3D Graph Examples

Measurement Studio delivers several examples for 3D Graph. Click on the following links for example descriptions and locations:

Axis (3D Graph)

Contours

Cursors (3D Graph)

Mouse Events (3D Graph)

Plot Styles

Region of Interest (3D Graph)

Simple 3D Graph

Spinning Light

Transparency



Graph3D Overview

Plots three-dimensional data on a user interface. For many real-world data sets, such as terrain contours, the motion of an airplane in three dimensions, the temperature distribution on a surface, and joint time-frequency analysis, you must visualize the data in three dimensions. With the Measurement Studio 3D Graph control, you can visualize three-dimensional data and modify the way that data appears by modifying graph, plot, and contour properties.

- Multiple plot styles: point, line, line-point, hidden-line, contour, surface, surface-line, surface-contour, and surface-normal
- Multiple plots with individual properties such as name, fill color, line style, and contour levels
- Configurable axes, including customizable ticks, labels, value pairs, captions, and autoscaling
- Cartesian, cylindrical, and spherical coordinate systems
- Custom color maps
- Transparency
- Plane projections
- Orthographic and perspective viewing
- Lighting
- Built-in format styles for labels, including scientific, symbolic engineering, scaling, time, and date
- Rotation, panning and zooming at runtime. Panning is useful when the graph displays only a subset of the data that has been plotted. You can scroll through all data plotted on the graph, essentially shifting the graph's display to different portions of the plot. You can use zooming to enlarge or diminish a portion of the plot displayed by the graph. For example, if you zoom on a section of a plot, the graph displays a smaller portion of the plot in the same amount of display area, which enlarges the detail of that section.
- 3D hardware acceleration
- Triangulation of arbitrarily arranged data (Plot3DMesh method)

Notes

• In addition to the examples and documentation included with the Measurement Studio Reference, go to the National Instruments Developer Zone, which provides access to the latest example programs, tutorials, and technical news, as well as a community of developers ready to share their techniques.

Tips

- To get information about any of the properties in the 3D graph control's property pages, right click on the property and select **What's This?**. For complete reference information about this control and its properties, click on the Visual Basic or Visual C++ Reference link.
- In Visual C++, if you select the **ALL** tab in the property pages, the What's This? help is disabled.

CWGraph3D

Rotating, Panning, and Zooming (concept)

Simple 3D Graph Example

CWAxes3D

Standard collection containing all CWAxis3D objects of the graph. A graph has one x-, y-, and z-axis. These different CWAxis3D objects are contained in the CWAxes3D collection and can be referenced by index. The x-axis is at index 1, the y-axis is at index 2, and the z-axis is at index 3.

Count

Returns the number of objects in the collection.

Methods

<u>Item</u>

Returns the specified object from the collection.

CWGraph3D.Axes
CWAxis3D

CWAxis3D

The CWAxis3D object defines an axis for the CWGraph3D control. The CWAxis3D object's properties determine how data values are scaled to the range displayed on the control, where and if ticks and grid lines are drawn, and what kinds of labels are placed on the axis.

<u>AutoScale</u> Determines if the minimum and maximum limits of the axis are automatically set.

<u>Caption</u> Specifies the title for the axis.

<u>CaptionColor</u> Specifies the color of the axis caption.

<u>CaptionFont</u> Specifies the font of the axis caption.

<u>CaptionNormal</u> Draws the axis caption at the normal position when set to True.

<u>CaptionOpposite</u> Draws the axis caption at the opposite position when set to True.

<u>FormatString</u> Specifies the format string for formatting the labels on this axis.

<u>Inverted</u> Specifies if the direction of an axis is inverted.

<u>Labels</u> Returns a CWLabels3D object, which specifies how labels appear on the axis.

Log Specifies if the axis has a Log10 scale.

Maximum Specifies the maximum value of the axis.

Minimum Specifies the minimum value of the axis.

Name Specifies the name of the axis.

<u>Ticks</u> Returns a CWTicks3D object, which specifies how divisions and ticks appear on this axis.

<u>ValuePairs</u> Returns a CWValuePairs collection of CWValuePair objects, which specify labels for particular

points on the axis.

<u>Visible</u> Specifies if the axis is visible or hidden.

Methods

<u>AutoScaleNow</u> Rescales the axis.

<u>SetMinMax</u> Sets both the minimum and the maximum values of the axis at the same time.

CWAxes3D

Axis (3D Graph) Example

CWContour

A CWContour defines a single contour for a plot.

<u>LabelColor</u> Specifies the color of the contour label.

<u>LabelFont</u> Specifies the contour label font.

<u>LabelFormat</u> Specifies the format string for formatting the label on this contour.

<u>LabelVisible</u> Specifies if the contour line has a visible label.

Level Specifies the number of contour levels.

LineColor Specifies the color of the contour line.

LineStyle Specifies the line style of the contour line.

<u>LineWidth</u> Specifies the width of the contour line in points.

CWContours

Contours Example

CWContours

CWContours is a collection of CWContour objects for a plot. It also has properties that affect the whole collection and methods that modify a property on all CWContour objects in the collection.

Anchor Specifies the value of the contour anchor.

AnchorEnabled Enables the contour anchor if set to True.

Basis Specifies the data coordinate that the contour levels are calculated from.

Count Returns the number of objects in the collection.

Interval Specifies the distance between each contour level.

LevelList Adds a contour level at a specified value on the plot.

Levels Specifies the number of contour levels for the plot.

Methods

Adds a contour at the specified level.

Remove Returns the specified object from the collection.

Remove Removes the specified item from the collection.

Removes all objects from the collection.

Sets the label color for all contours in the collection.

SetLabelFont Sets the label font for all contours in the collection.

SetLabelFormatSets the label format string for all contours in the collection.SetLabelVisibleSets the label visibility for all contours in the collection.SetLineColorSets the line color for all contours in the collection.SetLineStyleSets the line style for all contours in the collection.

<u>SetLineWidth</u> Sets the line width for all contours in the collection.

CWContour

CWPlot3D.Contours

CWCursor3D

The CWCursor3D object displays a crosshair on a CWGraph3D to mark a specific point or region on the graph or highlight data programmatically. CWCursor properties define the position and appearance of the cursor. The cursor position corresponds to the coordinate space of the associated plot or to the CWGraph3D.PlotTemplate if the cursor is not associated with a specific plot.

To interact with the cursors while the program is running, set the TrackMode property to cwG3DTrackZoomPanRotate or cwG3DTrackCursors.

<u>Column</u> Specifies the column index of the plot data point that the cursor is snapped

to.

Enabled Specifies if you can drag the cursor.

Font Specifies the font for the cursor text.

LineColor Specifies the color of the cursor lines.

LineStyle Specifies the style of the cursor lines.

LineWidth Specifies the width of the cursor lines.

Specifies the name of the cursor.

<u>NameVisible</u> Specifies if the plot name is draw next to the cursor.

<u>PlaneColor</u> Specifies the color of the cursor planes.

<u>PlaneTransparency</u> Specifies the transparency of the cursor planes.

<u>Plot</u> Specifies the plot that the cursor is snapped to.

PointColorSpecifies the color of the cursor point.PointSizeSpecifies the size of the cursor point.PointStyleSpecifies the style of the cursor point.

<u>PositionVisible</u> Specifies if the cursor position is displayed next to the cursor.

Specifies the row index of the plot data point that the cursor is snapped to.

SnapMode
Specifies if the cursor can be dragged and how it snaps to the plot data.

<u>TextBackColor</u> Specifies the color to paint behind the cursor text.

<u>TextBackgroundTransparency</u> Specifies how transparent to paint a box behind the cursor text.

TextColor Specifies the color of the cursor text.

Visible Specifies if the cursor is visible.

XPosition, YPosition, ZPosition Specifies the position of the cursor.

XYPlaneVisible, XZPlaneVisible, Specifies if the cursor planes are visible.

<u>YZPlaneVisible</u>

Methods

Sets the line, plane, point, and text colors of the cursor.

 $\underline{\text{SetPosition}} \qquad \qquad \text{Sets the x, y, and z position of the cursor.}$

CWCursors3D

CWCursors3D is a collection of CWCursor3D objects.

Count

Returns the number of objects in the collection.

Methods

Adds an object to the collection and returns the new object.

ItemReturns the specified object from the collection.RemoveRemoves the specified item from the collection.

RemoveAll Removes all objects from the collection.

CWCursor3D

CWGraph3D

The CWGraph3D object has properties that affect the overall appearance of the control. It also has collections of axes, plots, and lights. To set properties for a specific part of the control, set the properties on the object corresponding to that part.

AmbientLightColor Specifies the ambient light color for the graph.

Axes Specifies a collection of CWAxis3D objects.

<u>BackColor</u> Specifies the color for the background of the caption.

Caption Specifies the caption for the 3D graph.
CaptionColor Specifies the color of the caption text.

<u>ClipData</u> Indicates if the data is clipped to the axes ranges.

<u>Cursors</u> Specifies a collection of CWCursor3D objects.

<u>Dither</u> Specifies the dither mode of the graph.

<u>Enabled</u> Specifies if the graph generates any events.

<u>FastDraw</u> Specifies how the graph is drawn during interactive operations such as rotating, zooming,

and panning.

<u>Font</u> Specifies the font for the caption.

GraphFrameColor Specifies the color for the graph frame.

GraphFrameVisible Specifies if the graph frame is visible.

GridFrameColor Specifies the color of the graph's grid frame.

GridSmoothing Specifies the smoothing mode for grid lines.

GridXY, GridXZ, GridYZ GridXY draws the XY grid plane when set to True.

GridXZ draws the XZ grid plane when set to True.

GridYZ draws the YZ grid plane when set to True.

<u>ImmediateUpdates</u> Specifies if the graph draws new data as soon as it is available, or if the form refreshes

the graph when it draws other controls.

<u>KeyboardMode</u> Specifies how the control handles keyboard input from the user.

<u>Lighting</u> Enables graph lighting when set to True.

<u>Lights</u> Specifies a collection of CWLights.

<u>PlotAreaColor</u> Specifies the background color of the plot area.

<u>Plots</u> Specifies a collection of CWPlot3D objects.

<u>PlotTemplate</u> Returns the CWPlot3D object to use as a template for new plots.

<u>ProjectionStyle</u> Specifies the projection style of the graph.

ReadyState Returns the ready state.

<u>TrackMode</u> Determines the type of events generated and other automatic processing, such as how the

mouse interacts with the graph during run time.

<u>Use3DHardwareAcceleration</u> Determines if the 3D Graph control renders directly to the display adapter.

<u>ViewAutoDistance</u> Specifies how the viewing distance is set.

<u>ViewDistance</u> Specifies the distance of the viewing position from the center of the view as defined by

ViewXCenter, ViewYCenter, and ViewZCenter.

<u>ViewLatitude</u> Specifies the latitude in degees of the viewing position.

<u>ViewLongitude</u> Specifies the longitude in degrees of the viewing position.

<u>ViewMode</u> Specifies the viewing position of the graph.

ViewXCenter, ViewYCenter, ViewZCenter Specify the X, Y, and Z coordinates of the center of the view.

Specifies if the control has a window. Windowless

Methods

AboutBox Displays the About Box for the control.

<u>ClearData</u> Clears data in all plots.

<u>ControlImage</u> Returns an image of the entire control.

ExportStyle Exports the style of the control to a .cwx file.

<u>ImportStyle</u> Imports a previously exported style to the control.

Plot3DCurve Plots three (or four) 1D arrays of data as a parametric curve.

Plot3DMesh Plots three (or four) 1D arrays of data as a triangulated surface.

Plot3DParametricSurface Plots three (or four) 2D arrays of data as a parametric surface.

<u>Plot3DSimpleSurface</u> Plots one (or two) 2D array(s) of data as a surface.

<u>Plot3DSurface</u> Plots two 1D arrays and one (or two) 2D array(s) of data as a surface.

<u>SetDefaultView</u> Restores the default viewing parameters after panning, zooming, or rotating the graph.

Events

<u>Click</u> Generates a Click event when you click the mouse on the control.

<u>CursorChange</u> Generates the CursorChange event when you reposition a cursor with

the mouse.

<u>DblClick</u> Generates a DblClick event when you double-click the mouse on the

control.

<u>KeyDown, KeyUp</u> Generates the KeyUp event when you release a key while the control

has the input focus.

Generates the KeyDown event when you press a key while the control

has the input focus.

KeyPress Generated when the control has focus and you press a key.

MouseDown, MouseMove, MouseUp Generates the MouseDown event when you click the mouse on the

control.

Generates the MouseMove event when you move the mouse over the

control.

Generates the MouseUp event when you release the mouse on the

control

Pan Generates the Pan event when you pan the graph up and down or left

and right.

PlotAreaMouseDown, PlotAreaMouseMove,

<u>PlotAreaMouseUp</u>

Generates PlotAreaMouseDown when you click the mouse on the plot

area.

Generates PlotAreaMouseMove when you move the mouse over the

plot area.

Generates PlotAreaMouseUp when you release the mouse over the plot

area.

PlotMouseDown, PlotMouseMove, PlotMouseUp Generates PlotMouseDown when you click the mouse on a plot.

Generates PlotMouseMove when you move the mouse over a plot.

Generates PlotMouseUp when you release the mouse over a plot.

<u>ReadyStateChange</u>

Generates the event when the ready state changes.

<u>Rotate</u> Generates the Rotate event when you rotate the graph.

Zoom Generates the Zoom event when you zoom in or out on the plot.

Three-Dimensional Plots (concept)

CWLabels3D

Determines how axis labels are drawn. Labels are the numbers displayed next to the ticks. The Labels3D object properties specify where to draw the labels (normal or opposite), the font of the labels, and the color of the labels.

<u>Color</u> Specifies the color of the axis labels.

<u>Font</u> Specifies the font of the axis labels.

Normal Specifies if labels appear at the normal axis position.

Opposite Specifies if labels appear at the opposite axis position.

CWAxis3D.Labels

CWLight

A CWLight object defines a light for the CWGraph3D control. You can use Lights to create highlights and shadows that enhance visualization of 3D data.

Properties

<u>Attenuation</u> Specifies the attenuation style of the light source.

<u>Color</u> Specifies the color of the light source.

<u>Distance</u> Specifies the distance of the light source from the origin of the graph.

<u>Enabled</u> Enables the light when set to True.

Latitude Specifies the latitudinal position in degrees of the light source.

Longitude Specifies the longitudinal position in degrees of the light source.

CWGraph3D.Lights
CWLights

CWLights

CWLights is a standard collection containing exactly four CWLight objects.

Properties

Count

Returns the number of objects in the collection.

Methods

<u>Item</u>

Returns the specified object from the collection.

CWGraph3D.Lights
CWLight

CWPlot3D

Represents an individual plot on the graph. The object contains a number of different properties that determine the display of the plot. Each Plot3D object has a set of Plot methods similar to those of the Graph3D object. Calling these methods directly on the Plot3D object allows you to update a specific plot on the graph.

The PlotTemplate object is a special instance of a Plot3D object used to specify the default property values of new plots. The PlotTemplate object properties are identical to those of the Plot3D object and are set through the property pages or programmatically. The PlotTemplate property values are used as default property values for newly created plots when the Add method is called on the Plots3D collection. New plots use settings from the CWGraph3D. TemplatePlot object.

Properties

AutoScale Specifies if the maximum and minimum of the data in the plot affect the maximum and

minimum of an autoscaling axis.

CacheData Improves the drawing performance by caching intermediate calculations when set to True.

ColorMapAutoScale Specifies if the color map rescales its values when the plot data changes.

ColorMapColors, Specifies a custom color map when used with the ColorMapValues property. Each element in ColorMapValues

the value vector and its corresponding element in the color vector are a value-color pair in the

color map.

ColorMapInterpolate Specifies if the color map interpolates between colors. ColorMapLog

Specifies if the color map interpolates logarithmically. ColorMapStyle Specifies the color map style that is used by the plot.

Contours Specifies a collection of CWContour objects.

CoordinateSystem Specifies the coordinate system of the plot.

Enabled Specifies if the plot generates mouse events when CWGraph3D.TrackMode is set to

cwGTrackAllEvents and the plot is visible.

FillColor Specifies the color to use for drawing the plot surface.

FillStyle Specifies the fill style of the plot.

LineColor Specifies the color of lines for connecting points in the plot. LineStyle Specifies the style of lines for connecting points on a plot.

LineWidth Specifies the width of the plotting line. The width range is 0 to 100.

MultiPlot Determines if the CWGraph3D plot methods can use this plot.

Name Specifies the name of the plot.

PointColor Specifies the color for points on a plot.

PointFrequency Specifies how often points in the plot are drawn.

PointSize Specifies the size of the plot points in points.

PointStyle Specifies the image drawn at each point on a plot.

ProjectionXY, ProjectionXZ, ProjectionXY draws the XY plane projection of the plot when set to True.

ProjectionYZ ProjectionXZ draws the XZ plane projection of the plot when set to True.

ProjectionYZ draws the YZ plane projection of the plot when set to True.

ShowProjectionsOnly Displays only the specified projection of the plot.

Style Specifies the style of the plot.

Transparency Indicates the percentage of transparency of the plot.

Visible Specifies if the plot is visible or hidden.

XAxis, YAxis, ZAxis Specifies the x-, y-, and z-axes for the plot. These are read-only properties.

Methods

<u>ClearData</u> Clears the data currently displayed in the plot.

Plot3DCurve Plots three (or four) 1D arrays of data as a parametric curve.

Plot3DMesh Plots three (or four) 1D arrays of data as a triangulated surface.

Plot3DParametricSurface Plots three (or four) 2D arrays of data as a parametric surface.

Plot3DSimpleSurface Plots one (or two) 2D array(s) of data as a simple surface.

<u>Plot3DSurface</u> Plots two 1D arrays and one (or two) 2D array(s) of data as a 3D surface.

CWPlots3D

 $\underline{CWGraph3D.PlotTemplate}$

PlotTemplate Object (concept)

CWPlots3D

CWPlots3D is a collection of CWPlot3D objects.

Properties

Count

Returns the number of objects in the collection.

Methods

Adds an object to the collection and returns the new object.

ItemReturns the specified object from the collection.RemoveRemoves the specified item from the collection.

RemoveAll Removes all objects from the collection.

