Rasterex RxViewX Component Specification

October 8, 2003

Rasterex Software a.s

http://www.rasterex.com

Copyright © 2001-2003 Rasterex Software a.s. All rights reserved.

Introduction

RxViewX is a powerful ActiveX component implementing the new generation of the Rasterex View API, making it available to any environment supporting this technology. RxViewX has been designed from the bottom up in order to offer a consistent object model to read and visualize a wide variety of document types, and extensive functionality to manage the huge set of filter components that is an integral part of the RxViewX package.

Included in the RxViewX installation are sample executables and projects using the following Rapid Application Development (RAD) tools and environments:

Visual Basic 6.0
MFC
Delphi
HTML

You may modify and use the executables and sample code in your own product or environment.

RxViewX contains a redlining component. “Redlining”, “commenting” and “collaboration” are often referred to as “Markup”. Please see the MFC sample.

Distribution of the RxViewX ActiveX component, RimEngine, Rasterex filters and associated DLLs and libraries requires that you sign a licensing agreement with RasterexSoftware a.s. Details of licensing are available from Rasterex.

Contact Rasterex via the Rasterex website http://www.rasterex.com/
Email to sales@rasterex.com

RxViewX, RimEngine, RxFilter, RimServer and RxHighX are registered trademarks belonging to Rasterex Software a.s

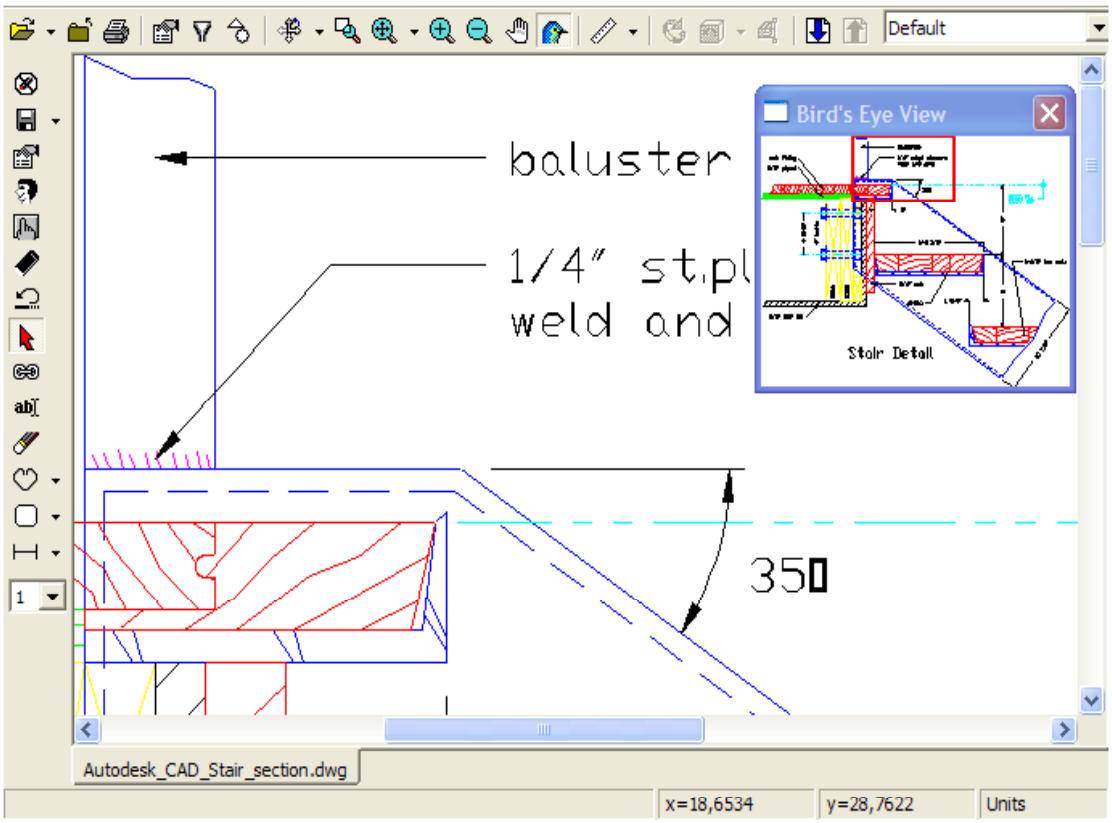
Disclaimer

Rasterex Software a.s makes no representations or warranties regarding the contents of this document, and specifically disclaims any implied warranties of merchantability of fitness for any particular purpose. Furthermore, Rasterex Software a.s reserves the right to revise this document and to make changes from time to time in the contents without an obligation by Rasterex Software a.s to notify any person of such revision changes.

Table of Contents

1 RxViewX Component

As illustrated below, the RxViewX component can be used as a complete viewer in itself. It can be implemented as a single object in a client program or HTLM page, and act as an independent complete viewer. The client program can hide any or all of the toolbars, individual buttons, the status bar and the file tabs. In which case the client program must handle the user interface itself, using the numerous methods and properties provided by RxViewX.



1.1 The coordinate systems

RxViewX operates with three coordinate systems as follows:
The **Original** coordinate system created by the original application that was used to create the document.
The **World** coordinate system used internally in RxViewX and RimEngine.
The **Screen** coordinate system used by Windows.

Hence, when a document is loaded, it is transformed and scaled from the Original coordinate system to the World coordinate System.
When a document is displayed, it is transformed and scaled from the World coordinate System to the Screen coordinate system.

2 RxViewX Methods

The following methods are available for the client application.
The order in which the methods are described within this document is the order in which they are normally used in an application.
NOTE Redline methods are only available when the RxRedlines module has been installed.

