AMChart Object Reference Guide

NetIQ AppManager

Version 6.0



Legal Notice

THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT ARE FURNISHED UNDER AND ARE SUBJECT TO THE TERMS OF A LICENSE AGREEMENT OR A NON-DISCLOSURE AGREEMENT. EXCEPT AS EXPRESSLY SET FORTH IN SUCH LICENSE AGREEMENT OR NON-DISCLOSURE AGREEMENT, NETIQ CORPORATION PROVIDES THIS DOCUMENT AND THE SOFTWARE DESCRIBED IN THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. SOME STATES DO NOT ALLOW DISCLAIMERS OF EXPRESS OR IMPLIED WARRANTIES IN CERTAIN TRANSACTIONS; THEREFORE, THIS STATEMENT MAY NOT APPLY TO YOU.

This document and the software described in this document may not be lent, sold, or given away without the prior written permission of NetIQ Corporation, except as otherwise permitted by law. Except as expressly set forth in such license agreement or non-disclosure agreement, no part of this document or the software described in this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, or otherwise, without the prior written consent of NetIQ Corporation. Some companies, names, and data in this document are used for illustration purposes and may not represent real companies, individuals, or data.

This document could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein. These changes may be incorporated in new editions of this document. NetIQ Corporation may make improvements in or changes to the software described in this document at any time.

Copyright © 1995-2004 NetIQ Corporation, all rights reserved.

U.S. Government Restricted Rights: If the software and documentation are being acquired by or on behalf of the U.S. Government or by a U.S. Government prime contractor or subcontractor (at any tier), in accordance with 48 C.F.R. 227.7202-4 (for Department of Defense (DOD) acquisitions) and 48 C.F.R. 2.101 and 12.212 (for non-DOD acquisitions), the government's rights in the software and documentation, including its rights to use, modify, reproduce, release, perform, display or disclose the software or documentation, will be subject in all respects to the commercial license rights and restrictions provided in the license agreement.

ActiveAgent, ActiveAnalytics, ActiveKnowledge, ActiveReporting, ADcheck, AppAnalyzer, Application Scanner, AppManager, AuditTrack, AutoSync, Chariot, ClusterTrends, CommerceTrends, Configuration Assessor, ConfigurationManager, the cube logo design, DBTrends, DiagnosticManager, Directory and Resource Administrator, Directory Security Administrator, Domain Migration Administrator, End2End, Exchange Administrator, Exchange Migrator, Extended Management Pack, FastTrends, File Security Administrator, Firewall Appliance Analyzer, Firewall Reporting Center, Firewall Suite, Ganymede, the Ganymede logo, Ganymede Software, Group Policy Administrator, Intergreat, Knowledge Scripts, Log Analyzer, Migrate.Monitor.Manage, Mission Critical Software, Mission Critical Software for E-Business, the Mission Critical Software logo, MP3check, NetIQ, the NetIQ logo, the NetIQ Partner Network design, NetWare Migrator, OnePoint, the OnePoint logo, Operations Manager, Qcheck, RecoveryManager, Security Analyzer, Security Manager, Server Consolidator, SQLcheck, VigilEnt, Visitor Mean Business, Visitor Relationship Management, Vivinet, W logo, WebTrends, WebTrends Analysis Suite, WebTrends Data Collection Server, WebTrends for Content Management Systems, WebTrends Intelligence Suite, WebTrends Live, WebTrends Network, WebTrends OLAP Manager, WebTrends Report Designer, WebTrends Reporting Center, WebTrends Warehouse, Work Smarter, WWWorld, and XMP are trademarks or registered trademarks of NetIQ Corporation or its subsidiaries in the United States and other jurisdictions.

All other company and product names mentioned are used only for identification purposes and may be trademarks or registered trademarks of their respective companies.

Contents

Chapter 1 AMChart object reference 5

AMChart 5

AMChart properties 6

ChartStyle 7

ChartTitle 8

ColorScheme 9

DataLabelAngle 9

DecimalPrecision 10

DemoMode 10

DrawBorder 11

DrawHLines 11

DrawVLines 12

HTTPServer 13

MaxDataPoints 13

ViewMode 14

XMLDataInfo 15

YAxisAutoRange 15

YAxisMaxRange 16

YAxisMinRange 16

AMChart methods 17

AddChartHighThreshold 21

AddData 22

AddSeriesHighThreshold 22

AddSeriesLowThreshold (not supported at this

time) 23
BeginSelection 24
EnableDataDetails 25
DefaultRotation 25
DeleteAllSeries 26
DeleteChartHighThresholds 26
DeleteSeries 27
EnableScrolling 27
EstablishSeries 28
EstablishSeriesADO 29
EstablishSeriesADO_DSN 31
EstablishSeriesADO_RS 32
GetBottomYAxisLabel 34
GetChartAggregation 34
GetChartTransparency 35
GetDataAtIndex 36
GetRenderHeight 37
GetRenderWidth 37
GetScrollMax 38
GetScrollMin 38
GetScrollPos 39
GetSeriesCount 39
GetSeriesScale 40
GetSeriesStyle 41
GetSeriesTransparency 41
GetTopYAxisLabel 42
GetYAxisLabel 42
InitXML 43

InsertData 43
PromptDeleteFromLegend 44
RebuildScene 45
RefreshData 45
RemoveAllDataLabels 46
RemoveDataLabel 46
RemoveSelection 47
RenderScene 47
RotateRealTime 48
SaveImage 48
SetBorderColor 49
SetBottomYAxisLabel 50
SetChartStyleArea 50
SetChartStyleAreaStacked 51
SetChartStyleBar 51
SetChartStylePie 52
SetChartStylePoint 52
SetChartStyleRibbon 53
SetChartTransparency 53
SetDataAtDate 54
SetDataAtTime 54
SetDataAtTimeEX 55
SetDataLabel 56
SetRotation 57
SetRenderSize 57
SetScrollPos 58
SetSeriesScale 58
SetSeriesStyle 59

SetSeriesTransparency 61
SetTopYAxisLabel 61
SetYAxisLabel 62
ShowDataViewer 62
ShowDockableLegend 63
ShowDockableTimeframe 63
ShowProperties 64
ShowStaticLegend 64
SortSeriesByLegend 65
SortSeriesByValue 66
UpdateChart 66
ZoomIn 67
ZoomNormal 67
ZoomOut 68
CHART_USER_SETTINGS XML Specification . . . 68
XML 68

Index. 73

Chapter 1

AMChart object reference

This chapter covers the AMChart object properties and methods.

AMChart

AMChart is an OLE automation-based API for rendering charts. It is used by the reporting engine, the Chart Console, the Web Operator Console, and for the XMP Custom Saved Views. It can be used from applications written in C++ and VisualBasic.

AMChart is designed to use data from different types of databases (for example, SQL, Oracle, or Access). Developers who want to include graphical analysis in reports can use AMChart for real-time rendering.

To render a chart:

1. Create the AMChart object, or the AMLayout::ChartGenerator object.
2. Add a series to the chart by making a connection to a database that holds data points. Use the `EstablishSeries()` or `EstablishSeriesADO_DSN()` method to make the connection.

   If you use `EstablishSeries()`:

   - Use `AddData()` to add data points that are not associated with a date/time
   - Use `SetDataAtTime()` or `SetDataAtDate()` to add data points that are associated with a data/time

   If you use `EstablishSeriesADO_DSN()`, there is no need to use `AddData()`, `SetDataAtTime()` or `SetDataAtDATE()` because the chart binds to the ADO recordset and fetches the data points

automatically as needed.

3 Call methods to change the chart settings (for example, rotation, view mode, image size, and color scheme).

4 Call `Update()` to rebuild the scene with the settings from Step 3 and render the chart.

AMChart is never used directly to create charts. For example, the Chart Console and Operator Web Console create charts by calling a method of the AMChartHost object; XMP Custom Saved Views create charts by calling a method of the OMChartHost object. AMChartHost and OMChartHost are automation objects that serve as wrappers to AMChart. These objects perform additional functions like creating trees based on AppManager repository data (AMChartHost) and connecting to the MOM database (OMChartHost). The idea is to keep AMChart as strictly a charting engine. Any new solution that requires application-specific code should be implemented separately to contain AMChart.

This section covers the following topics:

- AMChart properties
- AMChart methods
- CHART_USER_SETTINGS XML Specification

AMChart properties

The following properties are available for the AMChart object.

Property	Description
ChartStyle	Sets or returns the style of a chart.
ChartTitle	Sets or returns the title of a chart.
ColorScheme	Sets or returns the name of the color scheme used by a chart.
DataLabelAngle	Sets or returns the angle at which data labels along the X-axis are rotated.