CWPlot3D

CWGraph3D.Plots

Three-Dimensional Plots (concept)

CWTicks3D

Specifies how tick marks appear on a particular axis. You can set properties to specify the spacing between ticks as well as major and minor tick selection. The Ticks3D object also controls any grid displayed for a particular axis on the graph. Usually, tick properties are set during design though the property pages. If necessary, you can change them at run time with simple property calls.

Properties

<u>AutoDivisions</u> Specifies if divisions on the specified axis are automatically calculated to display the data.

<u>Inside</u> Specifies if tick marks appear on the inside of the axis plane.

<u>MajorDivisions</u> Specifies the number of major divisions of ticks on a specified axis.

MajorGridSpecifies if major grid lines appear.MajorGridColorSpecifies the color of major grid lines.MajorTickColorSpecifies the color of major ticks.

<u>MajorTicks</u> Specifies if major ticks appear on a specified axis.

<u>MajorUnitsBase</u> Specifies the base number for calculating ticks.

MajorUnitsInterval Specifies the number of units between major divisions.

Minor Divisions Specifies the number of minor divisions for each major division of ticks on a specified axis.

MinorGridSpecifies if minor grid lines appear.MinorGridColorSpecifies the color of minor grid lines.MinorTickColorSpecifies the color of minor ticks.

MinorTicksSpecifies if minor ticks appear on a specified axis.MinorUnitsIntervalSpecifies the number of units between minor divisions.NormalSpecifies if tick marks appear at the normal axis position.OppositeSpecifies if tick marks appear at the opposite axis position.OutsideSpecifies if tick marks appear on the outside of the axis plane.

CWAxis3D.Ticks

CWValuePair

Configures an individual value pair, which consists of a name and a value. Use value pairs on the axis of a graph control as custom ticks, labels, and grid lines. You can specify if the value pair's value or the value pair's index in the collection determines the position of the value pair on the axis and if the graphical representation of the value pair on the axis is its name or value.

Properties

 $\underline{\text{Name}} \hspace{1cm} \textbf{Specifies the name of the value pair.}$

<u>Value</u> Specifies the value of the value pair.

CWValuePairs

CWValuePairs

Standard collection of CWValuePair objects for an axis. Use the ValuePairs collection and ValuePair objects to mark specific points on any axis with a custom label. The ValuePairs collection contains a variable number of ValuePair objects on an axis.

Properties

<u>Count</u> Returns the number of objects in the collection.

<u>GridLines</u> Specifies if grid lines appear at value pair locations.

<u>LabelType</u> Specifies the type of labels to draw for the value pairs.

<u>Location</u> Specifies if value pairs are placed on the axis by their value or by their index.

MajorTicks Specifies if major ticks appear at the location of value pairs.

Methods

Adds an object to the collection and returns the new object.

ItemReturns the specified object from the collection.RemoveRemoves the specified item from the collection.

RemoveAll Removes all objects from the collection.

Swaps two CWValuePair elements, altering their indices.

<u>CWValuePair</u>

Example: Axis (3D Graph)

Explores different axis properties and shows how they interact with each other.

Description

Use this example to determine how different axis scaling properties affect the data you are visualizing. Experiment with the AutoScale, Minimum, Maximum, Inverted, and Log properties. Also see the effects of changing the CWPlot3D.AutoScale property and calling the AutoScaleNow method. This example also explores the properties affecting the axis caption, labels, ticks, and grid.

Controls, Properties, Methods, and Events

This example demonstrates the following controls, properties, methods, and events:

CWGraph3D

Plot3DParametricSurface, Plots, Axes

CWPlot3D

<u>AutoScale</u>

CWAxis3D

<u>AutoScaleNow</u>, <u>Minimum</u>, <u>Maximum</u>, <u>Inverted</u>, <u>Log</u>, <u>Ticks</u>, <u>Caption</u>, <u>Labels</u>, <u>CaptionNormal</u>, <u>CaptionOpposite</u>

MajorGrid, MinorGrid, Inside, MajorTicks, MinorTicks, Normal, Opposite, Outside

Example Location

Samples\UI\3DGraph\Axis

Example: Contours

Explores different plot contour properties and shows how they interact with each other.

Description

Press one of the plot buttons to plot data on both of the graphs, which contain the same data displayed from different angles. The plot on the left graph has a Surface Contour style. The plot on the right graph has a Contour style of. Try changing the number of contour levels, the contour interval, and the contour anchor. Also experiment with changing the contour label and contour line properties.

Controls, Properties, Methods, and Events

This example demonstrates the following controls, properties, methods, and events:

CWGraph3D

Plot3DSurface, Plot3DSimpleSurface, Plots

CWPlot3D

Contours

CWContours

Levels, Interval, AnchorEnabled, Anchor, SetLabelVisible, SetLabelFormat, SetLabelColor, SetLineColor, SetLineStyle

Example Location

Samples\UI\3DGraph\Contours

Example: Cursors (3D Graph)

Demonstrates all the cursor properties.

Description

Press the "Import File to Plot 1" button to plot some data. Move the mouse over the visible cursor and notice that the mouse icon changes. Click down with the left button on the cursor and drag to move it around. Modify the properties on the right side of the form to see how they affect the cursor.

This example demonstrates the following controls, properties, methods, and events:

CWGraph3D

CWCursors3D

CWPlot3D

 $Samples \verb|\UI|\\ 3DG raph \verb|\Cursors|$

Example: Mouse Events (3D Graph)

Shows which events are fired and what values they contain when you select different track modes on the 3D Graph.

Description

Press the Plot Surfaces button to plot two surfaces. Select the ZoomPanRotate track mode; zoom, pan, and rotate graph; and observe the corresponding events being fired.

- To rotate the graph, click and hold the left mouse button and move the mouse.
- To zoom in on a plot, press and hold the <Alt> key, click and hold with the left mouse button, and move the mouse. You also can zoom with the mouse wheel.
- To pan the graph, press and hold the <Shift> key, click and hold with the left mouse button, and move the mouse.

Select the PlotAreaEvents or the AllEvents track mode. Move the mouse over the graph and observe the effect of different mouse buttons and the <Alt> , <Ctrl> , and <Shift> keys.

This example demonstrates the following controls, properties, methods, and events:

CWGraph3D

TrackMode, Zoom, Pan, Rotate, PlotAreaMouseDown, PlotMouseDown

Samples\UI\3DGraph\Events

Example: Plot Styles

Demonstrates the plotting methods and plot styles.

Description

Press each of the plot buttons to plot data, and then experiment with each of the plot styles for that particular plot button. The Plot Curve button uses the Plot3DCurve method on the graph. The Plot Dual Sine Surface button uses the Plot3DSurface method. The Plot Torus button uses the Plot3DParametricSurface method. The Plot Random Surface button uses the Plot3DSimpleSurface method.

Note: Some plot styles are not supported on plots created with the Plot3DCurve method.

This example demonstrates the following controls, properties, methods, and events:

CWGraph3D

<u>Plots</u>, <u>Plot3DSurface</u>, <u>Plot3DSurface</u>, <u>Plot3DSimpleSurface</u>, <u>Plot3DParametricSurface</u> <u>CWPlot3D</u>

Style, ClearData, ColorMapStyle

Samples\UI\3DGraph\Plot Styles

Example: Region of Interest (3D Graph)

Uses plot events to select a region of interest on a plot.

Description

Click and drag across the plot in the left graph. Notice that the graph on the right displays the selected region of interest. This example uses the PlotMouseDown event to mark one corner of the region of interest, and the PlotMouseMove and PlotMouseUp events to mark the other corner of the region of interest.

This example demonstrates the following controls, properties, methods, and events:

CWGraph3D

TrackMode, PlotMouseDown, Plot3DSimpleSurface

CWAxis3D

Minimum, Maximum

CWPlot3D

ClearData, ColorMapStyle

Samples\UI\3DGraph\Region of Interest

Example: Simple 3D Graph

Demonstrates basic use of the 3D Graph.

Description

This example shows how to plot simple 3D curves or surfaces.

This example demonstrates the following controls, properties, methods, and events:

CWGraph3D

 $\underline{Plot3DCurve}, \ \underline{Plot3DSimpleSurface}$

CWPlot3D

Style, ColorMapStyle

Samples\UI\3DGraph\Simple 3D Graph

See Also

CWGraph3D Overview

Plot Styles Example

Three-Dimensional Plots (concept)

Example: Spinning Light

Demonstrates several lighting options available on the 3D Graph.

Description

Lighting is useful for visualizing the slopes of peaks and valleys in three-dimensional data. This example turns on lighting by setting the Lighting property on the graph to True and enabling one of the lights in the CWLights collection. It then rotates the light by changing the latitude and longitude of the light.

This example demonstrates the following controls, properties, methods, and events:

CWGraph3D

ViewLatitude, Lighting, Plots, Lights

CWPlot3D

FillColor, ColorMapStyle

CWLight

Enabled, Latitude, Longitude

Samples\UI\3DGraph\Spinning Light

Example: Transparency

Demonstrates the effect of turning on transparency for a plot.

Description

The plot transparency property has a range of 0% to 100%. 0% produces an opaque plot, 50% produces a partially transparent plot, and 100% a completely transparent plot.

This example demonstrates the following controls, properties, methods, and events:

CWGraph3D

<u>Plots</u>

CWPlot3D

Transparency, Plot3DSurface

Samples\UI\3DGraph\Transparency



Rotating, Panning, and Zooming

To rotate, pan, and zoom 3D graphs in an application that is running, you must set the <u>TrackMode</u> property for the graph as follows:

CWGraph3D1.TrackMode = cwG3DZoomPanRotate

Use the following steps to rotate, zoom, and pan a 3D graph:

- To rotate the graph, press and hold the left mouse button and drag.
- To zoom on the graph, press and hold the <Alt> key and the left mouse button while dragging the mouse forward and backward. If your mouse has a wheel, you also can zoom on the graph by rotating the wheel.
- To pan the graph, press and hold the <Shift> key and the left mouse button while dragging the mouse.

Note: If the CWGraph3D.Enabled property is set to False, all tracking and events are disabled.

See Also

CWGraph3D.TrackMode

CWGraph3D.Rotate

CWGraph3D.Pan

CWGraph3D.Zoom

Count Property (Read Only)

Syntax

Object.Count

Data Type

Long

Applies To

CWAxes3D CWLights

Purpose

Returns the number of objects in the collection.

Example

'The same syntax applies to other collections. numPlots = CWGraph3D1.Plots.Count

Item Method

Syntax

Object.**Item** ((Item))

Return Type

<u>Object</u>

Applies To

CWAxes3D CWLights

Purpose

Returns the specified object from the collection.

Parameters

Item As **Variant**

The name of the object or the index of the object in the collection. The index is one-based.

Example

'Assign object in collection to variable
Dim Plot1 As CWPlot3D
Set Plot1 = CWGraph3D1.Plots.Item(1)
'Call property or method on object in collection
CWGraph3D1.Plots.Item(1).Plot3DCurve x, y, z

Axes Property (Read Only) Syntax

CWGraph3D.Axes

Data Type

CWAxes3D

Purpose

Specifies a collection of CWAxis3D objects.

Remarks

The collection always has one X, one Y, and one Z CWAxis3D object.

See Also

CWAxes3D

CWAxis3D

AutoScale Property

Syntax

CWAxis3D.AutoScale

Data Type

Boolean

Purpose

Determines if the minimum and maximum limits of the axis are automatically set.

Remarks

It affects only CWPlot3D objects that have the AutoScale property set to True.

Example

CWGraph3D1.Axes.Item(1).AutoScale = True

See Also

 $\underline{CWPlot3D.AutoScale}$

AutoScaleNow

Caption Property

Syntax

CWAxis3D.Caption

Data Type

String

Purpose

Specifies the title for the axis.

Example

'Sets the caption for the first axis CWGraph3D1.Axes.Item(1).Caption = "Temperature" CWGraph3D1.Axes.Item(1).CaptionColor = vbBlue CWGraph3D1.Axes.Item(1).CaptionFont = "Arial" CWGraph3D1.Axes.Item(1).CaptionNormal = True

See Also

CaptionColor

CaptionFont

 $\underline{CaptionNormal}$

CaptionOpposite

CaptionColor Property

Syntax

CWAxis3D.CaptionColor

Data Type

Color

Purpose

Specifies the color of the axis caption.

Example

'Sets the caption for the first axis CWGraph3D1.Axes.Item(1).Caption = "Temperature" CWGraph3D1.Axes.Item(1).CaptionColor = vbBlue CWGraph3D1.Axes.Item(1).CaptionFont = "Arial" CWGraph3D1.Axes.Item(1).CaptionNormal = True

See Also

Caption

CaptionFont

 $\underline{CaptionNormal}$

CaptionOpposite

CaptionFont Property

Syntax

CWAxis3D. Caption Font

Data Type

<u>Font</u>

Purpose

Specifies the font of the axis caption.

Example

'Sets the caption for the first axis CWGraph3D1.Axes.Item(1).Caption = "Temperature" CWGraph3D1.Axes.Item(1).CaptionColor = vbBlue CWGraph3D1.Axes.Item(1).CaptionFont = "Arial" CWGraph3D1.Axes.Item(1).CaptionNormal = True

See Also

Caption

CaptionColor

 $\underline{CaptionNormal}$

CaptionOpposite

CaptionNormal Property

Syntax

CWAxis3D. Caption Normal

Data Type

Boolean

Purpose

Draws the axis caption at the normal position when set to True.

Remarks

You can set an axis caption to appear at two different positions. CaptionNormal draws the caption beside the specified axis label. CaptionOpposite draws the caption above the grid that is opposite the labeled axis.

These properties are independent of each other, and you can enable one or both properties for an axis.

Example

'Sets the caption for the first axis CWGraph3D1.Axes.Item(1).Caption = "Temperature" CWGraph3D1.Axes.Item(1).CaptionColor = vbBlue CWGraph3D1.Axes.Item(1).CaptionFont = "Arial" CWGraph3D1.Axes.Item(1).CaptionNormal = True

See Also

Caption

CaptionColor

CaptionFont

 $\underline{CaptionOpposite}$

CaptionOpposite Property

Syntax

CWAxis3D. Caption Opposite

Boolean

Purpose

Draws the axis caption at the opposite position when set to True.

Remarks

You can set an axis caption to appear at two different positions. CaptionNormal draws the caption beside the specified axis label. CaptionOpposite draws the caption above the grid that is opposite the labeled axis.

These properties are independent of each other, and you can enable one or both properties for an axis.

Example

'Sets the caption for the first axis CWGraph3D1.Axes.Item(1).Caption = "Temperature" CWGraph3D1.Axes.Item(1).CaptionColor = vbBlue CWGraph3D1.Axes.Item(1).CaptionFont = "Arial" CWGraph3D1.Axes.Item(1).CaptionOpposite = True

See Also

Caption

CaptionColor

CaptionFont

CaptionNormal

FormatString Property

Syntax

CWAxis3D.FormatString

String

Purpose

Specifies the format string for formatting the labels on this axis.

Remarks

The formatting string is similar to the string for the Format function in Visual Basic. The following formatting characters are available:

- . Specifies the beginning of the number within the label. Use # and 0 to (decimal the right of the decimal point to specify the precision. A decimal point point) is assumed to be the first character in the format string if you do not include a decimal point in the format string.
- O Specifies the precision to the right of the decimal point. For example ".#0" always produces two digits to the right of the decimal point even when given an exact number such as 1.0.
- # Specifies the precision to the right of the decimal point. For example ".##" produces up to two digits to the right of the decimal point. Thus 1.0 produces 1 while 1.025 produces 1.03.
- e or E Specifies exponential notation. "E" specifies the capital letter. "e" specifies the lower case letter.
- *nn Scales labels. For example ".*10" prints the value 1.0 as 10.
- +nn or Offsets labels. For example ".+20" prints the value 1.0 as 21.
- k Specifies symbolic notation. For the format string ".k" 1.0 prints as 1 and 1000 prints as 1k and .001 prints as 1m.
- "text" Specifies the addition of text to the label. For example '."V" adds a V to the right of every label. Thus 1.0 becomes 1 V.
- \$ or % Specifies to print these special characters.
- : (colon) Time separator. The time separator uses hours:minutes:seconds when time values are formatted.
- / Date separator. The date separator uses day/month/year when date values are formatted.
- d Displays the day as a number without a leading zero. For example: 1-3
- ,dd Displays the day as a number with a leading zero. For example: 01-31
- ddd Displays the day as an abbreviation. For example: Sun Sat.
- dddd Displays the day as a full name. For example: Sunday Saturday

W	Displays the day of the week as a number. For example: 1 for Sunday through 7 for Saturday
WW	Displays the week of the year as a number. For example: 1-53
m	Displays the month as a number without a leading zero. For example: 1 - 12
mm	Displays the month as a number with a leading zero. For example: 01 - 12
mmm	Displays the month as an abbreviation. For example: Jan - Dec
mmmm	Displays the month as a full month name January - December
q	Displays the quarter of the year as a number. For example: 1 - 4
y	Displays the day of the year as a number. For example: 1 - 366
уу	Displays the year as a 2-digit number. For example: 00 - 99
уууу	Displays the year as a 4-digit number. For example: 1000 - 9999
h	Displays the hour as a number without leading zeros. For example: 0 - 23
hh	Displays the hour as a number with leading zeros. For example: 00 - 23
n	Displays the minute as a number without leading zeros. For example: 0 - 59.
nn	Displays the minute as a number with leading zeros. For example: 00 - 59
S	Displays the second as a number without leading zeros. For example: $0-59$
SS	Displays the second as a number with leading zeros. For example: 00 - 59
AM/PM	Uses the 12-hour clock and displays "AM" with any hour before noon. Displays "PM" with any hour between noon and 11:59 PM
am/pm	Uses the 12-hour clock and displays "am" with any hour before noon. Displays "pm" with any hour between noon and 11:59 pm
A/P	Uses the 12-hour clock and displays "A" with any hour before noon. Displays "P" with any hour between noon and 11:59 PM.
a/p	Uses the 12-hour clock and displays "a" with any hour before noon.

Displays "p" with any hour between noon and 11:59 pm.

If you select a format string that includes date or time formatting, the property pages will display all values in a generic date and time format for editing.

