimum = 1
        .maximum = 182
    End With

End Sub
```

Doughnut Example

The following example changes the chart type to doughnut and dispays the labels inside the plot.

```
Private Sub Command1_Click()

    ' Set chart type to Doughnut
    VtChart1.chartType = VtChChartType3dDoughnut

    With VtChart1.Plot.Doughnut
    .sides = 1
    .interiorRatio = 0.5
    End With

    ' Display the sub label indside the plot
    VtChart1.Plot.SubPlotLabelPosition =
VtChSubPlotLabelLocationTypeCenter

End Sub
```

Elevation Example

The following example sets the surface parameters for a 3D Surface chart and applies row and column smoothing.

```
Private Sub Command1_Click()
    ' Set Chart Type to Surface
    VtChart1.chartType = VtChChartType3dSurface

    ' Set Elevation properties
    With VtChart1.Plot.Elevation
        .autoValues = True
        .colorType = VtChContourColorTypeGradient
        .rowSmoothing = 15
        .colSmoothing = 15
        .SeparateContourData = False
    End With

    ' Set Surface color
    With VtChart1.Plot.Elevation.surface.Brush
        .style = VtBrushStyleSolid
        .fillColor.Set 100, 200, 100
    End With

End Sub
```

Fill Example

The following example sets a gradient backdrop for a chart using the Fill object.

```
Private Sub Command1_Click()

    With VtChart1.backdrop.fill
        ' Set a gradient backdrop
        .style = VtFillStyleGradient
        .gradient.style = VtGradientStyleHorizontal
        .gradient.fromColor.Set 100, 0, 255     ' From Blue
        .gradient.toColor.Set 255, 0, 100       ' To Red
    End With

End Sub
```

Footnote Example

The following example sets the footnote location, text and color for a chart.

```
Private Sub Command1_Click()

    With VtChart1.Footnote
        ' Make Footnote Visible
        .location.visible = True
        .location.locationType = VtChLocationTypeBottomLeft

        ' Set Footnote properties
        .text = "Chart Footnote"
        .VtFont.VtColor.Set 255, 0, 0
    End With

End Sub
```

Frame Example

The following example sets a blue, doubleline frame on a chart backdrop.

```
Private Sub Command1_Click()

    With VtChart1.backdrop.frame
        .style = VtFrameStyleDoubleLine
        .width = 2
        .FrameColor.Set 0, 0, 255           ' Blue frame
        .spaceColor.Set 255, 0, 0           ' red spacing
    End With

End Sub
```

Gradient Example

The following example sets a horizontal gradient as the chart backdrop fill.

```
Private Sub Command1_Click()

    ' Select gradient backdrop
    VtChart1.backdrop.fill.style = VtFillStyleGradient

    ' Set Gradient properties
    With VtChart1.backdrop.fill.Gradient
        .style = VtGradientStyleHorizontal
        .fromColor.Set 100, 0, 255       ' From Blue
        .toColor.Set 255, 0, 100         ' To Red
    End With

End Sub
```

HiLo Example

The following example sets the gaincolor and losscolor on an Open-Hi-Lo-Close bar chart.

```
Private Sub Command1_Click()
    Dim column As Integer

    ' Set Chart Type to 2D Open Hilo Close Bar
    VtChart1.chartType = VtChChartType2dHiLo
    For column = 1 To VtChart1.columnCount
        VtChart1.Plot.SeriesCollection.Item(column).seriesType =
VtChSeriesType2dOHLCBar
    Next column

    ' Set HiLo properties
    For column = 1 To VtChart1.columnCount
        With VtChart1.Plot.SeriesCollection.Item(column).HiLo
            .gainColor.Set 255, 0, 0         ' GainColor = Red
            .lossColor.Set 0, 0, 255         ' LossColor = Blue
        End With
    Next column

End Sub
```

Intersection Example

The following example sets manual intersection position properties and displays labels with the axis.

```
Private Sub Command1_Click()

    'change chart type to 3D Line
    VtChart1.chartType = VtChChartType3dLine

    With VtChart1.Plot.axis(VtChAxisIdX).Intersection
        ' Set Intersection Properties
        .auto = False                   ' Set positioning to manual
        .Point = 20                         ' Set intersection with the Y
Axis to 20
        .labelsInsidePlot = True        ' Display Labels with Axis
instead of at the base
    End With

End Sub
```

Legend Example

The following example sets the text and backdrop parameters for a chart legend.

```
Private Sub Command1_Click()

    With VtChart1.Legend
        ' Make Legend Visible
        .location.visible = True
        .location.locationType = VtChLocationTypeRight

        ' Set Legend properties
        .TextLayout.HorzAlignment = VtHorizontalAlignmentRight    '
Right justify
        .VtFont.VtColor.Set 255, 255, 0                          ' Use
Yellow text

        .backdrop.fill.style = VtFillStyleBrush
        .backdrop.fill.Brush.style = VtBrushStyleSolid
        .backdrop.fill.Brush.fillColor.Set 255, 0, 255
    End With

End Sub
```

Light Example

The following example sets the ambient light and edge lighting intensity for a chart.

```
Private Sub Command1_Click()

    ' Changes the Lighting for 3D Chart
    VtChart1.chartType = VtChChartType3dBar
    With VtChart1.Plot.Light
        .AmbientIntensity = 1              ' 100 % Intensity
        .EdgeIntensity = 0.5       ' 50 % Intensity
        .EdgeVisible = True
    End With

End Sub
```

Location Example

The following example sets the title location for a chart using the TitleLocation object.

```
Private Sub Command1_Click()

    ' Set Title Text
    VtChart1.TitleText = "Test Title Location"

    With VtChart1.title.location
        ' Make Title Visible
        .visible = True
        ' Use Top Left locaiton to display the title
        .locationType = VtChLocationTypeTopLeft
    End With

End Sub
```

Marker Example

The following example sets a blue X marker style for a chart series.

```
Private Sub Command1_Click()

    ' Display Markers for Series 1
    For Index = 1 To VtChart1.rowCount
        With
VtChart1.Plot.SeriesCollection.Item(1).DataPoints.Item(Index).Marker
            .visible = True
            .size = 20
            .style = VtMarkerStyleX
            .fillColor.automatic = False
            .fillColor.Set 0, 0, 255
        End With
    Next Index

End Sub
```

Pen Example

The following example sets the pen attributes for a 2D XY chart series.

```
Private Sub Command1_Click()

    '' Set Guide Lines for 2D XY chart Series 1
    VtChart1.chartType = VtChChartType2dXY
    VtChart1.Plot.SeriesCollection.Item(1).ShowGuideLine(VtChAxisIdX) =
True

    With VtChart1.Plot.SeriesCollection.Item(1).GuideLinePen
        ' Set Pen attributes
        .VtColor.Set 255, 255, 0
        .width = 10
        .style = VtPenStyleDashDot
        .join = VtPenRound
        .cap = VtPenCapRound
    End With

End Sub
```

Pie Example

The following example sets the thickness ratio and topradius ratio for a pie chart.

```
Private Sub Command1_Click()

    '  Set Chart Type to 3d Pie
    VtChart1.chartType = VtChChartType3dPie

    With VtChart1.Plot.Pie
        ' Set Pie Properties
        .thicknessRatio = 0.5        ' 50% Thickness
        .topRadiusRatio = 0          ' Cone shaped
    End With

End Sub
```

Plot Example

The following example sets the chart viewing distance and axis division spacing.

```
Private Sub Command1_Click()

    ' change the chart type to 3D Bar
    Form1.VtChart1.chartType = VtChChartType3dBar
    With Form1.VtChart1.plot
        'changes 3d bar chart's viewing
        .depthToHeightRatio = 2
        .widthToHeightRatio = 2
        'changes the spacing between divisions on the X-Axis
        .xGap = 0
        'changes the spacing between divisions on the Z-Axis
        .zGap = 0.8
    End With

End Sub
```

PlotBase Example

The following example sets the chart base parameters on a 3D bar chart.

```
Private Sub Command1_Click()

    'Change the chart type to 3D
    VtChart1.chartType = VtChChartType3dBar

    With Form1.VtChart1.plot.PlotBase
        'Change the base height
        .baseHeight = 20
        'Use the pattern style for base
        .Brush.style = VtBrushStylePattern
        .Brush.Index = VtBrushPatternHorizontal
        .Brush.fillColor.Set 255, 160, 160
        .Brush.patternColor.Set 180, 180, 255
        .Pen.style = VtPenStyleSolid
        .Pen.VtColor.Set 72, 72, 255
    End With

End Sub
```

Position Example

The following example uses the position object to hide and exclude chart series.

