



---

# **Arbortext IsoView® Programmer's Reference**

**Arbortext IsoView 7.2  
April 2011**

---

**Copyright © 2011 Parametric Technology Corporation and/or Its Subsidiary Companies. All Rights Reserved.**

User and training guides and related documentation from Parametric Technology Corporation and its subsidiary companies (collectively "PTC") are subject to the copyright laws of the United States and other countries and are provided under a license agreement that restricts copying, disclosure, and use of such documentation. PTC hereby grants to the licensed software user the right to make copies in printed form of this documentation if provided on software media, but only for internal/personal use and in accordance with the license agreement under which the applicable software is licensed. Any copy made shall include the PTC copyright notice and any other proprietary notice provided by PTC. Training materials may not be copied without the express written consent of PTC. This documentation may not be disclosed, transferred, modified, or reduced to any form, including electronic media, or transmitted or made publicly available by any means without the prior written consent of PTC and no authorization is granted to make copies for such purposes.

Information described herein is furnished for general information only, is subject to change without notice, and should not be construed as a warranty or commitment by PTC. PTC assumes no responsibility or liability for any errors or inaccuracies that may appear in this document.

The software described in this document is provided under written license agreement, contains valuable trade secrets and proprietary information, and is protected by the copyright laws of the United States and other countries. It may not be copied or distributed in any form or medium, disclosed to third parties, or used in any manner not provided for in the software licenses agreement except with written prior approval from PTC.

**UNAUTHORIZED USE OF SOFTWARE OR ITS DOCUMENTATION CAN RESULT IN CIVIL DAMAGES AND CRIMINAL PROSECUTION.** PTC regards software piracy as the crime it is, and we view offenders accordingly. We do not tolerate the piracy of PTC software products, and we pursue (both civilly and criminally) those who do so using all legal means available, including public and private surveillance resources. As part of these efforts, PTC uses data monitoring and scouring technologies to obtain and transmit data on users of illegal copies of our software. This data collection is not performed on users of legally licensed software from PTC and its authorized distributors. If you are using an illegal copy of our software and do not consent to the collection and transmission of such data (including to the United States), cease using the illegal version, and contact PTC to obtain a legally licensed copy.

Important Copyright, Trademark, Patent, and Licensing Information: See the About Box, or copyright notice, of your PTC software.

**UNITED STATES GOVERNMENT RESTRICTED RIGHTS LEGEND**

This document and the software described herein are Commercial Computer Documentation and Software, pursuant to FAR 12.212(a)-(b) (OCT'95) or DFARS 227.7202-1(a) and 227.7202-3(a) (JUN'95), and are provided to the US Government under a limited commercial license only. For procurements predating the above clauses, use, duplication, or disclosure by the Government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software Clause at DFARS 252.227-7013 (OCT'88) or Commercial Computer Software-Restricted Rights at FAR 52.227-19(c)(1)-(2) (JUN'87), as applicable. 01012011

**Parametric Technology Corporation, 140 Kendrick Street, Needham, MA 02494 USA**

# Contents

About This Guide.....	9
Introduction .....	15
Specifying the Class ID or Object Type for Embedding .....	16
<object> Tag Attributes and Parameters for Embedding .....	16
Inserting the ActiveX Control Dynamically .....	17
General Functions .....	19
OpenFile .....	20
Iso3OpenFile .....	21
CloseFile.....	22
AboutBox .....	22
GetVersion .....	22
GetDocumentFlags .....	23
Iso3GetLicensee.....	23
SetPassword .....	24
SetZoomRange .....	24
ShutDownServer .....	24
Configuration Functions .....	27
ConfigEvents.....	28
ConfigTools .....	29
ConfigTools2 .....	29
Iso7ConfigTools.....	30
Iso4SetPreference .....	31
Iso4GetPreference.....	32
Iso4EditPreferences.....	32
ConvertTranslationMetadata.....	33
Print Functions .....	35
ConfigPrinting.....	36
DoPrint .....	37
SetPrintStrings .....	38
SetPrintStringStyle.....	39
Redlining Functions .....	41
Iso4StartRedlining .....	42
Iso4StopRedlining.....	42
Iso4ExportRedlining.....	42
Iso5ImportRedlining .....	43
HideRedlining .....	43
Viewport Functions .....	45
GetFirstViewPort .....	46

---

GetNextViewPort .....	46
DisposeViewPortList .....	46
GetViewPortName .....	47
GoHome .....	47
SetHome.....	48
Iso3SetHome .....	48
GetObjectViewPortFile .....	49
GetObjectViewPortName .....	49
 Layer Functions .....	51
GetLayerCount .....	52
GetLayerName .....	52
GetLayerVisibility .....	53
LayerHasObjects .....	53
SetLayerVisibility .....	54
DisposeLayerList .....	54
 Context Menu Functions .....	55
CreateContextMenu .....	56
AddContextMenuItem .....	56
EnableContextMenuItem .....	57
DoPopupMenu .....	57
DisposeContextMenu .....	58
 Rotation Functions .....	59
Iso3GetRotAngle .....	60
Iso3SetRotAngle.....	60
 Animation Functions .....	61
StartAnimation.....	62
StopAnimations .....	62
ConvertAnimationMetadata .....	63
HasAnimations .....	63
CountAnimations .....	64
GetAnimationObjectID.....	64
GetAnimationSequenceName.....	65
GetAnimationFlags .....	66
CountObjectAnimations.....	67
GetObjectAnimationSequenceName .....	67
GetObjectAnimationFlags.....	68
 Magnifier Functions .....	71
Iso4ConfigMagnifier .....	72
Iso4OpenMagnifier .....	72
Iso4CloseMagnifier .....	73
Iso4GetMagnifierState.....	73
Iso4GetMagnifierScale .....	73
Iso4SetMagnifierScale .....	74
 GlobalView Functions .....	75
Iso4ConfigGlobalView .....	76

---

Iso4OpenGlobalView .....	76
Iso4CloseGlobalView .....	77
Iso4GetGlobalViewState .....	77
 Object Functions .....	79
GetObjectCount.....	80
Iso3GetObjectCount .....	80
Iso3ShowObjects.....	80
GetObjectID .....	81
Iso3GetObjectID .....	82
Iso3GetObjectIDByIndex .....	82
GetObjectName.....	83
Iso3GetObjectByName .....	84
Iso3GetObjectByNameByIndex .....	84
GetObjectDataString .....	85
GetObjectFlags .....	85
GetObjectType .....	86
Iso3GetObjectLayer .....	87
HighlightHotspot .....	87
Iso3HighlightObject.....	88
DisposeObjectList.....	89
 Object Tip .....	91
GetObjectTip .....	92
Iso3GetObjectTip.....	92
SetObjectTip .....	93
Iso3SetObjectTip .....	93
GetObjectTipColor .....	94
SetObjectTipColor .....	94
SetObjectTipStyle .....	94
Iso3SetObjectTipStyle.....	95
 Object Attributes.....	97
Iso3GetAttrCount.....	98
Iso3GetAttrByIndex.....	98
Iso3GetAttrType.....	98
Iso3GetAttrValue .....	99
Iso3GetFloatAttrValue .....	99
Iso3SetFloatAttrValue .....	100
Iso3GetIntegerAttrValue .....	100
Iso3SetIntegerAttrValue.....	101
Iso3GetStringAttrValue .....	101
Iso3SetStringAttrValue .....	101
Iso3SetGrAttribute .....	102
 Traversing the Object Tree .....	105
Iso3GetRootObject .....	106
Iso3GetChildCount .....	106
Iso3GetFirstChild .....	106

---

Iso3GetParent .....	107
Iso3GetNextSibling .....	107
Iso3GetPreviousSibling .....	108
Object Links .....	109
Iso3CountLinks .....	110
Iso3GetLinkByIndex .....	110
Iso3SelectLink .....	110
Properties .....	113
AllowJumps .....	114
FileName .....	114
HsVisible .....	114
Preview .....	114
PropertyChangeMode .....	115
ResizeToFit .....	115
ViewPort .....	116
ViewPortCount .....	116
ViewSize .....	116
AcceptDroppedFiles .....	117
ActiveTool .....	117
ExtentX .....	117
ExtentY .....	118
HomeFileName .....	118
HomeViewPort .....	118
ObjectTipState .....	118
OffsetX .....	119
OffsetY .....	119
AutostartOnloadAnimations .....	119
Events .....	121
ContextMenuItemHit .....	122
FileChanged .....	122
HsHit .....	123
InitFinished .....	123
KeyDown .....	124
KeyPress .....	124
KeyUp .....	124
MouseDown .....	125
MouseMove .....	125
MouseUp .....	126
PropertyChanged .....	126
ObjectHit .....	126
ViewWindowChanged .....	127
Appendix A.Object Identifiers and Expressions .....	129
Appendix B.Constants .....	133
Appendix C.Standards Support.....	139

---

Appendix D.Files and File Formats.....	141
Appendix E.ArborText IsoView Basics.....	145



# About This Guide

This *Arbortext IsoView Programmer's Reference* provides the information you need to write application program interface (API) scripts for Arbortext IsoView. It contains descriptions of methods (functions), properties, and events you can use in these scripts, along with code syntax and examples.

You can use API scripts to:

- Customize the appearance and behavior of the Arbortext IsoView object (ActiveX control) embedded in an HTML document.
- Get and change information stored in the illustration displayed in the Arbortext IsoView window.

## Prerequisite Knowledge

To use the Arbortext IsoView API, you need some experience programming simple object-oriented, event-driven scripts in languages such as JavaScript or VBScript. For more information on using the Arbortext IsoView application, see the *Arbortext IsoView User's Reference*.

## Organization of This Guide

This *Arbortext IsoView Programmer's Reference* begins with an [Introduction on page 15](#) that describes how to embed the Arbortext IsoView ActiveX control in HTML documents using the `<object>` tag. It also covers mandatory and optional `<object>` tag parameter settings that affect how the ActiveX control appears and reacts to browser events.

---

The remaining sections of this guide provide code references for the methods (functions), properties, and events that comprise the Arbortext IsoView API. The content of each of these sections is summarized below:

Section	Description
<a href="#">General Functions on page 19</a>	File handling and runtime Arbortext IsoView application management.
<a href="#">Configuration Functions on page 27</a>	Change settings for Arbortext IsoView toolbar appearance and behavior; get and set Arbortext IsoView preferences.
<a href="#">Print Functions on page 35</a>	Get and set Arbortext IsoView printing preferences; specify which illustration content to print; print illustration.
<a href="#">Redlining Functions on page 41</a>	Toggle (start/stop) redlining mode in Arbortext IsoView; import and export redlining (*.rl.iso) files.
<a href="#">Viewport Functions on page 45</a>	Get viewport information for an illustration displayed in Arbortext IsoView; change and reset the <b>Home</b> viewport.
<a href="#">Layer Functions on page 51</a>	Get layer information for an illustration displayed in Arbortext IsoView, such as the layer name and whether it has objects or not; set layer visibility and specify the active layer.
<a href="#">Context Menu Functions on page 55</a>	Create, customize, and manage context menus in the Arbortext IsoView window. (A context menu is a pop-up menu that appears when you right-click inside the Arbortext IsoView window.)
<a href="#">Rotation Functions on page 59</a>	Get and set the rotation angle applied to illustrations when they are opened in Arbortext IsoView.
<a href="#">Animation Functions on page 61</a>	Start and stop animations in Arbortext IsoView; convert DOM-format animation metadata to ISO animation format before running it in Arbortext IsoView.
<a href="#">Magnifier Functions on page 71</a>	Open, close, resize, and specify the position of the <b>Magnifier</b> window in Arbortext IsoView; get and set its default magnification.

Section	Description
<a href="#">GlobalView Functions on page 75</a>	Open, close, resize, and specify the position of the <b>Global View</b> window in Arbortext IsoView; get and set its current view state.
<a href="#">Object Functions on page 79</a>	Get and set object metadata, such as object names and IDs; count and highlight objects in the illustration displayed in Arbortext IsoView.
<a href="#">Object Tip on page 91</a>	Show or hide object tips for the illustration displayed in Arbortext IsoView; get and set object tip text content and styling.
<a href="#">Object Attributes on page 97</a>	Get, set, and edit object attribute values for text, numeric, and graphic object attributes in the open illustration; count object attributes.
<a href="#">Traversing the Object Tree on page 105</a>	Get object tree structure information for objects in the open illustration.
<a href="#">Object Links on page 109</a>	Count, get, and select object links in the open illustration in Arbortext IsoView.
<a href="#">Properties on page 113</a>	Set and change properties of the open illustration in Arbortext IsoView.
<a href="#">Events on page 121</a>	Specify events to generate based on user and program actions in the Arbortext IsoView window, such as highlighting a hot spot when the mouse pointer enters the hot spot area, or displaying a notification when an illustration file changes.
<a href="#">Appendix A-E</a>	Defines parameters used in API methods, properties, and events; lists Arbortext IsoView toolbar tool functions; lists constants for tools, objects, and events; lists supported file formats and files to use with the Arbortext IsoView ActiveX control.

---

## Related Documentation

For more information on Arbortext IsoView and Arbortext IsoDraw products, refer to the following documentation found in the Arbortext IsoDraw Help Center. Help Center includes both HTML and PDF versions of the documentation. Choose **Help ▶ Help Center** to access it.

For Arbortext IsoView information:

Documentation	Description
<i>Arbortext IsoView Release Notes</i>	Information about new, changed, and deleted features in this Arbortext IsoView release.
<i>Arbortext IsoView Installation Guide</i>	Installation and licensing information for Arbortext IsoView.
<i>Arbortext IsoView User's Reference</i>	Instructions for using the tools and functions in Arbortext IsoView, and for setting preferences.
<i>Arbortext IsoView Programmer's Reference</i>	(This guide) Reference for writing API scripts that customize and extend the Arbortext IsoView ActiveX control.

For Arbortext IsoDraw information:

Documentation	Description
<i>Arbortext IsoDraw Release Notes</i>	Information about new, changed, and deleted features in this Arbortext IsoDraw release.
<i>Installing Arbortext IsoDraw</i>	Installation and licensing information for Arbortext IsoDraw.
<i>Drawing Basics Tutorial</i>	Hands-on examples for learning Arbortext IsoDraw basic functions.
<i>3D Mode Tutorial</i>	Hands-on examples for learning 3D CAD data editing functions using Arbortext IsoDraw.
<i>Arbortext IsoDraw User's Reference</i>	Comprehensive guide to using the tools and functions in Arbortext IsoDraw products.
<i>Arbortext IsoDraw Macro Language Reference</i>	Reference for writing macros that you can run in Arbortext IsoDraw.
<i>Arbortext IsoDraw Data Exchange Reference</i>	Instructions for importing and exporting graphics data in various formats to and from Arbortext IsoDraw.

---

## Technical Support

Contact PTC Technical Support via the PTC Web site, phone, fax, or e-mail if you encounter problems using your product or the product documentation.

For complete details, refer to Contacting Technical Support in the PTC Customer Service Guide. This guide can be found under the Related Resources section of the PTC Web site at:

<http://www.ptc.com/support/>

The PTC Web site also provides a search facility for technical documentation of particular interest. To access this search facility, use the URL above and select Search the Knowledge Base.

You must have a Service Contract Number (SCN) before you can receive technical support. If you do not have an SCN, contact PTC Maintenance Department using the instructions found in your *PTC Customer Service Guide* under Contacting Your Maintenance Support Representative.

## Global Services

PTC Global Services delivers the highest quality, most efficient and most comprehensive deployments of the PTC Product Development System including Pro/ENGINEER, Windchill, Arbortext and Mathcad. PTC's Implementation and Expansion solutions integrate the process consulting, technology implementation, education and value management activities customers need to be successful. Customers are led through Solution Design, Solution Development and Solution Deployment phases with the continuous driving objective of maximizing value from their investment.

Contact your PTC sales representative for more information on Global Services.

## Comments

PTC welcomes your suggestions and comments on our documentation. You can submit your feedback to the following email address:

[arbortext-documentation@ptc.com](mailto:arbortext-documentation@ptc.com)

Please include the following information in your email:

- Name
- Company
- Product
- Product Version
- Document or Online Help Topic Title

- 
- Level of Expertise in the Product (Beginning, Intermediate, Advanced)
  - Comments (including page numbers where applicable)

## Documentation Conventions

This guide uses the following notational conventions:

- **Bold text** represents exact text that appears in the program's user interface. This includes items such as button text, menu selections, and dialog box elements.  
For example,  
Click **OK** to begin the operation.
- A right arrow represents successive menu selections. For example,  
Choose **File > Print** to print the document.
- Monospaced text represents code, command names, file paths, or other text that you would type exactly as described. For example,  
At the command line, type `version` to display version information.
- *Italicized monospaced text* represents variable text that you would type. For example,  
`installation-dir\custom\ scripts\`
- Italicized text represents a reference to other published material. For example,  
If you are new to the product, refer to the Getting Started Guide for basic interface information.

# 1

## Introduction

Specifying the Class ID or Object Type for Embedding.....	16
<object> Tag Attributes and Parameters for Embedding.....	16
Inserting the ActiveX Control Dynamically.....	17

This section describes how to embed the Arbortext IsoView ActiveX control in HTML documents.

---

## Specifying the Class ID or Object Type for Embedding

When you embed the Arbortext IsoView ActiveX Control in an HTML document, specify the class ID or object type as shown below:

```
classid="CLSID:865B2280-2B71-11D1-BC01-006097AC382A"
```

or

```
type="application/x-isoview"
```

We strongly recommend use of the `<object>` tag in Microsoft Internet Explorer. Otherwise, formatting and scaling problems might occur when printing HTML documents with embedded Arbortext IsoView objects.

```
<object id="ivxl" classid="CLSID:865B2280-2B71-11D1-BC01-006097AC382A"  
       width="400" height="300">  
<param name="src" value="sample.iso">  
</object>
```

or

```
<object id="ivxl" type="application/x-isoview" width="400" height="300">  
<param name="src" value="sample.iso">  
</object>
```

## `<object>` Tag Attributes and Parameters for Embedding

The mandatory and optional `<object>` tag attributes and parameters below can be used for embedding the ActiveX control.