Property	Description
DecimalPrecision	Sets or returns the short integer value representing the precision of the datapoint value. The value corresponds to the number of digits after the decimal point.
DemoMode	Sets or returns the Boolean value to display demo data in the Chart Settings dialog box.
DrawBorder	Sets or returns the boolean value to control the drawing of a border around the chart.
DrawHLines	Sets or returns the boolean value to indicate whether the chart should draw horizontal grid lines.
DrawVLines	Sets or returns the boolean value to indicate whether the chart should draw vertical grid lines.
HTTPServer	Sets or returns the name of the Web server that is providing XML data to the chart control.
MaxDataPoints	Sets or returns the maximum number of data points that can be handled by the chart control. The number of points in each series is limited to the number of MaxDataPoints.
ViewMode	Sets or returns the mode of current view of the chart control.
XMLDataInfo	Returns the Chart_User_Settings XML, which describe a chart. This XML contains Title, Rotation, Zoom, Chart Style, etc.
YAxisAutoRange	Sets or returns the Boolean value that enables the chart to set the Y-axis maximum and minimum range automatically, based on the values from data points in the chart.
YAxisMaxRange	Sets or returns the maximum range for the Y-axis scale that a chart can plot. Any data points having higher values than YAxisMaxRange are clipped.
YAxisMinRange	Sets or returns the minimum range for the Y-axis scale that a chart can plot. Any data points having lower values than YAxisMinRange are clipped.

ChartStyle

Sets or returns the style of a chart. Possible styles are:

- -1 = Mixed (read-only, indicates all series are not the same style)
- 0 = Area

- 1 = Ribbon
- 2 = Line
- 3 = Bar
- 4 = Cylinder
- 5 = Area Stacked
- 6 = Pie

Example

VB

```
Property ChartStyle As Short
```

C/C++

```
short ChartStyle
```

Parameters

- Short/short - Style to set on a chart

Return values

- Short/short - Style that a chart is set to

Remarks

ChartTitle

Sets or returns the title of a chart.

Example

VB

```
Property ChartTitle As String
```

C/C++

```
BSTR ChartTitle
```

Parameters

- String/BSTR - Title to set on a chart

Return values

- String/BSTR - Returns the chart title

Remarks

ColorScheme

Sets or returns the name of the color scheme used by a chart.

Example

VB

```
Property ColorScheme As String
```

C/C++

```
BSTR ColorScheme
```

Parameters

- String/BSTR - Name of the color scheme to set on chart

Return values

- String/BSTR - Name of the color scheme used by a chart

Remarks

DataLabelAngle

Sets or returns the angle at which data labels along the X-axis are rotated. Possible values are 0-90. Data labels appear only in charts that use time-based data, and use SetDataLabel.

Example

VB

```
Property DataLabelAngle As Short
```

C/C++

```
short DataLabelAngle
```

Parameters

- Short/short - Angle in degrees which data labels are rotated

Return values

- Short/short - Angle in degrees which data labels are rotated

Remarks

DecimalPrecision

Sets or returns the short integer value representing the precision of the datapoint value. The value corresponds to the number of digits after the decimal point.

Example

VB

```
Property DecimalPrecision As Short
```

C/C++

```
short DecimalPrecision
```

Parameters

- Short/short - Number of digits after the decimal point

Return values

- Short/short - Number of digits after the decimal point

Remarks

DemoMode

Sets or returns the Boolean value to display demo data in the Chart Settings dialog box.

Example

VB

```
Property DemoMode As Boolean
```

C/C++

```
BOOL DemoMode
```

Parameters

- Boolean/BOOL - True to enable demo mode, FALSE otherwise

Return values

- Boolean/BOOL - True if demo mode is enabled, FALSE otherwise

Remarks

DrawBorder

Sets or returns the boolean value to control the drawing of a border around the chart.

Example

VB

```
Property DrawBorder As Boolean
```

C/C++

```
BOOL DrawBorder
```

Parameters

- Boolean/BOOL - True to add border to a chart, FALSE otherwise

Return values

- Boolean/BOOL - True if chart has border, FALSE otherwise

Remarks

DrawHLines

Sets or returns the boolean value to indicate whether the chart should draw horizontal grid lines.

Example

VB

```
Property DrawHLines AS Boolean
```

C/C++

```
BOOL DrawHLines
```

Parameters

- Boolean/BOOL - True to enable horizontal lines, FALSE otherwise

Return values

- Boolean/BOOL - True if horizontal lines are enabled, FALSE otherwise

Remarks

DrawVLines

Sets or returns the boolean value to indicate whether the chart should draw vertical grid lines.

Example

VB

```
Property DrawVLines AS Boolean
```

C/C++

```
BOOL DrawVLines
```

Parameters

- Boolean/BOOL - True to enable vertical lines, FALSE otherwise

Return values

- Boolean/BOOL - True if vertical lines are enabled, FALSE otherwise

Remarks

HTTPServer

Sets or returns the name of the Web server that is providing XML data to the chart control.

Example

VB

```
Property HTTPServer As String
```

C/C++

```
BSTR HTTPServer
```

Parameters

- String/BSTR - Name of the Web server that is providing XML data

Return values

- String/BSTR - Name of the Web server that is providing XML data

Remarks

MaxDataPoints

Sets or returns the maximum number of data points that can be handled by the chart control. The number of points in each series is limited to the number of MaxDataPoints.

Example

VB

```
Property MaxDataPoints As Long
```

C/C++

```
long MaxDataPoints
```

Parameters

- Long/long - Maximum number of points in a series (0 = unlimited points)

Return values

- Long/long - Maximum number of points in a series (0 = unlimited points)

Remarks

ViewMode

Sets or returns the mode of current view of the chart control. Possible values are:

- 0 = Perspective (approximately 35mm projection)
- 1 = Orthographic (parallel projection)

Example

```
VB
Property ViewMode As Short

C/C++
short ViewMode
```

Parameters

- Short/short - 0 to set perspective view, 1 otherwise

Return values

- Short/short - Returns the current view (0 if perspective, 1 otherwise)

Remarks

XMLDataInfo

Returns the Chart_User_Settings XML, which describe a chart. This XML contains Title, Rotation, Zoom, Chart Style, etc.

Example

VB

```
Property XMLDataInfo As String
```

C/C++

```
BSTR XMLDataInfo
```

Parameters

- String/BSTR - XML data that describes a chart

Return values

- String/BSTR - XML data that describes a chart

Remarks

YAxisAutoRange

Sets or returns the Boolean value that enables the chart to set the Y-axis maximum and minimum range automatically, based on the values from data points in the chart.

Example

VB

```
Property YAxisAutoRange As Boolean
```

C/C++

```
BOOL YaxisAutoRange
```

Parameters

- Boolean/BOOL - True to enable auto-range for a chart, FALSE otherwise

Return values

- Boolean/BOOL - True if chart control is set to auto-range, FALSE otherwise

Remarks

YAxisMaxRange

Sets or returns the maximum range for the Y-axis scale that a chart can plot. Any data points having higher values than YAxisMaxRange are clipped.

Example

VB

```
Property YAxisMaxRange As Double
```

C/C++

```
Double YAxisMaxRange
```

Parameters

- Double/double - Maximum range for the Y-axis

Return values

- Double/double - Maximum range for the Y-axis

Remarks

YAxisMinRange

Sets or returns the minimum range for the Y-axis scale that a chart can plot. Any data points having lower values than YAxisMinRange are clipped.

Example

VB

```
Property YAxisMinRange As Double
```

C/C++

```
Double YAxisMinRange
```

Parameters

- Double/double - Minimum range for the Y-axis

Return values

- Double/double - Minimum range for the Y-axis

Remarks

AMChart methods

The following methods are available for the AMLayout object.

Methods	Descriptions
AddChartHighThreshold	Adds a high threshold to the chart, with the specified value and color.
AddData	Adds data dValue to a series specified in nSeriesIndex.
AddSeriesHighThreshold	Adds a high threshold to the series, with the specified value and color.
AddSeriesLowThreshold (not supported at this time)	Adds a low threshold to the series, with the specified value and color.
BeginSelection	For AMChart.OCX use only (not AMLayout or AMChartLite). Sets the chart to selection mode.
EnableDataDetails	Enable display of the data point details.
DefaultRotation	Sets the chart engine to use preset rotation angles.
DeleteAllSeries	Deletes all series in the chart.
DeleteChartHighThresholds	Deletes all of the high thresholds from a chart.
DeleteSeries	Deletes a specified series from a chart.
EnableScrolling	Enables/disables scrolling in the chart pane.
EstablishSeries	Add a series to the chart.
EstablishSeriesADO	Add a series of time-based data to the chart.