```
Private Sub Command1_Click()

    'Stack two series and hide one series, exclude one
    Form1.VtChart1.columnCount = 4
    With Form1.VtChart1.plot.SeriesCollection
        .Item(1).Position.order = 2
        .Item(3).Position.hidden = True
        .Item(4).Position.excluded = True
    End With

End Sub
```

Series Example

The following example sets smoothing for all series in a 3D line chart.

```
Private Sub Command1_Click()
    Dim Series As Object

    'change the chart type to 3D line and smoothing each line
    Form1.VtChart1.chartType = VtChChartType3dLine
    Form1.VtChart1.columnCount = 4
    For Each Series In Form1.VtChart1.plot.SeriesCollection
        Series.smoothingType = VtSmoothingTypeCubicBSpline
        Series.smoothingFactor = 10
        Series.Pen.style = 4
    Next

End Sub
```

SeriesLabel Example

The following example sets the label parameters for a series on a chart.

```
Private Sub Command1_Click()

    'show series label for series 1
    Form1.VtChart1.chartType = VtChChartType3dLine
    With Form1.VtChart1.plot.SeriesCollection.Item(1).SeriesLabel
        .locationType = VtChLabelLocationTypeCenter
        .lineStyle = VtChLabelLineStyleBent
        .VtFont.name = "Courier New"
        .VtFont.size = 14
    End With

    'set series label backdrop
    With
Form1.VtChart1.plot.SeriesCollection.Item(1).SeriesLabel.backdrop.fill
        .style = VtFillStyleGradient
        .Gradient.style = VtGradientStyleOval
        .Gradient.fromColor.Set 255, 0, 0
        .Gradient.toColor.Set 0, 0, 255
    End With

End Sub
```

Series Marker Example

The following example sets marker parameters for all series in a chart.

```
Private Sub Command1_Click()
    Dim series As Object

    'show markers and unshow the lines for all series
    Form1.VtChart1.chartType = VtChChartType2dLine
    For Each series In Form1.VtChart1.plot.SeriesCollection
        series.SeriesMarker.Show = True
        series.ShowLine = False
    Next

End Sub
```

Shadow Example

The following example sets a shadow on a chart backdrop title.

```
Private Sub Command1_Click()

    'show shadow for title
    With Form1.VtChart1.title
        .location.visible = True
        .text = "Chart Title"
    End With

    With Form1.VtChart1.title.backdrop.frame
        .width = 1
        .FrameColor.Set 255, 0, 0
        .style = VtFrameStyleSingleLine
    End With

    With Form1.VtChart1.title.backdrop.shadow
        .style = VtShadowStyleDrop
        .offset.x = 10
        .offset.y = 10
    End With

End Sub
```

StatLine Example

The following example sets the color and pen parameters for a chart statistics line.

```
Private Sub Command1_Click()

    'show all statistic lines for series 2
    Form1.VtChart1.chartType = VtChChartType2dLine
    With Form1.VtChart1.plot.SeriesCollection.Item(2).StatLine
        .VtColor.Set 128, 128, 255
        .Flag = VtChStatsMinimum Or VtChStatsMaximum Or VtChStatsMean Or
VtChStatsStddev Or VtChStatsRegression
        .style(VtChStatsMinimum) = VtPenStyleDotted
        .width = 2
    End With

End Sub
```

Surface Example

The following example sets the parameters for a solid surface chart on a contour lines base.

```
Private Sub Command1_Click()

    'change to surface chart
    Form1.VtChart1.chartType = VtChChartType3dSurface
    With Form1.VtChart1.plot.Elevation.surface
        .base = VtChSurfaceBaseTypeStandardWithCLines
        .colWireframe = VtChSurfaceWireframeTypeMajor
        .DisplayType = VtChSurfaceDisplayTypeSolid
        .projection = VtChSurfaceProjectionTypeNone
        .Brush.fillColor.Set 255, 0, 0
    End With

End Sub
```

TextLayout Example

The following example sets the title text position and orientation for a chart.

```
Private Sub Command1_Click()

    'sets the title text position and orientation
    With Form1.VtChart1.title
        .location.visible = True
        .location.locationType = VtChLocationTypeLeft
        .text = "Title TextLayout"
    End With

    With Form1.VtChart1.title.TextLayout
        .orientation = VtOrientationUp
        .HorzAlignment = VtHorizontalAlignmentCenter
        .VertAlignment = VtVerticalAlignmentCenter
    End With

End Sub
```

Title Example

The following example sets a gradient backdrop for a chart title.

```
Private Sub Command1_Click()

    'Set the tick for y axis
    With Form1.VtChart1.plot.axis(VtChAxisIdY, 1).tick
        .length = 20
        .style = VtChAxisTickStyleOutside
    End With

End Sub
```

Value Scale Example

The following example sets the major and minor grid line color for a 2D bar chart using the Value Scale object.

```
Private Sub Command1_Click()

    'set chart type to 2d bar
    Form1.VtChart1.chartType = VtChChartType2dBar
    'use  manual scale to display y axis (value axis)
    With Form1.VtChart1.plot.axis(VtChAxisIdY).ValueScale
        .auto = False
        .MajorDivision = 2
        .MinorDivision = 5
    End With

    'show major grid line in red and minor grid line in blue
    With Form1.VtChart1.plot.axis(VtChAxisIdY).AxisGrid
        .majorPen.VtColor.Set 255, 0, 0
        .majorPen.width = 4
        .minorPen.VtColor.Set 0, 0, 255
        .minorPen.width = 2
    End With

End Sub
```

View3D Example

The following example sets the chart elevation and rotation for a 3D bar chart using the view object.

```
Private Sub Command1_Click()

    'set the chart type to 3d bar
    Form1.VtChart1.chartType = VtChChartType3dBar
    With Form1.VtChart1.plot.View3d
        .Elevation = 90  'look directly donw onto the top of the chart
        .rotation = 90
    End With
End Sub
```

VtChart Example

The following example displays a 3D chart with 8 columns and rows of data and sets the legend parameters.

```
Private Sub Command1_Click()

    With Form1.VtChart1
        'displays a 3d chart with 8 columns and 8 rows data
        .chartType = VtChChartType3dBar
        .columnCount = 8
        .rowCount = 8
        For column = 1 To 8
            For row = 1 To 8
                .column = column
                .row = row
                .Data = row * 10
            Next row
        Next column
        'use the chart as the backdrop of the legend
        .ShowLegend = True
        .SelectPart VtChPartTypePlot, index1, index2, index3, index4
        .EditCopy
        .SelectPart VtChPartTypeLegend, index1, index2, index3, index4
        .EditPaste
    End With

End Sub
```

VtColor Example

The following example sets the fill color for a chart backdrop brush.

```
Private Sub Command1_Click()

    ' Sets Backdrop to Fill - Brush Style
    VtChart1.backdrop.fill.style = VtFillStyleBrush

    ' Sets Chart Fill Color to Red
    With VtChart1.backdrop.fill.Brush.fillColor
        .red = 255                  '  Use properties to set color
        .green = 0
        .blue = 0
    End With

End Sub
```

Font Example

The following example sets the font parameters for a chart title.

```
Private Sub Command1_Click()

    ' Make Chart Title visible
    VtChart1.title.location.visible = True

    ' Set font for Chart Title
    With VtChart1.title.VtFont
        .name = "Times New Roman"
        .size = 18
        .style = VtfontStyleBoldItalic

        ' Use both StrikeThrough and Underline in the text
        .Effect = VtFontEffectStrikeThrough Or VtFontEffectUnderline

        ' Set text color to Blue
        .VtColor.Set 0, 0, 255
    End With

End Sub
```

VtPicture Example

The following example sets a picture as a chart backdrop.

```
Private Sub Command1_Click()

    ' Displays a picture for chart backdrop
    With VtChart1.backdrop.fill.VtPicture
        ' Set Picture properties
        .type = VtPictureTypeBMP
        .fileName = "C:\Windows\Argyle.bmp"
        .map = VtPictureMapTypeTiled
        .embedded = True
    End With

End Sub
```

Wall Example

The following example displays a colored wall for a 3D chart.

```
Private Sub Command1_Click()

    'displays a colored wall for a 3D chart
    Form1.VtChart1.chartType = VtChChartType3dBar
    With Form1.VtChart1.plot.wall
        .Brush.style = VtBrushStylePattern
        .Brush.index = VtBrushPatternChecks
        .Brush.fillColor.Set 255, 120, 120
        .Brush.patternColor.Set 120, 120, 0
        .width = 20
    End With

End Sub
```

Weighting Example

The following example shows the weighting of a 3D doughnut chart.