### Optional `<object>` Attributes

Name	Description
classid or type	Enforces usage of Arbortext IsoView; mandatory if no <code>src</code> file is specified
id	ID of the element (for use with JavaScript)

### Mandatory `<object>` Parameters

Name	Description
width	Width of the illustration in pixels
height	Height of the illustration in pixels

---

## Optional <object> Parameters

Name	Description
border	1 = border; 0 = no border
src	Filename of the illustration
view	Name of the home view
tools	1 = toolbar; 0 = no toolbar
events	Enables various types of events. (See <a href="#">ConfigEvents on page 28</a> for event parameter values.)
hsvisible	1=hotspots visible; 0=hotspots invisible. (See <a href="#">HsVisible on page 114</a> .)
allowjumps	1=internal viewport jumps allowed; 0=internal viewport jumps not allowed. (See <a href="#">AllowJumps on page 114</a> .)
offsetx	Sets the horizontal position of the representation as Double. (See <a href="#">OffsetX on page 119</a> .)
offsety	Sets the vertical position of the representation as Double. (See <a href="#">OffsetY on page 119</a> .)
scale	Sets the zoom factor of the representation as Double. (See <a href="#">ViewSize on page 116</a> .)
rotangle	Sets the rotation angle of the illustration (in degrees) on open as Double. (See <a href="#">Iso3SetRotAngle on page 60</a> .)
onload	WebCGM 2.0 object tag parameters. Definitions can be found at: <a href="http://www.w3.org/TR/webcgm/WebCGM20-IC.html#webcgm_3_4">http://www.w3.org/TR/webcgm/WebCGM20-IC.html#webcgm_3_4</a> .
fixed	
mapping	
halign	
valign	

## Inserting the ActiveX Control Dynamically

If an ActiveX Control is embedded into an HTML page utilizing the <applet> tag, <embed> tag, or <object> tag, it will require an additional activation click. This means that the graphic will be displayed, but stay inactive until the user has clicked on it.

In order to avoid this additional activation click, insert the ActiveX control dynamically as follows:

```
<html>
<head>
  <script language="JavaScript">
  <!--
    function Init()
```

---

```
{  
    gIv = document.ivx1;  
    gIv.Iso3OpenFile(  
        'six_cubes.iso#name(isovp_home,  
            view_context)' );  
}  
//-->  
</script>  
</head>  
<body onLoad="Init()">  
    <p align="center">  
        <script src="embed.js"></script>  
        <br>  
    </p>  
</body>  
</html>
```

This technique requires an additional external file, `embed.js` containing the following code:

```
document.write('<object id="ivx1"  
document.write('type="application/x-isoview"  
document.write('width="400" height="300">');  
document.write('<param name="border" value="1">');  
document.write('</object>');
```

# 2

## General Functions

OpenFile.....	20
Iso3OpenFile .....	21
CloseFile .....	22
AboutBox.....	22
GetVersion.....	22
GetDocumentFlags.....	23
Iso3GetLicensee .....	23
SetPassword.....	24
SetZoomRange.....	24
ShutDownServer .....	24

---

# OpenFile

## Note

This method is deprecated and supported for backward compatibility only. Use `Iso3Openfile` instead. (See [Iso3OpenFile on page 21](#).)

The `OpenFile` method opens an illustration and sets the specified viewport (View). The first call to `OpenFile` defines the Home view. From then on you can return to this initial view by clicking the **Home** button. (See [Toolbar on page 146](#).)

To set a new viewport in the current open illustration use an empty string "" as a *FileName*.

Introduced with Arbortext IsoView 1.0.

## Syntax

`OpenFile (lpszFileName, lpszView)`

Parameters:

<b>lpszFileName</b>	This parameter defines the file name or URL of the illustration to be opened as BSTR.
<b>lpszView</b>	This parameter defines the name of the view (ViewPort) as BSTR. Arbortext IsoView can generate viewports dynamically for a given object.

Return value: BOOL.

In all places where a string for a viewport is used you can also use the following:

<b>\$OBJ_ObjectName</b> or <b>\$NAME_Object-Name</b>	Where \$OBJ_ or \$NAME_ precedes the name of the object you want to show and ObjectName is the object's name. For example, if the name of the object is HS1 the string would be \$OBJ_HS1 or \$NAME_HS
<b>\$ID_ObjectID</b>	Where \$ID_ precedes the ID of the object you want to show and ObjectID is the object's ID. For example, if the name of the object is 2006 the string would be \$ID_2006.
<b>\$EXTENT</b>	Use this name to instruct Arbortext IsoView to build a viewport that fits around the extent of the file. You can use the <code>OpenFile</code> command to only change the viewport. In this case pass an empty string for the filename.

---

## Example

The following command opens a URL as viewport View1:

```
document.ivx1.OpenFile("http://www.ptc.com/demo/test.iso", "View1");
```

## Iso3OpenFile

Open a file using a URL. This URL may contain a fragment.

Introduced with Arbortext IsoView 3.0.

### Syntax

```
Iso3OpenFile (url)
```

The *url* parameter defines the url which should be opened as BSTR.

Allowed URLs are:

<b>http.../abc.iso</b>	absolute URL
<b>abc.iso</b>	relative URL, if a file is open only
<b>abc.iso#fragment</b>	fragment specifies object or viewport
<b>#fragment</b>	fragment points to current file

In case of success the method returns true, otherwise false.

## Example

Open the file at the specified location and navigate to the object myObj1:

```
http://www.ptc.com/abc.iso#id(myObj1)
```

Fragment samples using the ID of the object:

```
// navigate to object "myObj1"  
#id(myObj1)  
// zoom to the object's view context  
#id(myObj1,view_context)  
// highlight the object  
#id(myObj1,highlight)
```

Fragment samples using the name of the object:

```
// navigate to object "myName1"  
#name(myName1)  
// zoom to the object's view context  
#name(myName1,view_context)  
// highlight the object  
#name(myName1,highlight)  
// highlight all objects with this name  
#name(myName1,highlight_all)
```

---

The fragment syntax is defined in the WebCGM specification. It may also be used for CGM and Arbortext IsoDraw files; e.g., `http://www.ptc.com/abc.iso#id(myObj1)`.

## CloseFile

This method closes the current illustration.

Introduced with Arbortext IsoView 2.0.

### Syntax

```
CloseFile()
```

### Example

```
document.ivx1.CloseFile();
```

## AboutBox

This method displays an **AboutBox** dialog of the ActiveX control. Information on the version number and the license type can be taken from that dialog.

Introduced with Arbortext IsoView 1.0.

### Syntax

```
AboutBox()
```

### Example

```
document.ivx1.AboutBox();
```

## GetVersion

This method retrieves the version number of the plugin and returns it as a BSTR.

Introduced with Arbortext IsoView 2.0.

### Syntax

```
GetVersion()
```

### Example

```
edtVersion.value = document.ivx1.GetVersion();
```

---

## GetDocumentFlags

This method is used to determine if a document contains animations or 3D objects.

Introduced with Arbortext IsoView 7.1.

### Syntax

```
GetDocumentFlags()
```

Parameters / Return Values:

This method does not use parameters. It returns the values below in a bit-mapped array:

0x0002	The document contains 3D objects.
0x0004	The document contains animations of any type.  <b>Note</b> <i>To determine if a document contains a specific type of animation, use the HasAnimations method. (See <a href="#">HasAnimations on page 63</a>.)</i>

### Example

```
flags = GetDocumentFlags();
```

## Iso3GetLicensee

This method returns the string which identifies the licensee of Arbortext IsoView as BSTR.

Introduced with Arbortext IsoView 3.0.

### Syntax

```
Iso3GetLicensee()
```

### Example

```
licensee = document.ivx1.Iso3GetLicensee();
```

---

## SetPassword

This method enables Arbortext IsoView to read password protected Arbortext IsoDraw files.

Introduced with Arbortext IsoView 2.02.

### Syntax

`SetPassword (Password)`

Parameters:

<b>Password</b>	Defines the password to open protected files as BSTR.
-----------------	---

In case of success, the method returns `true`, otherwise `false`.

### Example

```
document.ivx1.SetPassword("TopSecret!");
```

## SetZoomRange

Use this method to limit the view size (set using the `ViewSize` property). (See [ViewSize on page 116](#).)

Introduced with Arbortext IsoView 2.02.

### Syntax

`SetZoomRange (minVal, maxVal)`

Parameters:

<b>minVal</b>	Defines the lower range for zoom factor (min. 1%) as double.
<b>maxVal</b>	Defines upper range for zoom factor (max. 6400%) as double.

### Example

```
document.ivx1.SetZoomRange(50, 200);
```

## ShutDownServer

This method tells Arbortext IsoView to close the appertaining Arbortext IsoView Server process.

---

### **Caution**

*No further processing is possible after this command. Use this method only when your container process is completed.*

Introduced with Arbortext IsoView 1.0.

### **Syntax**

```
ShutdownServer()
```

### **Example**

```
document.ivx1.ShutdownServer()
```



# 3

## Configuration Functions

ConfigEvents .....	28
ConfigTools .....	29
ConfigTools2 .....	29
Iso7ConfigTools.....	30
Iso4SetPreference.....	31
Iso4GetPreference .....	32
Iso4EditPreferences .....	32
ConvertTranslationMetadata .....	33

---

# ConfigEvents

This method allows you to select which events will be generated by Arbortext IsoView. Disabling unneeded events will increase performance. (See [Events on page 121](#). For constant values, see [Event Configuration Constants – ConfigEvents on page 137](#).)

Introduced with Arbortext IsoView 2.0.

## Syntax

`ConfigEvents (events)`

The `events` parameter defines which events are enabled. Allowed values are:

<b>1</b>	HsHit
<b>2</b>	FileChanged
<b>4</b>	ContextMenuHit
<b>8</b>	PropertyChanged
<b>16</b>	InitFinished
<b>32</b>	ObjectHit
<b>64</b>	ViewWindowChanged
<b>1024</b>	MouseMove
<b>2048</b>	MouseDown
<b>4096</b>	MouseUp
<b>8192</b>	DontHandleClicks  This prevents Arbortext IsoView from internal click handling (i.e., from bringing up a context menu). It must always be combined with MouseMove + MouseDown.

Add values to combine enabled events. The default `events` setting is `(1+2+4+8+16)`.

By default, mouse events are disabled; all others are enabled.

## Example

```
// enable default events + mouse move
document.ivx1.ConfigEvents(1+2+4+8+16+32+1024);
```

---

# ConfigTools

## Note

*This method is deprecated and supported for backward compatibility only. Use Iso7ConfigTools instead. (See [Iso7ConfigTools on page 30](#).)*

This method allows you to configure the Toolbar. (See [Toolbar on page 146](#).)

Introduced with Arbortext IsoView 1.0.

## Syntax

ConfigTools (*bWithTools*, *bAsPopup*, *bTop*, *bRight*)

Parameters:

<b>bWithTools</b>	Defines if the Toolbar is available as boolean (default: true).
<b>bAsPopup</b>	Defines if the Toolbar is displayed as popup menu as boolean (not supported at this time).
<b>bTop</b>	Defines if the Toolbar is displayed on the top window border as boolean (default: false).
<b>bRight</b>	Defines if the Toolbar is displayed right aligned as boolean (default: false).

## Example

```
// place tool bar in the upper left cornerdocument.  
ivx1.ConfigTools( true, false, true, false );
```

# ConfigTools2

## Note

*This method is deprecated and supported for backward compatibility only. Use Iso7ConfigTools instead. (See [Iso7ConfigTools on page 30](#).)*

This method allows you to configure the Toolbar. (See [Toolbar on page 146](#).)

Introduced with Arbortext IsoView 2.0.

## Syntax

ConfigTools2 (*ToolMode*, *ToolSet*, *SelectdTool*, *Top*, *Right*)

Parameters:

---

<b>ToolMode</b>	Sets the mode of the Arbortext IsoView tools as long. Allowed values are:	
	<b>0</b>	tools disabled
	<b>1</b>	default tools
	<b>2</b>	force all tools as popup
	<b>4</b>	Toolbar invisible; select using ActiveTool
	<b>8</b>	auto hide Redlining tools if not in Redlining mode
Use ActiveTool to switch between tools. (See <a href="#">ActiveTool on page 117</a> .)		
<b>ToolSet</b>	Configures the available tools as long. (For allowed values, see <a href="#">Tool Set/Selection Constants – ConfigTools2, ActiveTool on page 134</a> .)	
<b>SelectedTool</b>	Selects the active tool as long. (For allowed values, see <a href="#">Tool Set/Selection Constants – ConfigTools2, ActiveTool on page 134</a> .)	
<b>Top</b>	Defines if the Toolbar is anchored on the top of the window as boolean. The default value is <code>false</code> .	
<b>Right</b>	Defines if the Toolbar is right aligned as boolean. The default value is <code>false</code> .	

### Example

```
// place tool bar with pan, zoom in, zoom out and
// print button in the upper left corner and
// select pan tool as the active tool.
ivx1.ConfigTools2(1, 1+2+4+16, 1, true, false);
```

## Iso7ConfigTools

This method allows you to configure the Toolbar. (See [Toolbar on page 146](#).) It is the enhanced version of the `ConfigTools` and `ConfigTools2` methods. (See [ConfigTools on page 29](#) and [ConfigTools2 on page 29](#).)

### Note

*The old method, `ConfigTools2`, should only be used to configure tools available in versions before 7.0.*

Introduced first with Arbortext IsoView 7.0.

---

## Syntax

`Iso7ConfigTools (nToolmode, nBtnSet, nPopSet, nSelectedTool, bTop, bRight)`

Parameters:

<b>nToolMode</b>	Sets the mode of the Arbortext IsoView tools as long. Allowed values are: <table border="1"><tr><td>0</td><td>tools disabled</td></tr><tr><td>1</td><td>default tools</td></tr><tr><td>2</td><td>force all tools as popup</td></tr><tr><td>4</td><td>Toolbar invisible, select using ActiveTool</td></tr><tr><td>8</td><td>auto hide Redlining tools if not in Redlining mode</td></tr></table> Use ActiveTool to switch between tools. (See <a href="#">ActiveTool on page 117</a> .)	0	tools disabled	1	default tools	2	force all tools as popup	4	Toolbar invisible, select using ActiveTool	8	auto hide Redlining tools if not in Redlining mode
0	tools disabled										
1	default tools										
2	force all tools as popup										
4	Toolbar invisible, select using ActiveTool										
8	auto hide Redlining tools if not in Redlining mode										
<b>nBtnSet</b>	Configures the available tools as long. (For allowed values, see <a href="#">Tool Set/Selection Constants – Iso7ConfigTools and ActiveTool on page 135</a> .)										
<b>nPopSet</b>	Configures the available tools in the popup menu as long. (For allowed values, see <a href="#">Tool Set/Selection Constants – Iso7ConfigTools and ActiveTool on page 135</a> .)										
<b>nSelectedTool</b>	Selects the active tool as long. (For allowed values, see <a href="#">Tool Set/Selection Constants – Iso7ConfigTools and ActiveTool on page 135</a> .)										
<b>nTop</b>	Defines if the Toolbar is anchored on the top of the window as boolean. The default value is <code>false</code> .										
<b>nRight</b>	Defines if the Toolbar is right aligned as boolean. The default value is <code>false</code> .										

## Example

```
// place tool bar with pan, zoom in, zoom out and  
// print button, home, magnifier and globalview  
// buttons as popup in the upper left corner and  
// select pan tool as the active tooldocument.  
ivx1.Iso7ConfigTools(1, 1+2+4+16, 32+64+128, 1, true, false);
```

## Iso4SetPreference

Sets the data of a given preference. (See [Overview of Preference Identifiers on page 148](#) for constant values.)

---

Introduced with Arbortext IsoView 4.0.

### Syntax

`Iso4SetPreference (prefName, prefValue)`

Parameters:

<b>prefName</b>	Defines the name of the preference item as BSTR. (See <a href="#">Overview of Preference Identifiers on page 148</a> .)
<b>prefValue</b>	Defines the new data of the given preference as BSTR.

### Example

```
document.ivx1.Iso4SetPreferences ("default_toolset", (1+2+4+8));
```

## Iso4GetPreference

Retrieves the data of a given preference item and returns it as BSTR. (See [Overview of Preference Identifiers on page 148](#).)

Introduced with Arbortext IsoView 4.0.

### Syntax

`Iso4GetPreference (prefName)`

Parameters:

<b>prefName</b>	Defines the name of the preference item as BSTR. (See <a href="#">Overview of Preference Identifiers on page 148</a> .)
-----------------	---

### Example

```
pref1 = document.ivx1.Iso4GetPreference("default_toolset");
```

## Iso4EditPreferences

Opens the Arbortext IsoView preferences dialog.

Introduced with Arbortext IsoView 4.0.

### Syntax

`Iso4EditPreferences ()`

### Example

```
document.ivx1.Iso4EditPreferences ()
```

---

## ConvertTranslationMetadata

This method applies (converts) translation metadata which has been loaded using the applyCompanionFile method.

Introduced with Arbortext IsoView 7.1.

### Syntax

```
ConvertTranslationMetadata()
```

Parameters:

This method has no parameters.

### Example

```
document.ivx1.ConvertTranslationMetadata();
```



# 4

## Print Functions

ConfigPrinting .....	36
DoPrint .....	37
SetPrintStrings .....	38
SetPrintStringStyle .....	39

---

# ConfigPrinting

This method allows you to set preferences for printing. (See [Print Configuration Constants on page 137](#) for constant values.)

Return value: BOOL.

Introduced with Arbortext IsoView 2.0.