2.1 ActivateFile

Use this method to activate a file if more than one file is open in RxViewX.

Syntax	ActivateFile (BSTR psFileName).	
Parameters	**psFileName**	Path and name of file to be activated and displayed.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.2 ActivateLayerDialog

Use this method to open the layer dialog for files that support a layer structure.

Syntax	ActivateLayerDialog()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.3 AttachCaption

Use this method to attach a caption to an existing redline element. Use the PickElement event to get a valid Redline element Handle.

Syntax	AttachCaption(Long Handle, BSTR sCaption)	
Parameters	**Handle**	Handle to the redline element you want to attach a caption to.
	sCaption	Caption string you want attached to the redline element.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.4 CloseFile

Use this method to close files open in RxViewX.

Syntax	CloseFile (BOOL bAll, BSTR psFileName).	
Parameters	**bAll**	Set TRUE if all files should be closed
	psFileName	Path and name of file to be closed. If no file name is given, current active file will be closed
Returns	BOOL	Returns TRUE if the operation was successfully.

2.5 FileInfoDialog

Use this method to access the RxViewX File Information dialog.

Syntax	FileInfoDialog ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.6 FileOpen

Opens an existing file specified as an URL or UNC path name in the psFileName parameter.
Note: RxViewX files consists of one ore more layouts, and each layout has one or more pages, views or sheets.

This operation will load the first or default page/view/sheet from the first or default layout.
The opened file is set as the active file.

Syntax	FileOpen (BSTR psFileName, BSTR psDocName).	
Parameters	**psFileName**	Path and name of file to open.
	psDocName	name of documents as shown on tabs etc.
Returns	BOOL	Returns TRUE if the file was opened successfully.

2.7 FileOpenDialog

Use this method to open a file using the RxViewX internal file open dialog
Note: RxViewX files consists of one ore more layouts, and each layout has one or more pages, views or sheets.

This operation will load the first or default page/view/sheet from the first or default layout.
The opened file is set as the active file.

Syntax	FileOpen ().	
Parameters	**none**	
Returns	BOOL	Returns TRUE if a file was opened successfully.

2.8 FileVectorControlDialog

Use this method to access the RxViewX Vector Control dialog.

Syntax	FileVectorControlDialog ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.9 FilterInfoDialog

Use this method to access the filter information dialog in RxViewX.

Syntax	FilterInfoDialog ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.10 GetFileInfo

Use this method to get information about the currently active file.

Syntax	GetFileInfo (long* plFileType, BSTR* psFilterName, double* pdFileSize)	
Parameters	**plFileType**	Returns type of file. 0 = Text 1 = HTML 2 = Raster 3 = Vector_2D 4 = Vector_3D 5 = Workspace (hybrid) 6 = Spreadsheet
	psFilterName	Returns the name of the filter used to open the file
	pdFileSize	Returns the size of the file, in bytes

Returns	BOOL	Returns TRUE if the operation was successfully.

2.11 GetLayoutInfo

NOTE: For some documents, the number of pages is unknown until the whole document is loaded from file. In which case, the returned number of pages, in the GetLayoutInfo method above, will be set to –1. If this is the case, pages will automatically be loaded in the background in RxViewX until all pages are loaded and the number of pages is known

Syntax	GetLayoutInfo(long* plNoOfLayouts, long* plSelectedLayout, long* plNoOfPages, long* plSelectedPage)	
Parameters	**plNoOfLayouts**	Returns the number of layouts in the file
	plSelectedLayout	Returns index number of selected layout
	plNoOfPages	Returns number of pages in selected layout
	plSelectedPage	Returns index number of selected page
Returns	BOOL	Returns TRUE if the operation was successfully.

2.12 GetLayoutName

Use this method to get the layout name for a Layout given the Layout number.

Syntax	GetLayoutName(long lLayoutNo, BSTR* psLayoutName)	
Parameters	**lLayoutNo**	Layout index number. (from 0 to no. of Layouts – 1)
	psLayoutName	Returns the name of the layout
Returns	BOOL	Returns TRUE if the operation was successfully.

2.13 GetLoadedFileName

Use this method to retrieve the path and names of files currently loaded in RxViewX.
See also the **NumLoadedFiles** property.

Syntax	GetLoadedFileName (short nFileIndex, BSTR* psFileName)	
Parameters	**nFileIndex**	Index to required file
	psFileName	Returns the required filename
Returns	BOOL	Returns TRUE if the operation was successfully.

2.14 GetOriginalPageInfo

Use this method to get information about the currently active page.

Syntax	GetOriginalPageInfo (long* plOriginalUPI, double* pdOriginalScale, double* pdOriginalTransX, double* pdOriginalTransY)	
Parameters	**plOriginalUPI**	Returns the number of Units Per Inch of the page
	pdOriginalScale	Returns the scaling factor used to convert the contents from its original coordinates the RxViewX internal coordinate system
	pdOriginalTransX	Returns the X transmission factor used to convert the contents from its original coordinates the RxViewX internal coordinate system
	pdOriginalTransY	Returns the Y transmission factor used to convert the contents from its original coordinates the RxViewX internal coordinate system
Returns	BOOL	Returns TRUE if the operation was successfully.

2.15 GetPageName

Use this method to get the a Page name given the page number.

Syntax	GetPageName(long lPageNo, BSTR* psPageName)	
Parameters	**lPageNo**	The page no. index (from 0 to no. of Pages – 1)
	psPageName	Returns the name of the page, view or sheet
Returns	BOOL	Returns TRUE if the operation was successfully.