```
Private Sub Command1_Click()

    'show the weighting of the 3D doughnut
    Form1.VtChart1.chartType = VtChChartType3dDoughnut
    With Form1.VtChart1.plot.Weighting
        .basis = VtChPieWeightBasisTotal
        .style = VtChPieWeightStyleArea
    End With

End Sub
```

Tick Example

The following example sets the tick length and style for the y axis on a chart.

```
Private Sub Command1_Click()

    'Set the tick for y axis
    With Form1.VtChart1.plot.axis(VtChAxisIdY, 1).tick
        .length = 20
        .style = VtChAxisTickStyleOutside
    End With

End Sub
```

Coor Object Example

The following example sets a custom location for the chart title using the coordinate system.

```
Private Sub Command1_Click()

    ' Change the Chart Type to 3D Bar
    VtChart1.chartType = VtChChartType3dBar
    ' Set Chart perspective
    With VtChart1.Plot.perspective
        .x = 0.5
        .y = 0.5
        .z = 1
    End With

End Sub
```

Label Object Example

The following example sets X and Y axis label parameters for a chart.

```
Private Sub Command1_Click()

    ' Set variables to First Level Axis Labels
    With VtChart1.Plot.axis(VtChAxisIdX).Labels.Item(1)
        ' Set X Axis label properties
        .auto = True
        .standing = True
        .TextLayout.orientation = VtOrientationVertical

        ' Set X Axis label color to blue
        .backdrop.fill.style = VtFillStyleBrush
        .backdrop.fill.Brush.style = VtBrushStyleSolid
        .backdrop.fill.Brush.fillColor.Set 0, 0, 255
    End With

    With VtChart1.Plot.axis(VtChAxisIdY).Labels.Item(1)
        ' Set Y Axis label properties
        .auto = True
        .Format = "#.00[RED]"
        .TextLayout.orientation = VtOrientationHorizontal
    End With

End Sub
```

LightSource Collection Example

The following example sets the coordinates and intensity for a light source, then adds and removes a light source from a chart.

```
Private Sub Command1_Click()
Dim LightSource As Object
Dim Index As Integer

' Set variable to Light Source 1
Set LightSource = VtChart1.plot.light.LightSources.Item(1)

' Set coordinates for Light Source 1 as well as its intensity
LightSource.x = 1
LightSource.y = 0.5
LightSource.z = 1
LightSource.intensity = 1

' Add a new light source
   VtChart1.plot.light.LightSources.Add 0.5, 1, 1, 1
' Remove Light Source 1
VtChart1.plot.light.LightSources.Remove (1)

End Sub
```

XYZ Object Example

The following example sets the intersection points on an 3D XYZ chart.

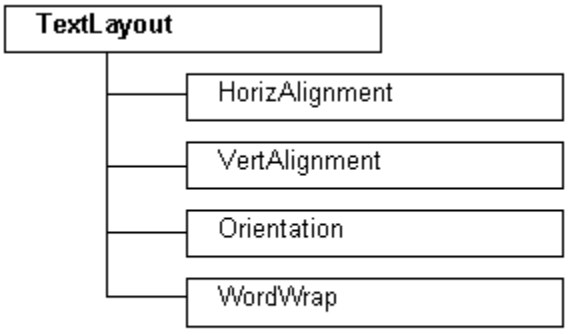
```
Private Sub Command1_Click()
    'set chart type to 3d XYZ
    Form1.VtChart1.chartType = VtChChartType3dXYZ
    'x, y, z intersects at 20
    With Form1.VtChart1.plot.xyz
        .xIntersection = 20
        .yIntersection = 20
        .zIntersection = 20
    End With

End Sub
```

VtFont Example

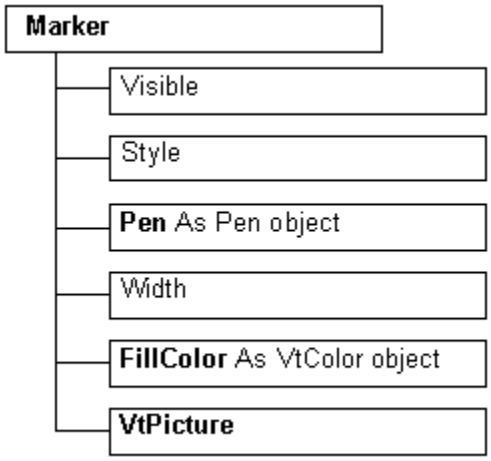
The following example sets the font parameters for a chart title.

```
Private Sub Command1_Click()

    ' Make Chart Title visible
    VtChart1.title.location.visible = True

    ' Set font for Chart Title
    With VtChart1.title.VtFont
        .name = "Times New Roman"
        .size = 18
        .style = VtfontStyleBoldItalic

        ' Use both StrikeThrough and Underline in the text
        .Effect = VtFontEffectStrikeThrough Or VtFontEffectUnderline

        ' Set text color to Blue
        .VtColor.Set 0, 0, 255
    End With

End Sub
```

The TextLayout Object Model



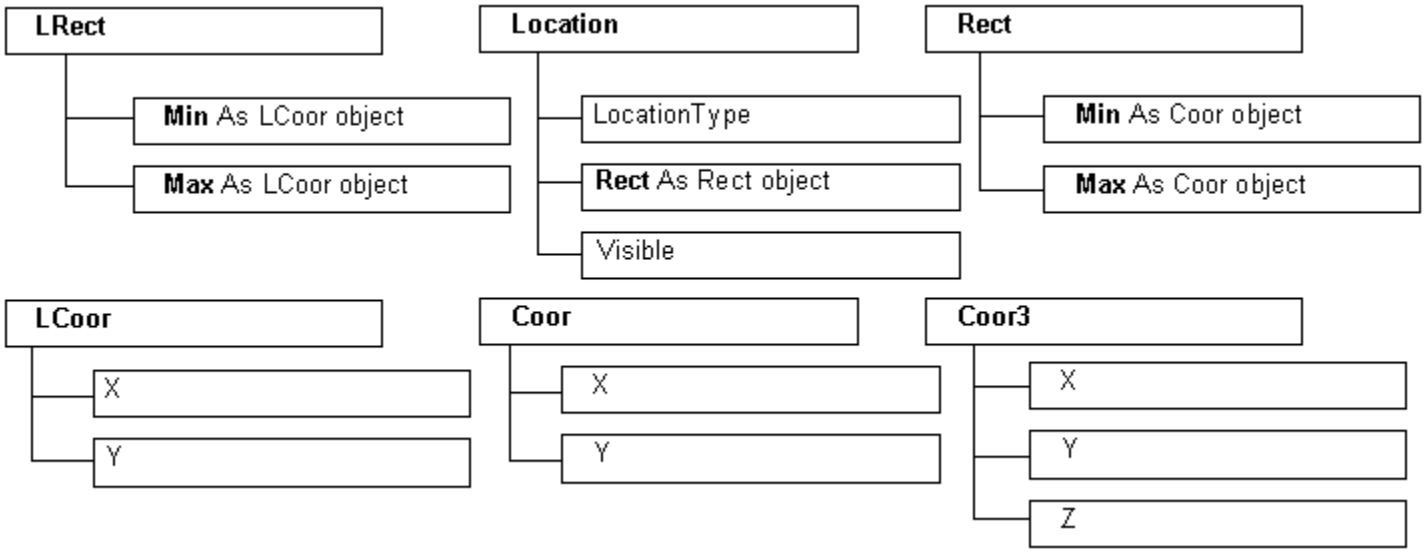
The Marker Object Model



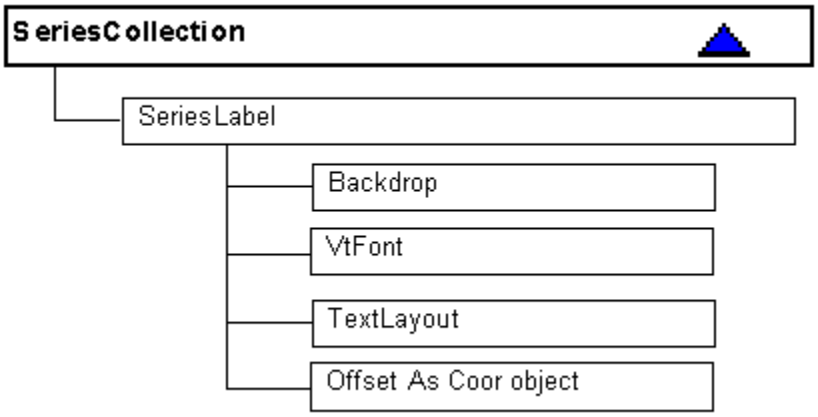
The VtFont Object Model

-

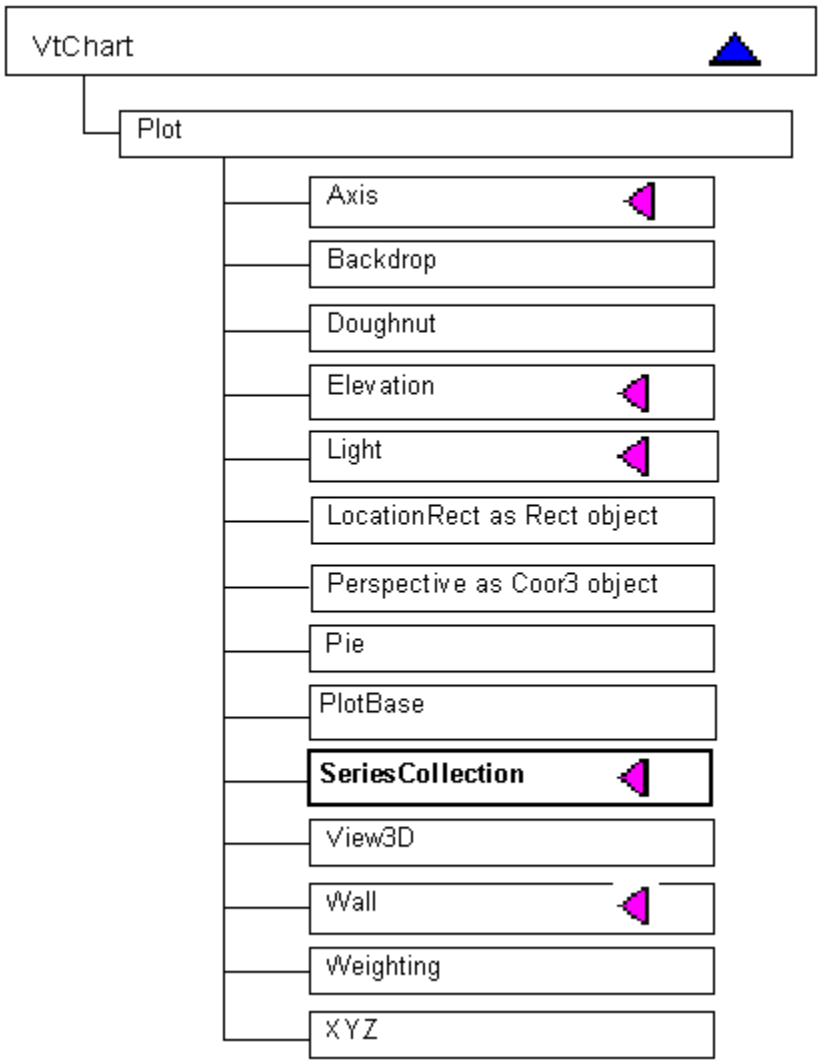
Coor and Location Objects Model



Offset As Coor Object



Plot Object Model



PrintInformation Object Example

The following example sets the print and page setup before printing a chart.