## Syntax

ConfigPrinting (*OnlyVisContent*, *SplitPages*, *Orientation*, *Scale*)

Parameters:

<b>OnlyVisContent</b>	Defines if only the visible content will be printed as short. Allowed values are:
-1	KEEPINTERSETTINGS
0	PRINTALL (the entire illustration will be printed)
1	PRINTVISIBLE (only the portion of the illustration which is visible inside the viewer will be printed (default))
<b>SplitPages</b>	Defines if the illustration will be split as short. Allowed values are:
-1	KEEPINTERSETTINGS
0	FITONPAGE (the illustration will be fitted onto one page (default))
1	SPLITPAGES (the illustration will be split across multiple pages if necessary)
<b>Orientation</b>	Selects the orientation of printed page as short. Allowed values are:
-1	KEEPINTERSETTINGS
1	PORTRAIT
2	LANDSCAPE
<b>Scale</b>	Defines the scale of the printed illustration as double. Allowed values are:
-1	KEEPINTERSETTINGS
<b>Value &gt; 0</b>	scale factor

---

## Example

```
document.ivx1.ConfigPrinting( 1, 0, -1, 100.0 );
```

# DoPrint

This method prints the open illustration to the default printer. You can set additional information using SetPrintStrings. (See [SetPrintStrings on page 38](#). Also see [Print Configuration Constants on page 137](#) for constant values.)

Introduced with Arbortext IsoView 2.0.

## Syntax

DoPrint (*OnlyVisContent*, *SplitPages*, *Orientation*, *Scale*)

Parameters:

<b>OnlyVisContent</b>	Defines if only the visible content will be printed as short. Allowed values are:	
	<b>-1</b>	KEEPINTERSETTINGS
	<b>0</b>	PRINTALL (the entire illustration will be printed)
	<b>1</b>	PRINTVISIBLE (only the portion of the illustration which is visible inside the viewer will be printed (default))
<b>SplitPages</b>	Defines if the illustration will be split as short. Allowed values are:	
	<b>-1</b>	KEEPINTERSETTINGS
	<b>0</b>	FITONPAGE (the illustration will be fitted onto one page (default))
	<b>1</b>	SPLITPAGES (the illustration will be split across multiple pages if necessary)

---

<b>Orientation</b>	Selects the orientation of printed page as short. Allowed values are:	
	-1	KEEPPRINTERSETTINGS
	1	PORTRAIT
	2	LANDSCAPE
<b>Scale</b>	Defines the scale of the printed illustration as double. Allowed values are:	
	-1	KEEPPRINTERSETTINGS
<b>Value &gt; 0</b>		scale factor

### Example

```
document.ivx1.DoPrint( 1, 0, -1, 100.0 );
```

## SetPrintStrings

Use this method to set the text for the header and footer strings placed on a printed page. Use [SetPrintStringStyle on page 39](#) to configure the style and position of these strings.

Introduced with Arbortext IsoView 2.0.

### Syntax

`SetPrintStrings (Header, Footer)`

Parameters:

<b>Header</b>	Defines the text for page header as BSTR.
<b>Footer</b>	Defines the text for the page footer as BSTR. Within this text, the following macros can be used:
<b>\$PATH</b>	filename and path of the illustration
<b>\$NAME</b>	filename of the illustration
<b>\$DATE</b>	current date
<b>\$TIME</b>	current time

### Example

```
document.ivx1.SetPrintStrings ("Parametric Technology Corporation", "PTC");
```

# SetPrintStringStyle

Use this method to set the text style used when printing header and footer strings.

Use SetPrintStrings to set the text for the header and footer. (See [SetPrintStrings on page 38](#).)

Introduced with Arbortext IsoView 2.0.

## Syntax

```
SetPrintStringStyle ( Target, Font, Size, Style, Alignment, TopMargin,  
BotMargin, TopGap, BotGap )
```

Parameters:

<b>Target</b>	Defines the target for the text on the printed page as short. Allowed values are: <table border="1"><tr><td><b>1</b></td><td>HEADER</td></tr><tr><td><b>2</b></td><td>FOOTER</td></tr><tr><td><b>3</b></td><td>HEADER AND FOOTER</td></tr></table>	<b>1</b>	HEADER	<b>2</b>	FOOTER	<b>3</b>	HEADER AND FOOTER		
<b>1</b>	HEADER								
<b>2</b>	FOOTER								
<b>3</b>	HEADER AND FOOTER								
<b>Font</b>	Defines the display name of the font as BSTR.								
<b>Size</b>	Defines the size of the text in points as short.								
<b>Style</b>	Defines the style of the text as short. Allowed values are: <table border="1"><tr><td><b>0</b></td><td>NORMAL</td></tr><tr><td><b>1</b></td><td>BOLD</td></tr><tr><td><b>2</b></td><td>ITALIC</td></tr><tr><td><b>3</b></td><td>BOLDITALIC</td></tr></table> <b>Tip</b> Add 256 if you want Arbortext IsoView to keep the tip window outside the object's area.	<b>0</b>	NORMAL	<b>1</b>	BOLD	<b>2</b>	ITALIC	<b>3</b>	BOLDITALIC
<b>0</b>	NORMAL								
<b>1</b>	BOLD								
<b>2</b>	ITALIC								
<b>3</b>	BOLDITALIC								
<b>Alignment</b>	Defines the alignment of the text as short. Allowed values are: <table border="1"><tr><td><b>0</b></td><td>LEFT</td></tr><tr><td><b>1</b></td><td>CENTER</td></tr><tr><td><b>2</b></td><td>RIGHT</td></tr></table>	<b>0</b>	LEFT	<b>1</b>	CENTER	<b>2</b>	RIGHT		
<b>0</b>	LEFT								
<b>1</b>	CENTER								
<b>2</b>	RIGHT								
<b>TopMargin</b>	Defines the margin (in mm) between top of the illustration and the upper border of the printable area (KEEPSETTINGS: -1) as double.								

---

<b>BotMargin</b>	Defines the margin (in mm) between bottom of the illustration and the lower border of the printable area (KEEPSETTINGS: -1) as double.
<b>TopGap</b>	Defines the gap (in mm) between top of the illustration and the lower border of the header (KEEPSETTINGS: -1) as double.
<b>BotGap</b>	Defines the gap (in mm) between bottom of the illustration and the upper border of the footer (KEEPSETTINGS: -1) as double.

## Example

```
document.ivx1.SetPrintStringsStyle( 1, "Courier", 12, 0, 1, 20.0, 20.0, 5.0, 5.0 );
```

# 5

## Redlining Functions

Iso4StartRedlining .....	42
Iso4StopRedlining .....	42
Iso4ExportRedlining .....	42
Iso5ImportRedlining.....	43
HideRedlining .....	43

---

## Iso4StartRedlining

Starts the Redlining mode. Redlining tools can only be selected if Redlining mode is enabled. This is controlled through the Iso7ConfigTools settings. (See [Iso7ConfigTools on page 30](#).) If the Redlining tools are disabled the Redlining mode cannot be started through this method.

Introduced with Arbortext IsoView 4.0.

### Syntax

```
Iso4StartRedlining()
```

In case of success the method returns true, otherwise false.

### Example

```
document.ivx1.Iso4StartRedlining();
```

## Iso4StopRedlining

Stops the Redlining mode.

Introduced with Arbortext IsoView 4.0.

### Syntax

```
Iso4StopRedlining()
```

### Example

```
document.ivx1.Iso4StopRedlining();
```

## Iso4ExportRedlining

Exports a pair of redlining and illustration files. To use the redlining file in Arbortext IsoDraw, open the illustration file and import the redlining file as an additional layer.

Introduced with Arbortext IsoView 4.0.

### Syntax

```
Iso4ExportRedlining(illuFile, rlFile)
```

Parameters:

---

<b>illuFile</b>	Defines the destination path and name of the illustration as BSTR.
<b>rlFile</b>	Defines the destination path and name of the redlining file as BSTR.

In case of success the method returns `true`, otherwise `false`.

### Example

```
document.ivx1.Iso4ExportRedlining("C:\\TEMP\\catalog.iso",
                                    "C:\\TEMP\\catalog_rl.iso");
```

## Iso5ImportRedlining

Imports a redlining file into the current illustration. Here the same file might be imported, which has been exported by the use of `Iso4ExportRedlining` ( . . . )

Introduced with Arbortext IsoView 5.0.

### Syntax

```
Iso5ImportRedlining(rlFile)
```

Parameters:

<b>rlFile</b>	Defines the destination path and name of the redlining file as BSTR.
---------------	--

In case of success the method returns `true`, otherwise `false`.

### Example

```
document.ivx1.Iso5ImportRedlining("C:\\TEMP\\catalog_rl.iso");
```

## HideRedlining

Hides or shows redlining information in the viewing area of the Arbortext IsoView window (if redlining information exists).

Introduced with Arbortext IsoView 7.1.

### Syntax

```
HideRedlining(state)
```

Parameters:

---

<b>state</b>	Determines if existing redlining information is hidden or shown. Allowed values are:
<b>true</b>	Hides redlining information
<b>false</b>	Shows redlining information

## Example

```
HideRedlining( true );
```

# 6

## Viewport Functions

GetFirstViewPort .....	46
GetNextViewPort .....	46
DisposeViewPortList .....	46
GetViewPortName .....	47
GoHome .....	47
SetHome .....	48
Iso3SetHome .....	48
GetObjectViewPortFile .....	49
GetObjectViewPortName .....	49

---

## GetFirstViewPort

This method provides the name of the first viewport in the graphic. All further names can be determined by [GetNextViewPort](#). (See [GetNextViewPort on page 46](#).)

It returns the name of the first viewport as BSTR.

Introduced with Arbortext IsoView 1.0.

### Syntax

```
GetFirstViewPort()
```

### Example

```
form1.edtFile.value = document.ivx1.GetFirstViewPort();
```

## GetNextViewPort

This method provides the name of the next viewport in the graphic. The first name can be obtained by selecting [GetFirstViewPort](#). (See [GetFirstViewPort on page 46](#).)

The method returns the Name of the viewport. At the end of the list, an empty string is returned.

Introduced with Arbortext IsoView 1.0.

### Syntax

```
GetNextViewPort()
```

### Example

```
form1.edtFile.value = document.ivx1.GetNextViewPort();
```

## DisposeViewPortList

This method disposes the viewport list. This list generated by calls to [GetViewPortName](#). (See [GetViewPortName on page 47](#).) Once you have retrieved your preferred viewport information it is recommended to call [DisposeViewPortList](#) to free the used memory.

Introduced with Arbortext IsoView 1.0.

### Note

*This method has been obsolete since version 3.0.*

---

## Syntax

```
DisposeViewPortList()
```

In case of success the method returns true, otherwise false.

## Example

```
document.ivx1.DisposeViewPortList();
```

# GetViewPortName

This method retrieves the name of the viewport with the given index. Use ViewPortCount to get the number of viewports in the illustration. (See [ViewPortCount on page 116](#).)

After you have retrieved your preferred viewport information, call DisposeViewPortList to free the memory used by Arbortext IsoView to keep the layer information for all viewports. (See [DisposeViewPortList on page 46](#).) This list will be rebuilt after the next call to GetViewPortName.

This method returns the name of this viewport as BSTR.

Introduced with Arbortext IsoView 2.0.

## Syntax

```
GetViewPortName (Index)
```

Parameters:

<b>Index</b>	Defines the index of the viewport as long (0 based).
--------------	--

## Example

```
for ( n = 0; n < document.ivx1.GetViewPortCount(); n++ ){
  form1.edtName.value = document.ivx1.GetViewPortName(n);
} document.ivx1.DisposeViewPortList();
```

# GoHome

This method instructs Arbortext IsoView to reset the view to the home viewport. This is the first viewport set with OpenFile, or the one set by SetHome. (See [OpenFile on page 20](#) and [SetHome on page 48](#).)

Introduced with Arbortext IsoView 2.0.

---

## Syntax

GoHome ()

### Example

```
document.ivx1.GoHome();
```

## SetHome

Sets the file and the name of the home viewport.

Introduced with Arbortext IsoView 2.0.

### Syntax

SetHome (*FileName*, *View*)

Parameters:

<b>FileName</b>	Defines the file name or URL of the illustration as BSTR.
<b>View</b>	Defines the name of the home view port as BSTR.

In case of success the method returns `true`, otherwise `false`.

### Example

```
document.ivx1.SetHome ("test1.iso", "home1");
```

## Iso3SetHome

Sets the home location. May use a fragment.

Introduced with Arbortext IsoView 3.0.

### Syntax

Iso3SetHome (*url*)

Parameters:

<b>url</b>	Defines the location as BSTR.
------------	-------------------------------

### Example

```
document.ivx1.Iso3SetHome ("http://www.ptc.com/abc.iso#id(myObj1)");
```

---

## GetObjectViewPortFile

This method retrieves the viewport file of the specified object and returns it as a string.

The method returns the Object Viewport Filename as BSTR.

Introduced with Arbortext IsoView 2.02.

### Syntax

GetObjectViewPortFile (*ID, Name*)

Parameters:

<b>ID</b>	Defines the ID of the object as long.
<b>Name</b>	Defines the name of the object as BSTR.

### Example

```
edtData.value = document.ivx1.GetObjectViewPortFile( id, name );
```

## GetObjectViewPortName

This method retrieves the viewport name of the specified object and returns it as a string.

The method returns the Object Viewport Name as BSTR.

Introduced with Arbortext IsoView 2.02.

### Syntax

GetObjectViewPortName (*ID, Name*)

Parameters:

<b>ID</b>	Defines the ID of the object as long.
<b>Name</b>	Defines the name of the object as BSTR.

### Example

```
edtData.value = document.ivx1.GetObjectViewPortName( id, name );
```



# 7

## Layer Functions

GetLayerCount.....	52
GetLayerName.....	52
GetLayerVisibility.....	53
LayerHasObjects.....	53
SetLayerVisibility .....	54
DisposeLayerList.....	54

---

## GetLayerCount

Use this method to retrieve the number of layers in the current illustration. It returns the number of layers as long.

To retrieve further layer information you can use [GetLayerName](#), [GetLayerVisibility](#). (See [GetLayerName on page 52](#) and [GetLayerVisibility on page 53](#).)

Introduced with Arbortext IsoView 2.0.

### Syntax

```
GetLayerCount ()
```

### Example

```
n = document.ivx1.GetLayerCount();
```

## GetLayerName

This method retrieves the name of the layer with the given index. It returns the name of the layer as BSTR.

Use GetLayer Count to get the number of layers in the illustration. (See [GetLayerCount on page 52](#).)

After you have retrieved your preferred layer information it is recommended to call [DisposeLayerList](#) to free the memory used by Arbortext IsoView to keep the layer information for all layers. (See [DisposeLayerList on page 54](#).) This list will be rebuilt after the next call to [GetLayerName](#).

Introduced with Arbortext IsoView 2.0.

### Syntax

```
GetLayerName (index)
```

Parameters:

<b>index</b>	Defines the index of the layer as long (0 based).
--------------	---

### Example

```
for ( n = 0; n < document.ivx1.GetLayerCount();  
n++ )  
{  
    form1.edtName.value =  
    document.ivx1.GetLayerName (n);  
}
```

---

```
document.ivx1.DisposeLayerList();
```

## GetLayerVisibility

This method retrieves the visibility of the layer with the given name. It returns `true` if layer if visible, `false` if not.

Use `SetLayerVisibility` to modify the visibility of a layer. (See [SetLayerVisibility on page 54](#)) Use `GetLayerName` to get the name of an layer. (See [GetLayerName on page 52](#).)

Introduced with Arbortext IsoView 2.0.

### Syntax

```
GetLayerVisibility(Name)
```

Parameters:

<b>Name</b>	Defines the name of the layer retrieve visibility from as BSTR.
-------------	---

### Example

```
form1.edtName.value = document.ivx1.GetLayerVisibility("Layer 1");
```

## LayerHasObjects

Use this method to retrieve if the given layer has objects. It returns `true` if the given layer contains objects. (See [Object Functions on page 79](#).)

Introduced with Arbortext IsoView 2.0.

### Syntax

```
LayerHasObjects(Name)
```

Parameters:

<b>Name</b>	Defines the name of the layer retrieve visibility from as BSTR.
-------------	---

### Example

```
form1.edtName.value = document.ivx1.LayerHasObjects ("Layer 1");
```

---

## SetLayerVisibility

Use this method to modify the visibility of a layer in the illustration. It returns `true` if successful; otherwise `false`.

Introduced with Arbortext IsoView 2.0.

### Syntax

```
SetLayerVisibility( Name, IsVisible )
```

Parameters:

<b>Name</b>	Defines the name of the layer as BSTR.
<b>IsVisible</b>	Defines if the layer should be visible. Select <code>true</code> for the <i>IsVisible</i> parameter, <code>false</code> otherwise

### Example

```
document.ivx1.SetLayerVisibility( "Layer 1", true );
```

## DisposeLayerList

This method disposes of the layer list. It returns `true` if successful; otherwise `false`.

The layer list is generated by calls to [GetLayerName](#). (See [GetLayerName on page 52](#).) Once you have retrieved your preferred layer information it is recommended to call `DisposeLayerList` to free the used memory.

Introduced with Arbortext IsoView 2.0.

### Note

*This method is obsolete since version 3.0.*

### Syntax

```
DisposeLayerList()
```

### Example

```
document.ivx1.DisposeLayerList();
```

# 8

## Context Menu Functions

CreateContextMenu .....	56
AddContextMenuItem .....	56
EnableContextMenuItem.....	57
DoPopupMenu .....	57
DisposeContextMenu .....	58

---

## CreateContextMenu

This method activates a context menu which has been created previously using AddContextMenuItem. (See [AddContextMenuItem on page 56](#).) From then on, this menu is opened and evaluated upon a right mouse button click. The menu item selected by the user is reported back by ContextMenuItemHit. (See [ContextMenuItemHit on page 122](#).)

The method returns the number of menu items as short.

Introduced with Arbortext IsoView 1.0.

### Syntax

```
CreateContextMenu()
```

### Example

```
// see a more complete example at the description of  
// AddContextMenuItem  
// document.ivx1.CreateContextMenu();
```

## AddContextMenuItem

This method adds a new context menu item to the list and returns the value as short.