2.16 GetPageCompression

Use this method to get information about the compression used for the active page.

Syntax	GetPageInfo(BSTR* psCompression)	
Parameters	**psCompression**	Returns compression type of page.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.17 GetPageSize

Use this method to get the width and height for the active page.

Syntax	GetPageSize (long* plWidth, long * plHeight)	
Parameters	**plWidth**	Returns the page width in World coordinates.
	plHeight	Returns the page height in World coordinates.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.18 GetRecentFileName

RxViewX will hold the names of the last loaded files in registry. Use this method to retrieve these filenames.
See also the **NumRecentFiles** property.

Syntax	GetRecentFileName (short nFileIndex, BSTR* psFileName)	
Parameters	**nFileIndex**	Index to wanted file in recent file list
	psFileName	Returns the wanted filename
Returns	BOOL	Returns TRUE if the operation was successfully.

2.19 OverlayDialog

Use this method to access the RxViewX file Overlay and Compare dialog.

Syntax	OverlayDialog ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.20 PreferenceDialog

Use this method to access the RxViewX Preferences dialog.

Syntax	PreferenceDialog ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.21 Print

Use this method to print active file using the RxViewX Print dialog.

Syntax	Print ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.22 RedlinePreferenceDialog

Use this method to access the RxViewX Redline Preference dialog.

Syntax	RedlinePreferenceDialog ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.23 RedlineControlDialog

Use this method to access the RxViewX Redline Control dialog.

Syntax	RedlineControlDialog ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.24 RedlineSave

Use this method to save current user redlines to file.

Syntax	RedlineSave ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.25 RedlineRefresh

Use this method to refresh (i.e. save and reload) the current redlines.

Syntax	RedlineUndo ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.26 RedlineUndo

Use this method to undo the last peformed redline action.

Syntax	RedlineUndo ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.27 RedlineDefineLayer

Use this method to define Redline Layers in RxViewX Redline.
See also the property **RedlineNoOfLayers**

Syntax	RedlineDefineLayer (long nIndex, BSTR sName, long bSwitchable)	
Parameters	**nIndex**	0-based index to the layer
	sName	Name of the indexed layer
	bSwitchable	TRUE if current user can turn indexed layer on and off
Returns	BOOL	Returns TRUE if the operation was successfully.

2.28 RedlineGetLayerState

Use this method to get the redline layer states of current active file
See also the property **RedlineNoOfLayers**

Syntax	RedlineGetLayerState (long nIndex, long* pbState)	
Parameters	**nIndex**	0-based index to the layer
	pbState	Returns TRUE or FALSE indicating layer ON or OFF
Returns	BOOL	Returns TRUE if the operation was successfully.

2.29 RedlineSetLayerState

Use this method to set the redline layer states of current active file
See also the property **RedlineNoOfLayers**

Syntax	RedlineSetLayerState (long nIndex, long bState)	
Parameters	**nIndex**	0-based index to the layer
	bState	TRUE sets the layer ON, FALSE sets the layer OFF
Returns	BOOL	Returns TRUE if the operation was successfully.

2.30 RedlineGetUserInfo

Use this method to get information of redline users of current active file
See also the property **RedlineNoOfUsers**

Syntax	RedlineGetUserInfo (long nIndex, BSTR* psName, BSTR* psDate, BSTR* psFile)	
Parameters	**nIndex**	0-based index to the layer
	psName	Returns the name of indexed user
	psDate	Returns the date when this users redlines was created
	psFile	Returns the file name from which this users redlines was loaded
Returns	BOOL	Returns TRUE if the operation was successfully.

2.31 RedlineSetUserSwitch

Use this method to determine which redline users the current user can turn on and off
See also the property **RedlineNoOfUsers**

Syntax	RedlineSetUserSwitch (long nIndex, long bSwitchEnable)	
Parameters	**nIndex**	0-based index to the layer
	bSwitchEnable	TRUE if current user can turn indexed user's redlines on and off
Returns	BOOL	Returns TRUE if the operation was successfully.

2.32 RedlineGetUserState

Use this method to get redline user state information of current active file
See also the property **RedlineNoOfUsers**

Syntax	RedlineGetUserState (long nIndex, long* pbEnable)	
Parameters	**nIndex**	0-based index to the layer
	pbEnable	Returns TRUE if this user's redlines are shown
Returns	BOOL	Returns TRUE if the operation was successfully.

2.33 RedlineSetUserState

Use this method to set redline user state information of current active file
See also the property **RedlineNoOfUsers**

Syntax	RedlineGetUserState (long nIndex, long bEnable)	
Parameters	**nIndex**	0-based index to the layer
	bEnable	TRUE turns this user's redlines ON, FALSE turns this user's redlines OFF
Returns	BOOL	Returns TRUE if the operation was successfully.

2.34 RedlineDelUser

Use this method to delete all redlines made by specified user

Syntax	RedlineDelUse (BSTR szSignature)	
Parameters	**szSignature**	The signature of the user to be deleted
Returns	BOOL	Returns TRUE if the operation was successfully.