```
Private Sub Command1_Click()

    With VtChart1.PrintInformation
        '  Print Setup
        .layoutForPrinter = True
        .orientation = VtPrintOrientationLandscape
        .scaleType = VtPrintScaleTypeStretched

        ' Page Setup
        .leftMargin = 1
        .rightMargin = 1
        .topMargin = 1
        .bottomMargin = 1

        .centerHorizontally = True
        .centerVertically = True
    End With

    ' Print Chart
    VtChart1.PrintChart

End Sub
```

Series Collection Example

The following example hides all the series in a chart.

```
Private Sub Command1_Click()
    Dim series As Object

    ' Hides All Series
    For Each series In VtChart1.plot.SeriesCollection
        series.Position.hidden = True
    Next

End Sub
```

Labels Collection Example

The following example sets standing labels for all levels of X axis labels on a chart.

```
Private Sub Command1_Click()

    Dim Label As Object

    ' Change chart type to 3d bar and set 2 levels x axis labels
    With VtChart1
        .chartType = VtChChartType3dBar
        .rowLabelCount = 2
        .RowLabelIndex = 2
        .RowLabel = "Level Two Label"
    End With

    ' Show labels standing for all levels of  X Axis labels
    For Each Label In VtChart1.Plot.axis(VtChAxisIdX).Labels
        Label.standing = True
    Next

End Sub
```

Rect Object Example

The following example sets the location of the chart plot using the Location object.

```
Private Sub Command1_Click()

    ' Sets the lcoation of the chart plot
    VtChart1.plot.AutoLayout = False
    With VtChart1.plot.LocationRect
        .min.x = 0.4
        .min.y = 0.4
        .max.x = 0.5
        .max.y = 0.5
    End With

End Sub
```

ContourGradient Example

The following example sets gradient contour band surface for a 3D surface chart using the ContourGradient object.

```
Private Sub Command1_Click()

    '' Change chart type to 3D Surface
    VtChart1.chartType = VtChChartType3dSurface

    With VtChart1.Plot.elevation
        '' Set the surface chart color type to gradient
        .colorType = VtChContourColorTypeGradient
        '' Turn off the auto values
        .autoValues = False
        '' Set surface display type to Bands
        .surface.DisplayType = VtChSurfaceDisplayTypeCBands
    End With

    With VtChart1.Plot.elevation.ContourGradient
        '' Change the from color and to color
        .FromBrushColor.Set 122, 122, 122
        .ToBrushColor.Set 255, 0, 120
    End With

End Sub
```

Item As Attribute Object Example

The following example changes the chart to a 3D Surface chart and displays contour lines.

```
Private Sub Command1_Click()

    '' Change chart type to 3D Surface
    VtChart1.chartType = VtChChartType3dSurface

    '' Turn off the auto value, use manual color
    With VtChart1.plot.Elevation
        .autoValues = False
        .colorType = VtChContourColorTypeManual
        .surface.DisplayType = VtChSurfaceDisplayTypeCLines
    End With

    '' Modify the third attribute's value and color
    With VtChart1.plot.Elevation.attributes.Item(3)
        .Pen.width = 5
        .Pen.VtColor.Set 255, 0, 0
    End With