A context menu created in this way is enabled by the CreateContextMenu method. (See [CreateContextMenu on page 56](#).) No additional menu items can be added to an existing context menu. It has to be disposed of first by DisposeContextMenu. (See [DisposeContextMenu on page 58](#).)

Introduced with Arbortext IsoView 1.0.

### Syntax

```
addContextMenuItem(ipszTitle, bEnabled)
```

Parameters:

<b>ipszTitle</b>	Sets the title (text) of the menu item as BSTR.
<b>bEnabled</b>	Defines if the menu item is enabled as boolean.

A menu item containing only a single hyphen ("") instructs Arbortext IsoView to insert a separator line at this position. A menu item prefixed by an underline ("\_") instructs Arbortext IsoView to insert this item as a sub menu item.

### Example

```
document.ivx1.AddContextMenuItem( "Highlight", true );
```

---

```
document.ivx1.AddContextMenuItem( "-", true );
document.ivx1.AddContextMenuItem( "ViewPorts", true );
document.ivx1.AddContextMenuItem( "_Home", true );
document.ivx1.AddContextMenuItem( "_Top", true );
document.ivx1.CreateContextMenu();
```

## EnableContextMenuItem

This method enables or disables menu items added by AddContextMenuItem. (See [AddContextMenuItem on page 56](#).) This is also possible after the menu has been created by CreateContextMenu. (See [CreateContextMenu on page 56](#).) This method returns `true` if successful; otherwise `false`.

Introduced with Arbortext IsoView 1.0.

### Syntax

```
EnableContextMenuItem( item, enabled )
```

Parameters:

<b>item</b>	Sets the number of the menu item (base = 1) as short.
<b>enabled</b>	Defines if the menu item is enabled as boolean.

### Example

```
document.ivx1.EnableContextMenuItem( 2, true );
```

## DoPopupMenu

This method opens an existing context menu at the specified position. It returns `true` if successful; otherwise `false`.

You can set the context menu items using CreateContextMenu and AddContextMenuItem. (See [CreateContextMenu on page 56](#) and [AddContextMenuItem on page 56](#).)

Introduced with Arbortext IsoView 2.0.

### Syntax

```
DoPopupMenu( x, y )
```

Parameters:

---

x	Defines the horizontal position for the upper left menu corner in pixels as long.
y	Defines the vertical position for the upper left menu corner in pixels as long.

### Example

```
document.ivx1.DoPopupMenu( 211, 102 );
```

## DisposeContextMenu

This method disposes a context menu. All menu items previously inserted by AddContextMenuItem will be removed. (See [AddContextMenuItem on page 56.](#))

Introduced with Arbortext IsoView 1.0.

### Syntax

```
DisposeContextMenu()
```

### Example

```
document.ivx1.DisposeContextMenu();
```

# 9

## Rotation Functions

Iso3GetRotAngle .....	60
Iso3SetRotAngle .....	60

---

## Iso3GetRotAngle

Gets the rotation angle of the illustration. This angle will rotate all illustrations when they have been opened. Use the rotation angle to rotate illustrations for display.

This method returns the rotation angle in degrees as double.

Introduced with Arbortext IsoView 3.0

### Syntax

```
Iso3GetRotAngle()
```

### Example

```
rotAngle = document.ivx1.Iso3GetRotAngle();
```

## Iso3SetRotAngle

Sets the rotation angle of the illustration. This angle will rotate all illustrations when they have been opened. Use the rotation angle to rotate illustrations for display.

Introduced with Arbortext IsoView 3.0

### Syntax

```
Iso3SetRotAngle(newValue)
```

Parameters:

<b>newValue</b>	Defines the rotation angle in degrees as double.
-----------------	--

### Example

```
document.ivx1.Iso3SetRotAngle(90);

//To apply die rotation angle to the current
//illustration you have the close the file
//and to reopen it:
document.ivx1.Iso3SetRotAngle(90);
document.ivx1.CloseFile();
document.ivx1.Iso3OpenFile("sample.iso");
```

The rotation angle will be applied to the next illustration that will be opened.

### Note

*This method should not be used to rotate animated or 3D objects.*

# 10

## Animation Functions

StartAnimation .....	62
StopAnimations .....	62
ConvertAnimationMetadata .....	63
HasAnimations .....	63
CountAnimations .....	64
GetAnimationObjectID .....	64
GetAnimationSequenceName .....	65
GetAnimationFlags .....	66
CountObjectAnimations .....	67
GetObjectAnimationSequenceName .....	67
GetObjectAnimationFlags .....	68

---

# StartAnimation

Starts a named animation sequence of an object.

## Note

*Only animations which are defined as “remote event” animations in Arbortext IsoDraw should be started with this function.*

Introduced `ISo7StartAnimation` with Arbortext IsoView 7.0; replaced by `StartAnimation` in Arbortext IsoView 7.1. `ISo7StartAnimation` is deprecated and supported for backward compatibility.

## Syntax

`StartAnimation (object, sequence, offset)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
<b>sequence</b>	Defines the name of the sequence which should be started (or stopped) as string. By using the keyword <code>\$STOP</code> instead of the name of a sequence, you can stop all running animations of the defined object.
<b>offset</b>	Defines the offset in seconds after which the animation will be started.

## Example

```
// starts SEQ_1 on object with ID "Z_01"  
result = ivx1.StartAnimation ("Z_01", "SEQ_1", 0);  
  
// starts SEQ_1 on object with ID "Z_02"  
result = ivx1.StartAnimation ("id(Z_02)", "SEQ_1", 0);  
  
// starts SEQ_1 on all objects with name "blue"  
result = ivx1.StartAnimation ("name(blue)", "SEQ_1", 0);  
  
// stops all animations on object with ID "Z_01"  
result = ivx1.StartAnimation ("Z_01", "$STOP", 0);  
  
// stops all animations on objects with name "blue"  
result = ivx1.StartAnimation ("name(blue)", "$STOP", 0);
```

# StopAnimations

Stops all running animations and optionally restores them to the state before animation.

---

Introduced as `Iso7StopAnimations` with Arbortext IsoView 7.0; replaced by `StopAnimations` in Arbortext IsoView 7.1. `Iso7StopAnimations` is deprecated and supported for backward compatibility.

### Syntax

`StopAnimations (bRestore)`

Parameters:

<b>bRestore</b>	Defines if the animation should be restored to the state before starting it as Boolean.
-----------------	---

### Example

```
ivx1.StopAnimations (false)
```

## ConvertAnimationMetadata

Converts animation metadata added using DOM calls into ISO animation format (required before running the animation).

Introduced as `Iso7ConvertAnimationMetadata` with Arbortext IsoView 7.0; replaced by `ConvertAnimationMetadata` in Arbortext IsoView 7.1. `Iso7ConvertAnimationMetadata` is deprecated and supported for backward compatibility.

### Syntax

`ConvertAnimationMetadata ()`

### Example

```
ivx1.ConvertAnimationMetadata ();
```

## HasAnimations

Determines if the illustration contains one or more animation sequences of the specified type. It returns the search result as Boolean: `true` if found; `false` if not.

Introduced with Arbortext IsoView 7.1.

### Syntax

`HasAnimations (sequence_type)`

Parameters:

---

<b>sequence_type</b>	Specifies the type of animation sequence to search for. Allowed values are:
1	<b>remote event</b> sequence
2	<b>on load</b> sequence
4	<b>disabled</b> sequence

### Example

```
// look for any kind of animations in the document:  
bAnyAnimations = document.ivx1.HasAnimations( 1+2+4 );
```

## CountAnimations

Counts all animation sequences in the current illustration. It returns the total number of sequences as Long. All types of animation sequences are counted, including **remote event**, **on load**, and **disabled**.

Introduced with Arbortext IsoView 7.1.

### Note

*To retrieve further information about animations in the current illustration, you can use GetAnimationObjectID, GetAnimationSequenceName and GetAnimationFlags. (See [GetAnimationObjectID on page 64](#), [GetAnimationSequenceName on page 65](#), and [GetAnimationFlags on page 66](#).)*

### Syntax

```
CountAnimations()
```

### Example

```
n = document.ivx1.CountAnimations();
```

## GetAnimationObjectID

Gets information about an enumerated animation sequence. It returns a string, objectID; the ID of the object on which this sequence acts.

---

After an illustration loads, the animation sequences it contains are in an unsorted order. This order will not change unless it is explicitly forced to by another method such as `ConvertAnimationMetadata`. (See [ConvertAnimationMetadata on page 63](#).)

### Note

The results of `GetAnimationObjectID`, `GetAnimationSequenceName` and `GetAnimationFlags` are only aligned if you use the same index. (See [GetAnimationObjectID on page 64](#), [GetAnimationSequenceName on page 65](#), and [GetAnimationFlags on page 66](#).)

Introduced with Arbortext IsoView 7.1.

### Syntax

`GetAnimationObjectID (index)`

Parameters:

<b>index</b>	Picks the sequence of interest. The value can be 0 or any positive number less than the result of <code>CountAnimations</code> . (See <a href="#">CountAnimations on page 64</a> .)
--------------	---

### Example

See the example for [GetAnimationFlags on page 66](#).

## GetAnimationSequenceName

Gets information about an enumerated animation sequence. It returns a string, `seqName`; the name of this sequence.

After an illustration loads, the animation sequences it contains are in an unsorted order. This order will not change unless it is explicitly forced to by another method such as `ConvertAnimationMetadata`. (See [ConvertAnimationMetadata on page 63](#).)

### Note

The results of `GetAnimationObjectID`, `GetAnimationSequenceName` and `GetAnimationFlags` are only aligned if you use the same index. (See [GetAnimationObjectID on page 64](#), [GetAnimationSequenceName on page 65](#), and [GetAnimationFlags on page 66](#).)

Introduced with Arbortext IsoView 7.1.

---

## Syntax

GetAnimationSequenceName (*index*)

Parameters:

<b>index</b>	Picks the sequence of interest. The value can be 0 or any positive number less than the result of CountAnimations. (See <a href="#">CountAnimations on page 64</a> .)
--------------	---

## Example

See the example for [GetAnimationFlags on page 66](#).

# GetAnimationFlags

Gets information about an enumerated animation sequence. Possible return values are:

1	<b>remote event</b> sequence
2	<b>on load</b> sequence
4	<b>disabled</b> sequence

After an illustration loads, the animation sequences it contains are in an unsorted order. This order will not change unless it is explicitly forced to by another method such as ConvertAnimationMetadata. (See [ConvertAnimationMetadata on page 63](#).)

## Note

The results of GetAnimationObjectID, GetAnimationSequenceName and GetAnimationFlags are only aligned if you use the same index. (See [GetAnimationObjectID on page 64](#), [GetAnimationSequenceName on page 65](#), and [GetAnimationFlags on page 66](#).)

Introduced with Arbortext IsoView 7.1.

## Syntax

GetAnimationFlags (*index*)

Parameters:

<b>index</b>	Picks the sequence of interest. The value can be 0 or any positive number less than the result of CountAnimations. (See <a href="#">CountAnimations on page 64</a> .)
--------------	---

---

## Example

```
// collect all remote-startable animations in a HTML form selection list
for ( i=0,j=0; i<document.ivx1.CountAnimations(); i++ )
{
    objID = document.ivx1.GetAnimationObjectID( i );
    seqName = document.ivx1.GetAnimationSequenceName( i );
    flags = document.ivx1.GetAnimationFlags( i );
    caption = objID + " ~> " + seqName
    if ( flags=="1" )
    {
        document.Form1.Animation.options[ j++ ] = new Option( caption );
    }
}
```

## CountObjectAnimations

Counts all animation sequences on the specified object. It returns the total number of all sequences on the specified object as Long. All types of animation sequences are counted, including **remote event**, **on load**, and **disabled**.

Introduced with Arbortext IsoView 7.1.

### Syntax

CountObjectAnimations (*objID*)

Parameters:

<b>objID</b>	The ID of the object this method acts upon.
--------------	---

## Example

```
n = document.ivx1.CountObjectAnimations( 'id(myObj)' );
```

## GetObjectAnimationSequenceName

Gets information about an enumerated animation sequence of an object. It returns a string, *seqName*; the name of this sequence.

---

After an illustration loads, the animation sequences it contains are in an unsorted order. This order will not change unless it is explicitly forced to by another method such as `ConvertAnimationMetadata`. (See [ConvertAnimationMetadata on page 63](#).)

### Note

*The results of `GetObjectAnimationSequenceName` and `GetObjectAnimationFlags` are only aligned if you use the same index.* (See [GetObjectAnimationSequenceName on page 67](#) and [GetObjectAnimationFlags on page 68](#).)

Introduced with Arbortext IsoView 7.1.

### Syntax

`GetObjectAnimationSequenceName (objID, index)`

Parameters:

<b>objID</b>	The ID of the object this method acts upon.
<b>index</b>	Picks the sequence of interest. The value can be 0 or any positive number less than the result of <code>CountObjectAnimations</code> . (See <a href="#">CountObjectAnimations on page 67</a> .)

### Example

See the example for [GetObjectAnimationFlags on page 68](#).

## GetObjectAnimationFlags

Gets information about an enumerated animation sequence of an object. Possible return values are:

1	<b>remote event</b> sequence
2	<b>on load</b> sequence
4	<b>disabled</b> sequence

---

After an illustration loads, the animation sequences it contains are in an unsorted order. This order will not change unless it is explicitly forced to by another method such as `ConvertAnimationMetadata`. (See [ConvertAnimationMetadata on page 63](#).)

### Note

The results of `GetObjectAnimationSequenceName` and `GetObjectAnimationFlags` are only aligned if you use the same index. (See [GetObjectAnimationSequenceName on page 67](#) and [GetObjectAnimationFlags on page 68](#).)

Introduced with Arbortext IsoView 7.1.

### Syntax

`GetObjectAnimationFlags (objID, index)`

Parameters:

<b>objID</b>	The ID of the object this method acts upon.
<b>index</b>	Picks the sequence of interest. The value can be 0 or any positive number less than the result of <code>CountAnimations</code> . (See <a href="#">CountAnimations on page 64</a> .)

### Example

```
strObject = "id(" + object + ")";
strInfo = "Animations of object " + strID + ":\n";
n = ivx1.CountObjectAnimations(strID);
for ( i = 0; i < n; i++ )
{
    seqName = ivx1.GetObjectAnimationSequenceName( strObject, i );
    flags = ivx1.GetObjectAnimationFlags( strObject, i );
    flags = document.ivx1.GetAnimationFlags( i );
    strInfo = strInfo + "Sequence: " + seqName + ", flags =" + flags + ".\n";
}
if ( n > 0 )
    window.alert( strInfo );
else
    window.alert( "Object does not have animations." );
```



# 11

## Magnifier Functions

Iso4ConfigMagnifier.....	72
Iso4OpenMagnifier .....	72
Iso4CloseMagnifier.....	73
Iso4GetMagnifierState .....	73
Iso4GetMagnifierScale.....	73
Iso4SetMagnifierScale .....	74

---

## Iso4ConfigMagnifier

Configures the display and position of the **Magnifier** window. This method returns true if successful; otherwise false.

Introduced with Arbortext IsoView 4.0.

### Syntax

```
Iso4ConfigMagnifier(styleFlags, width, height, scale, minScale,  
maxScale)
```

Parameters:

<b>styleFlags</b>	Defines the magnifier style as long. Allowed values are:  <table border="1"><tr><td>1</td><td>docked</td></tr><tr><td>2</td><td>show slider</td></tr><tr><td>4</td><td>show in b/w</td></tr><tr><td>256</td><td>top left</td></tr><tr><td>512</td><td>top right</td></tr><tr><td>768</td><td>bottom left</td></tr><tr><td>1024</td><td>bottom right</td></tr></table> <b>Note</b> <i>Add values to combine different attributes.</i>	1	docked	2	show slider	4	show in b/w	256	top left	512	top right	768	bottom left	1024	bottom right
1	docked														
2	show slider														
4	show in b/w														
256	top left														
512	top right														
768	bottom left														
1024	bottom right														
<b>width</b>	Defines the width of the <b>Magnifier</b> window in pixels as long.														
<b>height</b>	Defines the height of the <b>Magnifier</b> window in pixels as long.														
<b>scale</b>	Defines the initial scale used in the <b>Magnifier</b> window as double.														
<b>minScale</b>	Defines the minimum scale (used for slider) as double.														
<b>maxScale</b>	Defines the maximum scale (used for slider) as double.														

### Example

```
document.ivx1.Iso4ConfigGlobalView(257,100,100,400,200,1200);
```

## Iso4OpenMagnifier

Opens the **Magnifier** window. (See [Iso4ConfigMagnifier on page 72](#) for configuration options.)

This method returns true if successful; otherwise false.

Introduced with Arbortext IsoView 4.0.

---

## Syntax

```
Iso4OpenMagnifier()
```

## Example

```
document.ivx1.Iso4OpenMagnifier();
```

# Iso4CloseMagnifier

Closes the **Magnifier** window.

Introduced with Arbortext IsoView 4.0.

## Syntax

```
Iso4CloseMagnifier()
```

## Example

```
document.ivx1.Iso4CloseMagnifier();
```

# Iso4GetMagnifierState

Retrieves the current state of the **Magnifier** window.

The method returns a nonzero value indicating the settings of the open **Magnifier** window or zero indicating the window is not open.

Introduced with Arbortext IsoView 4.0.

## Syntax

```
Iso4GetMagnifierState()
```

## Example

```
fState = document.ivx1.Iso4GetMagnifierState();
```

# Iso4GetMagnifierScale

Retrieves the current scale of the **Magnifier** window.

This method returns the current scale of the magnifier window as double.

Introduced with Arbortext IsoView 4.0.

## Syntax

```
Iso4GetMagnifierScale()
```

---

## Example

```
fScale = document.ivx1.Iso4GetMagnifierScale();
```

## Iso4SetMagnifierScale

Sets the scale of **Magnifier** window.