2.35 RedLineDeleteElement

Use this method to delete a redline element. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedLineDeleteElement(Long Handle)	
Parameters	**Handle**	Handle to the redline element you want delete.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.36 RedLineGetColor

Use this method to get the color of an existing redline element. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedLineGetColor(Long Handle, Long Color)	
Parameters	**Handle**	Handle to the redline element.
	Color	Color of the selected redline element.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.37 RedlineGetDimensions

Use this method to get the dimension of an existing redline element in World coordinates. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedlineGetDimensions (Long Handle, Long lenght, Long area)	
Parameters	**Handle**	Handle to the redline element.
	lenght	Length of the redline element in World coordinates.
	area	Area of the redline element in World coordinates.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.38 RedLineGetElementFont

Use this method to get the font used by an existing text redline element. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedLineGetElementFont(Long Handle, BSTR Name, Long Height, Long Weight, Long Italic, Long Underline, Long Charset, Long Pitch)	
Parameters	**Handle**	Handle to the redline element.
	Name	Name of the font.
	Height	Height of the font.
	Weight	See windows API LOGFONT structure for details.
	Italic	1 if the font is Italic 0 if not..
	Underline	1 if the font is underlined 0 if not.
	Charset	See windows API LOGFONT structure for details.
	Pitch	See windows API LOGFONT structure for details.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.39 RedlineGetElementRect

Use this method to get the rectangle of an existing redline element in World coordinates. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedlineGetElementRect(Long Handle, Long X1, Long Y1, Long X2, Long, Y2)	
Parameters	**Handle**	Handle to the redline element.
	X1	Upper Left X coordinate in World coordinates.
	Y1	Upper Left Y coordinate in World coordinates.
	X2	Bottom Right X coordinate in World coordinates.
	Y2	Bottom Right X coordinate in World coordinates.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.40 RedLineGetFont

Use this method to get the font used by as the default font when creating new text redline elements.

Syntax	RedLineGetFont(BSTR Name, Long Height, Long Weight, Long Italic, Long Underline, Long Charset, Long Pitch)	
Parameters	**Name**	Name of the font.
	Height	Height of the font.
	Weight	See windows API LOGFONT structure for details.
	Italic	1 if the font is Italic 0 if not..
	Underline	1 if the font is underlined 0 if not.
	Charset	See windows API LOGFONT structure for details.
	Pitch	See windows API LOGFONT structure for details.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.41 RedlineGetInfo

Use this method to get information about an existing redline element. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedlineGetInfo(Long Handle, Long type, Long style, Long marker, Long Color, Long layer,Double Width, Long Rotation)	
Parameters	**Handle**	Handle to the redline element.
	type	Same as IMainType in SetMouseActiontype list.
	style	Same as ISubtype in SetMouseActiontype list.
	marker	1 if marker color is used 0 if not.
	Color	Color of redline element.
	layer	The markup layer that the redline element is on.
	Width	Width of the redline element in World coordinates.
	Rotation	Rotation angle of the redline element.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.42 RedlineGetInfo2

Use this method to get information about an existing redline element. Same as RedlineGetInfo but the width parameter is a **Long** parameter value. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedlineGetInfo2(Long Handle, Long type, Long style, Long alt, Long marker, Long Color, Long layer,Long Width, Long Rotation)	
Parameters	**Handle**	Handle to the redline element.
	type	Same as IMainType in SetMouseActiontype list.
	style	Same as ISubtype in SetMouseActiontype list.
	alt	Same as IAlternativeAlternative in SetMouseActiontype list.
	marker	1 if marker color is used 0 if not.
	Color	Color of redline element.
	layer	The markup layer that the redline element is on.
	Width	Width of the redline element in World coordinates.
	Rotation	Rotation angle of the redline element.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.43 RedLinePushElement

Use this method to change the drawing order of redline elements. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedlineGetInfo(Long Handle, Long Back)	
Parameters	**Handle**	Handle to the redline element.
	Back	If 1 the element is moved to the bottom if 0 to the top.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.44 RedLineSetColor

Use this method to set the color of an existing redline element. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedLineSetColor(Long Handle, Long Color)	
Parameters	**Handle**	Handle to the redline element.
	Color	Color of the selected redline element.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.45 RedLineSetElementFont

Use this method to set the font used by an existing text redline element. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedLineSetElementFont(Long Handle, BSTR Name, Long Height, Long Weight, Long Italic, Long Underline, Long Charset, Long Pitch)	
Parameters	**Handle**	Handle to the redline element.
	Name	Name of the font.
	Height	Height of the font.
	Weight	See windows API LOGFONT structure for details.
	Italic	1 if the font is Italic 0 if not..
	Underline	1 if the font is underlined 0 if not.
	Charset	See windows API LOGFONT structure for details.
	Pitch	See windows API LOGFONT structure for details.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.46 RedLineSetFont

Use this method to set the font used as the default font when creating new text redline elements.

Syntax	RedLineSetFont(BSTR Name, Long Height, Long Weight, Long Italic, Long Underline, Long Charset, Long Pitch)	
Parameters	**Name**	Name of the font.
	Height	Height of the font.
	Weight	See windows API LOGFONT structure for details.
	Italic	1 if the font is Italic 0 if not..
	Underline	1 if the font is underlined 0 if not.
	Charset	See windows API LOGFONT structure for details.
	Pitch	See windows API LOGFONT structure for details.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.47 RedlineSetInfo

Use this method to set information for an existing redline element. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedlineSetInfo(Long Handle, Long type, Long style, Long marker, Long Color, Long layer,Double Width, Long Rotation)	
Parameters	**Handle**	Handle to the redline element.
	type	Same as lMainType in SetMouseActiontype list.
	style	Same as lSubtype in SetMouseActiontype list.
	marker	1 if marker color is used 0 if not.
	Color	Color of redline element.
	layer	The markup layer that the redline element is on.
	Width	Width of the redline element in World coordinates.
	Rotation	Rotation angle of the redline element.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.48 RedlineSetInfo2

Use this method to set information about an existing redline element. Same as RedlineSetInfo but the width parameter is a **Long** parameter value. Use the PickElement event to get a valid Redline element Handle.