End Sub
```

Performance Tuning

First Impression provides true 3D drawings of text and objects in 3D charts. In order to accomplish this effect, each chart object may have to be broken down into hundreds of smaller objects for the most realistic 3D appearance. While this can provide a great deal of impact in your chart, frequent repaints or printing of 3D objects can cause performance problems on slower machines. The following tips can help you get the best performance from First Impression.

- **Limit the use of line smoothing**. Smoothing also dramatically increases the number of objects required to draw an element. If you want to use smoothing, turn it on right before printing. You can control the smoothing applied to a series with the Smoothing controls on the Options tab of the Format Series dialog box.
- **Choose line styles with dits rather than dots**. 3D line styles with dots or rounded end caps can take much longer to render because they dramatically increase the number of objects that make up the line. Use line styles with square end caps and dits instead of dots to design your chart. If necessary, you can switch to the dotted line styles at the end of the design process.
- **Limit the number of sides on elements**. Increasing the number of sides on elements such as bars and doughnuts can give them a more pleasing, rounded appearance. However, remember that each additional side can dramatically increase the number of objects in the element. Visually, there may not be that much difference between 10 sides and 12 sides, but the two extra sides can have a dramatic effect on the amount of time it takes to draw a doughnut. Use the lowest number of sides that gives you the appearance you want.
- **Turn off edge lighting**. Proper edge lighting can help to clearly delineate the faces of 3D elements. However, it also dramatically increases the number of objects that have to be drawn to create the element. You can improve performance by turning the edge lighting off. This is particularly true if you are drawing doughnut charts with many sides. If you require edge lighting on your chart, turn it on right before printing the chart. You can control the use of edge lighting by clicking the Use Edge Lighting check box on the 3D Lighting tab of the Format Plot dialog box.
- **Disable repainting when performing a series of operations**. When performing a number of sequential operations on a chart, disable repainting with the Repaint property so the screen does not repaint after each operation. This increases the speed of the operation and avoids unnecessary screen flashing.
- **Improve repaint speed for C/C++ programs**. You can improve repaint speed at the cost of some memory, by creating the window with class style CS_SAVEBITS.
- **Speed the code writing process with object variables.** You can simplify tasks by declaring object variables in code. By assigning a variable to the object, you can shorten the statements you must type. Refer to Working with Objects for more information.
- **Save time using the WITH clause**. You can slash time when writing code by using the WITH clause in your statements. The WITH clause allows you to declare the object one time and then set multiple properties for the same object in a single statement. Refer to Working with Objects for more information

AngleUnits Constants

Description

VtAngleUnits provides the valid units for measuring chart angles.

The following table lists the valid constants for **VtAngleUnits**:

Constants	Description
VtAngleUnitsDegrees	Angles are measured in degrees.
VtAngleUnitsRadians	Angles are measured in radians.
VtAngleUnitsGrads	Angles are measured in grads.

AxisId Constants

Description

VtAxisId provides options for identifying a chart axis.

The following table lists the valid constants for **VtChAxisId:**

Constants	Description
VtChAxisIdX	Identifies the X axis
VtChAxisIdY	Identifies the Y axis
VtChAxisIdY2	Identifies the secondary Y axis
VtChAxisIdZ	Identifies the Z axis

AxisTickStyle Constants

Description

VtChAxisTickStyle provides options for indicating axis tick mark location.

The following table lists the valid constants for **VtChAxisTickStyle**:

Constants	Description
VtChAxisTickStyleNone	No tick marks are displayed on the axis.
VtChAxisTickStyleCenter	Tick marks are centered across the axis.
VtChAxisTickStyleInside	Tick marks are displayed inside the axis.
VtChAxisTickStyleOutside	Tick marks are displayed outside the axis.

BrushStyle Constants

Description

VtBrushStyle provides valid brush types.

The following table lists the valid constants for **VtBrushStyle**:

Constant	Description

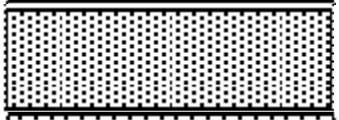
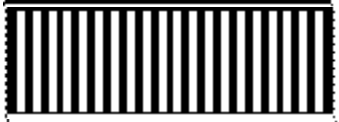
VtBrushStyleNull	No brush (background shows through).
VtBrushStyleSolid	Solid color brush.
VtBrushStylePattern	Bitmap patterned brush.
VtBrushStyleHatched	Hatched brush.

BrushPatterns Constants

Description

VtBrushPattern provides valid brush types if **VtBrushStyle** is set to **VtBrushStylePattern**.

The following table lists the valid constants for **VtBrushPattern**:

Constant	Description
VtBrushPattern94percent	
VtBrushPattern88percent	
VtBrushPattern75percent	
VtBrushPattern50percent	
VtBrushPattern25percent	
VtBrushPatternBoldHorizontal	
VtBrushPatternBoldVertical	
VtBrushPatternBoldDownDiagonal	
VtBrushPatternBoldUpDiagonal	
VtBrushPatternChecks	

VtBrushPatternWeave

VtBrushPatternHorizontal

VtBrushPatternVertical

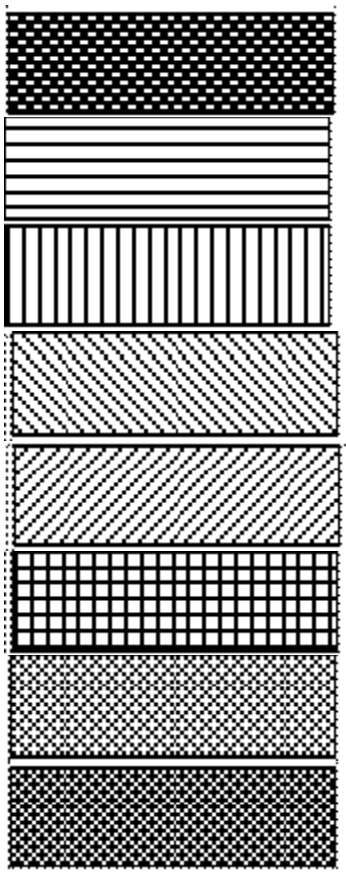
VtBrushPatternDownDiagonal

VtBrushPatternUpDiagonal

VtBrushPatternGrid

VtBrushPatternTrellis

VtBrushPatternInvertedTrellis



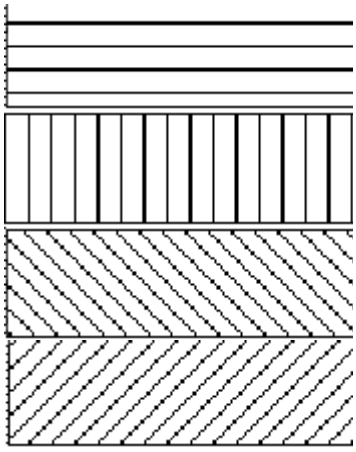
BrushHatches Constants

Description

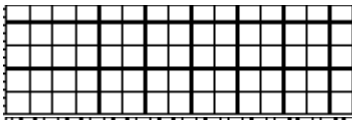
VtBrushHatch provides valid brush types.if **VtBrushStyle** is set to **VtBrushStyleHatch**.

The following table lists the valid constants for **VtBrushHatch**:

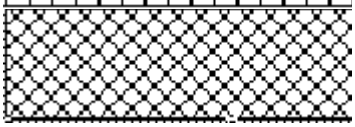
Constant	Description
VtBrushHatchHorizontal	
VtBrushHatchVertical	
VtBrushHatchDownDiagonal	
VtBrushHatchUpDiagonal	



VtBrushHatchCross



VtBrushHatchDiagonalCross



ChartType Constants

Description

VtChChartType provides chart type options.

The following table lists the valid constants for **VtChChartType:**

Constant	Description
VtChChartType3dBar	3D Bar
VtChChartType2dBar	2D Bar
VtChChartType3dLine	3D Line
VtChChartType2dLine	2D Line
VtChChartType3dArea	3D Area
VtChChartType2dArea	2D Area
VtChChartType3dStep	3D Step
VtChChartType2dStep	2D Step
VtChChartType3dCombination	3D Combination
VtChChartType2dCombination	2D Combination
VtChChartType3dHorizontalBar	3D Horizontal Bar
VtChChartType2dHorizontal Bar	2D Horizontal Bar
VtChChartType3dClusteredBar	3D Clustered Bar
VtChChartType3dPie	3D Pie
VtChChartType2dPie	2D Pie
VtChChartType3dDoughnut	2D Doughnut
VtChChartType2dXY	2D XY
VtChChartType2dPolar	2D Polar
VtChChartType2dRadar	2D Radar
VtChChartType2dBubble	2D Bubble
VtChChartType2dHiLo	2D Hi-Lo
VtChChartType2dGantt	2D Gantt
VtChChartType3dGantt	3D Gantt
VtChChartType3dSurface	3D Surface
VtChChartType2dContour	2D Contour
VtChChartType3dScatter	3D Scatter
VtChChartType3d XYZ	3D XYZ

ContourDisplayType Constants

Description

VtChContourDisplayType provides options for displaying chart contours.

The following table lists the valid constants for **VtChContourDisplayType:**

Constants	Description
VtChContourDisplayTypeCBand	The contour is displayed with contour bands.
VtChContourDisplayTypeCLines	The contour is displayed with contour lines.

ContourColorType Constants

Description

VtChContourColorType provides options for displaying contour colors.

The following table lists the valid constants for **VtChContourColorType:**

Type	Description
VtChContourColorTypeAutomatic	The contour colors are displayed as the default series colors.
VtChContourColorTypeGradient	The contours are displayed in an even transition of color.
VtChContourColorTypeManual	Custom contour colors can be specified and modified by the user. (Manual colors is only available when Automatic values is unchecked.)

DateIntervalType Constants

Description

VtChDateIntervalType provides options for displaying tick marks on a date axis.

The following table lists the valid constants for **VtChDateIntervalType:**

Constants	Description
VtChDateIntervalTypeNone	No Interval
VtChDateIntervalTypeDaily	A tick mark occurs each day.
VtChDateIntervalTypeWeekly	A tick mark occurs Monday of each week.
VtChDateIntervalTypeSemimonthly	A tick mark occurs on the 1st and 15th of each month.
VtChDateIntervalTypeMonthly	A tick mark occurs on the 1st of each month.
VtChDateIntervalTypeYearly	A tick mark occurs on January 1 of each year.

DrawMode Constants

Description

VtChDrawMode provides options for redisplaying a chart after it has been altered.

The following table lists the valid constants for **VtChDrawMode:**

Constants	Description
VtChDrawModeDraw	The chart is redrawn on the screen every time you change a setting.
VtChDrawModeBlit	The chart is redrawn off screen and displayed after the redraw is complete.

Blit mode stores a bitmap copy of the chart in memory when the chart is laid out. Repainting the chart uses the bitmap and draws very quickly. It requires more memory than Draw mode, but can save time waiting for the chart to redraw on screen. Blit mode is particularly useful when working with charts that contain many elements.