This method returns `true` if successful; otherwise `false`.

Introduced with Arbortext IsoView 4.0.

### Syntax

```
Iso4SetMagnifierScale(Scale)
```

Parameters:

<b>Scale</b>	Defines the scale (in percent) of the <b>Magnifier</b> window as double. Any <i>Scale</i> value in the [1 to 6400] range is allowed.
--------------	--

## Example

```
// scale magnifier window 206%:  
document.ivx1.Iso4SetMagnifierScale(206);
```

# 12

## GlobalView Functions

Iso4ConfigGlobalView.....	76
Iso4OpenGlobalView .....	76
Iso4CloseGlobalView.....	77
Iso4GetGlobalViewState .....	77

---

## Iso4ConfigGlobalView

Configures the display and position of the **Global View** window. This method returns `true` if successful; otherwise `false`.

Introduced with Arbortext IsoView 4.0.

### Syntax

```
Iso4ConfigGlobalView(styleFlags, width, height, useStroke, strokeWidth,  
color)
```

Parameters:

<b>styleFlags</b>	Defines the <b>Global View</b> window style as long. Allowed values are:  <table border="1"><tr><td>1</td><td>docked</td></tr><tr><td>4</td><td>show in b/w</td></tr><tr><td>256</td><td>top left</td></tr><tr><td>512</td><td>top right</td></tr><tr><td>768</td><td>bottom left</td></tr><tr><td>1024</td><td>bottom right</td></tr></table> <b>Note</b> <i>Add values to combine different attributes.</i>	1	docked	4	show in b/w	256	top left	512	top right	768	bottom left	1024	bottom right
1	docked												
4	show in b/w												
256	top left												
512	top right												
768	bottom left												
1024	bottom right												
<b>width</b>	Defines the width of the <b>Global View</b> window in pixels as long.												
<b>height</b>	Defines the height of the <b>Global View</b> window in pixels as long.												
<b>useStroke</b>	Is TRUE if stroke rect indicates current view position, otherwise a filled rect is used.												
<b>strokeWidth</b>	Defines the line width of the stroke rect (if chosen) as long.												
<b>color</b>	Defines the color of the stroke/fill rect as OLE_COLOR.												

### Example

```
document.ivx1.Iso4ConfigGlobalView(257,100,100,TRUE,2)
```

## Iso4OpenGlobalView

Opens the **Global View** window. This method returns `true` if successful; otherwise `false`. (See [Iso4ConfigGlobalView on page 76](#) for configuration options.)

Introduced with Arbortext IsoView 4.0.

---

## Syntax

```
Iso4OpenGlobalView()
```

## Example

```
document.ivx1.Iso4OpenGlobalView();
```

# Iso4CloseGlobalView

Closes the **Global View** window.

Introduced with Arbortext IsoView 4.0.

## Syntax

```
Iso4CloseGlobalView()
```

## Example

```
document.ivx1.Iso4CloseGlobalView();
```

# Iso4GetGlobalViewState

Retrieves the current state of the **Global View** window. This method returns a value indicating the settings of the **Global View** window as long. (See [Iso4ConfigGlobalView on page 76](#).)

Introduced with Arbortext IsoView 4.0.

## Syntax

```
Iso4GetGlobalViewState()
```

## Example

```
lState = document.ivx1.Iso4GetGlobalViewState();
```



# 13

## Object Functions

GetObjectCount .....	80
Iso3GetObjectCount .....	80
Iso3ShowObjects .....	80
GetObjectID .....	81
Iso3GetObjectID.....	82
Iso3GetObjectIDByIndex.....	82
GetObjectName .....	83
Iso3GetObjectName .....	84
Iso3GetObjectByNameByIndex .....	84
GetObjectDataString.....	85
GetObjectFlags .....	85
GetObjectType .....	86
Iso3GetObjectLayer.....	87
HighlightHotspot.....	87
Iso3HighlightObject .....	88
DisposeObjectList .....	89

---

# GetObjectCount

## Note

*This method is deprecated and supported for backward compatibility. Use Iso3GetObjectCount instead. (See [Iso3GetObjectCount on page 80](#).)*

Use this property to retrieve the number of objects in the current illustration. To retrieve further object information you can use see GetObjectID, GetObjectName, GetObjectDataString, and GetObjectTip.

This method returns the number of objects as long.

Introduced with Arbortext IsoView 2.0.

See:

- [GetObjectID on page 81](#)
- [GetObjectName on page 83](#)
- [GetObjectTip on page 92](#)

## Syntax

```
GetObjectCount ()
```

## Example

```
n = document.ivx1.GetObjectCount();
```

# Iso3GetObjectCount

Gets the number of objects in an illustration and returns the number as long.

Introduced with Arbortext IsoView 3.0.

## Syntax

```
Iso3GetObjectCount ()
```

## Example

```
objCount = document.ivx1.Iso3GetObjectCount();
```

# Iso3ShowObjects

Sets the kind of objects to be shown.

Introduced with Arbortext IsoView 3.0.

---

## Syntax

`Iso3ShowObjects (bShowObjects, bShowHotSpot)`

Parameters:

<b>bShowObjects</b>	Set this parameter to <code>true</code> if objects without hotspots shall be highlighted.
<b>bShowHotSpot</b>	Set this parameter to <code>true</code> if objects with hotspots shall be highlighted.

## Example

```
document.ivx1.Iso3ShowObjects(1,1);
```

# GetObjectID

### Note

*This method is deprecated and supported for backward compatibility. Use [Iso3GetObjectID on page 82](#) instead.*

This method retrieves the ID of the object with the given index. It returns the ID of the object as long.

Use `GetObjectName` to get the name of an object. With `GetObjectCount` you get the number of objects in the illustration. After you have retrieved your preferred object information, it is recommended to call `DisposeObjectList` to free the memory used by Arbortext IsoView to keep the object information of all objects. This list will be rebuilt after the next call to `GetObjectID` or `GetObjectName`.

Introduced with Arbortext IsoView 2.0.

See:

- [GetObjectName on page 83](#)
- [GetObjectCount on page 80](#)
- [DisposeObjectList on page 89](#)
- [GetObjectID on page 81](#)
- [GetObjectName on page 83.](#)

## Syntax

`GetObjectID (index)`

Parameters:

---

<b>index</b>	Defines the index of the object as long (0 based).
--------------	--

### Example

```
var ObjID = new Array();
var ObjName = new Array();

for ( n = 0; n < document.ivx1.GetObjectCount();
n++ )
{
    ObjsID[ n ] = document.ivx1.GetObjectID(n);
    ObjName[ n ] =
document.ivx1.GetObjectName(n);
}

document.ivx1.DisposeObjectList();
```

## Iso3GetObjectID

Returns ID of an object.

This method returns the ID of an object as BSTR. It returns the plain ID of the object, not a complete object identifier including the "id(...)" part.

Introduced with Arbortext IsoView 3.0.

### Syntax

`Iso3GetObjectID (object)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
---------------	--

### Example

```
strID =
document.ivx1.Iso3GetObjectID("name(myName1)");
```

## Iso3GetObjectIDByIndex

Returns ID of an object.

This method returns the ID of the indexed object as BSTR. It returns the plain ID of the object, not a complete object identifier including the "id(...)" part.

Introduced with Arbortext IsoView 3.0.

---

## Syntax

Iso3GetObjectIDByIndex (*index*)

Parameters:

<b>index</b>	Defines the index of the object as long (0 based).
--------------	--

## Example

```
strID =  
document.ivx1.Iso3GetObjectIDByIndex(n);
```

# GetObjectName

### Note

*This method is deprecated and supported for backward compatibility. Use Iso3GetObjectName instead. (See [Iso3GetObjectName on page 84](#).)*

This method retrieves the name of the object with the given index. It returns the name of the object as BSTR.

Use GetObjectID to get the ID of an object. With GetObjectCount you get the number of objects in the illustration. After you have retrieved your preferred object information, it is recommended to call DisposeObjectList to free the memory used by Arbortext IsoView to keep the object information of all objects. This list will be rebuilt after the next call to GetObjectName or GetObjectID.

Introduced with Arbortext IsoView 2.0.

See:

- [GetObjectName on page 83](#)
- [GetObjectCount on page 80](#)
- [DisposeObjectList on page 89](#)
- [GetObjectID on page 81](#)
- [GetObjectName on page 83](#).

## Syntax

GetObjectName (*index*)

Parameters:

<b>index</b>	Defines the index of the object as long (0 based).
--------------	--

---

## Example

```
var ObjID = new Array();
var ObjName = new Array();
for ( n = 0; n < document.ivx1.GetObjectCount();
n++ )
{
    ObjsID[ n ] = document.ivx1.GetObjectID(n);
    ObjName[ n ] =
document.ivx1.GetObjectName(n);
}

document.ivx1.DisposeObjectList();
```

## Iso3GetObjectName

Returns name of an object as BSTR. It returns the plain name of the object, not a complete object identifier including the "name(...)" part.

Introduced with Arbortext IsoView 3.0.

### Syntax

Iso3GetObjectName (*object*)

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
---------------	--

## Example

```
strName = document.ivx1.Iso3GetObjectName(n);
```

## Iso3GetObjectNameByIndex

Returns the name of an indexed object as BSTR. It returns the plain ID of the object, not a complete object identifier including the "name(...)" part.

Introduced with Arbortext IsoView 3.0.

### Syntax

Iso3GetObjectNameByIndex (*index*)

Parameters:

<b>index</b>	Defines the index of the object as long (0 based).
--------------	--

---

## Example

```
strName =  
document.ivx1.Iso3GetObjectByNameByIndex(n);
```

## GetObjectDataString

This method retrieves the data property of the specified object and returns it as a string (BSTR).

Introduced with Arbortext IsoView 2.0.

### Syntax

GetObjectDataString (*ID, Name*)

Parameters:

<b>ID</b>	Defines the id of the object as long.
<b>Name</b>	Defines the name of the object as BSTR.

## Example

```
edtData.value =  
document.ivx1.GetObjectDataString( id, name );
```

## GetObjectFlags

This method is used to determine if an object has animations or hotspots. It is also used to determine if an object is 3D.

Introduced with Arbortext IsoView 7.1.

### Syntax

GetObjectFlags (*objID*)

Parameters:

<b>objID</b>	Defines the ID of the object as BSTR
--------------	--------------------------------------

Return Values:

This method returns the values below in a bit-mapped array:

0x0001	The object has hotspots.
0x0002	The object is 3D.

---

0x0004	<p>The object has animations of any type.</p> <p><b>Note</b></p> <p><i>To determine if a object contains a specific type of animation, use the <a href="#">HasAnimations</a> method. (See <a href="#">HasAnimations</a> on page 63.)</i></p>
--------	--

### Example

```
flags = GetObjectFlags( 'id(obj_1)' );
```

## GetObjectType

This method retrieves the object type.

Introduced with Arbortext IsoView 7.1.

### Syntax

```
GetObjectType( objID )
```

Parameters:

<b>objID</b>	Defines the ID of the object as BSTR
--------------	--------------------------------------

Return Values:

This method returns a string that identifies the object type. Some examples are listed below. The object types available depend on the current DTD (Document Type Definition).

<b>String</b>	<b>Object Type</b>
grobject	Graphic object
gmode	Group
layer	Layer
para	Paragraph
subpara	Subparagraph (fragment of text within para object)

### Example

```
strType = GetObjectType( 'id(obj_1)' );
```

---

## Iso3GetObjectLayer

Returns the layer name of this object as BSTR. (See [Layer Functions on page 51.](#))

Introduced with Arbortext IsoView 3.0.

### Syntax

`Iso3GetObjectLayer (object)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
---------------	--

### Example

```
layerName =  
document.ivx1.Iso3GetObjectLayer("id(myObj1)");
```

## HighlightHotspot

This method highlights the specified hotspot (if it is visible in the current window).

In case of success the method returns true, otherwise false.

Introduced with Arbortext IsoView 1.0.

### Syntax

`HighlightHotspot (IID, lpszName, IFlags, oleColor)`

Parameters:

<b>IID</b>	Defines the ID of the hotspot (ID field in Arbortext IsoDraw object info, usage not recommended) as long.
<b>lpszName</b>	Defines the name of the hotspot as string.

---

<b>IFlags</b>	Defines the display attributes as long. Allowed values are:
<b>0</b>	HIDE
<b>2</b>	FRAME
<b>4</b>	FILL
<b>8</b>	FLASH
<b>16</b>	CENTER (not yet supported)
	To process ALL hotspots use ID = 32.
	<b>Note</b> <i>Add values to combine attributes.</i>
<b>oleColor</b>	Defines the display color (RGB color as 32-bit unsigned integer) as OLE_Color. This value can be calculated as follows: $\text{color} = r + g * 256 + b * 65536$ , with r, g, and b indicating the colors red, green and blue in a range between 0 and 255.

## Example

```
document.ivx1.HighlightHotspot( id, name, 8,
RGB(128,128,0) );

<script language="JavaScript">
function RGB( r, g, b )
{
    return ( r + g*256 + b*65536 );
}
</script>
```

## Iso3HighlightObject

Highlights an object. (See [Object Highlighting Constants – Iso3HighlightObject on page 136](#) for constant values.)

Introduced with Arbortext IsoView 3.0.

### Syntax

`Iso3HighlightObject (objExpr, flags, color)`

Parameters:

---

<b>objExpr</b>	defines the object expression identifying for one or multiple objects as BSTR.								
<b>flags</b>	Defines the display attributes as long. Allowed values are:  <table border="1"> <tr> <td><b>0</b></td><td>HIDE</td></tr> <tr> <td><b>2</b></td><td>FRAME</td></tr> <tr> <td><b>4</b></td><td>FILL</td></tr> <tr> <td><b>8</b></td><td>FLASH</td></tr> </table> <b>Note</b> <i>Add values to combine attributes.</i>	<b>0</b>	HIDE	<b>2</b>	FRAME	<b>4</b>	FILL	<b>8</b>	FLASH
<b>0</b>	HIDE								
<b>2</b>	FRAME								
<b>4</b>	FILL								
<b>8</b>	FLASH								
<b>color</b>	Defines the RGB color as 32-bit unsigned integer. This value can be calculated as follows: $\text{color} = r + g * 256 + b * 65536$ , with r, g and b indicating the colors red, green and blue in a range between 0 and 255.								

## Example

```
document.ivx1.Iso3HighlightObject("id(myObj1)",
8, RGB(128,128,0) );

<script language="JavaScript">
function RGB( r, g, b )
{
    return ( r + g*256 + b*65536 );
}
</script>
```

# DisposeObjectList

### Note

*This method is obsolete since version 3.0.*

This method disposes of the object list. When successful, it returns `true`, otherwise `false`.

The object list is generated by calls to `GetObjectID` / `GetObjectName`. (See [GetObjectID on page 81](#)/[GetObjectName on page 83](#).) Once you have retrieved your preferred object information it is recommended to call `DisposeObjectList` to free the used memory.

Introduced first with Arbortext IsoView 2.0.

### Syntax

`DisposeObjectList()`

---

## **Example**

```
document.ivx1.DisposeObjectList();
```

# 14

## Object Tip

GetObjectTip.....	92
Iso3GetObjectTip .....	92
SetObjectTip .....	93
Iso3SetObjectTip.....	93
GetObjectTipColor.....	94
SetObjectTipColor .....	94
SetObjectTipStyle.....	94
Iso3SetObjectTipStyle .....	95

(See [ObjectTipState](#) on page 118 for constant values.)

---

# GetObjectTip

## Note

*This method is deprecated and supported for backward compatibility. Use Iso3GetObjectTip instead. (See [Iso3GetObjectTip on page 92.](#))*

This method retrieves the text used in object tips for this object. It returns the tip text of the object as BSTR.

Introduced with Arbortext IsoView 2.0.

## Syntax

GetObjectTip (*ID, Name*)

Parameters:

<b>ID</b>	Defines the id of the object as long.
<b>Name</b>	Defines the name of the object as BSTR.

## Example

```
form1.edtTip.value =  
document.ivx1.GetObjectTip( id, name );
```

# Iso3GetObjectTip

Retrieves the object tip for the specified object. It returns the object tip text of the given object as BSTR.

Introduced with Arbortext IsoView 3.0.

## Syntax

Iso3GetObjectTip (*Object*)

Parameters:

<b>Object</b>	Defines the object identifier as BSTR.
---------------	--

## Example

```
strObjTip =  
document.ivx1.Iso3GetObjectTip("id(myObj1)");
```

---

# SetObjectTip

## Note

*This method is deprecated and supported for backward compatibility. Use Iso3SetObjectTip instead. (See [Iso3SetObjectTip on page 93](#).)*

Use this method to add tip information for an object in the illustration. The usage of these tips is controlled by ObjectTipState. (See [ObjectTipState on page 118](#).)

Introduced with Arbortext IsoView 2.0.

## Syntax

SetObjectTip (*ID, Name, Tip*)

Parameters:

<b>ID</b>	Defines the id of the object as long.
<b>Name</b>	Defines the name of the object as BSTR.
<b>Tip</b>	Defines the tip text of the object as BSTR.

## Example

```
document.ivx1.SetObjectTip( 0, "UR-2006", "ring" );
```

# Iso3SetObjectTip

Sets the object tip.

Introduced with Arbortext IsoView 3.0.

## Syntax

Iso3SetObjectTip (*Object, Tip*)

Parameters:

<b>Object</b>	Defines the object identifier as BSTR.
<b>Tip</b>	Defines the tip text of the object as BSTR.

## Example

```
document.ivx1.Iso3SetObjectTip  
("id(myObj1)","Click here to continue");
```

---

## GetObjectTipColor

This method retrieves the color used for the background or the text of object tips. It returns the RGB color as 32 bit unsigned integer.

Introduced with Arbortext IsoView 2.0.