Additionally, any property can be set programmatically with a string representing a date and/or time. Thus, a statement such as: CWSlide1.Value = "01/08/1923 9:12:33.14 pm" is a valid value. This statement can be abbreviated to represent just the time or just the date. For example, CWSlide1.Value = "1/1/73" and CWSlide1.Value = "9:15 pm" are valid as well.

With ActiveX controls, the date is implemented as a floating-point value, measuring days from midnight, 30 December 1899. So, midnight, 31 December 1899, is represented by 1.0. Similarly, 6 AM, 1 January 1900, is represented by 2.25, and midnight, 29 December 1899, is -1.0. However, 6 AM, 29 December 1899, is -1.25. To interpret the time portion, take the absolute value of the fractional part of the number. Thus, 1 second equals 1 / 24 hours / 60 minutes / 60 seconds, which is 1/86400 or approximately 1.157407e-5.

Example

'Up to two digits to the right of the decimal on 'the x-axis of the graph to 2 digits of precision CWGraph3D1.Axes.Item(1).FormatString = ".##"

Inverted Property

Syntax

CWAxis3D.Inverted

Boolean

Purpose

Specifies if the direction of an axis is inverted.

Remarks

If the axis is inverted, the positions of the minimum and the maximum are swapped.

Example

'Invert the first axis CWGraph3D1.Axes.Item(1).Inverted = True

Labels Property (Read Only)

Syntax

CWAxis3D.Labels

CWLabels3D

Purpose

Returns a CWLabels3D object, which specifies how labels appear on the axis.

See Also

CWLabels3D

<u>Ticks</u>

<u>ValuePairs</u>

Log Property

Syntax

CWAxis3D.Log

Boolean

Purpose

Specifies if the axis has a Log10 scale.

Remarks

When you change the Log property from a property page, you can change the settings for minimum values, maximum values, and tick mark placement to work better with a log scale.

Example

'Set the y-axis of the graph to Log mode CWGraph3D1.Axes.Item(2).Log = True

Maximum Property

Syntax

CWAxis3D. Maximum

<u>Variant</u>

Purpose

Specifies the maximum value of the axis.

Remarks

The maximum value always must be greater than the minimum value. Use the SetMinMax method to set the minimum and maximum values at the same time.

Example

'Set the minimum and maximum values of the x-axis CWGraph3D1.Axes.Item(1).Minimum = 0 CWGraph3D1.Axes.Item(1).Maximum = 100

'Alternately, call SetMinMax CWGraph3D1.Axes.Item(1).SetMinMax 0, 100

See Also

<u>SetMinMax</u>

Minimum

<u>Inverted</u>

Minimum Property

Syntax

CWAxis3D.Minimum

<u>Variant</u>

Purpose

Specifies the minimum value of the axis.

Remarks

The minimum value always must be less than the maximum value. Use the SetMinMax method to set the minimum and maximum values at the same time.

Example

'Set the minimum and maximum values of the x-axis CWGraph3D1.Axes.Item(1).Minimum = 0 CWGraph3D1.Axes.Item(1).Maximum = 100

'Alternately, call SetMinMax CWGraph3D1.Axes.Item(1).SetMinMax 0, 100

See Also

<u>SetMinMax</u>

Maximum

<u>Inverted</u>

Name Property

Syntax

CWAxis3D.Name

Data Type

String

Purpose

Specifies the name of the axis.

Remarks

Use this name for indexing the Axes collection on a CWGraph3D object.

Example

'Set the x-axis name CWGraph3D1.Axes.Item(1).Name = XAxis

Ticks Property (Read Only)

Syntax

CWAxis3D.Ticks

Data Type

CWTicks3D

Purpose

Returns a CWTicks3D object, which specifies how divisions and ticks appear on this axis.

Example

'Set divisions on the graph's x-axis with AutoDivisions CWGraph3D1.Axes.Item(1).Ticks.AutoDivisions

See Also

CWTicks3D

Labels

<u>ValuePairs</u>

ValuePairs Property (Read Only)

Syntax

CWAxis3D. ValuePairs

Data Type

CWValuePairs

Purpose

Returns a CWValuePairs collection of CWValuePair objects, which specify labels for particular points on the axis.

Remarks

A ValuePair object is a label paired with a value. For example, use a CWValuePair object to add a text label such as "Test" to a numeric axis at the value 212.

Example

'Add a value pair to the graph's x-axis CWGraph3D1.Axes.Item(1).ValuePairs.Add

'Set the name and value of the value pair just added CWGraph3D1.Axes.Item(1).ValuePairs.Item(1).Name = "Test" CWGraph3D1.Axes.Item(1).ValuePairs.Item(1).Value = 212

See Also

<u>CWValuePair</u>

Visible Property

Syntax

CWAxis3D.Visible

Data Type

Boolean

Purpose

Specifies if the axis is visible or hidden.

Remarks

Set Visible to False to hide the axis.

Example

'Make the x-axis visible CWGraph3D1.Axes.Item(1).Visible = True

AutoScaleNow Method

Syntax

CWAxis3D.AutoScaleNow

Purpose

Rescales the axis.

Remarks

The axis is autoscaled, regardless of the setting of the AutoScale property.

Example

'Force the axis to rescale CWGraph3D1.Axes.Item(1).AutoScaleNow

See Also

<u>AutoScale</u>

SetMinMax Method

Syntax

CWAxis3D.**SetMinMax** Minimum, Maximum

Purpose

Sets both the minimum and the maximum values of the axis at the same time.

Remarks

Use this method to avoid setting a minimum value greater than the maximum value or a maximum value less than the minimum value.

To make the axis appear inverted, set the Inverted property.

Parameters

Minimum As Variant

The new minimum value for the axis.

Maximum As Variant

The new maximum value for the axis.

Example

'Set the minimum and maximum values of the x-axis CWGraph3D1.Axes.Item(1).Minimum = 0 CWGraph3D1.Axes.Item(1).Maximum = 100

'Alternately, call SetMinMax CWGraph3D1.Axes.Item(1).SetMinMax 0, 100

See Also

Minimum

Maximum

<u>Inverted</u>

LabelColor Property

Syntax

CWContour.LabelColor

Data Type

Color

Purpose

Specifies the color of the contour label.

Example

'Set the contour label color to blue CWGraph3D1.Plots.Item(1).Contours.Item(1).LabelColor = vbBlue 'Set the contour label font to bold CWGraph3D1.Plots.Item(1).LabelFont.Bold = True

See Also

LabelFormat

LabelFont Property

Syntax

CWContour.LabelFont

Data Type

<u>Font</u>

Purpose

Specifies the contour label font.

Example

'Set the contour label color to blue CWGraph3D1.Plots.Item(1).Contours.Item(1).LabelColor = vbBlue 'Set the contour label font to bold CWGraph3D1.Plots.Item(1).LabelFont.Bold = True

See Also

LabelFormat

LabelFormat Property

Syntax

CWContour.LabelFormat

Data Type

String

Purpose

Specifies the format string for formatting the label on this contour.

Remarks

This format string behaves exactly like the CWAxis3D.FormatString with one added feature: Each "%level" substring inside of quotes is replaced with the contour index.

Example

```
'Up to two digits to the right of the decimal on
' the first contour line label
CWGraph3D1.Plots.Item(1).Contours.Item(1).LabelFormat = ".##"
'Display the index on the second contour line label
CWGraph3D1.Plots.Item(1).Contours.Item(2).LabelFormat
= chr(34) + " Index = %level" + chr(34)
```

See Also

CWAxis3D.FormatString

LabelVisible Property

Syntax

CWContour.LabelVisible

Data Type

Boolean

Purpose

Specifies if the contour line has a visible label.

Remarks

When the LabelVisible property is set to True, the label is visible.

Example

'Set a visible label on the contour line.

CWGraph3D1.Plots.Item(1).Contours.Item(1).LabelVisible = True

See Also

LabelFormat

Level Property

Syntax

 $\underline{\text{CWContour}}.Level$

Data Type

<u>Double</u>

Purpose

Specifies the number of contour levels.

Remarks

Contour levels can be based on the X, Y, Z, or magnitude data.

Example

'Creates three contour levels CWGraph3D1.Contours.Item(1).Level = 3

See Also

CWContours.Basis

LineColor Property

Syntax

CWContour.LineColor

Data Type

Color

Purpose

Specifies the color of the contour line.

Example

'Set the plot style to a line CWGraph3D1.Plots.Item(1).Style = cwLine 'Set the contour line color to blue CWGraph3D1.Contours.Item(1).LineColor = vbBlue 'Set the line width to 1.0 CWGraph3D1.Plots.Item(1).Contours.Item(1).LineWidth = 1.0

See Also

LineStyle

LineWidth

LineStyle Property

Syntax

CWContour.LineStyle

Data Type

CW3DLineStyles

You can use the following constants with this data type:

- cwLine3DDash–Dashed line
- cwLine3DDashDot-Dash-dot line
- cwLine3DDot–Dotted line
- cwLine3DNone–No line
- cwLine3DSolid-Solid line

Purpose

Specifies the line style of the contour line.

Example

'Set the contour line style to a dashed line CWGraph3D1.Plots.Item(1).Contours.Item(1).LineStyle = cwLine3DDash

LineWidth Property

Syntax

CWContour.LineWidth

Data Type

<u>Double</u>

Purpose

Specifies the width of the contour line in points.

Remarks

The valid range for the width of a contour line is 0.0 to 100.0.

Example

'Set the plot style to a line CWGraph3D1.Plots.Item(1).Style = cwLine 'Set the contour line color to blue CWGraph3D1.Contours.Item(1).LineColor = vbBlue 'Set the line width to 1.0 CWGraph3D1.Plots.Item(1).Contours.Item(1).LineWidth = 1.0

Anchor Property

Syntax

CWContours.Anchor

Data Type

<u>Double</u>

Purpose

Specifies the value of the contour anchor.

Remarks

If you want to see data at a known level, use the Anchor property to define a value that a contour is guaranteed to pass through.

Example

'Set the plot style to a surface with contour lines.

CWGraph3D1.Plots.Item(1).Style = cwSurfaceContour

'Set the contour anchor value to 5

CWGraph3D1.Plots.Item(1).Contours.Anchor = 5

'Enable contour anchor

CWGraph3D1.Plots.Item(1).Contours.AnchorEnabled = True

See Also

AnchorEnabled

AnchorEnabled Property

Syntax

CWContours.AnchorEnabled

Data Type

Boolean

Purpose

Enables the contour anchor if set to True.

Example

'Set the plot style to a surface with contour lines.

CWGraph3D1.Plots.Item(1).Style = cwSurfaceContour

'Set the contour anchor value to 5

CWGraph3D1.Plots.Item(1).Contours.Anchor = 5

'Enable contour anchor

CWGraph3D1.Plots.Item(1).Contours.AnchorEnabled = True

See Also

Anchor

Basis Property

Syntax

CWContours.Basis

Data Type

<u>CWBases</u>

You can use the following constants with this data type:

- cwMagnitude–Contour levels are based on the plot's magnitude (W) data.
- cwX–Contour levels are based on the plot's X data.
- cwY–Contour levels are based on the plot's Y data.
- cwZ–Contour levels are based on the plot's Z data.

Purpose

Specifies the data coordinate that the contour levels are calculated from.

Remarks

Contour levels can be based on the X, Y, Z, or magnitude data.

Example

'Set the contour level's basis on the x-axis CWGraph3D1.Plots.Item(1).Contours.Basis = cwX

Count Property (Read Only)

Syntax

Object.Count

Data Type

Long

Applies To

CWContours

CWCursors3D

CWPlots3D

<u>CWValuePairs</u>

Purpose

Returns the number of objects in the collection.

Example

'The same syntax applies to other collections. numLights = CWGraph3D1.Lights.Count

Interval Property

Syntax

CWContours.Interval

Data Type

<u>Double</u>

Purpose

Specifies the distance between each contour level.

Remarks

When the Interval property is set, the number of levels adjusts to accommodate the new interval. For example, if you want to set the Interval property to 2.5 and anchor a contour line value at 5, the number of contour levels adjust to maintain equal spacing between the levels.

Example

'Set the interval to 2.5

CWGraph3D1.Plots.Item(1).Contours.Interval = 2.5

'Set the contour anchor at 5

CWGraph3D1.Plots.Item(1).Contours.Anchor = 5

'Enable the anchor

CWGraph3D1.Plots.Item(1).Contours.AnchorEnabled = True

See Also

CWContour.Level

Levels

Anchor

LevelList Property

Syntax

CWContours.LevelList

Data Type

Variant

Purpose

Adds a contour level at a specified value on the plot.

Remarks

A vector specifies each contour level in the plot. Each element of the vector contains the position of a contour level. You can create a contour level by specifying a value for the LevelList property.

Example

'Create five contour levels CWGraph3D1.Plots.Item(1).Contours.Levels = 5 'Set a contour level at value 6 CWGraph3D1.Plots.Item(1).Contours.LevelList = 6

See Also

CWContour.Level

<u>Levels</u>

Levels Property Syntax

CWContours.Levels

Data Type

Long

Purpose

Specifies the number of contour levels for the plot.

Example

'Create five contour levels CWGraph3D1.Plots.Item(1).Contours.Levels = 5 'Set a contour level at value 6 CWGraph3D1.Plots.Item(1).Contours.LevelList = 6

See Also

LevelList

<u>Interval</u>

Anchor

Add Method

Syntax

CWContours.Add Level

Purpose

Adds a contour at the specified level.

Remarks

An error occurs if a contour with the same level already exists.

Parameters

Level As Double

The new level.

See Also

LevelList

CWContour.Level

Item Method

Syntax

Object.**Item** ((Item))

Return Type

<u>Variant</u>

Applies To

CWContours

CWCursors3D

CWPlots3D

<u>CWValuePairs</u>

Purpose

Returns the specified object from the collection.

Parameters

Item As Object

The object specified.

Example

'Assign object in collection to variable
Dim Plot1 As CWPlot3D
Set Plot1 = CWGraph3D1.Plots.Item(1)
'Call property or method on object in collection
CWGraph3D1.Plots.Item(1).Plot3DCurve x, y, z

Remove Method

Syntax

Object.Remove (Item)

Applies To

CWContours

CWCursors3D

CWPlots3D

<u>CWValuePairs</u>

Purpose

Removes the specified item from the collection.

Parameters

Item As **Variant**

The name of the object or the index of the object in the collection. The index of the object is one-based.

Example

'Remove third plot from the graph CWGraph3D1.Plots.Remove 3

RemoveAll Method

Syntax

Object.RemoveAll

Applies To

CWContours

CWCursors3D

CWPlots3D

<u>CWValuePairs</u>

Purpose

Removes all objects from the collection.

Example

'Remove all contours from a plot CWGraph3D1.Plots(1).Contours.RemoveAll

SetLabelColor Method Syntax

CWContours.SetLabelColor LabelColor

Purpose

Sets the label color for all contours in the collection.

Remarks

Contour labels are available only if the plot style is set to Contour in the property pages or set programmatically to cwContourLine on the Style property of the Plots object.

Parameters

LabelColor As Color

The new label color.

Example

'Set the plot style to contour CWGraph3D1.Plots(1).Style = cwContourLine 'Set the label color to blue CWGraph3D1.Plots(1).Contours.SetLabelColor vbBlue

See Also

 $\underline{CWContour.LabelColor}$

SetLabelFont Method

Syntax

CWContours.SetLabelFont LabelFont

Purpose

Sets the label font for all contours in the collection.

Remarks

Contour labels are available only if the plot style is set to Contour in the property pages or set programmatically to cwContourLine on the Style property of the Plots object.

Parameters

LabelFont As Font

The new label font.

See Also

CWContour.LabelFont

SetLabelFormat Method

Syntax

CWContours.SetLabelFormat Label

Purpose

Sets the label format string for all contours in the collection.

Remarks

Contour labels are available only if the plot style is set to Contour in the property pages or set programmatically to cwContourLine on the Style property of the Plots object.

Parameters

Label As String

The new label format string.

See Also

CWContour.LabelFormat

SetLabelVisible Method

Syntax

CWContours.SetLabelVisible LabelVisible

Purpose

Sets the label visibility for all contours in the collection.

Remarks

Contour labels are available only if the plot style is set to Contour in the property pages or set programmatically to cwContourLine on the Style property of the Plots object.

Parameters

LabelVisible As Boolean

The new label visibility.

Example

'Hide contour labels CWGraph3D1.Plots(1).Contours.SetLabelVisible False

See Also

CWContour.LabelVisible

SetLineColor Method

Syntax

CWContours.SetLineColor LineColor

Purpose

Sets the line color for all contours in the collection.

Parameters

LineColor As Color

The new line color.

Example

See Also

 $\underline{CWContour.LineColor}$

SetLineStyle Method

Syntax

CWContours.SetLineStyle LineStyle

Purpose

Sets the line style for all contours in the collection.

Parameters

LineStyle As CW3DLineStyles

The new line style.

Example

See Also

CWContour.LineStyle

SetLineWidth Method

Syntax

CWContours.SetLineWidth LineWidth

Purpose

Sets the line width for all contours in the collection.

Remarks

The valid range for the width of a contour line is 0.0 to 100.0.

Parameters

LineWidth As Double

The new line width.

Example

See Also

 $\underline{CWContour.LineWidth}$

Contours Property

Syntax

Set CWPlot3D.Contours

Data Type

CWContours

Purpose

Specifies a collection of CWContour objects.

Remarks

To turn on contours, set the CWPlot3D.Style property to either cwContourLine or cwSurfaceContour.

Example

'Set style to cwContourLine CWGraph3D1.Plots.Item(2).Style = cwContourLine 'You can now access contour lines and for example, 'set the contour interval to 5 CWGraph3D1.Plots.Item(2).Contours.Intervals = 5

See Also

CWContours

<u>Style</u>

Column Property

Syntax

CWCursor3D.Column

Long

Specifies the column index of the plot data point that the cursor is snapped to.

Remarks

On a simple surface, as you move across a column, you move in the direction of the y axis.

This property is valid only if CWCursor3D.SnapMode is set to cwSnapToPlot or cwSnapNearestPlot.