DcType Constants

Description

VtDcType identifies the type of context represented by hdc (a standard Windows device context).

The following table lists the valid constants for **VtDcType**.

Constants	Description
VtDcTypeNull	No device context represented.
VtDcTypeDisplay	Sends the file to the display identified by hdc.
VtDcTypePrinter	Sends the file to the printer identified by hdc.
VtDcTypeMetafile	Saves the file in Windows Metafile format to the device identified by hdc.

FillStyle Constants

Description

VtFillStyle provides options for indicating the type of fill used to paint a backdrop.

The following table lists the valid constants for **VtFillStyle**:

Constant	Description
VtFillStyleNull	No fill (background shows through)
VtFillStyleBrush	A solid color or pattern fill.
VtFillStyleGradient	A gradient fill.

FontEffect Constants

Description

VtFontEffect provides methods of altering fonts for the desired effect.

The following table lists the valid constants for **VtFontEffect**:

Constant	Description
VtFontEffectStrikeThrough	Applies the strike-through attribute to the font.
VtFontEffectUnderline	Applies the underscore attribute to the font.

FontStyle Constants

Description

VtFontStyle provides valid font attribute options.

The following table lists the valid constants for **VtFontStyle**:

Constant	Description
VtFontStyleBold	Applies the bold attribute to the font.
VtFontStyleItalic	Applies the italic attribute to the font.

VtFontStyleOutline	Applies the outline attribute to the font.

FrameStyle Constants

Description

VtFrameStyle provides options for displaying backdrop frames.

The following table lists the valid constants for **VtFrameStyle:**

Constant	Description
VtFrameStyleNull	No frame.
VtFrameStyleSingleLine	A Single line encloses the backdrop.
VtFrameStyleDoubleLine	Two equal width lines enclose the backdrop.
VtFrameStyleThickInner	A thick inner line and a thin outer line enclose the backdrop.
VtFrameStyleThickOuter	A thin inner line and a thick outer line enclose the backdrop.

GradientStyle Constants

Description

VtGradientStyle provides methods of displaying chart gradients.

The following table lists the valid constants for **VtGradientStyle**::

Constant	Description
VtGradientStyleHorizontal	The color changes from top to bottom.
VtGradientStyleVertical	The color changes from left to right.
VtGradientStyleRectangle	The color changes in concentric rectangles from the center outwards.
VtGradientStyleOval	The color changes in concentric ovals from the center outwards.

HorizontalAlignment Constants

Description

VtHorizontalAlignment provides options for text alignment.

The following table lists the valid constants for **VtHorizontalAlignment**:

Constants	Description
VtHorizontalAlignmentLeft	All lines of text are aligned on the left margin.
VtHorizontalAlignmentRight	All lines of text are aligned on the right margin.
VtHorizontalAlignmentCenter	All lines of text are centered horizontally.

LabelComponent Constants

Description

VtChLabelComponent provides options for displaying chart labels.

The following table lists the valid constants for **VtChLabelComponent:**

Constants	Description
VtChLabelComponentValue	The value of the data point appears in the label. Data points in XY, Polar, and Bubble charts actually have two or three values. The default label for these chart types display all values in a standard format. You can customize this format to highlight an individual data value if you so desire.
VtChLabelComponentPercent	The value of the data point is displayed in the label as a percentage of the total value of the series.
VtChLabelComponentSeriesName	The series name is used to label the data point. This name is taken from the label associated with the column in the data grid.
VtChLabelComponentPointName	The data point name is used to label the data point.

LabelLineStyle Constants

Description

VtChLabelLineStyle provides options for displaying lines connecting a label and series.

The following table lists the valid constants for **VtChLabelLineStyle:**

Constants	Description
VtChLabelLineStyleNone	No line connects the label and series.
VtChLabelLineStyleStraight	A straight line connects the label and series.
VtChLabelLineStyleBent	A bent line connects the label and series.

LabelLocationType Constants

Description

VtChLabelLocationType provides options for determining series label location.

The following table lists the valid constants for **VtChLabelLocationType:**

Constants	Description
VtChLabelLocationTypeNone	No label displayed.
VtChLabelLocationTypeAbovePoint	The label is displayed above the.data point.
VtChLabelLocationTypeBelowPoint	The label is displayed below the data point.
VtChLabelLocationTypeCenter	The label is displayed centered on the data point.
VtChLabelLocationTypeBase	The label is displayed at the base along the category axis, directly beneath the data point..
VtChLabelLocationTypeInside	The label is displayed inside a pie slice.
VtChLabelLocationTypeOutside	The label is displayed outside a pie slice..

VtChLabelLocationTypeLeft	The label is displayed to the left of the data point.
VtChLabelLocationTypeRight	The label is displayed to the right of the data point..

LocationType Constants

Description

VtChLocationType provides location options for chart elements.

The following table lists the valid constants for **VtChLocationType**:

Constants	Example
VtChLocationTypeTop	Top
VtChLocationTypeTopLeft	TopLeft
VtChLocationTypeTopRight	TopRight
VtChLocationTypeLeft	Left
VtChLocationTypeRight	Right
VtChLocationTypeBottom	Bottom
VtChLocationTypeBottomLeft	BottomLeft
VtChLocationTypeBottomRight	BottomRight
VtChLocationTypeCustom	Custom

MarkerStyle Constants

Description

VtMarkerStyle provides options for displaying datapoint markers.

The following table lists the valid constants for **VtMarkerStyle**:

Constants	Example
VtMarkerStyleNull	**suppressed**
VtMarkerStyleDash	
VtMarkerStylePlus	
VtMarkerStyleX	
VtMarkerStyleStar	
VtMarkerStyleCircle	
VtMarkerStyleSquare	
VtMarkerStyleDiamond	
VtMarkerStyleUpTriangle	
VtMarkerStyleDownTriangle	
VtMarkerStyleFilledCircle	
VtMarkerStyleFilled Square	
VtMarkerStyleFilledDiamond	

VtMarkerStyleFilledUpTriangle

VtMarkerStyleFilledDownTriangle

VtMarkerStyle3dBall

MouseFlag Constants

Description

VtChMouseFlag indicates what keyboard keys being held down while the mouse button is clicked.

The following table lists valid constants for **VtChMouseFlag**:

Constants	Description
VtChMouseFlagShiftKeyDown	The Shift key is held down when the mouse button is clicked.
VtChMouseFlagControlKeyDown	The Control key is held down when the mouse button is clicked.

Orientation Constants

Description

VtOrientation provides options for positioning text.

The following table lists the valid constants for **VtOrientation**:

Constants	Description
VtOrientationHorizontal	The text is displayed horizontally.
VtOrientationVertical	The letters of the text are drawn one on top of each other from the top down.
VtOrientationUp	The text is rotated to read from bottom to top.
VtOrientationDown	The text is rotated to read from top to bottom.

PartType Constants

Description

VtChPartType provides options for chart elements.

The following table lists the valid constants for **VtChPartType:**

Constants	Description
VtChPartTypeChart	Identifies the chart control.
VtChPartTypeTitle	Identifies the chart title.
VtChPartTypeFootnote	Identifies the chart footnote.
VtChPartTypeLegend	Identifies the chart legend.
VtChPartTypePlot	Identifies the chart plot.
VtChPartTypeSeries	Identifies a chart series.
VtChPartTypeSeriesLabel	Identifies a series label.
VtChPartTypePoint	Identifies an individual data point.
VtChPartTypePointLabel	Identifies a data point label.
VtChPartTypeAxis	Identifies an axis.
VtChPartTypeAxisLabel	Identifies an axis label.
VtChPartTypeAxisTitle	Identifies an axis title.

PercentAxisBasis Constants

Description

VtChPercentAxisBasis provides methods of displaying percentage axes.

The following table lists the valid constants for **VtChPercentAxisBasis:**

Constant	Description
VtChPercentAxisBasisMaxChart	The largest value in the chart is considered 100 percent and all other values on the chart are displayed as percentages of

	that value.
VtChPercentAxisBasisMaxRow	The largest value in each row is considered 100 percent and all other values in that row are displayed as percentages of that value.
VtChPercentAxisBasisMaxColumn	The largest value in each series is considered 100 percent and all other values in that series are displayed as percentages of that value.
VtChPercentAxisBasisSumChart	All values in the chart are added together, and that value is considered 100 percent. All other values are displayed as percentages of that value.
VtChPercentAxisBasisSumRow	All values in each row are added together and the total value for each row is considered 100 percent. All other values in that same row are displayed as percentages of that value. This is the basis for 100 percent stacked charts.
VtChPercentAxisBasisSumColumn	All values in each series are added together to give a total value for each series. All values are displayed as a percentage of their series total value.

PenCap Constants

Description

VtPenCap provides methods for displaying line endings.

The following table lists the valid constants for **VtPenCap**:

Constant	Description
VtPenCapButt	The line is squared off at the endpoint.
VtPenCapRound	A semicircle with the diameter of the line thickness is drawn at the end of the line.
VtPenCapSquare	The line continues beyond the endpoint for a distance equal to half the line thickness and is squared off.

PenJoin Constants

Description

VtPenJoin provides options for joining line segments in a series.

The following table lists the valid constants for **VtPenJoin**:

Constant	Description
VtPenJoinMiter	The outer edges of the two lines are extended until they meet.
VtPenJoinRound	A circular arc is drawn around the point where the two lines meet.
VtPenJoinBevel	The notch between the ends of two joining lines is filled.

PenStyle Constants

Description

VtPenStyle provides options for the pen used to draw chart lines.

The following table lists the valid constants for **VtPenStyle**:

Constant	Pattern
VtPenStyleNull	No pen is applied.
VtPenStyleSolid	
VtPenStyleDashed	
VtPenStyleDotted	
VtPenStyleDashDot	
VtPenStyleDashDotDot	
VtPenStyleDitted	
VtPenStyleDashDit	
VtPenStyleDashDitDit	

PictureOptions Constants

Description

VtPictureOptions provides options for saving the chart as a graphic.

The following table lists the valid constants for **VtPictureOptions**:

Constants	Description
VtPictureOptionNoSizeHeader	Does not save the Adobe Placeable header information with the metafile. Do not use this option if you intend to import the metafile into a Microsoft application such as Word or Excel. These applications expect size information to be included in metafiles.
VtPictureOptionTextAsCurves	Saves chart text as curves instead of text. Use this option if you are using an unusual font that may not be present on other systems that will be using the metafile. This option should also be used to properly display text if you are deforming the metafile by stretching it.

PictureMapType Constants

Description

VtPictureMapType describes how the picture is displayed.

The following table lists the valid constants for **VtPictureMapType**:

Constants	Description
VtPictureMapTypeActual	Displays the graphic at the original size it was created.
VtPictureMapTypeFitted	Scales the graphic proportionally to fit within the object.
VtPictureMapTypeStretched	Scales the graphic to fill the object regardless of its orininal proportions.

VtPictureMapTypeTiled	Duplicates the graphic repeatedly to fill the object.
VtPictueMapTypeCropFitted	Centers the graphic and scales it proportionally to fill the object. Since the original aspect ratio is maintained, any part of the image that falls outside the object is cropped.

PictureType Constants

Description

VtPictureType provides types of graphic files.

The following table lists the valid constants for **VtPictureType**:

Constants	Description
VtPictureTypeNull	No graphic.
VtPictureTypeBMP	Windows bitmap.
VtPictureTypeWMF	Microsoft Windows MetaFile. This metafile contains Adobe placeable header information that indicates the size of the picture.

PieWeightBasis Constants

Description

VtChPieWeightBasis provides options for displaying pie chart slices.

The following table lists the valid constants for **VtPieWeightBasis**:

Types	Description
VtChPieWeightBasis None	All pies are drawn the same size.
VtChPieWeightBasisTotal	The slice values in each pie are totaled and the pie with the highest total identified. The size of each pie in the chart is determined by the ratio of its total value compared to the largest pie.
VtChPieWeightBasisSeries	The first column of data in the data grid holds the relative size index. In other words, if you have 5 categories, you can control the size of the pies representing each category by using the first column of the data grid to number the rows 1 through 5. The size of the pie is determined by the ratio of its first column value and the largest value in the first column. The pie containing the 1 is the largest pie; the one containing the 5 the smallest. It is most common to exclude this first column of data so that the values are not drawn as a pie slice.

PieWeightStyle Constants

Description

VtChPieWeightStyle provides options for displaying individual pies within a single chart.

The following table lists the valid constants for **VtChPieWeightStyle**:

Constant	Description

VtChPieWeightStyleArea	The area of the individual pies changes based on their weighting.
VtChPieWeightStyleDiameter	The diameter of the individual pies changes based on their weighting.

PrintOrientation Constants

Description

VtPrintOrientation provides options for displaying the chart for printing.

The following table lists the valid constants for **VtPrintOrientation**:

Constant	Description
VtPrintOrientationPortrait	The chart is printed across the short side of the paper.
VtPrintOrientationLandscape	The chart is rotated to print across the long side of the paper.

PrintScaleType Constants

Description

VtPrintScaleType provides methods of scaling the chart for printing.

The following table lists the valid constants for **VtPrintScaleType**:

Constant	Description
VtPrintScaleTypeActual	The chart is printed at the original size it was created.
VtPrintScaleTypeFitted	The chart is scaled proportionally to fit the page.
VtPrintScaleTypeStretched	The chart is scaled to fit the page, regardless of its original proportions.

ProjectionType Constants

Description

VtProjectionType provides viewpoint and perspective options for displaying and viewing a chart.

The following table lists the valid constants for **VtProjectionType**:

Constants	Description
VtProjectionTypePerspective	This provides the most realistic 3D appearance. Objects farther away from you converge toward a vanishing point. This is the default projection.
VtProjectionTypeOblique	This is sometimes referred to as 2.5 dimensional. The chart does have depth, but the XY plane does not change when the chart is rotated or elevated.
VtProjectionTypeOrthogonal	Perspective is not applied in this 3D view. The major advantage of using this type of projection is that vertical lines remain vertical, making some charts easier to read.

ScaleType Constants

Description

VtChScaleType provides methods for plotting chart values and displaying the chart scale.

The following table lists the valid constants for **VtChScaleType:**

Constant	Description
VtChScaleTypeLinear	Chart values are plotted in a linear scale with values ranging from the minimum to the maximum chart range value.
VtChScaleTypeLogarithmic	Chart values are plotted in a logarithmic scale with values based on a specific log scale set with the logBase argument of this function.
VtChScaleTypePercent	Chart values are plotted in a linear scale with values based on the percentages of the chart range values.

SeriesType Constants

Description

VtChSeriesType provides options for types of series.

The following table lists the valid series constants for **VtChSeriesType**:

Constant	Series Type
VtChSeriesType3dBar	3D Bar
VtChSeriesType2dBar	2D Bar
VtChSeriesType3dHorizontalBar	3D Horizontal Bar
VtChSeriesType2dHorizontalBar	2D Horizontal Bar
VtChSeriesType3dClusteredBar	3D Clustered Bar
VtChSeriesType3dLine	3D Line
VtChSeriesType2dLine	2D Line
VtChSeriesType3dArea	3D Area
VtChSeriesType2dArea	2D Area
VtChSeriesType3dStep	3D Step
VtChSeriesType2dStep	2D Step
VtChSeriesType2dXY	XY
VtChSeriesType2dPolar	Polar
VtChSeriesType2dRadarLine	Radar Line
VtChSeriesType2dRadarArea	Radar Area
VtChSeriesType2dBubble	Bubble
VtChSeriesType2dHiLo	Hi-Lo
VtChSeriesType2dHLC	Hi-Lo Close
VtChSeriesType2dHLCRight	Hi-Lo-Close with Close Marker to the right.

VtChSeriesType2dOHLC	Open-Hi-Lo-Close
VtChSeriesType2dOHLCBar	Open-Hi-Lo-Close Bar
VtChSeriesType2dGantt	2D Gantt
VtChSeriesType3dGantt	3D Gantt
VtChSeriesType3dPie	3D Pie
VtChSeriesType2dPie	2D Pie
VtChSeriesType3dDoughnut	Doughnut
VtChSeriesType2dDates	Dates
VtChSeriesType3dBarHiLo	Floating 3D Bars
VtChSeriesType2dBarHiLo	Floating 2D Bars
VtChSeriesType3dHorizontalBarHiLo	Floating 3D Horizontal Bars
VtChSeriesType2dHorizontalBarHiLo	Floating 2D Horizontal Bars
VtChSeriesType3dClusteredBarHiLo	Floating 3D Clustered Bars
VtChSeriesType3dSurface	3D Surface
VtChSeriesType2dContour	2D Contour
VtChSeriesType3dXYZ	3D XYZ

ShadowStyle Constants

Description

VtShadowStyle provides shadow options.

The following table lists the valid constants for **VtShadowStyle**:

Constant	Description
VtShadowStyleNull	No shadow.
VtShadowStyleDrop	Drop shadow.

SmoothingType Constants

Description

VtSmoothingType provides methods for smoothing chart display data.

The following table lists the valid constants for **VtSmoothingType:**

Constant	Description
VtSmoothingTypeNone	No smoothing is applied to the data.
VtSmoothingTypeQuadraticBSpline	A quadratic B-spline formula is used to determine the smoothing applied to the data. This form of smoothing results in a less-smooth curve that stays closer to the data points.
VtSmoothingTypeCubicBSpline	A cubic B-spline formula is used to determine the smoothing applied to the data. This form of smoothing

	results in a smoother curve, but varies further from the data point than a QuadraticBspline curve.

SortType Constants

Description

VtSortType provides options for sorting pie charts.

The following table lists the valid constants for **VtSortType**:

Constants	Description