Methods	Descriptions
EstablishSeriesADO_DSN	Add a series of time-based data to the chart.
EstablishSeriesADO_RS	Add a series of time-based data to the chart.
GetBottomYAxisLabel	Returns a string label at the bottom of the Y-axis in a chart.
GetChartAggregation	Returns a string representing the aggregation interval.
GetChartTransparency	Finds if all series in a chart are transparent.
GetDataAtIndex	Given an index of a series and datapoint, it fills in pdValue with the datapoint value.
GetRenderHeight	Returns the height of the render area.
GetRenderWidth	Returns the width of the render area.
GetScrollMax	Gets the maximum scrolling position.
GetScrollMin	Gets the minimum scrolling position.
GetScrollPos	Gets the position of scroll box.
GetSeriesCount	Gets the number of series in a chart.
GetSeriesScale	Gets the scale factor of a series in the chart.
GetSeriesStyle	Gets the style of a given series.
GetSeriesTransparency	Checks if a series in a chart has transparency enabled.
GetTopYAxisLabel	Gets the string label attached to the top of the Y-axis in a chart.
GetYAxisLabel	Gets the string label attached to the Y-axis in a chart.
InitXML	Sets the ChartUserSettings XML for the chart.
InsertData	Inserts new datapoint in the series and optionally rebuilds the series to reflect the new point.
PromptDeleteFromLegend	Displays a confirmation dialog prompt to delete the selected series in the legend pane.
RebuildScene	Rebuilds the chart based on current information, but does not display any changes. Sometimes used in conjunction with RenderScene.

Methods	Descriptions
RefreshData	Starts the background thread to refresh the series in a chart. Used with AMChart only, not AMChartLite or AMLayout.
RemoveAllDataLabels	Deletes the array that contains all the data labels for a chart. Data labels appear on the X-axis of the chart for non-time-based data (zero interval). See EstablishSeries for more information.
RemoveDataLabel	Removes a data label in an array that is indexed by the argument. Data labels appear on the X-axis of the chart for non-time-based data (zero interval). See EstablishSeries for more information.
RemoveSelection	For use with AMChart.OCX only (not AMLayout or AMChartLite). Takes the chart out of selection mode. (see the BeginSelection method).
RenderScene	Tells OpenGL to render the chart. Sometimes used in conjunction with UpdateChart prior to calling RenderScene.
RotateRealTime	Initiates real-time rotation of the chart (controlled by mouse). User must click to stop the rotation. Not used by AMLayout.DLL.
SaveImage	Save the chart as an image file. The only supported format is PNG (Portable Network Graphics).
SetBorderColor	Sets the chart window frame to the specified color.
SetBottomYAxisLabel	Sets the label at the bottom of the Y-axis.
SetChartStyleArea	Sets the chart style to Area.
SetChartStyleAreaStacked	Sets the chart style to Area Stacked.
SetChartStyleBar	Sets the chart style to Bar.
SetChartStylePie	Sets the chart style to Pie.
SetChartStylePoint	Sets the chart style to Pie.
SetChartStyleRibbon	Sets the chart style to Ribbon.
SetChartTransparency	Enables or disables the transparency of all series in the chart. When enabled, each series is drawn with a translucent quality.

Methods	Descriptions
SetDataAtDate	Given a DATE, the value of datapoint, and an index to a series, this method adds a data point to the series at the specified DATE (date and time).
SetDataAtTime	Given UTC time, the value of datapoint, and an index to a series, this method adds the datapoint to the series.
SetDataAtTimeEX	Given date, the value of datapoint, and an index to a series, this method creates a time stamp and calls SetDataAtTime to add the datapoint to the series.
SetDataLabel	Inserts a data label in the label array. Data labels appear along the X-axis when using a chart with non-time-based data (zero interval). See EstablishSeries for more information.
SetRotation	Sets the camera angle.
SetRenderSize	Sets the rendering size (in pixels) of the control.
SetScrollPos	Sets the scroll block to a new position.
SetSeriesScale	Adjusts the scale factor of a series.
SetSeriesStyle	Sets the style of a given series.
SetSeriesTransparency	Enables or disables the transparency of a series and rebuilds it. When enabled, the series is drawn with a translucent quality.
SetTopYAxisLabel	Sets the label attached to the top of the Y-axis.
SetYAxisLabel	Set label for the Y-axis.
ShowDataViewer	Displays the data viewer dialog.
ShowDockableLegend	Shows the legend bar when TRUE; hides it otherwise.
ShowDockableTimeframe	Shows the time frame bar when TRUE; hides it otherwise.
ShowProperties	Shows the chart's properties.
ShowStaticLegend	Draws a static legend below or to the right of the chart, depending on nStyle.
SortSeriesByLegend	Sorts all series in the chart from front to back (either ascending or descending) based upon the text of the legend.

Methods	Descriptions
SortSeriesByValue	Sorts all series in the chart from front to back (either ascending or descending) based upon the most recent value in each series.
UpdateChart	Update the entire chart (re-build all series) and render the scene.
ZoomIn	Zoom in to the chart.
ZoomNormal	Set the zoom to default.
ZoomOut	Zoom out from the chart.

AddChartHighThreshold

Adds a high threshold to the chart, with the specified value and color.

Example

VB

```
sub AddChartHighThreshold(bShowHiWatermarkLine As Long,
dblHiWatermarkValue As Double,nRed As Integer,nGreen As
Integer,nBlue As Integer)
```

C/C++

```
void AddChartHighThreshold(long bShowHiWatermarkLine, double
dblHiWatermarkValue,short nRed,short nGreen,short nBlue)
```

Parameters

- bShowHiWatermarkLine - Boolean value to toggle threshold
- dblHiWatermarkValue - Value of the threshold
- nRed - Red RGB value for threshold color
- nGreen - Green RGB value for threshold color
- nBlue - Blue RGB value for threshold color

Return values

- void

Remarks

AddData

Adds data dValue to a series specified in nSeriesIndex. Set bRebuildSeries TRUE to rebuild OpenGL display lists. AddData is used for data that is not time-based, i.e. interval of zero. See the EstablishSeries method.

Example

VB

```
sub AddData(nSeriesIndex As Short,dValue As
Double,bRebuildSeries As Boolean)
```

C/C++

```
void AddData(short nSeriesIndex, double dValue, BOOL
bRebuildSeries)
```

Parameters

- nSeriesIndex - Index of series
- dValue - Value of the datapoint
- bRebuildSeries - TRUE to re-build series, FALSE otherwise

Return values

- void

Remarks

AddSeriesHighThreshold

Adds a high threshold to the series, with the specified value and color.

Example

VB

```
sub AddSeriesHighThreshold(nSeries As Short,
bShowHiWatermarkLine As Long, dblHiWatermarkValue As Double,
nRed As Short, nGreen As Short, nBlue As Short)
```

C/C++

```
void AddSeriesHighThreshold(short nSeries, long
bShowHiWatermarkLine, double dblHiWatermarkValue, short nRed,
short nGreen, short nBlue)
```

Parameters

- nSeries - Series to which the high threshold will be added
- bShowHiWatermarkLine - Switch to show high threshold on and off
- dblHiWatermarkValue - Value of the high threshold
- nRed - Red RGB value for threshold color
- nGreen - Green RGB value for threshold color
- nBlue - Blue RGB value for threshold color

Return values

- void

Remarks

AddSeriesLowThreshold (not supported at this time)

Adds a low threshold to the series, with the specified value and color.

Example

VB

```
sub AddSeriesLowThreshold(nSeries As Short,
bShowHiWatermarkLine As Long, dblLoWatermarkValue As Double,
nRed As Short, nGreen As Short, nBlue As Short)
```

C/C++

```
void AddSeriesLowThreshold(short nSeries, long
bShowHiWatermarkLine, double dblLoWatermarkValue, short nRed,
short nGreen, short nBlue)
```

Parameters

- nSeries - Series to which the low threshold will be added

- bShowLoWatermarkLine - Switch to show low threshold on and off
- dblLoWatermarkValue - Value of the low threshold
- nRed - Red RGB value for threshold color
- nGreen - Green RGB value for threshold color
- nBlue - Blue RGB value for threshold color

Return values

- void

Remarks

BeginSelection

For AMChart.OCX use only (not AMLayout or AMChartLite). Sets the chart to selection mode.

Performs the same function as **Select data** on the shortcut menu. Use the mouse to click-and-drag to highlight a range of data. Release the mouse button to select the range. Range is used by the Export feature to export a specific range of data. To deselect the data, use RemoveSelection, or deselect **Select data** on the shortcut menu.