Example

CWGraph3D1.Cursors.Add
Set CWGraph3D1.Cursors(1).Plot = CWGraph3D1.Plots(1)
'The cursor is snapped to the plot specified above
CWGraph3D1.Cursors(1).SnapMode = cwSnapToPlot
'Place the cursor at column 5, row 10
CWGraph3D1.Cursors(1).Column = 5
CWGraph3D1.Cursors(1).Row = 10

Row

<u>Plot</u>

<u>SnapMode</u>

Enabled Property

Syntax

CWCursor3D.Enabled

Boolean

Specifies if you can drag the cursor.

Remarks

By default, the Enabled property is set to True. If you click on a disabled cursor, the cursor does not respond.

<u>SnapMode</u>

CWGraph3D.TrackMode

Font Property

Syntax

CWCursor3D.Font

<u>Font</u>

Specifies the font for the cursor text.

Name

<u>NameVisible</u>

<u>PositionVisible</u>

LineColor Property

Syntax

CWCursor3D.LineColor

Color

Specifies the color of the cursor lines.

LineStyle

LineColor

LineStyle Property

Syntax

CWCursor3D.LineStyle

CW3DLineStyles

You can use the following constants with this data type:

- cwLine3DDash–Dashed line
- cwLine3DDashDot-Dash-dot line
- cwLine3DDot–Dotted line
- cwLine3DNone–No line
- cwLine3DSolid-Solid line

Specifies the style of the cursor lines.

LineColor

LineWidth

LineWidth Property

Syntax

CWCursor3D.LineWidth

<u>Double</u>

Specifies the width of the cursor lines.

LineColor

LineStyle

Name Property

Syntax

CWCursor3D.Name

String

Specifies the name of the cursor.

NameVisible

<u>Font</u>

<u>TextColor</u>

 $\underline{TextBackGroundTransparency}$

<u>TextBackColor</u>

NameVisible Property

Syntax

CWCursor3D.NameVisible

Boolean

Specifies if the plot name is draw next to the cursor.

Name

PositionVisible

PlaneColor Property

Syntax

CWCursor3D.PlaneColor

Color

Specifies the color of the cursor planes.

PlaneTransparency

 $\underline{XYPlaneVisible}$

 $\underline{XZPlaneVisible}$

YZPlaneVisible

PlaneTransparency Property

Syntax

CWCursor3D.PlaneTransparency

Long

Specifies the transparency of the cursor planes.

Remarks

The value of PlaneTransparency must be a percentage.

PlaneColor

 $\underline{XYPlaneVisible}$

XZPlaneVisible

YZPlaneVisible

Plot Property

Syntax

Set CWCursor3D.Plot

CWPlot3D

Specifies the plot that the cursor is snapped to.

Remarks

This property is valid only if the CWCursor3D.SnapMode is set to cwSnapToPlot or cwSnapNearestPlot.

Example

CWGraph3D1.Cursors.Add
Set CWGraph3D1.Cursors(1).Plot = CWGraph3D1.Plots(1)
'The cursor is snapped to the plot specified above
CWGraph3D1.Cursors(1).SnapMode = cwSnapToPlot
'Place the cursor at column 5, row 10
CWGraph3D1.Cursors(1).Column = 5
CWGraph3D1.Cursors(1).Row = 10

Row

Column

<u>SnapMode</u>

PointColor Property

Syntax

CWCursor3D.PointColor

Color

Specifies the color of the cursor point.

<u>PointSize</u>

<u>PointStyle</u>

PointSize Property

Syntax

CWCursor3D.PointSize

<u>Double</u>

Specifies the size of the cursor point.

<u>PointColor</u>

<u>PointStyle</u>

PointStyle Property

Syntax

CWCursor3D.PointStyle

CW3DPointStyles

You can use the following constants with this data type:

- cwPoint3DAsterisk–Asterisk
- cwPoint3DBoldX-Bold X
- cwPoint3DEmptyCircle–Empty circle
- cwPoint3DEmptySquare–Empty square
- cwPoint3DNone–No point
- cwPoint3DSolidCircle–Solid circle
- cwPoint3DSolidCube-Solid cube
- cwPoint3DSolidDiamond–Solid diamond
- cwPoint3DSolidSphere–Solid sphere
- cwPoint3DSolidSquare—Solid square
- cwPoint3DWireframeCube-Wireframe cube
- cwPoint3DWireframeSphere–Wireframe sphere

Specifies the style of the cursor point.

<u>PointColor</u>

<u>PointSize</u>

PositionVisible Property

Syntax

CWCursor3D.PositionVisible

Boolean

Specifies if the cursor position is displayed next to the cursor.

XPosition

YPosition

ZPosition

<u>Font</u>

<u>TextColor</u>

<u>TextBackColor</u>

 $\underline{TextBackgroundTransparency}$

Row Property

Syntax

CWCursor3D.Row

Long

Specifies the row index of the plot data point that the cursor is snapped to.

Remarks

On a simple surface, as you move down a row, you move in down the x axis. This property is valid only if the CWCursor3D.SnapMode is set to cwSnapToPlot or cwSnapNearestPlot.

Example

CWGraph3D1.Cursors.Add
Set CWGraph3D1.Cursors(1).Plot = CWGraph3D1.Plots(1)
'The cursor is snapped to the plot specified above
CWGraph3D1.Cursors(1).SnapMode = cwSnapToPlot
'Place the cursor at column 5, row 10
CWGraph3D1.Cursors(1).Column = 5
CWGraph3D1.Cursors(1).Row = 10

Column

<u>Plot</u>

<u>SnapMode</u>

SnapMode Property Syntax

CWCursor3D.SnapMode

Data Type

CWCursor3DSnapModes

You can use the following constants with this data type:

- cwSnapFixed–Fixes the cursor at the specified x, y, and z position. The cursor cannot be moved.
- cwSnapNearestPlot—Snaps the cursor to the nearest plot. You can move the cursor along the nearest plot.
- cwSnapToPlot–Snaps the cursor to a specified plot. You can move the cursor along the specified plot.

Specifies if the cursor can be dragged and how it snaps to the plot data.

Remarks

By default, CWGraph3D.TrackMode is set to cwG3DTrackZoomPanRotate. If you have changed the value of the TrackMode property, you must specify either cwG3DTrackZoomPanRotate or cwG3DTrackCursors to move cursors.

Example

CWGraph3D1.Cursors.Add
Set CWGraph3D1.Cursors(1).Plot = CWGraph3D1.Plots(1)
'The cursor is snapped to the plot specified above
CWGraph3D1.Cursors(1).SnapMode = cwSnapToPlot
'Place the cursor at column 5, row 10
CWGraph3D1.Cursors(1).Column = 5
CWGraph3D1.Cursors(1).Row = 10

See Also

<u>Plot</u>

Row

Column

XPosition

YPosition

ZPosition

TextBackColor Property

Syntax

CWCursor3D.TextBackColor

Data Type

Color

Specifies the color to paint behind the cursor text.

See Also

<u>TextBackgroundTransparency</u>

<u>NameVisible</u>

 $\underline{PositionVisible}$

TextBackgroundTransparency Property Syntax

CWCursor3D.TextBackgroundTransparency

Data Type

Long

Specifies how transparent to paint a box behind the cursor text.

See Also

<u>TextColor</u>

<u>TextBackColor</u>

TextColor Property

Syntax

CWCursor3D.TextColor

Data Type

Color

Specifies the color of the cursor text.

See Also

<u>TextBackgroundTransparency</u>

<u>NameVisible</u>

 $\underline{PositionVisible}$

Visible Property

Syntax

CWCursor3D.Visible

Data Type

Boolean

Specifies if the cursor is visible.

XPosition, YPosition, ZPosition Properties

Syntax

CWCursor3D.XPosition

CWCursor3D. **YPosition**

CWCursor3D.ZPosition

Data Type

<u>Double</u>

Specifies the position of the cursor.

Remarks

Use the values of the XPosition, YPosition, and ZPosition properties to set or get the position of the cursor.

See Also

<u>SnapMode</u>

XYPlaneVisible, XZPlaneVisible, YZPlaneVisible Properties

Syntax

CWCursor3D.XYPlaneVisible

CWCursor3D.XZPlaneVisible

CWCursor3D. YZPlaneVisible

Data Type

Boolean

Specifies if the cursor planes are visible.

See Also

PlaneColor

PlaneTransparency

SetColor Method

Syntax

CWCursor3D.**SetColor** Color

Sets the line, plane, point, and text colors of the cursor.

Parameters

Color As $\underline{\text{Color}}$

Color of the cursor.

See Also

LineColor

<u>PlaneColor</u>

<u>PointColor</u>

TextColor

SetPosition Method

Syntax

CWCursor3D.**SetPosition** X, Y, Z

Sets the x, y, and z position of the cursor.

Parameters

X As Double

X axis position of the cursor.

 $\mathbf{Y} \ As \ \underline{ ext{Double}}$

Y axis position of the cursor.

 ${\bf Z} \; As \; {\scriptstyle {\tiny {\it Double}}}$

Z axis position of the cursor.

See Also

XPosition

YPosition

ZPosition

Add Method

Syntax

Object.Add ()

Return Type

<u>Object</u>

Applies To

CWCursors3D

CWPlots3D

<u>CWValuePairs</u>

Purpose

Adds an object to the collection and returns the new object.

Example

'Add a value pair to the x-axis CWGraph3D1.Axes.Item(1).ValuePairs.Add

AmbientLightColor Property

Syntax

CWGraph3D.AmbientLightColor

Data Type

Color

Purpose

Specifies the ambient light color for the graph.

Remarks

You must set $\underline{\text{Lighting}}$ to True before you can set the ambient light color.

Example

'Enable Lighting CWGraph3D1.Lighting = True 'Set the ambient light color to yellow CWGraph3D1.AmbientLightColor = vbYellow

See Also

Lighting

<u>Lights</u>

Rotating, Panning, and Zooming (concept)

BackColor Property

Syntax

CWGraph3D.BackColor

Data Type

Color

Purpose

Specifies the color for the background of the caption.

Remarks

The caption appears above the 3D graph.

Example

'Create the caption "Temperature vs. Time"

CWGraph3D1.Caption = "Temperature vs. Time"

'Set the caption text color to red

CWGraph3D1.CaptionColor = vbRed

'Set the caption background color to green

CWGraph3D1.BackColor = vbGreen

'Set the caption font to bold

CWGraph3D1.Font.Bold = True

Caption Property

Syntax

CWGraph3D.Caption

Data Type

String

Purpose

Specifies the caption for the 3D graph.

Remarks

The caption appears above the 3D graph.

Example

'Create the caption "Temperature vs. Time"

CWGraph3D1.Caption = "Temperature vs. Time"

'Set the caption text color to red

CWGraph3D1.CaptionColor = vbRed

'Set the caption background color to green

CWGraph3D1.BackColor = vbGreen

'Set the caption font to bold

CWGraph3D1.Font.Bold = True

See Also

CaptionColor

<u>Font</u>

CaptionColor Property

Syntax

CWGraph3D.CaptionColor

Data Type

Color

Purpose

Specifies the color of the caption text.

Remarks

The caption appears above the 3D graph.

Example

'Create the caption "Temperature vs. Time"

CWGraph3D1.Caption = "Temperature vs. Time"

'Set the caption text color to red

CWGraph3D1.CaptionColor = vbRed

'Set the caption background color to green

CWGraph3D1.BackColor = vbGreen

'Set the caption font to bold

CWGraph3D1.Font.Bold = True

See Also

Caption

ClipData Property

Syntax

CWGraph3D.ClipData

Data Type

Boolean

Purpose

Indicates if the data is clipped to the axes ranges.

Remarks

You can set the ClipData property programmatically or from the property pages. When data is clipped, any data that is outside the minimum or maximum ranges is not displayed on the graph.

When this property is set to True, data is clipped to the axes ranges.

When this property is set to False, data is not clipped to the axes ranges.

See Also

CWAxis3D.Minimum

CWAxis3D.Maximum

Cursors Property

Syntax

CWGraph3D.Cursors

Data Type

CWCursors3D

Purpose

Specifies a collection of CWCursor3D objects.

See Also

CWCursors3D

CWCursor3D

Dither Property

Syntax

CWGraph3D.Dither

Data Type

Boolean

Purpose

Specifies the dither mode of the graph.

Remarks

When set to True, dithering is enabled. Dithering combines colors to approximate selected colors that cannot be directly displayed because of a limited number of display colors on the system.

When set to False, dithering is disabled.

Example

'Enable dithering CWGraph3D1.Dither = True

Enabled Property

Syntax

CWGraph3D.Enabled

Data Type

Boolean

Purpose

Specifies if the graph generates any events.

Remarks

If the CWGraph3D.Enabled property is set to False, all tracking and events are disabled.

Example

'Generate events on the graph CWGraph3D1.Enabled = True

See Also

<u>TrackMode</u>

FastDraw Property Syntax

CWGraph3D.FastDraw

Data Type

Boolean

Purpose

Specifies how the graph is drawn during interactive operations such as rotating, zooming, and panning.

Remarks

When set to True, draws a reduced representation of the graph and its plots. When set to False, draws the entire graph and its plots. If you use this mode for complex graphs, you might notice unresponsive interaction due to the large amount of data.

Example

'Enable FastDraw CWGraph3D1.FastDraw = True

See Also

Zoom

<u>Pan</u>

Rotate

Font Property

Syntax

CWGraph3D.Font

Data Type

<u>Font</u>

Purpose

Specifies the font for the caption.

Remarks

The caption appears above the 3D graph.

Example

'Create the caption "Temperature vs. Time"

CWGraph3D1.Caption = "Temperature vs. Time"

'Set the caption text color to red

CWGraph3D1.CaptionColor = vbRed

'Set the caption background color to green

CWGraph3D1.BackColor = vbGreen

'Set the caption font to bold

CWGraph3D1.Font.Bold = True

See Also

Caption

GraphFrameColor Property

Syntax

CWGraph3D.GraphFrameColor

Data Type

Color

Purpose

Specifies the color for the graph frame.

Remarks

You must first enable GraphFrameVisible before setting the GraphFrameColor property.

Example

'Make the graph frame visible CWGraph3D1.GraphFrameVisible = True 'Set the graph frame color to blue CWGraph3D1.GraphFrameColor = vbBlue

See Also

GraphFrameVisible

GraphFrameVisible Property

Syntax

CWGraph3D.GraphFrameVisible

Data Type

Boolean

Purpose

Specifies if the graph frame is visible.

Remarks

You can place a frame around the graph by setting GraphFrameVisible to True or using the Graph tab on the property pages.

Example

'Make the graph frame visible CWGraph3D1.GraphFrameVisible = True 'Set the graph frame color to blue CWGraph3D1.GraphFrameColor = vbBlue

See Also

GraphFrameVisible

GridFrameColor Property

Syntax

CWGraph3D.GridFrameColor

Data Type

Color

Purpose

Specifies the color of the graph's grid frame.

Remarks

When you set the grid frame color, the color of the grid lines does not change. Only the grid plane frame color changes.

Example

'Set the grid frame color to green CWGraph3D1.GridFrameColor = vbGreen

GridSmoothing Property

Syntax

CWGraph3D. GridSmoothing

Data Type

Boolean

Purpose

Specifies the smoothing mode for grid lines.

Remarks

When set to True, the graph uses a technique called anti-aliasing to create smoother grid lines. When set to False, the graph draws grid lines without smoothing.

Example

'Enable smoothing CWGraph3D1.GridSmoothing = True

GridXY, GridXZ, GridYZ Properties

Syntax

 $\underline{\mathsf{CWGraph3D}}.\mathbf{GridXY}$

CWGraph3D.GridXZ

CWGraph3D.GridYZ

Data Type

Boolean

Purpose

GridXY draws the XY grid plane when set to True. GridXZ draws the XZ grid plane when set to True. GridYZ draws the YZ grid plane when set to True.

Example

'Disable, or hide, the XY grid plane CWGraph3D1.GridXY = False

ImmediateUpdates Property

Syntax

 $\underline{\text{CWGraph3D}}. Immediate Updates$

Data Type

Boolean

Purpose

Specifies if the graph draws new data as soon as it is available, or if the form refreshes the graph when it draws other controls.

Remarks

Set this property to True to guarantee that the graph plots all available data.

Set this property to False to skip graph updates for other events. When this property is False, the container, such as a Visual Basic form, controls the update rate.

KeyboardMode Property

Syntax

Object.**KeyboardMode**

Data Type

 $\underline{CWKeyboardModes}$

You can use the following constants with this data type:

- cwKeyboardHandled–Keystrokes are processed by the control.
- cwKeyboardNone–Keystrokes are ignored by the control.

Applies To

CWGraph3D

Purpose

Specifies how the control handles keyboard input from the user.

Lighting Property

Syntax

CWGraph3D.Lighting

Data Type

Boolean

Purpose

Enables graph lighting when set to True.

Remarks

When you enable graph lighting, all lighting, including ambient lighting and individual lights, is enabled.

Example

'Enable Lighting CWGraph3D1.Lighting = True

See Also

 $\underline{AmbientLightColor}$

<u>Lights</u>

Lights Property

Syntax

Set CWGraph3D.Lights

Data Type

CWLights

Purpose

Specifies a collection of CWLights.

Remarks

This collection always has four CWLight objects.

Lights are used to create highlights and shadows that enhance visualization of 3D data. You can configure a Light object programmatically or in the property pages.

Example

'Select and enable a Light object CWGraph3D1.Lights.Item(4).Enabled = True

See Also

 $\underline{AmbientLightColor}$

Lighting

Spinning Light Example

PlotAreaColor Property

Syntax

CWGraph3D.PlotAreaColor

Data Type

Color

Purpose

Specifies the background color of the plot area.

Example

'Set the plot area color to cyan CWGraph3D1.PlotAreaColor = vbCyan

Plots Property (Read Only)

Syntax

CWGraph3D.Plots

Data Type

CWPlots3D

Purpose

Specifies a collection of CWPlot3D objects.

Remarks

Plots are automatically added to this collection if needed.

See Also

CWPlots3D

CWPlot3D

PlotTemplate Property

Syntax

Set CWGraph3D.PlotTemplate

Data Type

CWPlot3D

Purpose

Returns the CWPlot3D object to use as a template for new plots.

Remarks

Set properties on the template plot to control how new plots appear. When you create a new plot, all of its properties are copied from the template.