Syntax	RedlineSetInfo2(Long Handle, Long type, Long style, Long alt, Long marker, Long Color, Long layer,Long Width, Long Rotation)	
Parameters	**Handle**	Handle to the redline element.
	type	Same as lMainType in SetMouseActiontype list.
	style	Same as lSubtype in SetMouseActiontype list.
	alt	Same as lAlternativeAlternative in SetMouseActiontype list.
	marker	1 if marker color is used 0 if not.
	Color	Color of redline element.
	layer	The markup layer that the redline element is on.
	Width	Width of the redline element in World coordinates.
	Rotation	Rotation angle of the redline element.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.49 RedlineShowEditDialog

Use this method to activate the redline edit dialog for the currently selected redline element.

Syntax	RedlineShowEditDialog()	
Parameters	**None**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.50 Refresh

Use this method to refresh the redline information without saving and reloading the redline file. Use this method instead of the RedlineRefresh method.

Syntax	Refresh()	
Parameters	**None**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.51 SaveToRasterDialog

Use this method to access the RxViewX “save active file to raster” dialog.

Syntax	SaveToRasterDialog ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.52 SaveToVectorDialog

Use this method to access the RxViewX “save active file to vector” dialog.

Syntax	SaveToVectorDialog ()	
Parameters	**none**	
Returns	BOOL	Returns TRUE if the operation was successfully.

2.53 Screen2World

Use this method to convert screen coordinates to RxRimX world coordinates

Syntax	Screen2World (long* pX, long* pY)	
Parameters	**pX**	Holds the X-screen coordinate on input and the world coordinate on output.
	pY	Holds the Y-screen coordinate on input and the world coordinate on output.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.54 SelectLayout

Use this method to Set the active Layout using the layout number.

If the layout is selected for the first time, the first or default page/view will automatically be loaded.

Syntax	SelectLayout(long lLayoutNo)	
Parameters	**lLayoutNo**	The layout index no. to select for viewing (from 0 to no. of Layouts – 1)
Returns	BOOL	Returns TRUE if the operation was successfully.

2.55 SelectPage

Use this method to set the active Page using the page number.

Syntax	SelectPage(long lPageNo)	
Parameters	**lPageNo**	The page no. to select for viewing. (from 0 to no. of Pages – 1)
Returns	BOOL	Returns TRUE if the operation was successfully.

2.56 SetMouseActionType

Use this method for any type of interaction with RxViewX that involves use of the mouse.

Syntax	SetMouseActionType (long lMainType, long lSubType, long lAlternative)	
Parameters	**lMainType**	Determines the main action type. See table below
	lSubType	Determines the sub action type. See below
	lAlternative	Determines draw alternatives. See below
Returns	BOOL	Returns TRUE if the operation was successfully.

The following table describes main mouse action types and its different subtypes and alternatives

lMainType	lSubType	lAlternative
0 = No action to be performed	none	none
1 = Zoom window	none	none
2 = Pan	none	none
3 = Calibrate drawing	none	none
4 = Measure drawing	none	none
5 = Redline action: Edit redlines	none	none
6 = Redline action: Push link and note buttons	none	none
7 = Redline action: Rubout	none	none
8 = Redline action: Make link button	none	none
9 = Redline action: Draw text or note	0 = Normal text 1 = Arrow Pointed text 2 = Note wrapped text	Defines text enclosure: 0 = Unframed text 1 = outlined rectangle 2 = outlined rounded corner rectangle 3 = outlined talk bubble 4 = edged rectangle 5 = edged rounded corner rectangle 6 = edged talk bubble
10 = Redline action: Draw lines	0 = Freehand pen 1 = Polylines 2 = Polycurves 3 = Measurement area	0 = Open 1 = Closed 2 = Closed and filled 3 = Closed and filled with background color
11 = Redline action: Draw arrow	0 = Normal Arrow 1 = Dimension line	For normal arrow: 0 = Single headed outlined 1 = Single headed filled 2 = Double headed outlined 3 = Double headed filled For dimension line: 0 = Lines 90 degree on measurement line 1 = Filled circles 2 = Double headed outlined arrow plus 90 degree lines 3 = Double headed filled arrow plus 90 degree lines
12 = Redline action: Draw rectangles / ovals etc	0 = Rectangle 1 = RoundedRectangle 2 = Oval 3 = Bubble	0 = Outlined 1 = Outlined 2 = filled 3 = outlined width redline color and filled with background color
13 = copy to clipboard : Selected rectangle will be copied to clipboard using the given format.	0 = Metafile 1 = DIB	Not used

2.57 ShowToolButton

Use this method to show or hide individual buttons in the main toolbar

Syntax	ShowToolButton (enumToolButtons nButtonIndex, BOOL bShow)	
Parameters	**nButtonIndex**	0-based indexed button number
	bEnable	TRUEto show the button. FALSE to hide it.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.58 ShowMarkupButton

Use this method to show or hide individual buttons in the markup/red toolbar

Syntax	ShowMarkupButton (enumMarkupButtons nButtonIndex, BOOL bShow)	
Parameters	nButtonIndex	0-based indexed button number
	bEnable	TRUEto show the button. FALSE to hide it.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.59 Zoom

Use this method to zoom into a section of the drawing using the predefined zoom types.

Syntax	Zoom(long lZoomType)	
Parameters	**lZoomType**	The type of zoom to be performed 0 = UnZoomed (1 : 1) 1 = Zoom to extent 2 = Zoom to width 3 = Zoom to height 4 = Zoom in 5 = Zoom out
Returns	BOOL	Returns TRUE if the operation was successfully.