Example

VB

```
sub BeginSelection()
```

C/C++

```
void BeginSelection()
```

Parameters

Return values

- void

Remarks

EnableDataDetails

Enable display of the data point details.

Example

VB

```
sub EnableDataDetails(bEnable As BOOL)
```

C/C++

```
void EnableDataDetails(BOOL bEnable)
```

Parameters

- bEnable - Boolean value to toggle data detail

Return values

- void

Remarks

DefaultRotation

Sets the chart engine to use preset rotation angles.

Example

VB

```
sub DefaultRotation()
```

C/C++

```
void DefaultRotation()
```

Parameters

Return values

- void

Remarks

DeleteAllSeries

Deletes all series in the chart.

Example

VB

```
sub DeleteAllSeries()
```

C/C++

```
void DeleteAllSeries()
```

Parameters

Return values

- void

Remarks

DeleteChartHighThresholds

Deletes all of the high thresholds from a chart.

Example

VB

```
sub DeleteChartHighThresholds()
```

C/C++

```
void DeleteChartHighThresholds()
```

Parameters

Return values

- void

Remarks

DeleteSeries

Deletes a specified series from a chart.

Example

VB

```
sub DeleteSeries(nSeriesIndex As Short)
```

C/C++

```
void DeleteSeries(short nSeriesIndex)
```

Parameters

- nSeriesIndex - Series to delete

Return values

- void

Remarks

EnableScrolling

Enables/disables scrolling in the chart pane. Scrolling is useful for large amounts of data (hundreds or thousands of points). When scrolling is disabled, the data is all fit within the chart window (sometimes referred to as "Fit Data To Window").

Example

VB

```
sub EnableScrolling(bEnable As Long)
```

C/C++

```
void EnableScrolling(long bEnable)
```

Parameters

- bEnable - Boolean value to toggle scrolling

Return values

- void

Remarks

EstablishSeries

Add a series to the chart. The legend for the series is specified in strName. The expected time interval (in seconds) between each point should be specified in lInterval.

Two options are possible:

- Zero Interval: Your chart will not have a date/time associated with each point. You need to add your points to this series using the AddData method. Finally, after (or before) adding your series and data to the chart, you need to use SetDataLabel to specify your X-Axis labels. The chart will not draw any data until the data labels have been specified.
- Nonzero Interval: Your chart will have a data/time associated with each point. For example: Your application gathers a new data point every 5 minutes and records the time at which the point was taken. You would specify an lInterval of 300 seconds. Each series can have a different interval (as long as the interval is > 0). The chart will examine the intervals of all series and calculate a common interval so that many series with different intervals can be plotted correctly in the same chart. You would use SetDataAtTime or SetDataAtDate to add your points at a specified time.

Example

VB

```
sub EstablishSeries(strName As String, lInterval As Long)
```

C/C++

```
short EstablishSeries(BSTR strName,long lInterval)
```

Parameters

- strName - Name of the series to add
- lInterval - Running interval of the Knowledge Script

Return values

- Short/short - returns the index of the series in a chart

Remarks

EstablishSeriesADO

Add a series of time-based data to the chart. The legend for the series is specified in strName. A SQL statement is specified in strSQL. The DSN is specified in strDSN. The expected time interval (in seconds) between each point should be specified in lInterval. The minimum interval is 1 second.

For example, your application gathers a new data point every 5 minutes and records the time at which the point was taken. You would specify an lInterval of 300 seconds. Each series can have a different interval (as long as the interval is > 0). The chart will examine the intervals of all series and calculate a common interval so that many series with different intervals can be plotted correctly in the same chart.

There is no need to call SetDataAtTime or SetDataAtDate because the chart will bind to the ADO recordset and fetch the data points automatically as needed. The database table in which your data exists should have time and value columns. The time column should contain a date/time value in UTC (long) format.

Your SELECT statement must specify the time first, then the value, and the order of the data must be descending from newest to oldest. You must use the "ORDER BY x DESC" clause.

Example

VB

```
sub EstablishSeriesADO(strName As String, strSQL As String,
```

```
strServer As String, strDatabase As String, strUser As String,
strPassword As String, bTrusted As Long, lInterval As Long)
```

C/C++

```
short EstablishSeriesADO(BSTR strName,BSTR strSQL,BSTR
strServer,BSTR strDatabase,BSTR strUser,BSTR
strPassword,long bTrusted,long lInterval)
```

Parameters

- strName - Name of the data stream to add
- strSQL - Name of the record object retrieved from the AppManager repository with ADO
- strServer - Name of the SQL server
- strDatabase - Name of the AppManager repository
- strUser - Name of logon user
- strPassword - Password
- lTrusted - 1 for trusted connection, 0 otherwise.
- lInterval - Running interval of Knowledge Script

Return values

- Short/short - Index of a series in a chart

Remarks

Sample code

The following example connects to an AppManager repository on the computer "SCOTT", with a database "QDB", and fetches the points from a table called "Data". Only the points with a DataID of 1 are returned. In AppManager, all points are stored in the Data table, so we need to specify which series we are interested in. We do this by specifying a DataID with a specific value (of 1 in this case). Next, be sure the recordset is returned in reverse order (newest to oldest) by using the "ORDER BY x DESC" clause. Next, specify a DSN and a time interval (in seconds).

```
EstablishSeriesADO("SERVER001: PROCESSOR Utilization% - CPU#
```

```
0", "Select Time, Value from Data Where DataID=1 ORDER BY Time
DESC", "Provider=SQLOLEDB;Server=SCOTT;User
ID=sa;Pwd=;Database=QDB", 15)
```

EstablishSeriesADO_DSN

Add a series of time-based data to the chart. The legend for the series is specified in strName. A SQL statement is specified in strSQL. The DSN is specified in strDSN. The expected time interval (in seconds) between each point should be specified in lInterval. The minimum interval is 1 second.

For example, your application gathers a new data point every 5 minutes and records the time at which the point was taken. You would specify an lInterval of 300 seconds. Each series can have a different interval (as long as the interval is > 0). The chart will examine the intervals of all series and calculate a common interval so that many series with different intervals can be plotted correctly in the same chart.

There is no need to call SetDataAtTime or SetDataAtDate because the chart will bind to the ADO recordset and fetch the data points automatically as needed. The database table in which your data exists should have time and value columns. The time column should contain a date/time value in UTC (long) format.

Your SELECT statement must specify the time first, then the value, and the order of the data must be descending from newest to oldest. You must use the "ORDER BY x DESC" clause.

Example

VB

```
sub EstablishSeriesADO_DSN(strName As String, strSQL As
String, strDSN As String, lInterval As Long)
```

C/C++

```
short EstablishSeriesADO_DSN(BSTR strName,BSTR strSQL,BSTR
strDSN,long lInterval)
```

Parameters

- strName - Name of the series
- strSQL - Name of the record set
- strDSN - Name of DSN
- lInterval - Running interval of qml

Return values

- Short/short - Index of the series added to a chart

Remarks

Sample code

The following example connects to an AppManager repository on the computer "SCOTT", with a database "QDB", and fetches the points from a table called "Data". Only the points with a DataID of 1 are returned. In AppManager, all points are stored in the Data table, so we need to specify which series we are interested in. We do this by specifying a DataID with a specific value (of 1 in this case). Next, be sure the recordset is returned in reverse order (newest to oldest) by using the "ORDER BY x DESC" clause. Next, specify a DSN and a time interval (in seconds).

```
EstablishSeriesAD)_DSN("SERVER001: PROCESSOR Utilization% -
CPU# 0", "Select Time, Value from Data Where DataID=1 ORDER
BY Time DESC", "Provider=SQLOLEDB;Server=SCOTT;User
ID=sa;Pwd=;Database=QDB", 15)
```

EstablishSeriesADO_RS

Add a series of time-based data to the chart. The legend for the series is specified in strName. A SQL statement is specified in strSQL. The DSN is specified in strDSN. The expected time interval (in seconds) between each point should be specified in lInterval. The minimum interval is 1 second.

For example, your application gathers a new data point every 5 minutes and records the time at which the point was taken. You would

specify an lInterval of 300 seconds. Each series can have a different interval (as long as the interval is > 0). The chart will examine the intervals of all series and calculate a common interval so that many series with different intervals can be plotted correctly in the same chart.

There is no need to call SetDataAtTime or SetDataAtDate because the chart will bind to the ADO recordset and fetch the data points automatically as needed. The database table in which your data exists should have time and value columns. The time column should contain a date/time value in UTC (long) format.

Your SELECT statement must specify the time first, then the value, and the order of the data must be descending from newest to oldest. You must use the "ORDER BY x DESC" clause.