The axes of the template plots determine the coordinates for plot area events.

See Also

CWPlot3D

PlotTemplate Object (concept)

ProjectionStyle Property

Syntax

CWGraph3D.ProjectionStyle

Data Type

CWGraph3DProjectionStyles

You can use the following constants with this data type:

- cwOrthographic–Displays orthographic projection, in which all points appear to be the same size regardless of viewing distance.
- cwPerspective—Displays perspective projection, in which point size reflects the viewing distance of the point.

Purpose

Specifies the projection style of the graph.

Example

'Set the projection style to orthographic projection CWGraph3D1.ProjectionStyle = cwOrthographic

ReadyState Property (Read Only)

Syntax

Object.**ReadyState**

Data Type

Long

Applies To

CWGraph3D

Purpose

Returns the ready state.

Remarks

The ready state provides a mechanism for interacting with a control before all its data has been loaded. For example, if a CWGraph3D control has a plot with a lot of data, the control might take a long time to load.

See Also

ReadyStateChange

TrackMode Property

Syntax

CWGraph3D.TrackMode

Data Type

CWGraph3DTrackModes

You can use the following constants with this data type:

- cwG3DTrackAllEvents–Generates mouse events for plots and the plot area.
- cwG3DTrackCursors–Generates cursor events.
- cwG3DTrackPlotAreaEvents—Generates mouse events only for the plot area.
- cwG3DTrackZoomPanRotate–Generates Zoom, Pan, and Rotate events.

Purpose

Determines the type of events generated and other automatic processing, such as how the mouse interacts with the graph during run time.

Remarks

Some common modes on the graph generate events for mouse interaction with plots and the plot area, as well as rotating, zooming, and panning the graph.

To rotate, pan, and zoom 3D graphs in an application that is running, set TrackMode to cwG3DZoomPanRotate. Use the following steps to rotate, zoom, and pan a 3D graph:

- To rotate the graph, press and hold the left mouse button and drag.
- To zoom on the graph, press and hold the <Alt> key and the left mouse button while dragging the mouse forward and backward. If your mouse has a wheel, you also can zoom on the graph by rotating the wheel.
- To pan the graph, press and hold the <Shift> key and the left mouse button while dragging the mouse.
 If the property CWGraph3D.Enabled is set to False, all tracking and events are disabled.

Example

'Enable panning, zooming, rotating CWGraph3D1.TrackMode = cwG3DZoomPanRotate

See Also

Enabled

Rotating, Panning, and Zooming (concept)

Mouse Events (3D Graph) Example

Use3DHardwareAcceleration Property Syntax

CWGraph3D.Use3DHardwareAcceleration

Data Type

Boolean

Purpose

Determines if the 3D Graph control renders directly to the display adapter.

Remarks

When this property is set to True, the 3D Graph control renders directly to the display adapter, which maximizes rendering performance if your adapter supports OpenGL hardware acceleration. When this property is set to False, the 3D Graph control renders 3D graphs to an off-screen frame buffer, which is then copied to the screen. The 3D Graph control caches this frame buffer and updates it only when the graph needs to be rendered (such as adding a plot or changing the view position). For complex 3D graphs, the time to render the image into the frame buffer greatly exceeds the time to copy the frame buffer to the screen. As a result, simple screen updates that do not require the graph to re-render (such as scrolling the page) execute very quickly.

See Also

Accelerated OpenGL Graphics Adapter (concept)

ViewAutoDistance Property

Syntax

CWGraph3D.ViewAutoDistance

Data Type

Boolean

Specifies how the viewing distance is set.

Remarks

When this property is set to True, the viewing distance is automatically calculated.

When this property is set to False, the viewing distance is defined by the ViewDistance property.

See Also

ViewDistance

ViewDistance Property

Syntax

CWGraph3D.ViewDistance

Data Type

<u>Double</u>

Specifies the distance of the viewing position from the center of the view as defined by ViewXCenter, ViewYCenter, and ViewZCenter.

See Also

ViewAutoDistance

ViewLatitude Property

Syntax

CWGraph3D.ViewLatitude

Data Type

<u>Double</u>

Specifies the latitude in degees of the viewing position.

See Also

ViewLongitude

ViewLongitude Property

Syntax

CWGraph3D.ViewLongitude

Data Type

<u>Double</u>

Specifies the longitude in degrees of the viewing position.

See Also

<u>ViewLatitude</u>

ViewMode Property

Syntax

CWGraph3D.ViewMode

Data Type

CWGraph3DViewModes

You can use the following constants with this data type:

- cwViewUserDefined–Views a user-defined area.
- cwViewXYPlane–Views the XY plane.
- cwViewXZPlane–Views the XZ plane.
- cwViewYZPlane–Views the YZ plane.

Specifies the viewing position of the graph.

Example

'View only the XY plane of the graph CWGraph3D1.ViewMode = cwViewXYPlane

See Also

<u>ViewLatitude</u>

ViewLongitude

ViewXCenter, ViewYCenter, ViewZCenter Properties Syntax

CWGraph3D. ViewXCenter

CWGraph3D.ViewYCenter

CWGraph3D.ViewZCenter

Data Type

<u>Double</u>

Specify the X, Y, and Z coordinates of the center of the view.

Remarks

By default, the graph rotates around the center coordinates of the graph.

Use ViewXCenter, ViewYCenter, and ViewZCenter to set the graph to rotate around a point other than the center of the graph.

Windowless Property

Syntax

Object. Windowless

Data Type

Boolean

Applies To

CWGraph3D

Specifies if the control has a window.

Remarks

The CWGraph3D control behaves better when Windowless is set to False In some containers, Windowless controls will flicker when updating. Flickering is the result of container behavior and cannot be avoided by the control. In containers where Windowless controls flicker, you should turn the Windowless property off.

Example

'Set the property to False so that it has a window. CWGraph3D1.Windowless = False

See Also

CWGraph3D.BackColor

AboutBox Method

Syntax

Object. About Box

Applies To

CWGraph3D

Displays the About Box for the control.

Remarks

The About Box displays the version number of the control and information about contacting National Instruments. You can access the About Box from a separate tab on the property pages.

Example

'Display the graph's About Box CWGraph3D1.AboutBox

ClearData Method

Syntax

CWGraph3D.ClearData

Purpose

Clears data in all plots.

Example

'Clear data in the plot CWGraph3D1.ClearData

See Also

CWPlot3D.ClearData

ControlImage Method

Syntax

Object.ControlImage ()

Return Type

<u>Picture</u>

Applies To

CWGraph3D

Purpose

Returns an image of the entire control.

Remarks

You can use the image of the control for printing. The image returned is a Windows Metafile.

Example

'Set the image of the picturebox control in VB to 'an image of the entire graph Set Picture1.Picture = CWGraph3D1.ControlImage

ExportStyle Method

Syntax

Object.**ExportStyle** FileName

Applies To

CWGraph3D

Purpose

Exports the style of the control to a .cwx file.

Remarks

A style is the complete current state of a control.

Use ExportStyle to save the style of a control after you have configured it. For example, if you often create a new CWGraph3D control and set properties such as tick colors, axes ranges, and background colors, you can save the style of a graph once you have configured it. When you want to create a graph of the same style, create a new graph, access the property pages, and import the exported style.

You can import or export a style from the right click menu or set it programmatically using ExportStyle and ImportStyle.

Parameters

FileName As String

The name of the style file to save.

Example

'Export a style CWGraph3D1.ExportStyle "c:\LogGraph.cwx"

'Import a style into graph2
CWGraph3D2.ImportStyle "c:\LinearGraph.cwx"

See Also

<u>ImportStyle</u>

ImportStyle Method

Syntax

Object.ImportStyle FileName

Applies To

CWGraph3D

Purpose

Imports a previously exported style to the control.

Remarks

A style is the complete current state of a control.

Use ExportStyle to save the style of a control after you have configured it. For example, if you often create a new CWGraph3D control and set properties such as tick colors, axes ranges, and background colors, you can save the style of a graph once you have configured it. When you want to create a graph of the same style, create a new graph, access the property pages, and import the exported style.

You can import or export a style from the right click menu or set it programmatically using ExportStyle and ImportStyle.

Parameters

FileName As String

The name of the style file to load.

Example

'Export a style CWGraph3D1.ExportStyle "c:\LogGraph.cwx"

'Import a style into graph2
CWGraph3D2.ImportStyle "c:\LinearGraph.cwx"

See Also

ExportStyle

Plot3DCurve Method

Syntax

CWGraph3D.**Plot3DCurve** xVector, yVector, zVector [, wVector]

Purpose

Plots three (or four) 1D arrays of data as a parametric curve.

Remarks

Plot3DCurve uses the first available plot from the CWGraph3D.Plots collection that is marked as a multiplot. If there is not a plot available, it adds a new plot with MultiPlot set to True.

Parameters

xVector As Variant

1D array of X data.

yVector As variant

1D array of Y data.

zVector As Variant

1D array of Z data.

 $wVector\ As\ \underline{{}^{\text{Variant}}}$

[Optional] 1D array of magnitude data.

See Also

CWPlot3D.MultiPlot

CWPlot3D.Plot3DCurve

 $\underline{Plot 3D Simple Surface}$

Plot3DSurface

<u>Plot3DParametricSurface</u>

Plot3DMesh

Three-Dimensional Plots (concept)

Plot Styles Example

Plot3DMesh Method

Syntax

CWGraph3D.**Plot3DMesh** xVector, yVector, zVector [, wVector]

Purpose

Plots three (or four) 1D arrays of data as a triangulated surface.

Remarks

Plot3DMesh uses the first available plot from the CWGraph3D.Plots collection that is marked as a multiplot. If there is not a plot available, it adds a new plot with MultiPlot set to True.

Parameters

xVector As Variant

1D array of X data.

yVector As variant

1D array of Y data.

zVector As Variant

1D array of Z data.

 $wVector\ As\ \underline{{}^{\text{Variant}}}$

[Optional] 1D array of magnitude data.

See Also

CWPlot3D.MultiPlot

 $\underline{CWPlot3D.Plot3DMesh}$

<u>Plot3DCurve</u>

Plot3DSimpleSurface

Plot3DSurface

 $\underline{Plot 3DParametric Surface}$

Three-Dimensional Plots (concept)

Plot Styles Example

Plot3DParametricSurface Method

Syntax

CWGraph3D.**Plot3DParametricSurface** xMatrix, yMatrix, zMatrix [, wMatrix]

Purpose

Plots three (or four) 2D arrays of data as a parametric surface.

Remarks

Plot3DParametricSurface uses the first available plot from the CWGraph3D.Plots collection that is marked as a multiplot. If there is not a plot available, it adds a new plot with MultiPlot set to True.

Parameters

xMatrix As Variant

2D array of X data.

yMatrix As Variant

2D array of Y data.

zMatrix As Variant

2D array of Z data.

wMatrix As Variant

[Optional] 2D array of magnitude data.

See Also

CWPlot3D.MultiPlot

 $\underline{CWPlot3D.Plot3DParametricSurface}$

<u>Plot3DCurve</u>

Plot3DSimpleSurface

Plot3DSurface

Plot3DMesh

Three-Dimensional Plots (concept)

Plot Styles Example

Plot3DSimpleSurface Method

Syntax

CWGraph3D.Plot3DSimpleSurface zMatrix [, wMatrix]

Purpose

Plots one (or two) 2D array(s) of data as a surface.

Remarks

The array provides the Z data for the surface while the indices provide the X and Y data. The optional second array is used to specify magnitude data.

Plot3DSimpleSurface uses the first available plot from the CWGraph3D.Plots collection that is marked as a multiplot. If there is not a plot available, it adds a new plot with MultiPlot set to True.

Parameters

zMatrix As Variant

2D array of Z data.

wMatrix As Variant

[Optional] 2D array of magnitude data.

Example

```
'Generate data to plot
   Const n = 20\#
   Dim x(n), y(n), z(n, n)
   For i = 0 To n
   x(i) = (i / n - 0.5) * 8
   y(i) = (i / n - 0.5) * 8
   Next i
   For i = 0 To n
   For j = 0 To n
   z(i, j) = 8 * Exp(-(x(i) \land 2 + y(j) \land 2) / 4)
   Next i
'Plot the data on each graph
   CWGraph3D1.Plot3DSimpleSurface z
```

See Also

CWPlot3D.MultiPlot

 $\underline{CWPlot3D.Plot3DSimpleSurface}$

Plot3DCurve

Plot3DSurface

Plot3DParametricSurface

Plot3DMesh

Three-Dimensional Plots (concept)

Plot Styles Example

Plot3DSurface Method

Syntax

CWGraph3D.**Plot3DSurface** xVector, yVector, zMatrix [, wMatrix]

Purpose

Plots two 1D arrays and one (or two) 2D array(s) of data as a surface.

Remarks

Plot3DSurface uses the first available plot from the CWGraph3D.Plots collection that is marked as a multiplot. If there is not a plot available, it adds a new plot with MultiPlot set to True.

Parameters

xVector As Variant

1D array of X data.

yVector As variant

1D array of Y data.

zMatrix As Variant

2D array of Z data.

wMatrix As Variant

[Optional] 2D array of magnitude data.

Example

```
'Generate data to plot
    Const n = 20#
    Dim x(n), y(n), z(n, n)
    For i = 0 To n
        x(i) = (i / n - 0.5) * 2 * PI
        y(i) = (i / n - 0.5) * 2 * PI
        Next i
    For i = 0 To n
        For j = 0 To n
        z(i, j) = Sinh(x(i)) * Cos(y(j))
        Next i
'Plot the data on each graph
    CWGraph3D1.Plot3DSurface x, y, z
```

See Also

CWPlot3D.MultiPlot

 $\underline{CWPlot3D.Plot3DSurface}$

<u>Plot3DCurve</u>

Plot3DSimpleSurface

Plot3DParametricSurface

Plot3DMesh

Three-Dimensional Plots (concept)

Plot Styles Example

SetDefaultView Method

Syntax

CWGraph3D.SetDefaultView

Purpose

Restores the default viewing parameters after panning, zooming, or rotating the graph.

See Also

<u>Pan</u>

Zoom

Rotate

Click Event

Syntax

Sub ControlName_Click()

Applies To

CWGraph3D

Purpose

Generates a Click event when you click the mouse on the control.

Remarks

DblClick does not call this event. Double-clicking on a control generates the Click event for the first click and the DblClick event for the second click. For more information, refer to Visual Basic help.

See Also

DblClick

MouseDown

<u>MouseMove</u>

MouseUp

CursorChange Event

Syntax

Sub *ControlName_*CursorChange(CursorIndex As Long, XPosition As Double, YPosition As Double, ZPosition As Double, Tracking As Boolean)

Applies To

CWGraph3D

Purpose

Generates the CursorChange event when you reposition a cursor with the mouse.

Remarks

This event is generated only if CWGraph3D.TrackMode is cwG3DTrackZoomPanRotate or cwG3DTrackCursors.

Move a cursor by clicking on the center of the crosshair with the left mouse button and dragging.

Parameters

CursorIndex As Long

Contains the one-based index of the cursor generating the event.

XPosition As **Double**

Contains the X position of the cursor.

YPosition As **Double**

Contains the Y position of the cursor.

ZPosition As **Double**

Contains the Z position of the cursor.

Tracking As Boolean

Contains true if the mouse button was down when the event occurred.

See Also

Cursors

<u>TrackMode</u>

DblClick Event

Syntax

Sub ControlName_DblClick()

Applies To

CWGraph3D

Purpose

Generates a DblClick event when you double-click the mouse on the control.

Remarks

Double-clicking on a control generates the Click event for the first click and the DblClick event for the second click. For more information, refer to Visual Basic help.

See Also

Click

MouseDown

<u>MouseMove</u>

 $\underline{MouseUp}$

KeyDown, KeyUp Events

Syntax

Sub *ControlName_*KeyDown(KeyCode As Integer, Shift As Integer) Sub *ControlName_*KeyUp(KeyCode As Integer, Shift As Integer)

Applies To

CWGraph3D

Purpose

Generates the KeyUp event when you release a key while the control has the input focus.

Generates the KeyDown event when you press a key while the control has the input focus.

Remarks

For more information, refer to Visual Basic help.

Parameters

KeyCode As Integer

A code representing the key that was pressed on the keyboard rather than the ASCII value of the key. For example, pressing the <A> key produces the value 65. Pressing <#> or <Shift-3> produces the code for "3".

Shift As Integer

The state of the <Shift> key, <Alt> key, and <Ctrl> key. The Shift argument can be any combination of the following values: 1 for <Shift> , 2 for <Ctrl> , or 4 for <Alt> . For example, if the <Shift> and <Ctrl> keys are pressed, the value is 3 (1 + 2).

See Also

<u>KeyPress</u>

KeyPress Event

Syntax

Sub ControlName_KeyPress(KeyAscii As Integer)

Applies To

CWGraph3D

Purpose

Generated when the control has focus and you press a key.

Remarks

For more information, refer to Visual Basic help.

Parameters

KeyAscii As Integer

The ASCII value of the key that was pressed. In Visual Basic, use the Chr() function to translate KeyAscii into a character.

See Also

KeyDown

<u>KeyUp</u>

MouseDown, MouseMove, MouseUp Events

Syntax

Sub *ControlName*_MouseDown(Button As Integer, Shift As Integer, x As OLE_XPOS_PIXELS, y As OLE_YPOS_PIXELS)

Sub *ControlName*_MouseMove(Button As Integer, Shift As Integer, x As OLE_XPOS_PIXELS, y As OLE_YPOS_PIXELS)

Sub *ControlName*_MouseUp(Button As Integer, Shift As Integer, x As OLE_XPOS_PIXELS, y As OLE_YPOS_PIXELS)

Applies To

CWGraph3D

Purpose

Generates the MouseDown event when you click the mouse on the control.

Generates the MouseMove event when you move the mouse over the control.

Generates the MouseUp event when you release the mouse on the control.

Remarks

For more information, refer to Visual Basic help.

Parameters

Button As Integer

The state of the mouse buttons. The Button argument can be any combination of the following values: 1 for the left button, 2 for the right button, or 4 for the middle button. For example, if both the left and right mouse buttons are pressed, the Button argument is 3(1 + 2).