### Syntax

`GetObjectTipColor (GetBackCol)`

Parameters:

<b>GetBackCol</b>	Defines which color is retrieved as BOOL. If <code>true</code> , the background color is retrieved; if <code>false</code> , the text color.
-------------------	---

### Example

```
colTip =  
document.ivx1.GetObjectTipColor(true);
```

## SetObjectTipColor

Use this method to set the color attributes used for object tips. By default the system color for tool tips is used.

The color value can be calculated as follows:  $\text{color} = r + g*256 + b*65536$ , with r, g and b indicating the colors red, green and blue in a range between 0 and 255.

Introduced with Arbortext IsoView 2.0.

### Syntax

`SetObjectTipColor (Background, Text)`

Parameters:

<b>Background</b>	Defines the background color as 32 bit unsigned integer.
<b>Text</b>	Defines the text color as 32 bit unsigned integer.

### Example

```
document.ivx1.SetObjectTipColor( 8454143, 0 );
```

## SetObjectTipStyle

Use this method set the text style used for object tips. The system font for tool tips is used by default. The usage of these tips is controlled by `ObjectTipState`.

---

Introduced with Arbortext IsoView 2.0.

## Syntax

`SetObjectTipStyle (Font, Size, Style, Alignment)`

Parameters:

<b>Font</b>	Defines the display name of the font as BSTR.								
<b>Size</b>	Defines the size of the text (in points) as short.								
<b>Style</b>	Defines the style of the text as short. Allowed values are: <table border="1"><tr><td><b>0</b></td><td>NORMAL</td></tr><tr><td><b>1</b></td><td>BOLD</td></tr><tr><td><b>2</b></td><td>ITALIC</td></tr><tr><td><b>3</b></td><td>BOLDITALIC</td></tr></table>	<b>0</b>	NORMAL	<b>1</b>	BOLD	<b>2</b>	ITALIC	<b>3</b>	BOLDITALIC
<b>0</b>	NORMAL								
<b>1</b>	BOLD								
<b>2</b>	ITALIC								
<b>3</b>	BOLDITALIC								
	<b>Note</b> Add 256 if you want Arbortext IsoView to keep the tip window outside the object's area.								
<b>Alignment</b>	Defines the alignment of the text as short. Allowed values are: <table border="1"><tr><td><b>0</b></td><td>LEFT</td></tr><tr><td><b>1</b></td><td>CENTER</td></tr><tr><td><b>2</b></td><td>RIGHT</td></tr></table>	<b>0</b>	LEFT	<b>1</b>	CENTER	<b>2</b>	RIGHT		
<b>0</b>	LEFT								
<b>1</b>	CENTER								
<b>2</b>	RIGHT								

## Example

```
document.ivx1.SetObjectTipStyle( "Arial", 9, 0, 1 );
```

# Iso3SetObjectTipStyle

Sets the style of the object tips.

Introduced with Arbortext IsoView 3.0.

## Syntax

`Iso3SetObjectTipStyle (Font, Size, Style, Alignment, Delay, ShowTime)`

Parameters:

<b>Font</b>	Defines the Display name of the font as BSTR.
<b>Size</b>	Defines the Size of the text (in points) as short.

---

<b>Style</b>	Defines the Style of the text as short. Allowed values are:
<b>0</b>	NORMAL
<b>1</b>	BOLD
<b>2</b>	ITALIC
<b>3</b>	BOLDITALIC
<b>Note</b>	
Add 256 if you want Arbortext IsoView to keep the tip window outside the object's area.	
<b>Alignment</b>	Defines the alignment of the text as short. Allowed values are:
<b>0</b>	LEFT
<b>1</b>	CENTER
<b>2</b>	RIGHT
<b>Delay</b>	Defines the delay as the number of 0.1-second intervals that elapse before a tip comes up as short.
<b>ShowTime</b>	Defines the time as the number of 0.1-second intervals that elapse before a tip disappears automatically as short.

### Example

```
document.ivx1.Iso3SetObjectTipStyle("Arial",12,0,0,5,10);
```

# 15

## Object Attributes

Iso3GetAttrCount.....	98
Iso3GetAttrByIndex .....	98
Iso3GetAttrType .....	98
Iso3GetAttrValue .....	99
Iso3GetFloatAttrValue .....	99
Iso3SetFloatAttrValue .....	100
Iso3GetIntegerAttrValue .....	100
Iso3SetIntegerAttrValue .....	101
Iso3GetStringAttrValue.....	101
Iso3SetStringAttrValue .....	101
Iso3SetGrAttribute .....	102

---

## Iso3GetAttrCount

Returns the number of attributes of an object as long.

Introduced with Arbortext IsoView 3.0.

### Syntax

`Iso3GetAttrCount (object)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
---------------	--

### Example

```
attrCount =  
document.ivx1.Iso3GetAttrCount ("id(myObj1)");
```

## Iso3GetAttrByIndex

Returns the name of an attribute as BSTR.

You can get the actual count of attributes on an object using `Iso3GetAttrCount`. (See [Iso3GetAttrCount on page 98](#).)

Introduced with Arbortext IsoView 3.0.

### Syntax

`Iso3GetAttrByIndex (object, index)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
<b>index</b>	Defines the index of the object as long (1-based).

### Example

```
attrName =  
document.ivx1.Iso3GetAttrByIndex ("id(myObj1)",1);
```

## Iso3GetAttrType

Returns the type of a specific attribute as long. The following number could be returned: 1 = integer, 2 = float, 3 = string, 4 = undefined, 5 = link.

Introduced with Arbortext IsoView 3.0.

---

## Syntax

`Iso3GetAttrType (object, attrName)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
<b>attrName</b>	Defines the attribute name as BSTR.

## Example

```
attrType =  
document.ivx1.Iso3GetAttrType("id(myObj1)", "PartNo");
```

## Iso3GetAttrValue

Returns the value of an attribute as a string (BSTR). The string is empty if the attribute doesn't exist.

Introduced with Arbortext IsoView 3.0.

## Syntax

`Iso3GetAttrValue (object, attrName)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
<b>attrName</b>	Defines the attribute name as BSTR.

## Example

```
attrValue = document.ivx1.Iso3GetAttrValue("id(myObj1)", "PartNo");
```

## Iso3GetFloatAttrValue

Returns the value of a floating number attribute.

Introduced with Arbortext IsoView 3.0.

## Syntax

`Iso3GetFloatAttrValue (object, attrName)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
<b>attrName</b>	Defines the attribute name as BSTR.

---

## Example

```
floatValue =  
document.ivx1.Iso3GetFloatAttrValue("id(myObj1)","Diameter");
```

## Iso3SetFloatAttrValue

Set the value of a floating number attribute. In case of success the method returns true, otherwise false. If the attribute does not exist it will be added to the object.

Introduced with Arbortext IsoView 3.0.

### Syntax

```
Iso3SetFloatAttrValue (object, attrName, value)
```

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
<b>attrName</b>	Defines the attribute name as BSTR.
<b>value</b>	Defines the new float value.

## Example

```
document.ivx1.Iso3SetFloatAttrValue("id(myObj1)","Diameter",0.5);
```

## Iso3GetIntegerAttrValue

Returns the value of an integer attribute as long.

Introduced with Arbortext IsoView 3.0.

### Syntax

```
Iso3GetIntegerAttrValue (object, attrName)
```

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
<b>attrName</b>	Defines the attribute name as BSTR.

## Example

```
intValue =  
document.ivx1.Iso3GetIntegerAttrValue("id(myObj1)","Count");
```

---

## Iso3SetIntegerAttrValue

Set the value of an integer attribute. If successful, the method returns `true`, otherwise `false`. If the attribute does not exist it will be added to the object.

Introduced with Arbortext IsoView 3.0.

### Syntax

```
Iso3SetIntegerAttrValue (object, attrName, value)
```

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
<b>attrName</b>	Defines the attribute name as BSTR.
<b>value</b>	Defines the new longinteger value.

### Example

```
document.ivx1.Iso3SetIntegerAttrValue("id(myObj1)", "Count", 5);
```

## Iso3GetStringAttrValue

Returns the value of a string attribute as BSTR.

Introduced with Arbortext IsoView 3.0.

### Syntax

```
Iso3GetStringAttrValue (object, attrName)
```

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
<b>attrName</b>	Defines the attribute name as BSTR.

### Example

```
strValue = document.ivx1.Iso3GetStringAttrValue("id(myObj1)", "Description");
```

## Iso3SetStringAttrValue

Sets the value of a string attribute. It returns `true` if successful; otherwise `false`. If the attribute does not exist it will be added to the object.

Introduced with Arbortext IsoView 3.0.

---

## Syntax

`Iso3SetStringValue (object, attrName, value)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
<b>attrName</b>	Defines the attribute name as BSTR.
<b>value</b>	Defines the new String value as BSTR.

## Example

```
document.ivx1.Iso3SetStringValue("id(myObj1)","Description","Bracket");
```

## Iso3SetGrAttribute

Sets a graphic attribute for an object and enables you to change the appearance of an object and its children on screen. The content of the graphics file is not changed.

Introduced with Arbortext IsoView 3.0.

## Syntax

`Iso3SetGRAttribute (objExpr, attrName, value)`

Parameters:

---

<b>objExpr</b>	Defines the object expression identifying for one or multiple objects as BSTR.
<b>attrName</b>	Defines the attribute name as BSTR. There are several predefined attribute names:
<b>stroke-width</b>	line weight of stroked elements
<b>stroke</b>	stroke color of stroked elements
<b>fill</b>	fill color of filled elements
<b>text-color</b>	text color
<b>color</b>	sets stroke, fill, and text colors
<b>backColor</b>	sets background color
<b>visibility</b>	visibility of the object
<b>greeked-size</b>	minimum text size before greeking is applied (negative value disables greeking)
<b>click-radius</b>	selection proximity for line hotspots
<b>value</b>	Defines a new value for the attribute; use empty string to reset original values.
<b>Allowed values for stroke-width</b>	positive value in mm or percentage
<b>Allowed values for colors</b>	RGB values separated by commas or percentage
<b>Allowed values for visibility</b>	use "hidden" to hide; all other values will show the object

## Example

```

objExpr = IsoViewX1.Iso3GetRootObject
Call IsoViewX1.Iso3SetGrAttribute(objExpr, "stroke", "20%")
Call IsoViewX1.Iso3SetGrAttribute(objExpr, "text-color", "20%")
Call IsoViewX1.Iso3SetGrAttribute(objExpr, "fill", "20%")
Call IsoViewX1.Refresh

//Examples for stroke-width:
Call IsoViewX1.Iso3SetGrAttribute(objExpr, "stroke-width", "2")
Call IsoViewX1.Iso3SetGrAttribute(objExpr, "stroke-width", "200%")

//Examples for colors:
Call IsoViewX1.Iso3SetGrAttribute(objExpr, "color", "255,0,0")
Call IsoViewX1.Iso3SetGrAttribute(objExpr, "color", "20%")

//Other examples:

```

---

```
Call IsoViewX1.Iso3SetGrAttribute("", "greeked-size", "2")
```

# 16

## Traversing the Object Tree

Iso3GetRootObject.....	106
Iso3GetChildCount .....	106
Iso3GetFirstChild.....	106
Iso3GetParent.....	107
Iso3GetNextSibling.....	107
Iso3GetPreviousSibling.....	108

---

## Iso3GetRootObject

Returns object identifier of root node as BSTR.

Introduced with Arbortext IsoView 3.0.

### Syntax

`Iso3GetRootObject ()`

### Example

```
objExpr = document.ivx1.Iso3GetRootObject();
```

## Iso3GetChildCount

Returns count of children, if any, as long. If object is the root, the layer count will be returned.

Introduced with Arbortext IsoView 3.0.

### Syntax

`Iso3GetChildCount (object)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
---------------	--

### Example

```
childCount =
document.ivx1.Iso3GetChildCount("id(myObj1)");
```

## Iso3GetFirstChild

Returns an object identifier of the first child of object, if any, as BSTR. If object is the root the first layer will be returned. The object identifier always uses the ID of the object. (See [Iso3GetObjectID on page 82](#).)

Introduced with Arbortext IsoView 3.0.

### Syntax

`Iso3GetFirstChild (object)`

Parameters:

---

<b>object</b>	Defines the object identifier as BSTR.
---------------	--

### Example

```
firstChild =  
document.ivx1.Iso3GetFirstChild("id(myObj1)");
```

## Iso3GetParent

Returns the object identifier of the parent object. An empty string is returned for the root node.

Introduced with Arbortext IsoView 3.0.

### Syntax

Iso3GetParent (*object*)

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
---------------	--

### Example

```
strParent =  
document.ivx1.Iso3GetParent("id(myObj1)");
```

## Iso3GetNextSibling

Returns an object identifier of next sibling, if any, as BSTR. If object is a layer the next layer will be returned. The object identifier always uses the ID of the object. (See [Iso3GetObjectID on page 82](#).)

Introduced with Arbortext IsoView 3.0.

### Syntax

Iso3GetNextSibling (*object*)

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
---------------	--

### Example

```
next =  
document.ivx1.Iso3GetNextSibling("id(myObj1)");
```

---

# Iso3GetPreviousSibling

Returns the object identifier of the previous sibling as BSTR. If the object is a layer the previous layer will be returned.

Introduced with Arbortext IsoView 3.0.

## Syntax

`Iso3GetPreviousSibling (object)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
---------------	--

## Example

```
previous = document.ivx1.Iso3GetPreviousSibling("id(myObj1)");
```

# 17

## Object Links

Iso3CountLinks .....	110
Iso3GetLinkByIndex .....	110
Iso3SelectLink.....	110

---

## Iso3CountLinks

Gets the number of links of an object. It returns the number of links of the object as long.

Introduced with Arbortext IsoView 3.0.

### Syntax

`Iso3CountLinks (object)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
---------------	--

### Example

```
document.ivx1.Iso3CountLinks("id(myObj1)");
```

## Iso3GetLinkByIndex

Gets the specified link of an object and returns the link as BSTR. This method will only retrieve `linkuri` attributes.

You can retrieve the number of links on an object using `Iso3CountLinks`. (See [Iso3CountLinks on page 110](#).)

Introduced with Arbortext IsoView 3.0.

### Syntax

`Iso3GetLinkByIndex (object, index)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
<b>index</b>	Defines the index of the object as long (0 based).

### Example

```
link =
document.ivx1.Iso3GetLinkByIndex("id(myObj1)",1);
```

## Iso3SelectLink

Lets the user select one of the links of an object in a dialog. It returns the index of the link chosen by the user as long.

---

This method is useful if multiple links have been associated with an object. A dialog opens showing all possible links. The user selects a link, which then will get executed.

Introduced with Arbortext IsoView 3.0.

## Syntax

`Iso3SelectLink (object)`

Parameters:

<b>object</b>	Defines the object identifier as BSTR.
---------------	--

## Example

```
linkNo =  
document.ivx1.Iso3SelectLink("id(myObj1)");
```



# 18

## Properties

AllowJumps .....	114
FileName .....	114
HsVisible .....	114
Preview .....	114
PropertyChangeMode .....	115
ResizeToFit .....	115
ViewPort .....	116
ViewPortCount .....	116
ViewSize .....	116
AcceptDroppedFiles .....	117
ActiveTool .....	117
ExtentX .....	117
ExtentY .....	118
HomeFileName .....	118
HomeViewPort .....	118
ObjectTipState .....	118
OffsetX .....	119
OffsetY .....	119
AutostartOnloadAnimations .....	119

---

## AllowJumps

With this property you determine whether internal jumps are allowed. An internal jump exists if a viewport has been selected in Arbortext IsoDraw as a target for an object. If the user clicks on the object the specified viewport will be set.

This property enables or disables internal jumping; default: `true`.

Introduced with Arbortext IsoView 1.0.

### Example

```
document.ivx1.AllowJumps = false; or document.ivx1.setAllowJumps( false );
```

## FileName

This property defines the name or the URL of the current illustration. It sets the current filename as BSTR.

Introduced with Arbortext IsoView 1.0.

### Example

```
document.ivx1.FileName = "http://www.ptc.com/demo/sample.iso";  
//or  
form1.edtFile.value = document.ivx1.FileName;
```

## HsVisible

This property controls the visibility of hotspots. Visible means that all hotspots in the current file are highlighted.

It sets the visible / invisible status of hotspots as Boolean (default: `true`).

Introduced with Arbortext IsoView 1.0.

### Example

```
document.ivx1.HsVisible = false;
```

## Preview

### Note

*This property is obsolete effective Arbortext IsoView 3.0.*

This property determines whether the illustration is displayed in preview mode or not. It sets if the preview mode is turned on or off as Boolean (default: `true`).

---

Preview has the same meaning here that it has inside Arbortext IsoDraw.

Introduced with Arbortext IsoView 1.0.

### Example

```
document.ivx1.Preview = true;
```

## PropertyChangeMode

With this property you start / end a sequence of property changes. It sets the property change mode status as Boolean.

This is advantageous as no redrawing will be performed after changing a single property. Setting the mode to `false` will automatically rebuild and redraw the image.

Introduced with Arbortext IsoView 1.0.

### Example

```
document.ivx1.PropertyChangeMode = true;
document.ivx1.Scale = 10;
document.ivx1.OffsetX = 47.5;
document.ivx1.OffsetY = 12.5;
document.ivx1.Preview = False;
document.ivx1.PropertyChangeMode = false;
```

## ResizeToFit

### Note

*This property is obsolete since version 3.0.*

This property enables / disables the resizing of the illustration to the viewer window. It is recommended to disable this before changing the position and size using `OffsetX`, `OffsetY`, `ViewSize`. It should be enabled if viewports are used.

This property sets the resize to fit status as Boolean.

See also:

- [OffsetX on page 119](#).
- [OffsetY on page 119](#).
- [ViewSize on page 116](#).

Introduced with Arbortext IsoView 1.0.

---

## Example

```
document.ivx1.ResizeToFit = true;  
or document.ivx1.setResizeToFit( true )
```

## ViewPort

This property sets the name of the current viewport (case sensitive). If this property changes, Arbortext IsoView tries to establish this viewport based on the information found in the current illustration.