Example

VB

```
sub EstablishSeriesADO_RS(strName As String, lInterval As
Long, ADORecordset As IDispatch::ADORecordSet)
```

C/C++

```
short EstablishSeriesADO_RS(BSTR strName,long
lInterval,IDispatch ADORecordset)
```

Parameters

- strName - Name of the series to add to a chart
- lInterval - Running interval of qml
- ADORecordSet - ADO record set retrieved from the AppManager repository.

Return values

- Short/short- Index of the series in a chart

Remarks

Sample code

The following example adds a time-based series to the chart using the given ADO recordset. Be sure the recordset is returned in reverse order (newest to oldest) by using the "ORDER BY x DESC" clause in your SQL statement.

```
EstablishSeriesADO_RS("SERVER001: PROCESSOR Utilization% -
CPU# 0", 15, pRecordset);
```

GetBottomYAxisLabel

Returns a string label at the bottom of the Y-axis in a chart.

Example

VB

```
sub GetBottomYAxisLabel()
```

C/C++

```
BSTR GetBottomYAxisLabel()
```

Parameters

Return values

- String/BSTR - Returns the string label attached at the bottom of Y axis

Remarks

GetChartAggregation

Returns a string representing the aggregation interval. Possible values are:

- None
- 5 Minutes
- 10 Minutes

- 15 Minutes
- 30 Minutes
- 1 Hour
- 2 Hours
- 4 Hours
- 8 Hours
- 12 Hours
- 1 Day
- 1 Week

Example

VB

```
sub GetChartAggregation()
```

C/C++

```
BSTR GetChartAggregation()
```

Parameters

Return values

- String/BSTR - String specified in 'Aggregation' combo box

Remarks

GetChartTransparency

Finds if all series in a chart are transparent.

Example

VB

```
sub GetChartTransparency()
```

C/C++

```
BOOL GetChartTransparency()
```

Parameters

Return values

- Boolean/BOOL - TRUE if all series are transparent, FALSE otherwise

Remarks

GetDataAtIndex

Given an index of a series and datapoint, it fills in pdValue with the datapoint value.

Example

VB

```
sub GetDataAtIndex(nSeriesIndex As Short, lDataIndex As
Long,pointer type pdValue)
```

C/C++

```
BOOL GetDataAtIndex(short nSeriesIndex,long
lDataIndex,pointer type pdValue)
```

Parameters

- nSeriesIndex - Index of the series
- lDataIndex - Index of the datapoint
- pdValue - Variable to fill in the value of datapoint specified as lDataIndex

Return values

- Boolean/BOOL - TRUE if the call is successful, FALSE otherwise

Remarks

GetRenderHeight

Returns the height of the render area. Usually used in conjunction with the SaveImage method, primarily with AMLayout and AMChartLite, and rarely with AMChart.

Example

VB

```
sub GetRenderHeight()
```

C/C++

```
long GetRenderHeight()
```

Parameters

Return values

- Long/long - Height of AMChart control

Remarks

GetRenderWidth

Returns the width of the render area. Usually used in conjunction with the SaveImage method, primarily with AMLayout and AMChartLite, and rarely with AMChart.

Example

VB

```
sub GetRenderWidth()
```

C/C++

```
long GetRenderWidth()
```

Parameters

Return values

- Long/long - Width of AMChart control

Remarks

GetScrollMax

Gets the maximum scrolling position.

Example

VB

```
sub GetScrollMax()
```

C/C++

```
long GetScrollMax()
```

Parameters

Return values

- Long/long - Returns the maximum scrolling position

Remarks

GetScrollMin

Gets the minimum scrolling position.

Example

VB

```
sub GetScrollMin()
```

C/C++

```
long GetScrollMin()
```

Parameters

Return values

- Long/long - Returns the minimum scrolling position

Remarks

GetScrollPos

Gets the position of scroll box.

Example

VB

```
sub GetScrollPos()
```

C/C++

```
long GetScrollPos()
```

Parameters

Return values

- Long/long - Returns the position of the scroll box

Remarks

GetSeriesCount

Gets the number of series in a chart.

Example

VB

```
sub GetSeriesCount()
```

C/C++

```
short GetSeriesCount()
```

Parameters

Return values

- Short/short- number of series in a chart

Remarks

GetSeriesScale

Gets the scale factor of a series in the chart. Possible values are:

- 100000.0
- 10000.0
- 1000.0
- 100.0
- 10.0
- 1.0
- 0.1
- 0.01
- 0.001
- 0.0001
- 0.00001
- 0.000001
- 0.0000001

Example

VB

```
sub GetSeriesScale(nSeries As Int)
```

C/C++

```
double GetSeriesScale(int nSeries)
```

Parameters

- nSeries - Index of the series in a chart

Return values

- Double/double - Returns a scale value if a series has any. Returns 1.0 otherwise

Remarks

GetSeriesStyle

Gets the style of a given series. Possible styles are:

- 0 = Area
- 1 = Ribbon
- 2 = Line
- 3 = Bar
- 4 = Cylinder
- 5 = Area Stacked
- 6 = Pie

Example

VB

```
sub GetSeriesStyle(nSeriesIndex As Integer)
```

C/C++

```
short GetSeriesStyle(short nSeriesIndex)
```

Parameters

- nSeriesIndex - Index of series of which get the style

Return values

- Short/short-Style of a series

Remarks

GetSeriesTransparency

Checks if a series in a chart has transparency enabled.

Example

VB

```
sub GetSeriesTransparency(nSeries As Short)
```

C/C++

```
BOOL GetSeriesTransparency(short nSeries)
```

Parameters

- Short/short - Index of the series in a chart

Return values

- Boolean/BOOL - Returns TRUE if transparency is turned on, FALSE otherwise

Remarks

GetTopYAxisLabel

Gets the string label attached to the top of the Y-axis in a chart.

Example

VB

```
sub GetTopYAxisLabel()
```

C/C++

```
BSTR GetTopYAxisLabel()
```

Parameters

Return values

- String/BSTR-Returns the string label attached to the top of the Y-axis

Remarks

GetYAxisLabel

Gets the string label attached to the Y-axis in a chart.

Example

VB

```
sub GetYAxisLabel()
```

C/C++

```
BSTR GetYAxisLabel()
```

Parameters

Return values

- String/BSTR-Returns the string label attached to the Y-axis

Remarks

InitXML

Sets the ChartUserSettings XML for the chart. The ChartUserSettings XML is used to describe a chart, including the title, rotation, series, etc.

Example

VB

```
sub InitXML(strXML As String)
```

C/C++

```
BOOL InitXML(BSTR strXML)
```

Parameters

- strXML - XML data to describe a chart

Return values

- Boolean/BOO L -Returns TRUE if initialization is successful, FALSE otherwise

Remarks

InsertData

Inserts new datapoint in the series and optionally rebuilds the series to reflect the new point.

When adding many points, an optimization is to not rebuild the series after adding each point by passing FALSE to the bRebuildSeries parameter.

Example

VB

```
sub InsertData(nSeriesIndex As Short, lDataIndex As Long,
double dValue, bRebuildSeries long)
```

C/C++

```
void InsertData(short nSeriesIndex, long lDataIndex, double
dValue,long bRebuildSeries)
```

Parameters

- nSeriesIndex - Index of series
- lDataIndex - Index of series where datapoint should be inserted
- dValue - Value of the datapoint
- bRebuildSeries - TRUE to rebuild series, FALSE otherwise

Return values

- void

Remarks

PromptDeleteFromLegend

Displays a confirmation dialog prompt to delete the selected series in the legend pane.

Example

VB

```
sub PromptDeleteFromLegend()
```

C/C++

```
void PromptDeleteFromLegend()
```

Parameters

Return values

Remarks

RebuildScene

Rebuilds the chart based on current information, but does not display any changes. Sometimes used in conjunction with RenderScene.

Example

VB

```
sub RebuildScene()
```

C/C++

```
void RebuildScene()
```

Parameters

Return values

- void

Remarks

RefreshData

Starts the background thread to refresh the series in a chart. Used with AMChart only, not AMChartLite or AMLayout.

Example

VB

```
sub RefreshData()
```

C/C++

```
void RefreshData()
```

Parameters

Return values

- void

Remarks

RemoveAllDataLabels

Deletes the array that contains all the data labels for a chart. Data labels appear on the X-axis of the chart for non-time-based data (zero interval). See EstablishSeries for more information.

Example

VB

```
sub RemoveAllDataLabels()
```

C/C++

```
void RemoveAllDataLabels()
```

Parameters

Return values

- void

Remarks

RemoveDataLabel

Removes a data label in an array that is indexed by the argument. Data labels appear on the X-axis of the chart for non-time-based data (zero interval). See EstablishSeries for more information.