Shift As Integer

The state of the <Shift> key, <Alt> key, and <Ctrl> key. The Shift argument can be any combination of the following values: 1 for <Shift> , 2 for <Ctrl> , or 4 for <Alt> . For example, if the <Shift> and <Ctrl> keys are pressed, the value is 3 (1 + 2).

x As ole xpos pixels

The X-coordinate of the current location of the pointer.

y As ole ypos pixels

The Y-coordinate of the current location of the pointer.

See Also

Click

DblClick

Pan Event

Syntax

Sub *ControlName_*Pan(NewXCenter As Variant, NewYCenter As Variant, NewZCenter As Variant)

Applies To

CWGraph3D

Purpose

Generates the Pan event when you pan the graph up and down or left and right.

Remarks

This event is generated only if CWGraph3D.TrackMode is cwG3DTrackZoomPanRotate.

Panning is performed by holding down the key and the left mouse button and moving the mouse.

Parameters

NewXCenter As Variant

New x coordinate of the center of the view.

NewYCenter As Variant

New y coordinate of the center of the view.

NewZCenter As Variant

New z coordinate of the center of the view.

Example

'Enables zooming, panning, and rotating the graph CWGraph3D1.TrackMode = cwG3DTrackZoomPanRotate

See Also

ViewXCenter

<u>ViewYCenter</u>

<u>ViewZCenter</u>

Zoom

Rotate

<u>TrackMode</u>

Rotating, Panning, and Zooming (concept)

Mouse Events (3D Graph) Example

PlotAreaMouseDown, PlotAreaMouseMove, PlotAreaMouseUp Events

Syntax

Sub *ControlName_*PlotAreaMouseDown(Button As Integer, Shift As Integer, XNear As Variant, YNear As Variant, XFar As Variant, XFar As Variant, YFar As Variant, ZFar As Variant)

Sub *ControlName_*PlotAreaMouseMove(Button As Integer, Shift As Integer, XNear As Variant, YNear As Variant, ZNear As Variant, XFar As Variant, YFar As Variant, ZFar As Variant)

Sub *ControlName_*PlotAreaMouseUp(Button As Integer, Shift As Integer, XNear As Variant, YNear As Variant, ZNear As Variant, XFar As Variant, YFar As Variant, ZFar As Variant)

Applies To

CWGraph3D

Purpose

Generates PlotAreaMouseDown when you click the mouse on the plot area.

Generates PlotAreaMouseMove when you move the mouse over the plot area.

Generates PlotAreaMouseUp when you release the mouse over the plot area.

Remarks

These events are generated only if CWGraph3D.TrackMode is set to cwGTrackAllEvents or cwGTrackPlotAreaEvents.

In Visual Basic, use the following constants with the Button and Shift parameters:

vbLeftButton1Left button is pressedvbRightButton2Right button is pressedvbMiddleButton4Middle button is pressedvbShiftMask1<Shift> key is pressedvbCtrlMask2<Ctrl> key is pressedvbAltMask4<Alt> key is pressed

When you display a three-dimensional object on a two-dimensional computer screen, you see a projection of the 3D space to the 2D space. For example, visualize a three-dimensional object, like a streetlight, casting a shadow on the ground. The shadow, which is being cast on a flat (or two-dimensional) surface, actually loses a dimension. Because of this loss of a dimension, multiple points in the 3D space are projected onto one point in the 2D space. That is, the points Pn = (XNear, YNear, ZNear) and Pf = (XFar, YFar, ZFar) are two 3D points that get mapped to the 2D mouse location. With these two coordinates, you can find the line on which every 3D point passing through Pn and Pf is mapped to the same 2D location. Mathematically, a 3D point Pn and Pn and

Note: If you are trying to find the coordinates of your mouse pointer on a plot, use one of the PlotMouse events instead.

Parameters

Button As Integer

The button that is pressed, released, or, in the case of a mouse move, the buttons that are down. The Button argument is a bit field with bits corresponding to the left button (Bit 0), right button (Bit 1), and middle button (Bit 2).

Shift As Integer

The state of the <Shift>, <Ctrl>, and <Alt> keys when the button specified in the Button argument is pressed, released, or moved. The Shift argument is a bit field with the least-significant bits corresponding to the <Shift> key (Bit 0), the <Ctrl> key (Bit 1), and the <Alt> key (Bit 2). Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed.

XNear As Variant

The X value of a point under the mouse pointer that was closer to the user when the event occurred.

YNear As Variant

The Y value of a point under the mouse pointer that was closer to the user when the event occurred.

ZNear As Variant

The Z value of a point under the mouse pointer that was closer to the user when the event occurred.

XFar As Variant

The X value of a point under the mouse pointer that was further from the user when the event occurred.

YFar As Variant

The Y value of a point under the mouse pointer that was further from the user when the event occurred.

ZFar As Variant

The Z value of a point under the mouse pointer that was further from the user when the event occurred.

See Also

<u>TrackMode</u>

 $\underline{PlotMouseDown}$

<u>PlotMouseMove</u>

<u>PlotMouseUp</u>

Mouse Events (3D Graph) Example

PlotMouseDown, PlotMouseMove, PlotMouseUp Events

Syntax

Sub *ControlName_*PlotMouseDown(Button As Integer, Shift As Integer, XData As Variant, YData As Variant, ZData As Variant, PlotIndex As Integer, PointI As Long, PointJ As Long)

Sub *ControlName_*PlotMouseMove(Button As Integer, Shift As Integer, XData As Variant, YData As Variant, ZData As Variant, PlotIndex As Integer, PointI As Long, PointJ As Long)

Sub *ControlName_*PlotMouseUp(Button As Integer, Shift As Integer, XData As Variant, YData As Variant, ZData As Variant, PlotIndex As Integer, PointI As Long, PointJ As Long)

Applies To

CWGraph3D

Purpose

Generates PlotMouseDown when you click the mouse on a plot.
Generates PlotMouseMove when you move the mouse over a plot.
Generates PlotMouseUp when you release the mouse over a plot.

Remarks

These events are generated only if CWGraph3D.TrackMode is set to cwGTrackAllEvents.

In Visual Basic, use the following constants with the Button and Shift parameters:

vbLeftButton
vbRightButton
vbMiddleButton
d Middle button is pressed
vbShiftMask
vbCtrlMask
vbAltMask
Left button is pressed
Middle button is pressed
vbShift> key is pressed
vbCtrlMask
4 <Alt> key is pressed
vbAltMask
4 <Alt> key is pressed

Parameters

Button As Integer

The button that is pressed, released, or in the case of a mouse move, the buttons that are down. The Button argument is a bit field with bits corresponding to the left button (Bit 0), right button (Bit 1), and middle button (Bit 2).

Shift As Integer

The state of the <Shift>, <Ctrl>, and <Alt> keys when the button specified in the Button argument is pressed, released, or moved. The Shift argument is a bit field with the least-significant bits corresponding to the <Shift> key (Bit 0), the <Ctrl> key (Bit 1), and the <Alt> key (Bit 2). Some, all, or none of the bits can be set, indicating that some, all, or none of the keys are pressed.

XData As Variant

The X value of the point nearest to the mouse pointer when the event occurred.

YData As Variant

The Y value of the point nearest to the mouse pointer when the event occurred.

ZData As Variant

The Z value of the point nearest to the mouse pointer when the event occurred.

PlotIndex As Integer

The index of the plot (one-based).

PointI As Long

The i index of the point (zero-based).

PointJ As Long

The j index of the point (zero-based).

See Also

<u>Plots</u>

<u>TrackMode</u>

Mouse Events (3D Graph) Example

ReadyStateChange Event

Syntax

Sub ControlName_ReadyStateChange()

Applies To

CWGraph3D

Purpose

Generates the event when the ready state changes.

Remarks

When this event is generated, use the ReadyState property to determine the new ready state of the control.

See Also

ReadyState

Rotate Event

Syntax

Sub ControlName_Rotate(NewLatitude As Variant, NewLongitude As Variant)

Applies To

CWGraph3D

Purpose

Generates the Rotate event when you rotate the graph.

Remarks

This event is generated only if CWGraph3D.TrackMode is set to cwG3DTrackZoomPanRotate.

You can rotate a graph by holding down the left mouse button and moving the mouse.

Parameters

NewLatitude As Variant

New latitude of the viewing position.

NewLongitude As **Variant**

New longitude of the viewing position.

Example

'Enables zooming, panning, and rotating the graph CWGraph3D1.TrackMode = cwG3DTrackZoomPanRotate

See Also

<u>ViewLatitude</u>

ViewLongitude

Zoom

<u>Pan</u>

<u>TrackMode</u>

Rotating, Panning, and Zooming (concept)

Mouse Events (3D Graph) Example

Zoom Event

Syntax

Sub *ControlName*_Zoom(NewDistance As Variant)

Applies To

CWGraph3D

Purpose

Generates the Zoom event when you zoom in or out on the plot.

Remarks

This event is generated only if CWGraph3D.TrackMode is set to cwG3DTrackZoomPanRotate.

You can zoom a graph by pressing and holding the key and the left mouse button and moving the mouse forward and backward. Zooming also can be performed by rotating the mouse wheel if your mouse is equipped with one. Press and hold the key to decrease the granularity of zooming with the mouse wheel.

Parameters

NewDistance As Variant

New distance of the viewing position from the origin.

Example

'Enables zooming, panning, and rotating the graph CWGraph3D1.TrackMode = cwG3DTrackZoomPanRotate

See Also

ViewDistance

<u>Pan</u>

Rotate

<u>TrackMode</u>

Rotating, Panning, and Zooming (concept)

Mouse Events (3D Graph) Example



Three-Dimensional Plots

Plot on a three-dimensional graph. The 3D Graph control can visualize data as a curve or surface using the Plot methods. A curve is comprised of a one-dimensional array of individual points on the graph, each point having an X, Y, and Z coordinate. Those points are then connected with a line. A curve is ideal for visualizing the path of a moving point, such as the flight path of a bullet. The 3D Graph control provides one method for plotting a curve: the cwgraph3D.Plot3DCurve method.

A surface plot is comprised of a two-dimensional array of points on the graph, each having an X, Y, and Z coordinate. Those points are then connected, forming a three-dimensional surface view of the data. For example, you might use a surface plot for terrain mapping. The 3D Graph control provides three methods for plotting a surface, depending on the type of data you have:

- CWGraph3D.Plot3DSimpleSurface: Use this method to plot one (or two) 2D array(s) of data, where the array provides the Z data for the surface while the indices provide the X and Y data. The optional second array is used to specify magnitude data.
- CWGraph3D.Plot3DParametricSurface: Use this method when you have three (or four) 2D arrays of data. The optional fourth array is used to specify magnitude data.
- CWGraph3D.Plot3DSurface: Use this method to plot two 1D arrays and one (or two) 2D array(s) of data, where the first two arrays provide the X and Y data and the third array provides the Z data. The optional fourth array is used to specify magnitude data.

See Also

CWGraph3D

CWPlot3D

Plot Styles Example

CWGraph3D Overview

PlotTemplate Object (concept)

Color Property

Syntax

CWLabels3D.Color

Data Type

Color

Purpose

Specifies the color of the axis labels.

Example

'Make the x-axis labels red CWGraph3D1.Axes.Item(1).Labels.Color = vbRed

Font Property

Syntax

CWLabels3D.Font

Data Type

<u>Font</u>

Purpose

Specifies the font of the axis labels.

Example

'Set the x-axis label's font to bold CWGraph3D1.Axes.Item(1).Labels.Font.Bold = True

Normal Property

Syntax

CWLabels3D.Normal

Data Type

Boolean

Purpose

Specifies if labels appear at the normal axis position.

Remarks

You can set an axis label to appear at two different positions. For example, when you set a label for the x-axis in the normal position, the label appears along the bottom edge of the grid panel. If you set the x-axis label in the opposite position, the label appears above the grid panel that is opposite the normal position axis.

These properties operate independently of each other, and you can enable one or both properties for an axis.

Example

'Label the second axis in the normal position CWGraph3D1.Axes.Item(2).Labels.Normal = True

See Also

<u>Opposite</u>

Opposite Property

Syntax

CWLabels3D.Opposite

Boolean

Specifies if labels appear at the opposite axis position.

Remarks

You can set an axis label to appear at two different positions. For example, when you set a label for the x-axis in the normal position, the label appears along the bottom edge of the grid panel. If you set the x-axis label in the opposite position, the label appears above the grid panel that is opposite the normal position axis.

These properties operate independently of each other, and you can enable one or both properties for an axis.

Example

'Label the x-axis in the opposite position CWGraph3D1.Axes.Item(1).Labels.Opposite = True

See Also

<u>Normal</u>

Attenuation Property

Syntax

CWLight. Attenuation

<u>CWLightAttenuationStyles</u>

You can use the following constants with this data type:

- cwLightNone–Specifies that the light does not dim.
- cwLinear–Specifies that the light dims linearly.
- cwQuadratic–Specifies that the light dims quadratically.

Specifies the attenuation style of the light source.

Remarks

Attenuation controls how the light dims as the distance from the light source changes. For example, setting Attenuation to cwQuadratic causes the light to dim more quickly than setting Attenuation to cwLinear.

Example

'Set the attenuation style to quadratic CWGraph3D1.Lights.Item(1).Attenuation = cwQuadratic

Color Property

Syntax

CWLight.Color

Color

Specifies the color of the light source.

Example

CWGraph3D1.Lights.Item(1).Color = vbBlue

See Also

CWGraph3D.AmbientLightColor

Distance Property

Syntax

CWLight. Distance

<u>Double</u>

Specifies the distance of the light source from the origin of the graph.

Example

CWGraph3D1.Lights.Item(1).Distance = 10

Enabled Property

Syntax

CWLight.Enabled

Boolean

Enables the light when set to True.

Example

'Enable light source CWGraph3D1.Lights.Item(1).Enabled = True

See Also

CWGraph3D.Lighting

Latitude Property

Syntax

CWLight.Latitude

<u>Double</u>

Specifies the latitudinal position in degrees of the light source.

Example

'Set the latitude of the light source to 35 degrees CWGraph3D1.Lights.Item(1).Latitude = 35 'Set the longitude of the light source to 40 degrees CWGraph3D1.Lights.Item(1).Longitude = 40

Longitude Property

Syntax

CWLight.Longitude

<u>Double</u>

Specifies the longitudinal position in degrees of the light source.

Example

'Set the latitude of the light source to 35 degrees CWGraph3D1.Lights.Item(1).Latitude = 35 'Set the longitude of the light source to 40 degrees CWGraph3D1.Lights.Item(1).Longitude = 40

AutoScale Property

Syntax

CWPlot3D.AutoScale

Boolean

Specifies if the maximum and minimum of the data in the plot affect the maximum and minimum of an autoscaling axis.

Remarks

To make the graph autoscale to a plot, set the CWAxis3D.AutoScale property to True.

Example

'Autoscale the graph to a plot CWGraph3D1.Axes.Item(1).AutoScale = True

See Also

CWAxis3D.AutoScale

CacheData Property

Syntax

CWPlot3D.CacheData

Data Type

Boolean

Purpose

Improves the drawing performance by caching intermediate calculations when set to True.

Remarks

When caching data, more memory is used. When the CacheData property is set to False, the plot does not cache intermediate calculations, which reduces memory usage.

Example

'Do not cache intermediate calculations CWGraph3D1.Plots.Item(1).CacheData = False

ColorMapAutoScale Property

Syntax

CWPlot3D.ColorMapAutoScale

Data Type

Boolean

Purpose

Specifies if the color map rescales its values when the plot data changes.

Remarks

When this property is set to True, the color map rescales its values so that the minimum value of the color map maps to the minimum data value, and the maximum value of the color map maps to the maximum data value. Predefined color maps always autoscale, regardless of this property value.

See Also

<u>ColorMapValues</u>

 $\underline{ColorMapColors}$

 $\underline{ColorMapStyle}$

ColorMapColors, ColorMapValues Properties

Syntax

 $\underline{\text{CWPlot3D}}. Color Map Colors$

 $\underline{\text{CWPlot3D}}. Color Map Values$

Data Type

<u>Variant</u>

Purpose

Specifies a custom color map when used with the ColorMapValues property. Each element in the value vector and its corresponding element in the color vector are a value-color pair in the color map.

Remarks

To specify a custom color map with ColorMapValues and ColorMapColors, first set ColorMapStyle to cwCustom. The ColorMapColors and ColorMapValues vectors must have at least two elements, and the ColorMapValues must have no duplicates and be sorted in increasing value.

Example

```
Dim values(1 To 3)
Dim colors(1 To 3)
values(1) = 0.0
values(2) = 0.5
values(3) = 1.0
colors(1) = vbRed
colors(2) = vbGreen
colors(3) = vbBlue
CWGraph3D1.Plots(1).ColorMapStyle = cwCustom
CWGraph3D1.Plots(1).ColorMapValues = values
CWGraph3D1.Plots(1).ColorMapColors = colors
```

See Also

ColorMapStyle

ColorMapInterpolate Property

Syntax

CWPlot3D. ColorMapInterpolate

Data Type

Boolean

Purpose

Specifies if the color map interpolates between colors.

Remarks

When this property is set to True, the color map interpolates between colors in the map when a value is between two color map values. When this property is set to False, all values get mapped to the color of the next lowest value.

See Also

ColorMapStyle
ColorMapValues

ColorMapLog Property

Syntax

CWPlot3D.ColorMapLog

Data Type

Boolean

Purpose

Specifies if the color map interpolates logarithmically.

Remarks

When this property is set to True, the color map interpolates logarithmically between colors. When this property is set to False, the color map interpolates linearly.

Use this property with logarithmic z-axis. This property only has effect when ColorMapInterpolate is set to True.

See Also

 $\underline{Color Map Interpolate}$

ColorMapStyle
CWAxis3D.Log

ColorMapStyle Property

Syntax

CWPlot3D.ColorMapStyle

Data Type

CWColorMapStyles

You can use the following constants with this data type:

- cwColorSpectrum—The color map is a color spectrum.
- cwCustom—The color map is custom-defined. Use the CWPlot3D.ColorMapValues and CWPlot3D.ColorMapColors properties to specify a custom color map.
- cwGrayScale—The color map is grayscale.
- cwNone–No color map is applied.
- cwShaded–The color map is shaded.

Purpose

Specifies the color map style that is used by the plot.