2.60 ZoomToCoordinates

Use this method to zoom into particular section of the drawing specified by a rectangle given in world coordinates.

Syntax	ZoomToCoordinates (long X1, long Y1, long X2, long Y2)	
Parameters	**X1**	The leftposition in world coordinates
	Y1	The top position in world coordinates
	X2	The right position in world coordinates
	Y2	The bottom position in world coordinates
Returns	BOOL	Returns TRUE if the operation was successfully.

2.61 World2Original

Use this method to convert RxRimX world coordinates to the original application coordinates. Note that the conversion is based on the filter measurement settings for the filter used to load active file.

Syntax	World2Original (double* pdX, double* pdY, short* nPresition, BSTR* sUnit)	
Parameters	**pdX**	Holds the X- world coordinate on input and the screen coordinate on output.
	pdY	Holds the X- world coordinate on input and the screen coordinate on output.
	nPresition	Returns the decimal precision required.
	sUnit	Returns the coordinate system unit. I.e. mm or inch.
Returns	BOOL	Returns TRUE if the operation was successfully.

2.62 World2Screen

Use this method to convert RxRimX world coordinates to screen coordinates.

Syntax	World2Screen (long* pX, long* pY)	
Parameters	**pX**	Holds the X- world coordinate on input and the screen coordinate on output.
	pY	Holds the X- world coordinate on input and the screen coordinate on output.
Returns	BOOL	Returns TRUE if the operation was successfully.

3 RxViewX Properties

The following RxViewX properties are exposed for the client application.

3.1 BackgroundColor (BOOL) (write only)
Use this property to set the background color of all file types.

3.2 Calibration (Double)
This property can hold a Calibration value to calibrate the current drawing. Use in combination with the MeasureUnit property. The property is only applied if measure settings are set to Custom for the filter.

3.3 HaveRedlineModule (BOOL)
Use this property to determine whether or not a Redline module has been installed.

3.4 KeepVectorColors (BOOL) (write only)
The *KeepVectorColor*s property is used to draw vectors with their correct colors. Normally, vectors with colors close to the background color are displayed inverted.

3.5 LayoutOnTabs (BOOL)
Use this property to control if AutoCAD layouts are displayed in separate tabs.

3.6 LocalHelp (BOOL)
Set this property to TRUE in order to have the Rasterex RxViewX shown when the user clicks a help button.
Set it FALSE to have the help event described below to occur.

3.7 MeasureUnit (BSTR)
This property can hold a Measure unit to use with the current calibration setting for the current drawing. Use in combination with the Calibration property. The property is only applied if measure settings are set to Custom for the filter.

3.8 Monochrome (BOOL)

The *Monochrome* property is used to set the display color to mono when displaying vector file types. If this option is not set then files will be shown in color. This property applies only to vectors and document files.

3.9 NumLoadedFiles (short)
This property holds the number of files currently loaded in RxViewX.

3.10 NumRecentFiles (short)
RxViewX will hold the names of the last loaded files in registry. This property holds the number of recent files currently held.

3.11 PrintMonochrome (BOOL)

The *PrintMonochrome* property is used to set the print color vector and document files using monochrome only. The PrintMonochromeColor property must contain the color that should be used when printing.

3.12 PrintMonochromeColor (BOOL)

If the *PrintMonochrome* property is enabled, this property will hold the color that should be used when printing.

3.13 RasterScaleType (enumRasterScale)

Normal raster scaling is the fastest method of displaying during zooming. Some black areas may disappear during zooming.
Scale to gray raster scaling is the slowest method of displaying during zooming. It gives the best display results. If you have problems reading, for example TIF files, use this option to improve clarity.
Preserve black raster scaling is the next fastest display method during zooming. Some black areas may disappear during zooming.

3.14 RedlineChanged (BOOL)

Use this property to check whether or not the user has made changes to redlines.

3.15 RedlineColor (long)

Use this property to set the redline color for current user.

3.16 RedlineFileExtension (BSTR)

Use this property to set the redline file extension for multi redline files.
Default is XCM.

3.17 RedlineFileType (short)

Use this property to set the redline file type to:
0 = multifile (XCM).
1 = 000 type file
2 = X00 type file

3.18 RedlineFolderPath (BSTR)

Use this property to set sub or absolute path to where to save redlines.
See RedlineFolderType above.

3.19 RedlineFolderType (short)

Use this property to set the redline folder type which determines where to save redlines.
0 = same folder as viewed file.
1 = subfolder to viewed file. RedlineFolderPath holds the relative sub-path
3 = separate folder. RedlineFolderPath holds the absolute path

3.20 RedlineLayer (long)

Use this property to set the redline layer for current user.

3.21 RedlineNoOfLayers (long)
Use this property to set the number of redline layer to be used.

3.22 RedlineNoOfUsers (long)
Use this property to get the number of redline users of active file.

3.23 RedlineOk2Undo (BOOL)
Use this property to check whether or not the user has made changes to redlines that can be undone.

3.24 RedLineShowPopup (BOOL)
If this property is disabled (set to false), the redline edit popup menu will not be shown when user clicks the right mouse button.

3.25 RedlineWidth (long)
Use this property to set the width of redlines to be drawn.

3.26 Rotation (long)
Use this property to set the Rotation on the viewed file in 90-degree increments.

3.27 ShowBirdsEyeView (BOOL)
Use this property to turn Birdseye view on/off.

3.28 ShowContolToolbar (BOOL)
This property controls whether or not the control toolbar is shown or not.
Set it FALSE to hide the control toolbar.