Example

VB

```
sub RemoveDataLabel(lDataIndex As Index)
```

C/C++

```
void RemoveDataLabel(long lDataIndex)
```

Parameters

- lDataIndex - Index of the data label to remove

Return values

- void

Remarks

RemoveSelection

For use with AMChart.OCX only (not AMLayout or AMChartLite). Takes the chart out of selection mode. (see the BeginSelection method).

Example

VB

```
sub RemoveSelection()
```

C/C++

```
void RemoveSelection()
```

Parameters

Return values

- void

Remarks

RenderScene

Tells OpenGL to render the chart. Sometimes used in conjunction with UpdateChart prior to calling RenderScene.

Example

VB

```
sub RenderScene()
```

C/C++

```
void RenderScene()
```

Parameters

Return values

- void

Remarks

RotateRealTime

Initiates real-time rotation of the chart (controlled by mouse). User must click to stop the rotation. Not used by AMLayout.DLL.

Example

VB

```
sub RotateRealTime()
```

C/C++

```
void RotateRealTime()
```

Parameters

Return values

- void

Remarks

SaveImage

Save the chart as an image file. The only supported format is PNG (Portable Network Graphics).

Example

VB

```
sub SaveImage(strFilePathName As String, bAddLegend As Long,
bAddBorder As Long)
```

C/C++

```
BOOL SaveImage(BSTR strFilePathName, long bAddLegend, long
bAddBorder)
```

Parameters

- strFilePathName - Full path to where the image file should be save
- bAddLegend - 1 to add legend in the image file, 0 otherwise
- bAddBorder - 1 to add border around the image, 0 otherwise

Return values

- Boolean/BOO L - TRUE if image saved successfully, FALSE otherwise

Remarks

SetBorderColor

Sets the chart window frame to the specified color. The default color is gray. This feature is typically used to help the chart blend more naturally with colored backgrounds, such as Web pages. Used with AMChart only, not AMChartLite or AMLayout.

Example

VB

```
sub SetBorderColor(nRed As Short, nGreen As Short, nBlue AS
Short)
```

C/C++

```
void SetBorderColor(short nRed,short nGreen,short nBlue)
```

Parameters

- nRed - Red RGB value for border color
- nGreen - GreenRGB value for border color
- nBlue - Blue RGB value for border color

Return values

- void

Remarks

SetBottomYAxisLabel

Sets the label at the bottom of the Y-axis.

Example

VB

```
sub SetBottomYAxisLabel(strLabel As String)
```

C/C++

```
void SetBottomYAxisLabel(BSTR strLabel)
```

Parameters

- strLabe l- Name to set as the label at the bottom of the Y-axis

Return values

- void

Remarks

SetChartStyleArea

Sets the chart style to Area.

Example

VB

```
sub SetChartStyleArea()
```

C/C++

```
void SetChartStyleArea()
```

Parameters

Return values

- void

Remarks

SetChartStyleAreaStacked

Sets the chart style to Area Stacked.

Example

VB

```
sub SetChartStyleAreaStacked()
```

C/C++

```
void SetChartStyleAreaStacked()
```

Parameters

Return values

- void

Remarks

SetChartStyleBar

Sets the chart style to Bar.

Example

VB

```
sub SetChartStyleBar()
```

C/C++

```
void SetChartStyleBar()
```

Parameters

Return values

- void

Remarks

SetChartStylePie

Sets the chart style to Pie.

Example

VB

```
sub SetChartStylePie()
```

C/C++

```
void SetChartStylePie()
```

Parameters

Return values

- void

Remarks

SetChartStylePoint

Sets the chart style to Pie.

Example

VB

```
sub SetChartStylePoint()
```

C/C++

```
void SetChartStylePoint()
```

Parameters

Return values

- void

Remarks

SetChartStyleRibbon

Sets the chart style to Ribbon.

Example

VB

```
sub SetChartStyleRibbon()
```

C/C++

```
void SetChartStyleRibbon()
```

Parameters

Return values

- void

Remarks

SetChartTransparency

Enables or disables the transparency of all series in the chart. When enabled, each series is drawn with a translucent quality.

Example

VB

```
sub SetChartTransparency(bTransparent As Long)
```

C/C++

```
void SetChartTransparency(long bTransparent)
```

Parameters

- bTransparent - TRUE to enable transparency, FALSE otherwise

Return values

- void

Remarks

SetDataAtDate

Given a DATE, the value of datapoint, and an index to a series, this method adds a data point to the series at the specified DATE (date and time).

Example

VB

```
sub SetDataAtDATE(nSeriesIndex As Short, dblValue As Double,
dtTimeStamp As Date, bRebuildSeries As Boolean)
```

C/C++

```
BOOL SetDataAtDATE(short nSeriesIndex, double dblValue, DATE
dtTimeStamp, BOOL bRebuildSeries)
```

Parameters

- nSeriesIndex - Index of series
- dblValue - Value of the datapoint
- dtTimeStamp - Time of datapoint
- bRebuildSeries - TRUE to rebuild series in the chart, FALSE otherwise

Return values

- Boolean/BOOL - TRUE if datapoint is added successfully, FALSE otherwise

Remarks

SetDataAtTime

Given UTC time, the value of datapoint, and an index to a series, this method adds the datapoint to the series.

Example

VB

```
Boolean SetDataAtTime(nSeriesIndex As Short, dblValue As
Double, lTimeStamp As Long, bRebuildSeries As Long)
```

C/C++

```
BOOL SetDataAtTime(short nSeriesIndex, double dblValue, long
lTimeStamp, long bRebuildSeries)
```

Parameters

- nSeriesIndex - Index of a series
- dblValue - Value of a datapoint
- lTimeStamp - Timestamp of datapoint
- bRebuildSeries - TRUE to rebuild series, FALSE otherwise

Return values

- Boolean/BOOL - TRUE if datapoint is added successfully, FALSE otherwise

Remarks

SetDataAtTimeEX

Given date, the value of datapoint, and an index to a series, this method creates a time stamp and calls 'SetDataAtTime' to add the datapoint to the series.

Example

VB

```
sub SetDataAtTimeEx(nSeriesIndex As Short, dblValue As
Double, lYear As Long, nMonth As Short, nDay As Short, nHour
As Short, nMinute AS Short, nSecond AS Short, bRebuildSeries
As Long)
```

C/C++

```
BOOL SetDataAtTimeEx(short nSeriesIndex, double dblValue,
long lYear, short nMonth, short nDay, short nHour, short
nMinute, short nSecond, long bRebuildSeries)
```

Parameters

- nSeriesIndex - Index of series
- dblValue - Value of the datapoint
- lYear - Year
- nMonth - Month
- nDay - Date
- nHour - Hour
- nMinute - Minute
- nSecond - Second
- bRebuildSeries - TRUE to rebuild series in the chart, FALSE otherwise

Return values

- Boolean/BOOL - TRUE if datapoint is added successfully, FALSE otherwise

Remarks

SetDataLabel

Inserts a data label in the label array. Data labels appear along the X-axis when using a chart with non-time-based data (zero interval). See EstablishSeries for more information.

Example

VB

```
sub SetDataLabel(lDataIndex As Long, strLabel As String)
```

C/C++

```
BOOL SetDataLabel(long lDataIndex, BSTR strLabel)
```

Parameters

- lDataIndex - Index of the label array where string is to be inserted
- strLabel - String to insert in the data label array

Return values

- Boolean/BOOL - TRUE if string inserted into the array successfully, FALSE otherwise

Remarks

SetRotation

Sets the camera angle.

Example

VB

```
sub SetRotation(nXDegrees As Short, nYDegrees As Short)
```

C/C++

```
void SetRotation(short nXDegrees,short nYDegrees)
```

Parameters

- nXDegrees - Degree of rotation for X axis
- nYDegrees - Degree of rotation for Y axis

Return values

- void

Remarks

SetRenderSize

Sets the rendering size (in pixels) of the control. Output is generated in the given dimensions. Usually used in conjunction with the SaveImage method, primarily with AMLayout and AMChartLite, and rarely with AMChart.

Example

VB

```
sub SetRenderSize(lXSize As Long,lYSize As Long)
```

C/C++

```
void SetRenderSize(long lXSize,long lYSize)
```

Parameters

- lXSize - Size for the width
- lYSize - Size for the height

Return values

- void

Remarks

SetScrollPos

Sets the scroll block to a new position.