Remarks

The color map is applied to data points by mapping the magnitude (W) value of each data point to a color in the color map. If magnitude (W) data is not specified for the plot, the Z data is used for the magnitude data.

See Also

 $\underline{ColorMapInterpolate}$

 $\underline{ColorMapLog}$

 $\underline{ColorMapAutoScale}$

ColorMapValues

ColorMapColors

CoordinateSystem Property

Syntax

CWPlot3D. Coordinate System

Data Type

 $\underline{CWCoordinateSystems}$

You can use the following constants with this data type:

- cwCartesian—Cartesian
- cwCylindrical—Cylindrical
- cwSpherical—Spherical

Specifies the coordinate system of the plot.

Example

'Plot the data using a spherical coordinate system CWGraph3D1.Plots.Item(1).CoordinateSystem = cwSpherical

Enabled Property

Syntax

CWPlot3D.Enabled

Data Type

Boolean

Specifies if the plot generates mouse events when CWGraph3D.TrackMode is set to cwGTrackAllEvents and the plot is visible.

Example

'Set TrackMode to cwG3DTrackAllEvents CWGraph3D1.TrackMode = cwG3DTrackAllEvents 'Set plot to visible CWGraph3D1.Plots.Item(1).Visible = True 'Enable mouse events CWGraph3D1.Plots.Item(1).Enabled = True

See Also

CWGraph3D.TrackMode

FillColor Property

Syntax

CWPlot3D.FillColor

Data Type

Color

Specifies the color to use for drawing the plot surface.

Remarks

The fill color is used in the following plot styles:

cwSurface

cwSurfaceLine

cwSurfaceContour

cwSurfaceNormal

Example

'Make the plot surface green CWGraph3D1.Plots.Item(1).FillColor = vbGreen 'Use a flat fill style CWGraph3D1.Plots.Item(1).FillStyle = cwFlat

See Also

<u>Style</u>

<u>FillStyle</u>

FillStyle Property

Syntax

CWPlot3D.FillStyle

Data Type

CWFillStyles

You can use the following constants with this data type:

- cwFlat–Flat
- cwSmooth–Smooth

Specifies the fill style of the plot.

Example

'Make the plot surface green CWGraph3D1.Plots.Item(1).FillColor = vbGreen 'Use a flat fill style CWGraph3D1.Plots.Item(1).FillStyle = cwFlat

See Also

FillColor

LineColor Property

Syntax

CWPlot3D.LineColor

Data Type

Color

Specifies the color of lines for connecting points in the plot.

Example

CWGraph3D1.Plots.Item(1).LineColor = vbBlue

See Also

LineStyle

LineWidth

LineStyle Property

Syntax

CWPlot3D.LineStyle

Data Type

CW3DLineStyles

You can use the following constants with this data type:

- cwLine3DDash–Dashed line
- cwLine3DDashDot-Dash-dot line
- cwLine3DDot–Dotted line
- cwLine3DNone–No line
- cwLine3DSolid-Solid line

Specifies the style of lines for connecting points on a plot.

Example

CWGraph3D1.Plots.Item(1).LineStyle = cwLine3DDash

See Also

LineColor

LineWidth

LineWidth Property

Syntax

CWPlot3D.LineWidth

Data Type

<u>Double</u>

Specifies the width of the plotting line. The width range is 0 to 100.

Example

'Set the line width of the first plot to 0.5 CWGraph3D1.Plots.Item(1).LineWidth = 0.5

See Also

LineColor

LineStyle

MultiPlot Property

Syntax

CWPlot3D.MultiPlot

Data Type

Boolean

Determines if the CWGraph3D plot methods can use this plot.

Remarks

The CWGraph3D Plot methods will plot data to the first plot in the array when MultiPlot is set to True. If none of the plots have MultiPlot set to True, then the graph adds a new plot.

Set this property to False to create a plot that the CWGraph3D plot methods cannot use.

Example

'Add a new plot if a plot is not available CWGraph3D1.Plots.Add.MultiPlot =False 'The next event causes the graph to add a plot or use another plot CWGraph3D1.Plot3DSurface x y z

See Also

CWGraph3D.Plot3DCurve

Name Property

Syntax

CWPlot3D.Name

Data Type

String

Purpose

Specifies the name of the plot.

Remarks

Use this name for indexing the Plots collection on a CWGraph3D object. If more than one plot has the same name, the first matching plot is used.

PointColor Property

Syntax

CWPlot3D.PointColor

Data Type

Color

Purpose

Specifies the color for points on a plot.

Remarks

The PointColor property is used with the following plot styles: cwPoint cwLinePoint

Example

'Set the plot style to cwLinePoint CWGraph3D1.Plots.Item(1).Style = cwLinePoint 'Set the points'color to blue CWGraph3D1.Plots.Item(1).PointColor = vbBlue

See Also

<u>PointStyle</u>

<u>PointSize</u>

PointFrequency

PointFrequency Property

Syntax

CWPlot3D.PointFrequency

Data Type

Long

Purpose

Specifies how often points in the plot are drawn.

Remarks

A frequency of one indicates that all points must be drawn. A frequency of two indicates that every other point must be drawn. The frequency must always be greater than zero.

Example

'Draw every other point CWGraph3D1.Plots.Item(1).PointFrequency = 2

See Also

<u>PointColor</u>

<u>PointSize</u>

<u>PointStyle</u>

PointSize Property

Syntax

CWPlot3D.PointSize

Data Type

<u>Double</u>

Purpose

Specifies the size of the plot points in points.

Remarks

The valid point size range is 0 to 100.

Example

'Set the point size to 5 CWGraph3D1.Plots.Item(1).PointSize = 5

See Also

PointColor

<u>PointStyle</u>

PointFrequency

PointStyle Property

Syntax

CWPlot3D.PointStyle

Data Type

CW3DPointStyles

You can use the following constants with this data type:

- cwPoint3DAsterisk–Asterisk
- cwPoint3DBoldX-Bold X
- cwPoint3DEmptyCircle–Empty circle
- cwPoint3DEmptySquare–Empty square
- cwPoint3DNone–No point
- cwPoint3DSolidCircle–Solid circle
- cwPoint3DSolidCube-Solid cube
- cwPoint3DSolidDiamond–Solid diamond
- cwPoint3DSolidSphere–Solid sphere
- cwPoint3DSolidSquare—Solid square
- cwPoint3DWireframeCube-Wireframe cube
- cwPoint3DWireframeSphere–Wireframe sphere

Purpose

Specifies the image drawn at each point on a plot.

Example

'Place an asterisk at each point on the plot.
CWGraph3D1.Plots.Item(1).PointStyle = cwPoint3DAsterisk

See Also

<u>PointColor</u>

<u>PointSize</u>

PointFrequency

ProjectionXY, ProjectionXZ, ProjectionYZ Properties

Syntax

 $\underline{\text{CWPlot3D}}. Projection XY$

 $\underline{\text{CWPlot}3D}. Projection XZ$

 $\underline{\text{CWPlot}3D}. Projection YZ$

Data Type

Boolean

Purpose

ProjectionXY draws the XY plane projection of the plot when set to True. ProjectionXZ draws the XZ plane projection of the plot when set to True. ProjectionYZ draws the YZ plane projection of the plot when set to True.

Example

'Draw the YZ plane projection CWGraph3D1.Plots.Item(1).ProjectionYZ = True 'Show only the YZ projection and hide the plot CWGraph3D1.Plots.Item(1).ShowProjectionOnly = True

See Also

ShowProjectionsOnly

ShowProjectionsOnly Property

Syntax

CWPlot3D. ShowProjectionsOnly

Data Type

Boolean

Displays only the specified projection of the plot.

Remarks

When set to True, the ShowProjectionsOnly property draws all the projections that are currently enabled, but does not draw the plot.

When set to False, the ShowProjectionsOnly property draws all the projections that are currently enabled and draws the plot.

Example

'Draw the YZ plane projection CWGraph3D1.Plots.Item(1).ProjectionYZ = True 'Show only the YZ projection and hide the plot CWGraph3D1.Plots.Item(1).ShowProjectionOnly = True

See Also

ProjectionXY

<u>ProjectionXZ</u>

 $\underline{ProjectionYZ}$

Style Property Syntax

CWPlot3D.Style

Data Type

CWPlot3DStyles

You can use the following constants with this data type:

- cwContourLine-Contour
- cwHiddenLine-Hidden line
- cwLine–Line
- cwLinePoint–Line-point
- cwPoint–Point
- cwSurface–Surface
- cwSurfaceContour–Surface-Contour
- cwSurfaceLine–Surface-line
- cwSurfaceNormal-Surface-normal

Specifies the style of the plot.

Example

'Set the plot style to cwContourLine CWGraph3D1.Plots.Item(1).Style = cwContourLine

Transparency Property

Syntax

CWPlot3D. Transparency

Data Type

Long

Indicates the percentage of transparency of the plot.

Remarks

A value of 0 specifies opaque, and a value of 100 specifies completely transparent.

Example

'Make the plot 50% transparent CWGraph3D1.Plots.Item(1).Transparency = 50

See Also

Transparency Example

Visible Property

Syntax

CWPlot3D.Visible

Data Type

Boolean

Specifies if the plot is visible or hidden.

Remarks

Set this property to False to hide a plot without discarding its data. An invisible plot does not generate mouse events.

Example

'Hide the plot CWGraph3D1.Plots.Item(1).Visible = False

See Also

Enabled

XAxis, YAxis, ZAxis Properties (Read Only)

Syntax

CWPlot3D.XAxis

CWPlot3D. YAxis

CWPlot3D.ZAxis

Data Type

CWAxis3D

Specifies the x-, y-, and z-axes for the plot. These are read-only properties.

Example

'Autoscale the x-axis for the plot CWGraph3D1.Plots.Item(1).XAxis.AutoScale = True

See Also

CWAxis3D

ClearData Method

Syntax

CWPlot3D.ClearData

Clears the data currently displayed in the plot.

Example

'Clear the plot data CWGraph3D1.Plots.Item(1).ClearData

Plot3DCurve Method

Syntax

CWPlot3D. Plot3DCurve xVector, yVector, zVector [, wVector]

Plots three (or four) 1D arrays of data as a parametric curve.

Remarks

The first array is the X data, the second array is the Y data, and the third array is the Z data. The optional fourth array is used to specify magnitude data.

Parameters

xVector As Variant

1D array of X data

yVector As variant

1D array of Y data

zVector As Variant

1D array of Z data

 $wVector\ As\ \underline{{}^{\text{Variant}}}$

[Optional] 1D array of magnitude data

See Also

CWGraph3D.Plot3DCurve

 $\underline{Plot 3D Simple Surface}$

Plot3DSurface

<u>Plot3DParametricSurface</u>

Plot3DMesh

<u>Plot Styles Example</u>

Three-Dimensional Plots (concept)

Plot3DMesh Method

Syntax

CWPlot3D. Plot3DMesh xVector, yVector, zVector [, wVector]

Plots three (or four) 1D arrays of data as a triangulated surface.

Remarks

The first array is the X data, the second array is the Y data, and the third array is the Z data. The optional fourth array is used to specify magnitude data.

Parameters

xVector As Variant

1D array of X data

yVector As variant

1D array of Y data

zVector As Variant

1D array of Z data

 $wVector\ As\ \underline{{}^{\text{Variant}}}$

[Optional] 1D array of magnitude data

CWGraph3D.Plot3DMesh

Plot3DCurve

 $\underline{Plot 3D Simple Surface}$

Plot3DSurface

<u>Plot3DParametricSurface</u>

Plot3DParametricSurface Method

Syntax

CWPlot3D. Plot3DParametricSurface xMatrix, yMatrix, zMatrix [, wMatrix]

Purpose

Plots three (or four) 2D arrays of data as a parametric surface.

Remarks

The first array is the X data, the second array is the Y data, and the third array is the Z data. The optional fourth array is used to specify magnitude data.

Parameters

xMatrix As Variant

2D array of X data

yMatrix As Variant

2D array of Y data

zMatrix As Variant

2D array of Z data

wMatrix As Variant

[Optional] 2D array of magnitude data

 $\underline{CWGraph3D.Plot3DParametricSurface}$

Plot3DCurve

 $\underline{Plot 3D Simple Surface}$

Plot3DSurface

Plot3DMesh

<u>Plot Styles Example</u>

Three-Dimensional Plots (concept)

Plot3DSimpleSurface Method

Syntax

CWPlot3D. Plot3DSimpleSurface zMatrix [, wMatrix]

Purpose

Plots one (or two) 2D array(s) of data as a simple surface.

Remarks

The array provides the Z data for the surface while the indices provide the X and Y data. The optional second array is used to specify magnitude data.

Parameters

zMatrix As Variant

2D array of Z data

wMatrix As Variant

[Optional] 2D array of magnitude data

 $\underline{CWGraph3D.Plot3DSimpleSurface}$

Plot3DCurve

Plot3DSurface

 $\underline{Plot 3DParametric Surface}$

Plot3DMesh

<u>Plot Styles Example</u>

Three-Dimensional Plots (concept)

Plot3DSurface Method

Syntax

CWPlot3D. Plot3DSurface xVector, yVector, zMatrix [, wMatrix]

Purpose

Plots two 1D arrays and one (or two) 2D array(s) of data as a 3D surface.

Remarks

The first two arrays provide the X and Y data, while the third array provides the Z data. The optional fourth array is used to specify magnitude data.

Parameters

xVector As Variant

1D array of X data

yVector As variant

1D array of Y data

zMatrix As Variant

2D array of Z data

wMatrix As Variant

[Optional] 2D array of magnitude data

CWGraph3D.Plot3DSurface

Plot3DCurve

 $\underline{Plot 3D Simple Surface}$

Plot3DParametricSurface

Plot3DMesh

<u>Plot Styles Example</u>

Three-Dimensional Plots (concept)



PlotTemplate Object

Special instance of a CWPlot3D object used to specify the default property values of new plots. The PlotTemplate object properties are the identical to those of the Plot3D object and are set through the property pages or programmatically.

The PlotTemplate property values are used as default property values for newly created plots when the Add method is called on the Plots3D collection. If you call one of the Plot methods on CWGraph3D and none of the plots have MultiPlot set to True, CWGraph3D automatically creates a new plot with the PlotTemplate object properties.

CWPlot3D

 $\underline{CWGraph3D.PlotTemplate}$

AutoDivisions Property

Syntax

CWTicks3D. Auto Divisions

Data Type

Boolean

Purpose

Specifies if divisions on the specified axis are automatically calculated to display the data.

Example

'Automatically calculate divisions on the graph's x-axis CWGraph3D1.Axes.Item(1).Ticks.AutoDivisions = True

MajorDivisions

 $\underline{Major Units Interval}$

Inside Property

Syntax

CWTicks3D.Inside

Data Type

Boolean

Purpose

Specifies if tick marks appear on the inside of the axis plane.

Example

'Draw ticks on the inside and outside CWGraph3D1.Axes.Item(1).Ticks.Inside = True

CWGraph3D1.Axes.Item(1).Ticks.Outside = True

Outside

MajorDivisions Property

Syntax

CWTicks3D. Major Divisions

Data Type

Variant

Purpose

Specifies the number of major divisions of ticks on a specified axis.

Remarks

Set MajorDivisions to 10 to split the axis range into 10 equal parts.

Setting this property overrides the MajorUnitsInterval, MinorUnitsInterval, and MajorUnitsBase properties and causes the MinorDivisions property to take effect.

Example

'Set 10 major divisions on the graph's x-axis CWGraph3D1.Axes.Item(1).Ticks.MajorDivisions = 10

'Set 2 minor divisions per major division CWGraph3D1.Axes.Item(1).Ticks.MinorDivisions = 2

See Also

MajorUnitsInterval
MinorDivisions

MajorGrid Property

Syntax

CWTicks3D.MajorGrid

Data Type

Boolean

Purpose

Specifies if major grid lines appear.

Example

'Enable Major and Minor grid lines on the graph's x-axis CWGraph3D1.Axes.Item(1).Ticks.MajorGrid = True CWGraph3D1.Axes.Item(1).Ticks.MinorGrid = True

'Set the grid line colors CWGraph3D1.Axes.Item(1).Ticks.MajorGridColor = vbRed CWGraph3D1.Axes.Item(1).Ticks.MinorGridColor = vbYellow

MajorGridColor Property

Syntax

CWTicks3D.MajorGridColor

Data Type

Color

Purpose

Specifies the color of major grid lines.

Example

'Enable Major and Minor grid lines on the graph's x-axis CWGraph3D1.Axes.Item(1).Ticks.MajorGrid = True CWGraph3D1.Axes.Item(1).Ticks.MinorGrid = True

'Set the grid line colors CWGraph3D1.Axes.Item(1).Ticks.MajorGridColor = vbRed CWGraph3D1.Axes.Item(1).Ticks.MinorGridColor = vbYellow

See Also

MinorGridColor

MajorTickColor Property

Syntax

CWTicks3D.MajorTickColor

Data Type

Color

Purpose

Specifies the color of major ticks.

Example

'Set the tick colors of the graph's x-axis CWGraph3D1.Axes.Item(1).Ticks.MajorTickColor = vbRed CWGraph3D1.Axes.Item(1).Ticks.MinorTickColor = vbYellow

See Also

MinorTickColor

MajorTicks Property

Syntax

CWTicks3D. MajorTicks

Data Type

Boolean

Purpose

Specifies if major ticks appear on a specified axis.

Example

'Turn major and minor ticks on CWGraph3D1.Axes.Item(1).MajorTicks = True CWGraph3D1.Axes.Item(1).MinorTicks = True

See Also

MinorTicks

MajorUnitsBase Property

Syntax

CWTicks3D.MajorUnitsBase

Data Type

<u>Variant</u>

Purpose

Specifies the base number for calculating ticks.

Remarks

When ticks are spaced by units, ticks are placed at locations that fit the following equation:

MajorUnitsBase + (n * MajorUnitsInterval)

where n is any integer.

Setting this property overrides the MajorDivisions and MinorDivisions properties and causes the MajorUnitsInterval and MinorUnitsInterval properties to take effect.