VtSortTypeNone	Pie slices are drawn in the order the data appears in the data grid.
VtSortTypeAscending	Pie slices are drawn, in order, from the smallest to the largest slice, starting at the defined starting angle and in the defined plot direction.
VtSortTypeDescending	Pie slices are drawn, in order, from the largest to the smallest slice, starting at the defined starting angle and in the defined plot direction.

SsLinkMode Constants

Description

VtChSsLinkMode provides methods for linking and maintaining a connection to a Formula One spreadsheet.

The following table lists the valid constants for **VtChSsLinkMode**:

Constants	Description
VtChSsLinkModeOff	The connection to the spreadsheet is not active.
VtChSsLinkModeOn	The spreadsheet is active.First Impression makes no attempt to interpret the spreadsheet data. It uses the values set by the Column, Row, ColumnLabelCount, and RowLabelCount properties to determine the data grid dimensions and then fills those areas with data from the spreadsheet.
VtChSsLinkModeAutoParse	The spreadsheet connection is active. First Impression examines the spreadsheet data and tries to determine what is a label and what is data. It determines what it thinks the dimensions of the data grid should be and adjusts the values of the Column, Row, ColumnLabelCount, and RowLabelCount properties accordingly.

StatsType Constants

Description

VtChStats provides methods of displaying statistic lines on a chart.

The following table lists the valid constants for **VtChStats**:

Constant	Description
VtChStatsMinimum	Shows the minimum value in the series.
VtChStatsMaximum	Shows the maximum value in the series.

VtChStatsMean	Shows the mathematical mean of the values in the series
VtChStatsStddev	Shows the standard deviation of the values in the series
VtChStatsRegression	Shows a trend line indicated by the values in a series.

SubPlotLabelLocationType Constants

Description

VtSubPlotLabelLocationType provides methods for displaying the subplot label.

The following table lists the valid constants for **VtSubPlotLabelLocationType**:

Constants	Description
VtChSubPlotLabelLocationTypeNone	No subplot label is displayed.
VtChSubPlotLabelLocationTypeAbove	The subplot label is displayed above the pie.
VtChSubPlotLabelLocationTypeBelow	The subplot label is displayed below the pie.
VtChSubPlotLabelLocationTypeCenter	The subplot label is centered on the pie.

SurfaceBaseType Constants

Description

VtChSurfaceBaseType provides methods for displaying the base of surface charts.

The following table lists the valid constants for **VtChSurfaceBaseType**

Constant	Description
VtChSurfaceBaseTypePedestal	The chart is displayed with a base that extends all the way to the surface.
VtChSurfaceBaseTypeStandard	The chart is displayed with a base like those in 3D bar charts.
VtChSurfaceBaseTypeStandardWithCBands	The chart is displayed as a standard base with contour bands on its top.
VtChSurfaceBaseTypeStandardWithCLines	The chart is displayed as a standard base with contour lines on its top.

SurfaceDisplayType Constants

Description

VtChSurfaceDisplayType provides methods for displaying chart surfaces.

The following table lists the valid constants for **VtChSurfaceDisplayType**:

Constant	Description
VtChSurfaceDisplayTypeNone	No surface is displayed.
VtChSurfaceDisplayTypeCBands	The surface is displayed with contour bands.
VtChSurfaceDisplayTypeCLines	The surface is displayed with contour lines.
VtChSurfaceDisplayTypeSolid	The surface is displayed with the surface brush.

VtChSurfaceDisplayTypeSolidWithCLines	The surface is displayed with the surface brush overlaid with contour lines.

SurfaceProjectionType Constants

Description

VtChSurfaceProjectionType provides options for projecting contours above a chart surface.

The following table lists the valid constants for **VtChSurfaceProjectionType**:

Constant	Description
VtChSurfaceProjectionTypeNone	No projection displayed.
VtChSurfaceProjectionTypeCBands	Contour bands are projected above the surface.
VtChSurfaceProjectionTypeCLines	Contour lines are projected above the surface.

SurfaceWireframeType Constants

Description

VtChSurfaceWireframeType provides methods for displaying wireframe surfaces.

The following table lists the valid constants for **VtChSurfaceWireframeType**:

Constant	Description
VtChSurfaceWireframeTypeNone	No wireframe is displayed.
VtChSurfaceWireframeTypeMajor	The wireframe is based on major divisions in the axis.
VtChSurfaceWireframeTypeMajorAndMinor	The wireframe is based on major and minor divisions in the axis.

TextLengthType Constants

Description

VtTextLengthType provides options for optimizing text layout for the screen or for the printer.

The following table lists the valid constants for **VtTextLengthType**:

Constants	Description
VtTextLengthTypeVirtual	Choose this constant to use TrueType virtual font metrics to optimize text layout for printing. TrueType virtual font metrics may not be very accurate for text displayed on the screen. Text displayed on the screen may be a larger or smaller than the virtual metrics requested. Larger text may not fit where it is supposed to and part of a character, a whole character, or even in some cases words may be clipped.
VtTextLengthTypeDevice	Choose this constant to optimize text layout for the screen. Text in charts laid out for screen display always fits correctly within its chart area. The printed text is generally a bit smaller and so the text may appear in slightly different places.

TextOutputType Constants

Description

VtTextOutputType provides methods of outputting text.

The following table lists the valid constants for **VtTextOutputType**:

Constants	Description
VtTextOutputTypeHardware	Device context type Null.
VtTextOutputTypePolygon	Device context type Metafile.

VerticalAlignment Constants

Description

VtVerticalAlignment provides methods of vertically aligning text.

The following table lists the valid constants for **VtVerticalAlignment**:

Constants	Description
VtVerticalAlignmentTop	All lines of text are aligned at the top margin.
VtVerticalAlignmentBottom	All lines of text are aligned at the bottom margin.
VtVerticalAlignmentCenter	All lines of text are centered vertically.

Handling Errors

First Impression errors that occur during program execution are handled like other errors. You must write your own error handling routine to trap and manage errors.

The **ErrorOffset** property sets or returns an adjustment to trappable error numbers returned by First Impression. The adjustment is the amount by which First Impression error numbers should by incremented so that the numbers do not conflict with other control errors in the container. By default, the ErrorOffset value is set to 20000.

Use this property only if the First Impression error range conflicts with values returned by the Err statement.

The following table list the errors that can be caught by exception handling.

Error Name	Description
VtFail	A general failure error.
VtErrorDeletingUsedObject	An internal error indicating that First Impression attempted to delete an object that is still in use.
VtErrorDeletingDeletedObject	An internal error indicating that First Impression attempted to delete an object that has already been deleted.
VtErrorCorruptData	An internal error indicating that First Impression has encountered corrupt data and is unable to continue the requested action.
VtErrorNotImplemented	Feature is not yet implemented.
VtErrorNoMemory	There is not enough memory to accomplish the requested action.
VtErrorInvalidArgument	One of the function arguments was bad.
VtErrorNotFound	Cannot find the item for which the function is looking.
VtErrorTooSmall	The returned result is longer than the return buffer size. A NULL string is placed in the buffer.
VtErrorInvalidRequest	The requested action is inappropriate or otherwise invalid.
VtErrorStreamIo	An error was encountered when trying to read or write a file.
VtErrorUserIo	An attempt to write a file to a user created function failed.
VtErrorCorruptArchive	Persistent storage is corrupted.
VtErrorArchiveVersion	Tried to load incompatible version.
VtErrorArchiveTypeMismatch	Encountered unexpected class type.
VtErrorArchivePointerMismatch	Tried to load a pointer into a reference.
VtErrorCannotOpenFile	Unable to open the requested file.
VtErrorUnableToLoadString	Unable to load a string, the string may have been too long.
VtErrorBufferTooSmall	String buffer is not large enough to hold the requested string. String has been truncated to fit the buffer.
VtErrorDisplay	First Impression is unable to properly communicate with display drivers.
VtErrorInvalidFontName	Requested font name is not available.
VtErrorInvalidFont	Requested font description is invalid.
VtErrorNoDisplayResources	First Impression does not have enough display resources to properly display the chart.
VtChError	Function failed with a non-specific error.
VtChErrorInvalidHandle	Invalid chart handle passed.
VtChErrorNoData	Data necessary for the completion of the task is unavailable.

VtChErrorInvalidSeriesNumber	One of the arguments in a function is a reference to a series that does not exist on the chart.
VtChErrorInvalidAxis	One of the arguments in a function is a reference to an axis that does not exist on the chart.
VtErrorRestrictedVersion	This feature not supported by the Borland Edition of First Impression.