Introduced with Arbortext IsoView 1.0.

## Example

```
document.ivx1.ViewPort = "View A";  
//or  
form1.edtVP.value = document.ivx1.ViewPort;
```

Arbortext IsoView can generate viewports dynamically for a given object. In all places where a string for a viewport is used you can also use the following:

`$OBJ_ObjectName` or `$NAME_ObjectName`

where `$OBJ_` or `$NAME_` precedes the name of the object you want to show and `ObjectName` is the object's name. If the name of the object is HS1 the string would be `$OBJ_HS1` or `$NAME_HS1`, `$ID_ObjectID`

## ViewPortCount

### Note

*This property is supported for compatibility with Arbortext IsoView 1.0.*

This read-only property returns the number of viewports contained in the illustration as Long.

Introduced with Arbortext IsoView 1.0.

## Example

```
lVp = document.ivx1.ViewPortCount;
```

## ViewSize

This property sets the scale (zoom factor) of the representation as double (1..6400%). If you use this, we recommend to disable the `ResizeToFit` feature to avoid recursive effects. (See [ResizeToFit on page 115](#).)

---

Introduced with Arbortext IsoView 1.0.

### Example

```
document.ivx1.ResizeToFit = false;  
document.ivx1.ViewSize = 50;
```

## AcceptDroppedFiles

With this property you instruct Arbortext IsoView to handle or to ignore files dropped on the viewer window. It sets how files are handled when dropped on the viewer window as Boolean (accept, default: true).

This property works with the Arbortext IsoView ActiveX Control only.

Introduced with Arbortext IsoView 2.0.

### Example

```
document.ivx1.AcceptDroppedFiles = false;
```

## ActiveTool

With this property you determine which tool is currently selected. It returns the active tool. (See [Tool Set/Selection Constants – Iso7ConfigTools and ActiveTool on page 135](#) for allowed values.)

Use Iso7ConfigTools to configure the behavior of the toolbar and the set of selectable tools. (See [Iso7ConfigTools on page 30](#).)

Introduced with Arbortext IsoView 2.0.

### Example

```
// select ZoomIn tool  
document.ivx1.ActiveTool = 2;  
//or  
document.ivx1.setActiveTool( 2 );
```

## ExtentX

Use this read-only property to retrieve the horizontal extent of illustration in mm as double.

Introduced with Arbortext IsoView 2.0.

### Example

```
form1.edtX.value = document.ivx1.ExtentX
```

---

## ExtentY

Use this read-only property to retrieve the vertical extent of illustration in mm as double.

Introduced with Arbortext IsoView 2.0.

### Example

```
form1.edtY.value = document.ivx1.ExtentY
```

## HomeFileName

Use this read-only property to retrieve the filename of the home illustration set by the first call to [Iso3OpenFile](#) or [SetHome](#). It returns the filename of the home illustration as BSTR. (See [Iso3OpenFile](#) on page 21 and [SetHome](#) on page 48.)

Introduced with Arbortext IsoView 2.0.

### Example

```
form1.edtHF.value = document.ivx1.HomeFileName
```

## HomeViewPort

Use this read-only property to retrieve the home viewport set by the first call to [Iso3OpenFile](#) or [SetHome](#). It returns the home viewport as BSTR. (See [Iso3OpenFile](#) on page 21 and [SetHome](#) on page 48.)

Introduced with Arbortext IsoView 2.0.

### Example

```
form1.edtHV.value = document.ivx1.HomeViewPort
```

## ObjectTipState

Use this property to determine the content of the object tips displayed when an object is 'touched' by the mouse pointer. It returns the content type of the object tip as Long. The default value is 0. (For more information, see [Object Tip](#) on page 91 and [Tip Constants - Object Tip](#) on page 136.)

Allowed values are:

0	NONE (no object tips are shown)
1	TIP (only object tips will be shown which have been assigned previously by SetObjectTip)

---

2	ID (the ID of the object will be shown)
3	TIP OR ID (shows a preassigned tip if available or the ID otherwise)
4	NAME (the name of the object will be shown)
5	TIP OR NAME (shows a preassigned tip if available or the name otherwise)
16	HOTSPOTS ONLY (forces that object tips only are displayed on hot spots)

Introduced with Arbortext IsoView 2.0.

## OffsetX

This property describes the horizontal position of the representation. It sets the horizontal position of the representation as double.

If you use this, we recommend to disable the `ResizeToFit` feature to avoid recursive effects. (See [ResizeToFit on page 115](#).)

Introduced with Arbortext IsoView 1.0.

### Example

```
document.ivx1.OffsetX = 20.0;
```

## OffsetY

This property describes the vertical position of the representation. It returns the vertical position of the representation as double.

If you use this, we recommend to disable the `ResizeToFit` feature to avoid recursive effects. (See [ResizeToFit on page 115](#).)

Introduced with Arbortext IsoView 1.0.

### Example

```
document.ivx1.OffsetY = 20.0;
```

## AutostartOnloadAnimations

Enables or disables the ability of `onload` animation sequences to start automatically when a document loads.

Introduced with Arbortext IsoView 7.1.

---

## Example

```
AutostartOnloadAnimations = false;
```

# 19

## Events

ContextMenuItemHit .....	122
FileChanged .....	122
HsHit .....	123
InitFinished .....	123
KeyDown .....	124
KeyPress .....	124
KeyUp .....	124
MouseDown .....	125
MouseMove .....	125
MouseUp .....	126
PropertyChanged .....	126
ObjectHit .....	126
ViewWindowChanged .....	127

---

## ContextMenuItemHit

This event is generated when the user selects an item from a context menu. (See [Context Menu Functions on page 55](#).)

Introduced with Arbortext IsoView 1.0.

### Syntax

ContextMenuItemHit (*item*, *Title*)

Parameters:

<b>item</b>	Indicates the No. of the menu item (1 based) as short.
<b>Title</b>	Indicates the Title of the menu item as BSTR.

### Example

```
<!-- Internet Explorer -->
<script language="VBScript" for="ivx1"
event="ContextMenuItemHit(item, Title)">
<!--
    call ContextMenuItemHit( item, Title )
//-->
</script>
```

## FileChanged

This event is generated when the file has been changed. This may occur either from outside (via `Iso3OpenFile` or `FileName`) or when Arbortext IsoView has executed internal “jumps” to another file. (See [Iso3OpenFile on page 21](#) and [FileName on page 114](#).)

Introduced with Arbortext IsoView 1.0.

### Syntax

FileChanged (*FileName*)

Parameters:

<b>FileName</b>	Indicates the current filename as BSTR.
-----------------	---

### Example

```
<!-- Internet Explorer -->
<script language="VBScript" for="ivx1"
event="FileChanged(strFileName)">
<!--
    call FileHasChanged( strFileName )
//-->
```

---

```
</script>
```

## HsHit

This event is generated if you move the mouse over a hotspot or click on it with one of the two mouse buttons.

As soon as the mouse leaves the hotspot area, another HsHit event is generated and all parameters are set to 0 or empty.

Introduced with Arbortext IsoView 1.0.

### Syntax

```
HsHit (MouseBtn, RefID, Name, ViewPortName, ViewPortFile)
```

Parameters:

<b>MouseBtn</b>	Indicates which mouse button has been clicked as short. Allowed values are:
0	no mouse button
1	left mouse button
2	right mouse button
<b>RefID</b>	Indicates the ID of the hotspot (may be 0) as long.
<b>Name</b>	Indicates the Name of the hotspot (may be empty) as BSTR.
<b>ViewPortName</b>	Indicates the Name of the viewport connected to the hotspot (may be empty) as BSTR.
<b>ViewPortFile</b>	Indicates the Name of the file in which the viewport is located (may be empty) as BSTR.

### Example

```
<!-- Internet Explorer -->
<script language="VBScript" for="ivx1"
event="HsHit(nMouseBtn, lRefID, strName, strViewPortName, strViewPortFile)">
<!--
    call MyHsHit( nMouseBtn, lRefID, strName, strViewPortName, strViewPortFile )
//-->
</script>
```

## InitFinished

This event is generated when the initialization of Arbortext IsoView is finished. This can be used in products like PowerPoint to initialize Arbortext IsoView before a slide is shown on screen.

---

Introduced with Arbortext IsoView 2.0.

### Syntax

InitFinished()

### Example

```
<!-- Internet Explorer -->
<script language="VBScript" for="ivx1"
event="InitFinished()">
<!--
    call MyInitFinished()
//-->
</script>
```

## KeyDown

Introduced with Arbortext IsoView 3.0.

### Syntax

KeyDown (*Char*, *ShiftState*)

Parameters:

<b>Char</b>	Indicates which key has been pressed as short.
<b>ShiftState</b>	Indicates if the SHIFT key was pressed as short.

## KeyPress

Introduced with Arbortext IsoView 3.0.

### Syntax

KeyPress (*Char*)

Parameters:

<b>Char</b>	Indicates which key has been pressed as short.
-------------	--

## KeyUp

Introduced with Arbortext IsoView 3.0.

---

## Syntax

`KeyUp (Char, ShiftState)`

Parameters:

<b>Char</b>	Indicates which key has just been released as short.
<b>ShiftState</b>	Indicates if the SHIFT key was pressed as short.

## MouseDown

This event is disabled by default. Use `ConfigEvents ()` to enable it.

Introduced with Arbortext IsoView 2.0.

## Syntax

`MouseDown (Button, Shift, x, y)`

Parameters:

<b>Button</b>	Indicates which mouse button has been clicked as short.
<b>Shift</b>	Indicates if the shift key has been pressed as short.
<b>x</b>	Indicates the horizontal position as OLE_XPOS_PIXELS.
<b>y</b>	Indicates the vertical position as OLE_YPOS_PIXELS.

## MouseMove

This event is disabled by default. Use `ConfigEvents` to enable it. (See [ConfigEvents on page 28](#).)

Introduced with Arbortext IsoView 2.0.

## Syntax

`MouseMove (Button, Shift, x, y)`

Parameters:

<b>Button</b>	Indicates which mouse button has been clicked as short.
<b>Shift</b>	Indicates if the shift key has been pressed as short.
<b>x</b>	Indicates the horizontal position as OLE_XPOS_PIXELS.
<b>y</b>	Indicates the vertical position as OLE_YPOS_PIXELS.

---

## MouseUp

This event is disabled by default. Use `ConfigEvents` to enable it. (See [ConfigEvents on page 28](#).)

Introduced with Arbortext IsoView 2.0.

### Syntax

`MouseUp (Button, Shift, x, y)`

Parameters:

<b>Button</b>	Indicates which mouse button has been clicked as short.
<b>Shift</b>	Indicates if the shift key has been pressed as short.
<b>x</b>	Indicates the horizontal position as OLE_XPOS_PIXELS.
<b>y</b>	Indicates the vertical position as OLE_YPOS_PIXELS.

## PropertyChanged

This event is generated when a property (`scale`, `offset`, `filename`, `viewport`, `flags`) has been changed.

Introduced with Arbortext IsoView 2.0.

### Syntax

`PropertyChanged ()`

## ObjectHit

This event is generated if you move the mouse over an object or click on it with one of the two mouse buttons.

As soon as the mouse leaves the object's area, another `ObjectHit` event is generated and all parameters are set to 0 or empty.

Introduced with Arbortext IsoView 3.0.

### Syntax

`ObjectHit (MouseBtn, object)`

Parameters:

---

<b>MouseBtn</b>	Indicates which mouse button has been pressed as short. Allowed values are:
0	no mouse button
1	left mouse button
2	right mouse button
<b>object</b>	Indicates the object identifier as BSTR.

### Example

```
<!-- Internet Explorer -->
<script language="VBScript" for="ivx1"
event="ObjectHit(nMouseBtn, strObject)">
<!--
    call MyObjectHit( nMouseBtn, strObject )
//-->
</script>
```

## ViewWindowChanged

This event occurs when one of the additional view windows (**Magnifier**, **Global View**) is opened or closed.

Introduced with Arbortext IsoView 4.0.

### Syntax

`ViewWindowChanged ( Window, Action )`

Parameters:

<b>Window</b>	Indicates the type of the window which has changed as short. Allowed values are:
1	MAGNIFIER
2	GLOBAL
<b>Action</b>	Indicates the action as short. Allowed values are:
1	WINDOW OPENED
2	WINDOW CLOSED

### Example

```
<script language="JavaScript" for="ivx1"
event="ViewWindowChanged(nWindow,nAction)">
<!--
    handleViewWindowChanged(nWindow,nAction);
//-->
```

---

```
</script>
```

# A

## Object Identifiers and Expressions

Object identifier .....	130
objExpr.....	130
Extended Object Expression .....	130

---

## Object identifier

The object identifier is used to identify a single object. The expression must uniquely identify the object. (See [Object Functions on page 79](#).)

Use an object expression to identify multiple objects with a common name. (See [objExpr on page 130](#).)

### Parameters

<b>id (objID)</b>	The expression consists of <code>id</code> followed by (in parentheses) the actual ID of the object.
<b>name (objName)</b>	The expression consists of <code>name</code> followed by the actual name of the object in parentheses. If the name is used on multiple objects the first object with that name will be identified.
<b>objID</b>	The actual ID of the object is a shortcut for the identifier <code>id (objID)</code> .
<b>\$ROOT</b>	Special ID for the root node of the entire illustration.
<b>id(myID1)</b>	Identifies the object with the ID <code>myID1</code> .
<b>name(myName1)</b>	Identifies the object with the name <code>myName1</code> .

## objExpr

The object expression is used to identify multiple objects with the same name.

Use an object identifier to identify a single object by its ID or name.

### Parameters

<b>name(myName1, all)</b>	The expression consists of <code>name</code> followed by (in parentheses; comma separated) the actual name of the object and <code>all</code> . This will identify all objects with this name.
<b>name (myObj1, all)</b>	Identifies all objects with the name <code>myObj1</code> .

## Extended Object Expression

The extended object expression is used to identify multiple objects or elements with the same attributes.

### Syntax

A valid term for the extended object expression can be either of a `specialForm`, the `normalForm` or the `isoForm`. For further reference to the `specialForm` and the `normalForm`, please refer to the WebCGM 2.0 fragment syntax.

---

```

objterm          ::=  specialForm
|               normalForm
|               isoForm

```

Please find below a description of the `isoForm` syntax. The `isoForm` is always introduced by the `isoeval` keyword.

<code>isoForm</code>	<code>::= "isoeval("evalTerm(" objbehavior)? ")"</code>
<code>evalTerm</code>	<code>::= objInfoTerm   objIDTerm   objNameTerm   objTipTerm   objAttrTerm   textTerm</code>

and can then address either Object ID, Name, Tip. It can also check for the existence of object information and check any other added attribute. Furthermore it is possible to access text elements.

<code>objInfoTerm</code>	<code>::= "obj_ex"</code>
<code>objIDTerm</code>	<code>::= "id_cont=" objid</code>

To address objects by their the name attribute you have three possible methods. The `objNameTerm` supports check for existing name attribute and the nonexistence of a name attribute. Objects may be addressed by parts of the value of the name attribute.

<code>objNameTerm</code>	<code>::= objNameExTerm   objNameNExTerm   objNameContTerm</code>
--------------------------	---

The keywords below need to be used to build the object expression.

<code>objNameExTerm</code>	<code>::= "name_ex"</code>
<code>objNameNExTerm</code>	<code>::= "name_nex"</code>
<code>objNameContTerm</code>	<code>::= "name_cont=" objname</code>

Objects may also be addressed by their object `tip` attribute. Once more, objects can be addressed by either having a `tip` attribute connected, and by certain values of the `tip` attribute.