Example

VB

```
sub SetScrollPos(lNewPos As Long)
```

C/C++

```
void SetScrollPos(long lNewPos)
```

Parameters

- lNewPos - New position for the scroll block

Return values

- void

Remarks

SetSeriesScale

Adjusts the scale factor of a series. Possible values are:

- 100000.0
- 10000.0

- 1000.0
- 100.0
- 10.0
- 1.0
- 0.1
- 0.01
- 0.001
- 0.0001
- 0.00001
- 0.000001
- 0.0000001

Example

VB

```
sub SetSeriesScale(nSeries As Int, dScale As Double)
```

C/C++

```
void SetSeriesScale(int nSeries,double dScale)
```

Parameters

- nSeries - Index of series to adjust the scaling
- dScale - Scale factor(Range:100000.0~0.0000001)

Return values

- void

Remarks

SetSeriesStyle

Sets the style of a given series. Possible styles are:

- 0 = Area
- 1 = Ribbon

- 2 = Line
- 3 = Bar
- 4 = Cylinder
- 5 = Area Stacked
- 6 = Pie

Example

VB

```
sub SetSeriesStyle(nSeriesIndex As Short, nNewValue As Short)
```

C/C++

```
void SetSeriesStyle(short nSeriesIndex,short nNewValue)
```

Parameters

- nSeriesIndex - Index of the series
- nNewValue - Range from -1 to 6

  Use:

  TYPE_MIXED=mixed style

  TYPE_AREA=area style

  TYPE_RIBBON=ribbon style

  TYPE_LINE=line style

  TYPE_BAR=bar style

  TYPE_CYLINDER=cylinder style

  TYPE_AREA_STACKED=area stacked style

  TYPE_PIE=pie style

Return values

- void

Remarks

SetSeriesTransparency

Enables or disables the transparency of a series and rebuilds it. When enabled, the series is drawn with a translucent quality.

Example

VB

```
sub SetSeriesTransparency(short nSeries,long bTransparent)
```

C/C++

```
void SetSeriesTransparency(short nSeries,long bTransparent)
```

Parameters

- nSeries - Index of series
- bTransparent - TRUE to turn on transparency, FALSE otherwise

Return values

- void

Remarks

SetTopYAxisLabel

Sets the label attached to the top of the Y-axis.

Example

VB

```
sub SetTopYAxisLabel(strLabel As String)
```

C/C++

```
void SetTopYAxisLabel(BSTR strLabel)
```

Parameters

- strLabel - Name of the label to attach

Return values

- void

Remarks

SetYAxisLabel

Set label for the Y-axis.

Example

VB

```
sub SetYAxisLabel(strLabel As String)
```

C/C++

```
void SetYAxisLabel(BSTR strLabel)
```

Parameters

- strLabel - Label to attach to the Y-axis

Return values

- void

Remarks

ShowDataViewer

Displays the data viewer dialog.

Example

VB

```
sub ShowDataViewer()
```

C/C++

```
void ShowDataViewer()
```

Parameters

Return values

- void

Remarks

ShowDockableLegend

Shows the legend bar when TRUE; hides it otherwise.

Example

VB

```
sub ShowDockableAggregation(bShow As Boolean)
```

C/C++

```
void ShowDockableAggregation(BOOL bShow)
```

Parameters

- bShow - TRUE to show the legend, FALSE otherwise

Return values

- void

Remarks

ShowDockableTimeframe

Shows the time frame bar when TRUE; hides it otherwise.

Example

VB

```
sub ShowDockableTimeframe(bShow As Boolean)
```

C/C++

```
void ShowDockableTimeframe(BOOL bShow)
```

Parameters

- bShow - TRUE to show the time frame bar, FALSE otherwise

Return values

- void

Remarks

ShowProperties

Shows the chart's properties.

Example

VB

```
sub ShowProperties()
```

C/C++

```
void ShowProperties()
```

Parameters

Return values

- void

Remarks

ShowStaticLegend

Draws a static legend below or to the right of the chart, depending on nStyle.

Example

VB

```
sub ShowStaticLegend(nStyle As Short)
```

C/C++

```
void ShowStaticLegend(short nStyle)
```

Parameters

- nStyle - Value to set the legend. Possible values for nStyle:

  0 = Do not draw static legend

  1 = Draw static legend below the chart

  2 = Draw static legend at the right side of the chart

Return values

- void

Remarks

SortSeriesByLegend

Sorts all series in the chart from front to back (either ascending or descending) based upon the text of the legend.

Example

VB

```
sub SortSeriesByLegend(bEnable As Long, bAscending As Long)
```

C/C++

```
void SortSeriesByLegend(long bEnable,long bAscending)
```

Parameters

- bEnable - 1 to enable sorting, 0 otherwise.
- bAscending - Positive value for ascending sort (A to Z), 0 or negative value for descending sort (Z to A)

Return values

- void

Remarks

This method is not available for AppManager 4.0.

SortSeriesByValue

Sorts all series in the chart from front to back (either ascending or descending) based upon the most recent value in each series.

Example

VB

```
sub SortSeriesByValue(bEnable As Long,bAscending As Long)
```

C/C++

```
void SortSeriesByValue(long bEnable,long bAscending)
```

Parameters

- bEnable - 1 to enable sorting, 0 otherwise
- bAscending - Positive value to sort by ascending value, 0 or negative value to sort by descending value

Return values

- void

Remarks

UpdateChart

Update the entire chart (re-build all series) and render the scene.

Example

VB

```
sub UpdateChart()
```

C/C++

```
void UpdateChart()
```

Parameters

Return values

- void

Remarks

ZoomIn

Zoom in to the chart.

Example

VB

```
sub ZoomIn()
```

C/C++

```
void ZoomIn()
```

Parameters

Return values

- void

Remarks

ZoomNormal

Set the zoom to default.

Example

VB

```
sub ZoomNormal()
```

C/C++

```
void ZoomNormal()
```

Parameters

Return values

- void

Remarks

ZoomOut

Zoom out from the chart.

Example

VB

```
sub ZoomOut()
```

C/C++

```
void ZoomOut()
```

Parameters

Return values

- void

Remarks

CHART_USER_SETTINGS XML Specification

XML used to describe a chart, including the title, rotation, series, etc. See the method InitXML.