3.29 ShowFileTabs (BOOL)
This property controls whether or not the view window file tabs is shown or not.
Set it FALSE to hide the file tabs bar.

3.30 ShowHelpButtons (BOOL)
Set this property to TRUE in order to have the HELP button shown in RxViewX dialogs.
When this property is set true and the user clicks a help button, either the help event described below will occur, or RxViewX help will be shown, depending on the state of the LocalHelp property below.

3.31 ShowRedlineToolbar (BOOL)
This property controls whether or not the Redline toolbar is shown or not.
Set it FALSE to hide the Redline toolbar.
Note that the Redline Toolbar and redlines as a whole is only available when the RxRedlines component is available.

3.32 ShowStatusbar (BOOL)
This property controls whether or not the Status bar is shown or not.
Set it FALSE to hide the Status bar.

3.33 Signature (BSTR)
Use this property to set the current user name/signature.

3.34 Source (BSTR)
This property can be used to set the path to file. The file will then be opened in RxViewX. Can be used as a substitute for the FileOpen method.

3.35 Version (BSTR)
This property returns RxViewX version and build number as a string.

3.36 XmlUrl (BSTR)
This property can hold a URL to a XML file that contains redlining configuration information. See Appendix B for more information about the XML file content.

4 RxViewX Events

4.1 Stock Events

RxViewX supports the following standard stock events:

MouseMove(short *nButton***, short** *nShiftState***, long** *x***, long** *y* **)**
MouseDown(**short** Button, **short** Shift, **long** x, **long** y);
MouseUp(**short** Button, **short** Shift, **long** x, **long** y);
DblClick()
KeyDown(**short*** KeyCode, **short** Shift);
KeyPress(**short*** KeyAscii);
KeyUp(**short*** KeyCode, **short** Shift);

4.2 Custom Events

RxViewX supports the following custom events:

4.2.1 ViewError

This event occurs whenever RxViewX has detected an error.

Syntax	**ViewError**(short nErrorNumber)	
Parameters	nErrorNumber	NONE 0 // no error NOMEM 1 // Out of memory MAXFILES 2 // Max files loaded NOFILE 3 // Unknown file format FWRITE 4 // File write error FREAD 5 // File read error FOPEN 6 // File open error FCREATE 7 // File create error RED_NOMEM 11 // Redline: Out of memory RED_MAXFILES 12 // Redline: Max files loaded RED_NOFILE 13 // Redline: Unknown file format RED_FWRITE 14 // Redline: File write error RED_FREAD 15 // Redline: File read error RED_FOPEN 16 // Redline: File open error RED_FCREATE 17 // Redline: File create error

4.2.2 HelpWanted

This event occurs whenever the user clicks a help button in one of RxViewX dialogs.
See also the property **ShowHelpButtons**

Syntax	**HelpWanted**(short nDialogID)	
Parameters	nDialogID	Index to the dialog for which help is wanted as follows: 1 = Preference dialog 2 = Overlay dialog 3 = Filter dialog 4 = Save to raster dialog 5 = Print dialog 6 = vector control dialog 7 = Redline Edit dialog 8 = Redline Text dialog 9 = Redline User and Layer control dialog 10 = Redline Preference dialog

4.2.3 StatusInformation

This event occurs whenever RxViewX wants to have status information displayed in the bottom status bar.

Syntax	**StatusInformation**(BSTR sStatusText);	
Parameters	sStatusText	Holds the status text

4.2.4 ViewComplete

This event occurs whenever RxViewX completes a view update.

Syntax	**ViewComplete**(short nStatus)	
Parameters	nStatus	Reserved for future use

4.2.5 Progress

Use this event to show progress when loading and saving files

Syntax	**Progress**(short nPercent, long *pbAbort)	
Parameters	nPercent	Holds percent loaded/saved
	pbAbort	Return TRUE in this parameter if the loading/saving should be aborted.

4.2.6 XRef

Use this event to control loading of external referenced files in vector files.
Either change the file name or return one of the actions shown.

Syntax	XRef (BSTR* psName, long* plAction)	
Parameters	psName	Holds path and name of Xref file
	plAction	Return one of the following: RX_EVENT_NO_ACTION = 0, RX_EVENT_USE_CURRENT_PATH = 3, RX_EVENT_USE_NEW_PATH = 4 Default is RX_EVENT_USE_CURRENT_PATH

4.2.7 FontSupport

Use this event to control loading of fonts files in vector files.
Either change the font file name or return one of the actions shown.

Syntax	FontSupport (BSTR* psName, long* plAction)	
Parameters	psName	Holds percent loaded/saved
	plAction	Return one of the following: RX_EVENT_NO_ACTION = 0, RX_EVENT_USE_DEFAULT_FONT = 1, RX_EVENT_USE_CURRENT_PATH = 3, RX_EVENT_USE_NEW_PATH = 4 Default is RX_EVENT_USE_CURRENT_PATH

4.2.8 BeforePrint

This event occurs prior to any printout from RxViewX.

Syntax	**BeforePrint**(BSTR sFileName, long lDC, long lLayout, long lPage)	
Parameters	sFileName	Holds the path/name of file to be printed
	lDC	Holds the printed device context
	lLayout	Holds the layout number to be printed
	lPage	Holds the page number to be printed

4.2.9 AfterPrint

This event occurs after all printouts from RxViewX

Syntax	**AfterPrint** (BSTR sFileName, long lDC, long lLayout, long lPage)	
Parameters	sFileName	Holds the path/name of file printed
	lDC	Holds the printed device context
	lLayout	Holds the layout number printed
	lPage	Holds the page number printed

4.2.10 FileOpenEvent

This event occurs prior to any file open in RxViewX.