Example

'Place major ticks every 2 units (even numbers) CWGraph3D1.Axes.Item(1).MajorUnitsInterval = 2

'Place minor ticks every .5 units CWGraph3D1.Axes.Item(1).MajorUnitsInterval = .5

'The unit's base is 0 CWGraph3D1.Axes.Item(1).MajorUnitsBase = 0

"To specify ticks at odd numbers, change the base to 1 CWGraph3D1.Axes.Item(1).MajorUnitsBase = 1

'Now, there will be major ticks at ..., -3, -1, 1, 3, 5, ...

See Also

MajorUnitsInterval

MajorDivisions

MajorUnitsInterval Property

Syntax

CWTicks3D.MajorUnitsInterval

Data Type

<u>Variant</u>

Purpose

Specifies the number of units between major divisions.

Remarks

Set this property to 5 and MajorUnitsBase to 0 to place major tick marks at multiples of five.

Setting this property overrides the MajorDivisions and MinorDivisions properties and causes the MajorUnitsBase and MinorUnitsInterval properties to take effect.

Example

'Place major ticks every 2 units (even numbers) CWGraph3D1.Axes.Item(1).MajorUnitsInterval = 2

'Place minor ticks every .5 units CWGraph3D1.Axes.Item(1).MajorUnitsInterval = .5

'The unit's base is 0 CWGraph3D1.Axes.Item(1).MajorUnitsBase = 0

"To specify ticks at odd numbers, change the base to 1 CWGraph3D1.Axes.Item(1).MajorUnitsBase = 1

'Now, there will be major ticks at ..., -3, -1, 1, 3, 5, ...

See Also

MajorUnitsBase

 $\underline{Minor Units Interval}$

 $\underline{Major Divisions}$

MinorDivisions Property

Syntax

CWTicks3D.MinorDivisions

Data Type

Variant

Purpose

Specifies the number of minor divisions for each major division of ticks on a specified axis.

Remarks

Setting this property overrides the MinorUnitsInterval, MajorUnitsInterval, and MajorUnitsBase properties and causes the MajorDivisions property to take effect if AutoDivisions is off (False).

Example

'Set 10 major divisions on the graph's x-axis CWGraph3D1.Axes.Item(1).Ticks.MajorDivisions = 10

'Set 2 minor divisions per major division CWGraph3D1.Axes.Item(1).Ticks.MinorDivisions = 2

See Also

MajorDivisions

 $\underline{Minor Units Interval}$

MinorGrid Property

Syntax

CWTicks3D.MinorGrid

Data Type

Boolean

Purpose

Specifies if minor grid lines appear.

Example

'Enable Major and Minor grid lines on the graph's x-axis CWGraph3D1.Axes.Item(1).Ticks.MajorGrid = True CWGraph3D1.Axes.Item(1).Ticks.MinorGrid = True

'Set the grid line colors CWGraph3D1.Axes.Item(1).Ticks.MajorGridColor = vbRed CWGraph3D1.Axes.Item(1).Ticks.MinorGridColor = vbYellow

See Also

<u>MajorGrid</u>

MinorGridColor Property

Syntax

CWTicks3D.MinorGridColor

Data Type

Color

Purpose

Specifies the color of minor grid lines.

Example

'Enable Major and Minor grid lines on the graph's x-axis CWGraph3D1.Axes.Item(1).Ticks.MajorGrid = True CWGraph3D1.Axes.Item(1).Ticks.MinorGrid = True

'Set the grid line colors CWGraph3D1.Axes.Item(1).Ticks.MajorGridColor = vbRed CWGraph3D1.Axes.Item(1).Ticks.MinorGridColor = vbYellow

See Also

 $\underline{MajorGridColor}$

MinorTickColor Property

Syntax

CWTicks3D.MinorTickColor

Data Type

Color

Purpose

Specifies the color of minor ticks.

Example

'Set the tick colors of the graph's x-axis CWGraph3D1.Axes.Item(1).Ticks.MajorTickColor = vbRed CWGraph3D1.Axes.Item(1).Ticks.MinorTickColor = vbYellow

See Also

 $\underline{MajorTickColor}$

MinorTicks Property

Syntax

CWTicks3D.MinorTicks

Data Type

Boolean

Purpose

Specifies if minor ticks appear on a specified axis.

Example

'Turn major and minor ticks on CWGraph3D1.Axes.Item(1).MajorTicks = True CWGraph3D1.Axes.Item(1).MinorTicks = True

See Also

<u>MajorTicks</u>

MinorUnitsInterval Property

Syntax

CWTicks3D.MinorUnitsInterval

Data Type

<u>Variant</u>

Purpose

Specifies the number of units between minor divisions.

Remarks

Setting this property overrides the MinorDivisions and MajorDivisions properties and causes the MajorUnitsInterval and MajorUnitsBase properties to take effect.

Example

'Place major ticks every 2 units (even numbers) CWGraph3D1.Axes.Item(1).MajorUnitsInterval = 2

'Place minor ticks every .5 units CWGraph3D1.Axes.Item(1).MajorUnitsInterval = .5

'The unit's base is 0 CWGraph3D1.Axes.Item(1).MajorUnitsBase = 0

"To specify ticks at odd numbers, change the base to 1 CWGraph3D1.Axes.Item(1).MajorUnitsBase = 1

'Now, there will be major ticks at ..., -3, -1, 1, 3, 5, ...

See Also

MinorDivisions

 $\underline{MajorUnitsInterval}$

Normal Property

Syntax

CWTicks3D.Normal

Data Type

Boolean

Purpose

Specifies if tick marks appear at the normal axis position.

Remarks

You can set tick marks to appear at two different positions. For example, if you enable tick marks for the x axis in the normal position, the marks appear along the bottom edge of the grid panel. If you set tick marks for the x axis in the opposite position, the marks appear above the grid panel that is opposite the normal position.

These properties operate independently of each other, and you can enable one or both properties for an axis.

Example

'Enable tick marks on the x-axis in the opposite position CWGraph3D1.Axes.Item(1).Ticks.Opposite = True 'Enable tick marks on the x-axis in the normal position CWGraph3D1.Axes.Item(1).Ticks.Normal = True

See Also

<u>Opposite</u>

Opposite Property

Syntax

CWTicks3D.Opposite

Data Type

Boolean

Purpose

Specifies if tick marks appear at the opposite axis position.

Remarks

You can set tick marks to appear at two different positions. For example, if you enable tick marks for the x axis in the normal position, the marks appear along the bottom edge of the grid panel. If you set tick marks for the x axis in the opposite position, the marks appear above the grid panel that is opposite the normal position.

These properties operate independently of each other, and you can enable one or both properties for an axis.

Example

'Enable tick marks on the x-axis in the opposite position CWGraph3D1.Axes.Item(1).Ticks.Opposite = True 'Enable tick marks on the x-axis in the normal position CWGraph3D1.Axes.Item(1).Ticks.Normal = True

See Also

<u>Normal</u>

Outside Property

Syntax

CWTicks3D.Outside

Data Type

Boolean

Purpose

Specifies if tick marks appear on the outside of the axis plane.

Example

'Draw ticks on the outside of the y-axis plane CWGraph3D1.Axes.Item(1).Ticks.Outside = True

See Also

<u>Inside</u>

Name Property

Syntax

CWValuePair.Name

Data Type

String

Purpose

Specifies the name of the value pair.

Remarks

For example, you can use the Name property to add a text label such as "Boiling Point" to a numeric axis at the value 212.

Example

'Create a value pair CWGraph3D1.Axes.Item(1).ValuePairs.Item(1).Add

'Set the name to "Boiling Point" and the value to 212 CWGraph3D1.Axes.Item(1).ValuePairs.Item(1).Name = "Boiling Point" CWGraph3D1.Axes.Item(1).ValuePairs.Item(1).Value = "212"

See Also

<u>Value</u>

Value Property

Syntax

CWValuePair. Value

Data Type

<u>Variant</u>

Purpose

Specifies the value of the value pair.

Remarks

For example, you can use the Value property to set the value on a numeric axis to 212. You can then add a text label, such as "Boiling Point" at the set value.

Example

'Create a value pair CWGraph3D1.Axes.Item(1).ValuePairs.Item(1).Add

'Set the name to "Boiling Point" and the value to 212 CWGraph3D1.Axes.Item(1).ValuePairs.Item(1).Name = "Boiling Point" CWGraph3D1.Axes.Item(1).ValuePairs.Item(1).Value = "212"

See Also

Name

GridLines Property

Syntax

CWValuePairs. GridLines

Data Type

Boolean

Purpose

Specifies if grid lines appear at value pair locations.

Example

'Turn on major ticks and grid lines at value pair locations CWGraph3D1.Axes.Item(1).ValuePairs.MajorTicks = True CWGraph3D1.Axes.Item(1).ValuePairs.GridLines = True

See Also

Location

CWTicks3D

LabelType Property Syntax

CWValuePairs.LabelType

Data Type

<u>CWValuePairLabels</u>

You can use the following constants with this data type:

- cwVPLabelName—The axis draws the name of the value pairs.
- cwVPLabelNone—The axis does not draw the value pair.
- cwVPLabelValue—The axis draws the value of the value pair.

Purpose

Specifies the type of labels to draw for the value pairs.

Example

'Display the value pair's name, not value CWGraph3D1.Axes.Item(1).ValuePairs.LabelType = cwVPLabelName

Location Property

Syntax

CWValuePairs. Location

Data Type

CWValuePairLocations

You can use the following constants with this data type:

- cwVPLocationIndex—The axis draws the value pairs at their index on the axis.
- cwVPLocationValue—The axis draws the value pairs at their value on the axis.

Purpose

Specifies if value pairs are placed on the axis by their value or by their index.

Example

'Locate the value pairs by value, not by index CWGraph3D1.Axes.Item(1).ValuePairs.Location = cwVPLocationValue

MajorTicks Property

Syntax

CWValuePairs. MajorTicks

Data Type

Boolean

Purpose

Specifies if major ticks appear at the location of value pairs.

Example

'Turn on major ticks and grid lines at value pair locations CWGraph3D1.Axes.Item(1).ValuePairs.MajorTicks = True CWGraph3D1.Axes.Item(1).ValuePairs.GridLines = True

Location

<u>GridLines</u>

CWTicks3D

Swap Method

Syntax

CWValuePairs. Swap element1, element2

Purpose

Swaps two CWValuePair elements, altering their indices.

Parameters

element1 As Variant

Index of a value pair to be swapped

element2 As variant

Index of a second value pair to be swapped

Example

'Swap value pairs 1 and 2 CWGraph3D1.Axes.Item(1).ValuePairs.Swap 1, 2

Data Types for CWGraph3D

Visual Basic C/C++ Description

Boolean VARIANT_BOOL Has the value True (-1) or False (0).

Color OLECOLOR A color. In many containers, colors are treated as 32-bit integers.

Double double 64-bit floating point number.

Double Array LPSAFEARRAY An array of doubles.

Font LPFONTDISP An OLE Automation Interface to a font.

Integer short 16-bit signed integer.
Long long 32-bit signed integer.

LPUNKNOWN LPUNKNOWN

Object LPDISPATCH A generic OLE Automation Interface.

OLE_XPOS_PIXELS long

OLE_YPOS_PIXELS long

Picture LPPICTUREDISP An OLE Automation Interface to a picture or image.

Single single 32-bit floating point number.

String BSTR A string.

Variant LPVARIANT A variant.

Void void Nothing.

Window OLE_HANDLE A handle to a window.

CW3DLineStyles Enumeration

CW3DLineStyles are the constants for the CWPlot3D.LineStyle and CWContour.LineStyle properties.

- cwLine3DDash-Dashed line
- cwLine3DDashDot-Dash-dot line
- cwLine3DDot-Dotted line
- cwLine3DNone-No line
- cwLine3DSolid-Solid line

CWPlot3D.LineStyle

 $\underline{CWContour.LineStyle}$

CWBases Enumeration

CWBases are the constants for the CWContours. Basis property.

- cwMagnitude–Contour levels are based on the plot's magnitude (W) data.
- cwX–Contour levels are based on the plot's X data.
- cwY–Contour levels are based on the plot's Y data.
- cwZ–Contour levels are based on the plot's Z data.

CWContours.Basis

CW3DPointStyles Enumeration

CW3DPointStyles are the constants for the CWPlot3D.PointStyle property.

- cwPoint3DAsterisk–Asterisk
- cwPoint3DBoldX-Bold X
- cwPoint3DEmptyCircle–Empty circle
- cwPoint3DEmptySquare–Empty square
- cwPoint3DNone-No point
- cwPoint3DSolidCircle–Solid circle
- cwPoint3DSolidCube-Solid cube
- cwPoint3DSolidDiamond-Solid diamond
- cwPoint3DSolidSphere–Solid sphere
- cwPoint3DSolidSquare-Solid square
- cwPoint3DWireframeCube–Wireframe cube
- cwPoint3DWireframeSphere–Wireframe sphere

CWPlot3D.PointStyle

CWCursor3DSnapModes Enumeration

CWCursor3DSnapModes are the constants for the CWCursor3D.SnapMode property.

- cwSnapFixed–Fixes the cursor at the specified x, y, and z position. The cursor cannot be moved.
- cwSnapNearestPlot—Snaps the cursor to the nearest plot. You can move the cursor along the nearest plot.
- cwSnapToPlot–Snaps the cursor to a specified plot. You can move the cursor along the specified plot.

CWCursor3D.SnapMode

${\bf CWKey board Modes\ Enumeration}$

Constants that specify how a control responds to keyboard input.

- cwKeyboardHandled–Keystrokes are processed by the control.
- cwKeyboardNone–Keystrokes are ignored by the control.

KeyboardMode

CWGraph3DProjectionStyles Enumeration

CWGraph3DProjectionStyles are the constants for the CWGraph3D.ProjectionStyle property.

- cwOrthographic–Displays orthographic projection, in which all points appear to be the same size regardless of viewing distance.
- cwPerspective—Displays perspective projection, in which point size reflects the viewing distance of the point.

 $\underline{CWGraph3D.ProjectionStyle}$

CWGraph3DTrackModes Enumeration

CWGraph3DTrackModes are the constants for the CWGraph3D.TrackMode property.

- cwG3DTrackAllEvents–Generates mouse events for plots and the plot area.
- cwG3DTrackCursors—Generates cursor events.
- cwG3DTrackPlotAreaEvents—Generates mouse events only for the plot area.
- cwG3DTrackZoomPanRotate–Generates Zoom, Pan, and Rotate events.

CWGraph3D.TrackMode



Accelerated OpenGL Graphics Adapter

Graphics adapter that supports OpenGL hardware acceleration. If your computer has an accelerated OpenGL Graphics Adapter, you can enable the Use 3D Hardware Acceleration option in the General subtab on the Graph property page. Enabling this option forces the 3D Graph control to render 3D graphs directly to the display adapter, which maximizes rendering performance.

Disabling this option forces the control to render 3D graphs to an off-screen frame buffer, which is then copied to the screen. The 3D Graph control caches this frame buffer and updates it only when the graph needs to be rendered (such as adding a plot or changing the view position). For complex 3D graphs, the time to render the image into the frame buffer greatly exceeds the time to copy the frame buffer to the screen. As a result, simple screen updates that do not require the graph to re-render (such as scrolling the page) execute very quickly.

The following table summarizes the advantages and disadvantages of enabling and disabling this option.

Enabled Advantages: Faster rendering, uses less memory

Disadvantages: Always renders the graph, even for simple screen

updates

Disabled Advantages: Fast redraw for simple screen updates

Disadvantages: Uses more memory, slower rendering

 $\underline{CWGraph 3D. Use 3DHardware Acceleration}$

$CWG raph 3DView Modes\ Enumeration$

CWGraph3DViewModes are the constants for the CWGraph3D.ViewMode property.

- cwViewUserDefined–Views a user-defined area.
- cwViewXYPlane–Views the XY plane.
- cwViewXZPlane–Views the XZ plane.
- cwViewYZPlane–Views the YZ plane.

CWGraph3D.ViewMode

CWLightAttenuationStyles Enumeration

CWLightAttenuationStyles are the constants for the CWLight.Attenuation property.

- cwLightNone–Specifies that the light does not dim.
- cwLinear–Specifies that the light dims linearly.
- cwQuadratic–Specifies that the light dims quadratically.

CWLight.Attenuation

CWColorMapStyles Enumeration

CWColorMapStyles are the constants for the CWPlot3D.ColorMapStyle property.

- cwColorSpectrum—The color map is a color spectrum.
- cwCustom—The color map is custom-defined. Use the CWPlot3D.ColorMapValues and CWPlot3D.ColorMapColors properties to specify a custom color map.
- cwGrayScale—The color map is grayscale.
- cwNone–No color map is applied.
- cwShaded–The color map is shaded.

CWPlot3D.ColorMapStyle

${\bf CWCoordinate Systems\ Enumeration}$

 $\ensuremath{\mathsf{CWCoordinateSystems}}$ are the constants for the $\ensuremath{\mathsf{CWPlot3D.CoordinateSystem}}$ property.

- cwCartesian–Cartesian
- cwCylindrical—Cylindrical
- cwSpherical—Spherical

 $\underline{CWPlot3D.CoordinateSystem}$

CWFillStyles Enumeration

CWFillStyles are the constants for the CWPlot3D.FillStyle property.

- cwFlat–Flat
- cwSmooth–Smooth

CWPlot3D.FillStyle

CWPlot3DStyles Enumeration

CWPlot3DStyles are the constants for the CWPlot3D.Style property.

- cwContourLine-Contour
- cwHiddenLine-Hidden line
- cwLine–Line
- cwLinePoint–Line-point
- cwPoint-Point
- cwSurface–Surface
- cwSurfaceContour—Surface-Contour
- cwSurfaceLine–Surface-line
- cwSurfaceNormal–Surface-normal

CWPlot3D.Style

CWValuePairLabels Enumeration

CWValuePairLabels specify how value pairs appear on an axis.

- cwVPLabelName—The axis draws the name of the value pairs.
- cwVPLabelNone—The axis does not draw the value pair.
- cwVPLabelValue—The axis draws the value of the value pair.

 $\underline{CWValuePairs.LabelType}$

CWValuePairLocations Enumeration

CWValuePairLocations specify where value pairs appear on an axis.

- cwVPLocationIndex—The axis draws the value pairs at their index on the axis.
- cwVPLocationValue—The axis draws the value pairs at their value on the axis.