XML

```
<CHART_USER_SETTINGS Version="1">
<UniqueID></UniqueID>
<GraphTitle>Scott's Chart</GraphTitle>
<ColorScheme>Sunshine</ColorScheme>
<VerticalMaxVal>10.000000</VerticalMaxVal>
<AutoVerticalMax>1</AutoVerticalMax>
<PointDensity>2.400000</PointDensity>
<OrthoMode>1</OrthoMode>
<CameraXAngle>10</CameraXAngle>
<CameraYAngle>353</CameraYAngle>
<DefaultChartStyle>3</DefaultChartStyle>
<YAxisLabel></YAxisLabel>
<TopYAxisLabel></TopYAxisLabel>
<BottomYAxisLabel></BottomYAxisLabel>
```

```
<Aggregation>12 Hours</Aggregation>
<FitDataToWindow>1</FitDataToWindow>
<ShowDockTimeframeBar>0</ShowDockTimeframeBar>
<ShowDockLegendBar>1</ShowDockLegendBar>
<ShowDockAggregateBar>1</ShowDockAggregateBar>
<StaticLegendFlags>0</StaticLegendFlags>
<SERIES>
<Server></Server>
<User></User>
<Password></Password>
<Database></Database>
<Legend>% Privileged Time(_Total)</Legend>
<MCComputer>SCOTTJ</MCComputer>
<HTTPSourceID></HTTPSourceID>
<Interval>-1</Interval>
<Style>3</Style>
<Hide>0</Hide>
<TrustedConnection>0</TrustedConnection>
<Transparent>0</Transparent>
<HighTH>0</HighTH>
<LowTH>0</LowTH>
<ShowHighTHLine>0</ShowHighTHLine>
<ShowLowTHLine>0</ShowLowTHLine>
<HighTHValue>0</HighTHValue>
<LowTHValue>0</LowTHValue>
<HighTHColor>255</HighTHColor>
<LowTHColor>32512</LowTHColor>
<Scale>1</Scale>
</SERIES>
<SERIES>
<Server></Server>
<User></User>
<Password></Password>
<Database></Database>
<Legend>% User Time(_Total)</Legend>
<MCComputer>SCOTTJ</MCComputer>
<HTTPSourceID></HTTPSourceID>
<Interval>-1</Interval>
<Style>0</Style>
<Hide>0</Hide>
<TrustedConnection>0</TrustedConnection>
<Transparent>0</Transparent>
<HighTH>0</HighTH>
<LowTH>0</LowTH>
<ShowHighTHLine>0</ShowHighTHLine>
```

```
<ShowLowTHLine>0</ShowLowTHLine>
<HighTHValue>0</HighTHValue>
<LowTHValue>0</LowTHValue>
<HighTHColor>255</HighTHColor>
<LowTHColor>32512</LowTHColor>
<Scale>1</Scale>
</SERIES>
<SERIES>
<Server></Server>
<User></User>
<Password></Password>
<Database></Database>
<Legend>% System Time(_Total)</Legend>
<MCComputer>SCOTTJ</MCComputer>
<HTTPSourceID></HTTPSourceID>
<Interval>-1</Interval>
<Style>3</Style>
<Hide>0</Hide>
<TrustedConnection>0</TrustedConnection>
<Transparent>0</Transparent>
<HighTH>0</HighTH>
<LowTH>0</LowTH>
<ShowHighTHLine>0</ShowHighTHLine>
<ShowLowTHLine>0</ShowLowTHLine>
<HighTHValue>0</HighTHValue>
<LowTHValue>0</LowTHValue>
<HighTHColor>255</HighTHColor>
<LowTHColor>32512</LowTHColor>
<Scale>1</Scale>
</SERIES>
</CHART_USER_SETTINGS>

Possible values for CHART_USER_SETTINGS:
<GraphTitle>: Any String Value
<ColorScheme>: Any String Value
<VerticalMaxVal>: Any Double Value
<AutoVerticalMax>: Integer value 0 or 1
<PointDensity>Any Double Value greater than 0
<OrthoMode>: Integer value 0 or 1
<CameraXAngle>: Integer value 0 - 359
<CameraYAngle>: Integer value 0 - 359
<ChartStyle>: Possible numeric values 0 – 6:
0 = AREA
1 = RIBBON
2 = LINE
```

```
3 = BAR
4 = CYLINDER
5 = AREA STACKED
6 = PIE

<YAxisLABEL>: Any String Value
<TopYAxisLABEL>: Any String Value
<BottomYAxisLABEL>: Any String Value
<Aggregation>: Possible String Values are:
"None"
"5 Minutes"
"10 Minutes"
"15 Minutes"
"30 Minutes"
"1 Hour"
"2 Hours"
"4 Hours"
"8 Hours"
"12 Hours"
"1 Day"
"1 Week"

<FitDataToWindow>: Integer value 0 or 1
0 = Do not force all data to fit in the window.  This allows
large amounts of data to scroll horizontally
1 = Yes, squash data to fit in window - best used in
conjunction with an <Aggregation> value
<ShowDockTimeframeBar>: Integer value 0 or 1
<ShowDockLegendBar>: Integer value 0 or 1
0 = do not show dockable legend bar.  You may want to consider
using <StaticLegendFlags> below to draw a static legend.
1 = show dockable legend bar.  You may want to set
<StaticLegendFlags> below to 0.
<ShowDockAggregateBar>: Integer value 0 or 1
<StaticLegendFlags>: Integer value 0 - 2:
0 = no static legend - you may want to set <ShowDockLegendBar>
above to 1
1 = draw static legend on bottom of chart - you may want to
set <ShowDockLegendBar> above to 0
2 = draw static legend at right of chart - you may want to set
<ShowDockLegendBar> above to 0

 <THRESHOLD>
<IsHighTH>: Only value allowable: 1
<ShowTHLine>: Integer value 0 or 1
```

```
0 = do not show threshold Line
1 = draw threshold line

<THVALUE>: Any Double value representing the value of the
threshold
<THColor>:Long value, must be a MS COLORREF Value
</THRESHOLD>
```

Index

A

AddChartHighThreshold 21
AddData 22
AddSeriesHighThrehsold 22
AddSeriesLowThrehsold 23
AMChart 5
  methods 17
    AddChartHighThreshold 21
    AddData 22
    AddSeriesHighThreshold 22
    AddSeriesLowThreshold 23
    BeginSelection 24
    DefaultRotation 25
    DeleteAllSeries 26
    DeleteChartHighThresholds 26
    DeleteSeries 27
    EnableDataDetails 25
    EnableScrolling 27
    EstablishSeries 28
    EstablishSeriesADO 29
    EstablishSeriesADO_DSN 31
    EstablishSeriesADO_RS 32
    GetBottomYAxisLabel 34
    GetChartAggregation 34
    GetChartTransparency 35
    GetDataAtIndex 36
    GetRenderHeight 37
    GetRenderWidth 37
    GetScrollMax 38
    GetScrollMin 38
    GetScrollPos 39
    GetSeriesCount 39
    GetSeriesScale 40
    GetSeriesStyle 41
    GetSeriesTransparency 41
    GetTopYAxisLabel 42
    GetYAxisLabel 42
    InitXML 43
    InsertData 43
    PromptDeleteFromLegend 44
    RebuildScene 45
    RefreshData 45
    RemoveAllDataLabels 46
    RemoveDataLabel 46
    RemoveSelection 47
    RenderScene 47
    RotateRealTime 48
    SaveImage 48
    SetBorderColor 49
    SetBottomYAxisLabel 50
    SetChartStyleArea 50
    SetChartStyleAreaStacked 51
    SetChartStyleBar 51
    SetChartStylePie 52
    SetChartStylePoint 52
    SetChartStyleRibbon 53
    SetChartTransparency 53
    SetDataAtDate 54
    SetDataAtTime 54
    SetDataAtTimeEX 55
    SetDataLabel 56

SetRenderSize 57
SetRotation 57
SetScrollPos 58
SetSeriesScale 58
SetSeriesStyle 59
SetSeriesTransparency 61
SetTopYAxisLabel 61
SetYAxisLabel 62
ShowDataViewer 62
ShowDockableLegend 63
ShowDockableTimeframe 63
ShowProerties 64
ShowStaticLegend 64
SortSeriesByLegend 65
SortSeriesByValue 66
UpdateChart 66
ZoomIn 67
ZoomNormal 67
ZoomOut 68
properties 6
ChartStyle 7
ChartTitle 8
ColorScheme 9
DataLabelAngle 9
DecimalPrecision 10
DemoMode 10
DrawBorder 11
DrawHLines 11
DrawVLines 12
HTTPServer 13
MaxDataPoints 13
ViewMode 14
XMLDataInfo 15
YAxisAutoRange 15
YAxisMaxRange 16
YAxisMinRange 16

B

BeginSelection 24

C

ChartStyle 7
ChartTitle 8
ColorScheme 9

D

DataLabelAngle 9
DecimalPrecision 10
DefaultRotation 25
DeleteAllSeries 26
DeleteChartHighThresholds 26
DeleteSeries 27
DemoMode 10
DrawBorder 11
DrawHLines 11
DrawVLines 12

E

EnableDataDetails 25
EnableScrolling 27
EstablishSeries 28
EstablishSeriesADO 29
EstablishSeriesADO_DSN 31
EstablishSeriesADO_RS 32

G

GetBottomYAxisLabel 34
GetChartAggregation 34
GetChartTransparency 35
GetDataAtIndex 36
GetRenderHeight 37
GetRenderWidth 37
GetScrollMax 38
GetScrollMin 38

GetScrollPos 39
GetSeriesCount 39
GetSeriesScale 40
GetSeriesStyle 41
GetSeriesTransparency 41
GetTopYAxisLabel 42
GetYAxisLabel 42

H
HTTPServer 13

I
InitXML 43
InsertData 43

M
MaxDataPoints 13

O
objects
    AMChart 5

P
PromptDeleteFromLegend 44

R
RebuildScene 45
RefreshData 45
RemoveAllDataLabels 46
RemoveDataLabel 46
RemoveSelection 47
RenderScene 47
RotateRealTime 48

S
SaveImage 48
SetBorderColor 49
SetBottomYAxisLabel 50
SetChartStyleArea 50
SetChartStyleAreaStacked 51
SetChartStyleBar 51
SetChartStylePie 52
SetChartStylePoint 52
SetChartStyleRibbon 53
SetChartTransparency 53
SetDataAtDate 54
SetDataAtTime 54
SetDataAtTimeEX 55
SetDataLabel 56
SetRenderSize 57
SetRotation 57
SetScrollPos 58
SetSeriesScale 58
SetSeriesStyle 59
SetSeriesTransparency 61
SetTopYAxisLabel 61
SetYAxisLabel 62
ShowDataViewer 62
ShowDockableLegend 63
ShowDockableTimeframe 63
ShowProperties 64
ShowStaticLegend 64
SortSeriesByLegend 65
SortSeriesByValue 66

U
UpdateChart 66

V
ViewMode 14

X
XMLDataInfo 15

Y
YAxisAutoRange 15
YAxisMaxRange 16
YAxisMinRange 16

Z
ZoomIn 67
ZoomNormal 67
ZoomOut 68