Syntax	**FileOpenEvent**(BSTR***** psFileName, long***** bOK)	
Parameters	psFileName	Holds the path/name of file to be opened. Change this if another file is to be opened
	BOK	Return TRUE if file open should take place. Else FALSE.

4.2.11 FileCloseEvent

This event occurs prior to any file close in RxViewX

Syntax	**FileCloseEvent**(BSTR sFileName, long***** bOK);	
Parameters	sFileName	Holds the path/name of file to be closed.
	bOK	Return TRUE if file closing should take place. Else FALSE

4.2.12 FileActivate

This event occurs when a file is activated in RxViewX.

Syntax	**FileActivate**(BSTR***** psFileName)	
Parameters	psFileName	Holds the path/name of file to be opened. Change this if another file is to be opened

4.2.13 RedlineFileSave

This event occurs prior to any Redline file save in RxViewX

Syntax	RedlineFileSave(BSTR sFileName, BSTR* psRedName, long* bOK);	
Parameters	sFileName	Holds the name of viewed file.
	psRedName	Name of redline file to be saved, can be changed by event handler.
	bOK	Return TRUE if file The saving should take place. Else FALSE

4.2.14 RedlineFileOpen

This event occurs prior to any Redline file open in RxViewX

Syntax	**RedlineFileOpen**(BSTR sFileName, BSTR* psRedName, long* bOK);	
Parameters	sFileName	Holds the name of viewed file.
	psRedName	Name of redline file to be opened, can be changed by event handler.
	bOK	Return TRUE if file The opening should take place. Else FALSE

4.2.15 RedlineFileSaved

This event occurs after the redline file has been saved.

Syntax	RedlineFileSaved(BSTR sFileName);	
Parameters	sFileName	Holds the path/name of the redline file that was saved.

4.2.16 NewElement

This event occurs after a new redline element has been added.

Syntax	NewElement(Long Handle);	
Parameters	Handle	Handle to the new redline element.

4.2.17 PageChange

This event occurs after a page up or page down.

Syntax	PageChange(BSTR sPage);	
Parameters	sPage	Name of current page.

4.2.18 PickElement

This event occurs when you click on a redline element with the mouse.

Syntax	PickElement(Long Handle);	
Parameters	Handle	Handle to the redline element.

4.2.19 SnapFound

This event occurs when you click on a redline element with the mouse.

Syntax	SnapFound(Long xw, Long yw, Long zw, Long xs, Long ys);	
Parameters	xw	Point x in World coordinate.
Parameters	yw	Point y in World coordinate.
Parameters	zw	Point z in World coordinate.
Parameters	xs	Point x in World coordinate.
Parameters	ys	Point y in World coordinate.

5 Appendix A

Notes on Watermark, Header and Footer print texts.
To print specific information as part of the watermark, header or footer, include the following characters as part of the text.

Type this	**To print this**
&F	Path and name of printed file
&f	Name only of printed file
&L	Layout name of printed layout
&U	User signature
&v	Name of printed page, view or spreadsheet
&d	Date in short format (as specified by Regional Settings in Control Panel)
&D	Date in long format (as specified by Regional Settings in Control Panel)
&t	Time in the format specified by Regional Settings in Control Panel
&T	Time in 24-hour format
&p	Current page number
&P	Total number of pages
&&	A single ampersand (&)
&b	The text immediately following these characters will be centered.
&b&b	The text immediately following the first "&b" will be centered, and the text following the second "&b" will be right-justified.

6 Appendix B

Notes on XML redline configuration file.
To control the redlining information on a web system you can set transfer configuration in an XML file

These are the available tags and their meaning

```
<!—Top node →
<XML_RXVIEWX_INFO>
<!— Config info sub tag. →
  <CONFIG_INFO>
<!— Contain the following sub tags →
<!— User layer contain a numerical value equal to the user layer number to use →
    <USER_LAYER>1</USER_LAYER>
<!— User color contain a hex RGB value value equal to the color to use →
    <USER_COLOR>0xff0000</USER_COLOR>
<!— User signature contain a string value for the user signature →
    <USER_SIGNATURE>Frank</USER_SIGNATURE>
<!— User markupfile is an URL to the redline file to save. Need to include user id and
password→
    <USER_EDITOLDCOMMENTS >FALSE</USER_EDITOLDCOMENTS >
<!—  If not FALSE, the user can change previous saved comments →
<USER_MARKUPFILE>http://uid:pwd@www.myweb.com/test.xcm</USER_MARKUPFILE>
<!— Other configuration settings will be supported. →
  </CONFIG_INFO>
<!— Load info sub tag. →
  <LOAD_INFO>
<!— Image info sub tag holds one sub tag URL with the URL to the file to load. →
    <IMAGE_INFO>
      <URL>http://www.rasterex.com/activex/test.dwf</URL>
    </IMAGE_INFO>
<!— Markup info sub tag holds MARKUP_URL sub tags with the URLs to the redline files to
load. →
    <MARKUP_INFO>
      <MARKUP_URL>http://test-3-xp.rasterex.com/Markup/test.xcm</MARKUP_URL>
      <MARKUP_URL>http://test-3-xp.rasterex.com/Markup/test2.xcm</MARKUP_URL>
      <MARKUP_URL>http://test-3-xp.rasterex.com/Markup/new.xcm</MARKUP_URL>
    </MARKUP_INFO>
  </LOAD_INFO>
</XML_RXVIEWX_INFO>
```