<code>objTipTerm</code>	<code>::= objTipEqTerm</code>
-------------------------	-------------------------------

---

```

|      objTipNEqTerm
|      objTipContTerm
|      objTipExTerm
|      objTipNExTerm

objTipEqTerm      ::= "tip_eq=" objtip
objTipNEqTerm     ::= "tip_neq=" objtip
objTipContTerm    ::= "tip_cont=" objtip
objTipExTerm      ::= "tip_ex"
objTipNExTerm     ::= "tip_nex"

objtip            ::= (char) +

```

The attributes can be checked for their content. Either the attribute will be checked if equal or not equal to a certain value. It is also possible to check whether an attribute content contains a certain value.

```

objAttrTerm       ::= objAttrEqTerm
|      objAttrNEqTerm
|      objAttrContTerm

```

Please use the keywords and syntax given below.

```

objAttrEqTerm     ::= "attr_eq=" objattr ",cont="
attrcontent
objAttrNEqTerm    ::= "attr_neq=" objattr ",cont="
attrcontent
objAttrContTerm   ::= "attr_cont=" objattr ",cont="
attrcontent
objattr           ::= (char) +
attrcontent        ::= (char) +

```

Text elements can also be addressed. As a criteria their value is taken.

```

textTerm          ::= textEqTerm
|      textNEqTerm
|      textContTerm

```

Here we can test either if a text element is exactly equal to a certain value,

```

textEqTerm         ::= "text_eq=" textcontent
textNEqTerm        ::= "text_neq=" textcontent
textContTerm       ::= "text_cont=" textcontent

```

# B

## Constants

Tool Set/Selection Constants – ConfigTools2, ActiveTool .....	134
Tool Set/Selection Constants – Iso7ConfigTools and ActiveTool .....	135
Tip Constants - Object Tip .....	136
Object Highlighting Constants – Iso3HighlightObject .....	136
Event Configuration Constants – ConfigEvents .....	137
Print Configuration Constants .....	137

---

## Tool Set/Selection Constants – ConfigTools2, ActiveTool

Use ConfigTools2 and ActiveTool to enable tool and/or let tool appear in main toolbar:

(See [ConfigTools2 on page 29](#).)

kIV4_Btn_Hand	1	0x00000001L
kIV4_Btn_Zoom	2	0x00000002L
kIV4_Btn_ZoomOut	4	0x00000004L
kIV4_Btn_Hotspots	8	0x00000008L
kIV4_Btn_Print	16	0x00000010L
kIV4_Btn_Home	32	0x00000020L
kIV4_Btn_Magnifier	64	0x00000040L
kIV4_Btn_GlobalView	128	0x00000080L
kIV4_Btn_RedLining	256	0x00000100L
kIV4_Btn_RLDraw	512	0x00000200L
kIV4_Btn_RLNNote	1024	0x00000400L
kIV4_Btn_RLDelete	2048	0x00000800L
kIV4_Btn_RLEExport	4096	0x00001000L
kIV4_Btn_Prefs	8192	0x00002000L
kIV4_Btn_RLImport	16384	0x00004000L
kIV4_AllButtons	65535	0x000FFFFL

Use these constants to let tool appear in popup menu:

kIV4_Pop_Hand	65536	0x00010000L
kIV4_Pop_Zoom	131072	0x00020000L
kIV4_Pop_ZoomOut	262144	0x00040000L
kIV4_Pop_Hotspots	524288	0x00080000L
kIV4_Pop_Print	1048576	0x00100000L
kIV4_Pop_Home	2097152	0x00200000L
kIV4_Pop_Magnifier	4194304	0x00400000L
kIV4_Pop_GlobalView	8388608	0x00800000L
kIV4_Pop_RedLining	16777216	0x01000000L
kIV4_Pop_RLDraw	33554432	0x02000000L
kIV4_Pop_RLNNote	67108864	0x04000000L
kIV4_Pop_RLDelete	134217728	0x08000000L
kIV4_Pop_RLEExport	268435456	0x10000000L

---

kIV4_Pop_Prefs	536870912	0x20000000L
----------------	-----------	-------------

Add values to combine options:

kIV4_DefaultTools	566255167	0x21C05E3FL
kIV4_ToolsNone	0	No tools to interact with viewer
kIV4_ToolsButtons	1	Default Arbortext IsoView toolbar with buttons
kIV4_ToolsPopup	2	All tools as popup menu
kIV4_ToolsRemote	4	Remote control of tools (using ActiveTool)
kIV4_ToolsRLAutoHide	8	Automatically hide/show redlining tools

## Tool Set/Selection Constants – Iso7ConfigTools and ActiveTool

Use these constants to enable a tool and/or add it to the main toolbar. (See [Iso7ConfigTools on page 30](#) and [ActiveTool on page 117](#).)

kIV7_Tool_Hand	1	0x00000001L
kIV7_Tool_Zoom	2	0x00000002L
kIV7_Tool_ZoomOut	4	0x00000004L
kIV7_Tool_Hotspots	8	0x00000008L
kIV7_Tool_Print	16	0x00000010L
kIV7_Tool_Home	32	0x00000020L
kIV7_Tool_Magnifier	64	0x00000040L
kIV7_Tool_GlobalView	128	0x00000080L
kIV7_Tool_RedLining	256	0x00000100L
kIV7_Tool_RLDraw	512	0x00000200L
kIV7_Tool_RLNote	1024	0x00000400L
kIV7_Tool_RLDelete	2048	0x00000800L
kIV7_Tool_RLExport	4096	0x00001000L
kIV7_Tool_RLImport	8192	0x00002000L
kIV7_Tool_Prefs	16384	0x00004000L
kIV7_Tool_Search	131072	0x00020000L
kIV7_Tool_RLCircle	262144	0x00040000L
kIV7_Tool_RLRectangle	524288	0x00080000L
kIV7_Tool_RLLine	1045876	0x00100000L

---

kIV7_Tool_AniStart*	2097152	0x00200000L
kIV7_Tool_AniStop*	4194304	0x00400000L
kIV7_Tool_AniRewind <sup>1</sup>	8388608	0x00800000L
kIV7_Tool_3DRotate <sup>2</sup>	16777216	0x01000000L

1. Only visible if file contains animation.
2. Only visible if file contains 3D data.

Use these constants to enable tools and/or display them on a popup menu.

kIV4_DefaultTools	566255167	0x21C05E3FL
kIV4_ToolsNone	0	No tools to interact with viewer
kIV4_ToolsButtons	1	Default Arbortext IsoView toolbar with buttons
kIV4_ToolsPopup	2	All tools as popup menu
kIV4_ToolsRemote	4	Remote control of tools (using ActiveTool)
kIV4_ToolsRLAutoHide	8	Automatically hide/show redlining tools

## Tip Constants - Object Tip

See [ObjectTipState on page 118](#).

kIV_ObjTip_Hide	0	hide object tips
kIV_ObjTip_ShowTip	1	show tips if available on object
kIV_ObjTip_ShowID	2	use ID of object in tip
kIV_ObjTip_ShowTipOr-ID	3	use tip of object or ID if none
kIV_ObjTip_ShowName	4	use name of object in tip
kIV_ObjTip_ShowTipOr-Name	5	use tip of object or name if none
kIV_ObjTip_HotspotsOnly	16	add to get objects tips on hotspots only

## Object Highlighting Constants – Iso3HighlightObject

See [Iso3HighlightObject on page 88](#).

kIV_HotSpotHide	0	hide hot spot highlight
kIV_HotSpotFramed	2	hot spot is framed

---

kIV_HotSpotFilled	4	hot spot is filled
kIV_HotSpotFlash	8	hot spot will be flashed
kIV_HotSpotCtr	16	hot spot will be centered (not yet supported)
kIV_HotSpotAll	32	process all hotspots with given ID
kIV_HotSpotCircled	64	a circle around will indicate hotspot

## Event Configuration Constants – ConfigEvents

See [ConfigEvents on page 28](#).

(Default: 1+2+4+8+16+32)

kIV_HsHit	1	Enables HsHit event (default)
kIV_FileChanged	2	Enables FileChanged event (default)
kIV_ContextMenuHit	4	Enables ContextMenuHit event (default)
kIV_PropertyChanged	8	Enables PropertyChanged event (default)
kIV_InitFinished	16	Enables InitFinished event
kIV_ObjectHit	32	Enables ObjectHit event (default)
kIV_MouseMove	1024	Enables MouseMove event
kIV_MouseDown	2048	Enables MouseDown event
kIV_MouseUp	4096	Enables MouseUp event
kIV_DontHandleClicks	8192	Prevents Arbortext IsoView from handling mouse clicks internally

## Print Configuration Constants

See [ConfigPrinting on page 36](#) and [DoPrint on page 37](#).

kIV_Printer_KeepSettings	-1	Do not change existing
--------------------------	----	------------------------

---

kIV_Printer_PrintAll	0	setting
kIV_Printer_PrintVisible	1	Print entire illustration
kIV_Printer_FitOnPage	0	Print only visible portion of illustration
kIV_Printer_SplitPages	1	Fit printed image onto one page
kIV_Printer_Portrait	1	Split printed image across multiple pages if necessary
kIV_Printer_Landscape	2	Print in portrait orientation
		Print in landscape orientation

# C

## Standards Support

WebCGM 2.0 DOM.....	140
Unicode Support .....	140

---

## WebCGM 2.0 DOM

Arbortext IsoView 7.0 and above supports WebCGM 2.0 and the WebCGM 2.0 DOM (Document Object Model).

For details regarding the WebCGM 2.0 DOM, please refer to the WebCGM 2.0 standard at: <http://www.oasis-open.org/specs/index.php#webcgmv2.0>, or the WebCGM 2.0 recommendation at <http://www.w3.org/Graphics/WebCGM/WG/>.

## Unicode Support

Arbortext IsoView can handle WebCGM and Arbortext IsoDraw files containing Unicode characters. Thus, it is possible to store East-Asian language characters and western characters in one file. Object identifiers, attributes, and screen tips use Unicode as well.

### **Note**

*The Windows 95/98/Me operating systems do not support Unicode—so Arbortext IsoView cannot display Unicode characters if it is running on one of these old Windows platforms.*

# D

## Files and File Formats

List of Viewable File Formats .....	142
Required Files .....	142

---

## List of Viewable File Formats

- Arbortext IsoDraw files (\*.iso)
- CGM
- ProductView files (\*.ol, \*.ed, \*.edz, \*.pvs, \*.pvz)
- SVG
- TIFF
- JPEG
- PNG
- PCX
- CALS Raster

## Required Files

You need the files listed below to use the Arbortext IsoView ActiveX control:

### PTC Files Required

Filename	Required for Arbortext IsoView Version:	
	7.0	7.1
IsoViewX7.ocx	✓	
IsoViewX71.ocx		✓
IsoEngine7.exe	✓	
IsoEngine71.exe		✓
OptCGM.prf	✓	✓
DefaultHatchings.iso	✓	✓
IsoView7License.iso	✓	✓
pvGlue.dll	✓	
pvGlue10.dll		✓
PV2ISO.pvr	✓	✓

### CGM Open Files Required

All CGM Open DLLs in the indicated version (or higher) should be located in the Windows ..\system32 directory.

Filename	Required for Arbortext IsoView Version:	
	7.0	7.1
cgmopenbho.dll (version 1.0.0.5)	✓	✓

<b>Filename</b>	<b>Required for Arbortext IsoView Version:</b>	
	<b>7.0</b>	<b>7.1</b>
cgmopenbho.cfg	✓	✓

For more information about the CGM Open BHO please refer to the CGM Open website: <http://www.cgmopen.org/technical/bho/index.html>

### **Microsoft Windows Operating System Files Required**

All Microsoft Windows DLLs in the indicated version (or higher) should be located in the Windows ..\system32 directory.

<b>Filename</b>	<b>Required for Arbortext IsoView Version:</b>	
	<b>7.0</b>	<b>7.1</b>
mfc42u.dll (version 6.0.8665.0)	✓	✓
msvcrt.dll (version 6.0.8797.0)	✓	✓

### **Microsoft Internet Explorer Files Required**

The following components of Microsoft Internet Explorer (IE) (version 4.01 or higher) are also required. Microsoft IE must be installed on the processing computer. All Microsoft IE DLLs in the indicated version (or higher) should be located in the Windows ..\system32 directory.

<b>Filename</b>	<b>Required for Arbortext IsoView Version:</b>	
	<b>7.0</b>	<b>7.1</b>
shlwapi.dll (version 4.70.1300)	✓	✓
urlmon.dll (version 4.70.1300)	✓	✓



# E

## Arbortext IsoView Basics

Toolbar .....	146
Redlining .....	148
Finding Elements on the Illustration.....	148
Overview of Preference Identifiers.....	148
Error Codes .....	149
How to Prepare Arbortext IsoView for Distribution.....	150
Compatibility .....	150

---

## Toolbar

The toolbar is optionally shown in the Arbortext IsoView window. (See [Iso7ConfigTools on page 30](#).)

	Pan	Activates the “pan,, mode
	3D Rotate	Rotate 3D (only visible if the file includes 3D data)
	ZoomIn	Activates the “zoom in,, mode (“zoom out,, with depressed shift key) Press CTRL to zoom inside hotspots.
	ZoomOut	Activates the “zoom out,, mode (“zoom in,, with depressed shift key) Press CTRL to zoom inside hotspots.
	Hotspot	Highlights the hotspots of the view as long as the mouse key is depressed
	Animation Start	Starts the animation (only visible if file contains an animation)
	Animation Stop	Stops the animation (only visible if file contains an animation)
	Animation Rewind	Rewinds the animation (only visible if file contains an animation)
	Print	Prints the current illustration
	Find	Finds Elements on the illustration.
	Home	Returns to the “Home,, view (see OpenFile, SetHome)
	Magnifier	Toggles the magnifier window state
	Global view	Toggles the global view window state

---

	Redlining	Toggles the redlining mode
	Redline element	Activate the redlining element drawing mode only selectable if redlining mode is active
	Redline circle	Activate the redlining circle drawing mode only selectable if redlining mode is active
	Redline rectangle	Activate the redlining rectangle drawing mode only selectable if redlining mode is active
	Redline line	Activate the redlining line drawing mode only selectable if redlining mode is active
	Annotation	Activate the redlining annotation mode only selectable if redlining mode is active
	Delete	Activate the redlining delete mode only selectable if redlining mode is active
	Export	Exports the illustration and the redlining information only selectable if redlining mode is active
	Import	Imports redlining information.
	Preferences	Opens the preferences dialog
	More tools	Opens the popup tools

Since the toolbar is configurable through the **Preferences** dialog, some tools might appear in the toolbar, in the popup tools, or be disabled. Using the **auto hide redlining tools** option it is possible to hide the redlining tool buttons until the redlining mode is activated.

# Redlining

This technology enables you to add “information” to an existing illustration.

To do so, Arbortext IsoView version 4 introduced some new tools. To use these redlining tools you have to start the redlining mode either using the  toolbar button, or by calling the `Iso4StartRedlining` method. (See [Iso4StartRedlining on page 42](#).)

While running in Redlining mode Arbortext IsoView you can draw freehand lines and other graphical primitives and to add textual annotations.

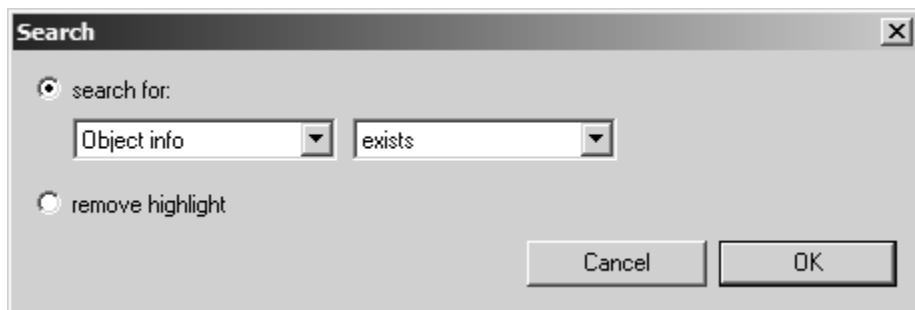
After adding redlining information to the illustration you have can export these for later use.

With Arbortext IsoView version 5 the `Iso5ImportRedlining` method was introduced to re-import the redlining layers for display later. (See [Iso5ImportRedlining on page 43](#).)

## Finding Elements on the Illustration

This functionality was introduced with Arbortext IsoView version 6 and is used to find elements on the illustration.

To do so, click the  toolbar button. You will be prompted to specify the search options in a **Search** dialog box:



Elements can be highlighted similar to the Arbortext IsoDraw **Search** dialog box. Here, it is not necessary to have object information assigned.

## Overview of Preference Identifiers

See [Iso4SetPreference on page 31](#).

<b>anti_aliasing</b>	Switches the anti-aliasing mode on (1) or off (0).
<b>default_toolset</b>	Default set of toolbar buttons. (See <a href="#">Iso7ConfigTools on page 30</a> for values.)
<b>default_toolflags</b>	Default flags for toolbar configuration. (See <a href="#">Iso7ConfigTools on page 30</a> for values.)

<b>redlining_linecolor</b>	Color of RedLining line.						
<b>redlining_line-width</b>	Line width (in points) of RedLining line.						
<b>redlining_font-size</b>	Font size (in points) of RedLining annotations.						
<b>redlining_font-name</b>	Font name of RedLining annotations.						
<b>use_default_printer</b>	Automatically use default printer (1) or popup printer selection dialog box (0).						
<b>object_behavior</b>	<p>Change the behavior of the <a href="#">ObjectHit</a> on page 126 event.</p> <p>Values:</p> <table border="1"> <tr> <td><b>0</b></td> <td>Arbortext IsoView version 4: left click on hotspots only, right click, move on all objects.</td> </tr> <tr> <td><b>1</b></td> <td>Hotspots only: left click, right click, move on hotspots only.</td> </tr> <tr> <td><b>2</b></td> <td>All objects: left click, right click, move on all objects.</td> </tr> </table>	<b>0</b>	Arbortext IsoView version 4: left click on hotspots only, right click, move on all objects.	<b>1</b>	Hotspots only: left click, right click, move on hotspots only.	<b>2</b>	All objects: left click, right click, move on all objects.
<b>0</b>	Arbortext IsoView version 4: left click on hotspots only, right click, move on all objects.						
<b>1</b>	Hotspots only: left click, right click, move on hotspots only.						
<b>2</b>	All objects: left click, right click, move on all objects.						

## Error Codes

<b>5</b>	Invalid function call
<b>7</b>	Insufficient free memory
<b>53</b>	File not found
<b>75</b>	Error in file access
<b>380</b>	Invalid parameter values
<b>1000</b>	Unknown error
<b>1001</b>	Error when starting Arbortext IsoView server
<b>1002</b>	Invalid command line parameter
<b>1003</b>	Prohibited extension of existing context menu
<b>1004</b>	Error when creating context menu
<b>1005</b>	Illegal try to open <b>Global View</b> or <b>Magnifier</b> window twice
<b>1006</b>	Selected tool/function not enabled for use

---

## How to Prepare Arbortext IsoView for Distribution

1. Open the self-extracting archive `isoviewx7.exe` (for Arbortext IsoView version 7.0) or `isoviewx71.exe` (for Arbortext IsoView version 7.1) using WinZip and extract all files to a new folder.
2. For both Arbortext IsoView version 7.0 and 7.1: Add the file `IsoView7License.iso`—which you received with your Arbortext IsoView Distribution Kit—to that folder. The installer will look for this file at install time. If it is not in the correct place the installer will stop and ask for it.
3. Add all files in the folder to a new self-extracting archive `isoviewx7.exe` (for Arbortext IsoView version 7.0) or `isoviewx71.exe` (for Arbortext IsoView version 7.1) using WinZip Self-Extractor. Make sure you select **Self-extracting Zip file for Software Installation** as the new archive type.

## Compatibility

In principle, the Arbortext IsoView ActiveX control may be used in all ActiveX-capable environments. We have tested the following applications:

- Microsoft Internet Explorer 5.01, 5.5, 6.0, 7.0
- Microsoft Visual Basic 6.0
- Microsoft Access 2000

When using Microsoft Internet Explorer, PTC strongly recommends version 5.01 or newer. Older versions do not support all the navigation methods available in Arbortext IsoView.

