

Parametric Technology Corporation

**Arbortext IsoView 7.0
Functional Reference**

Arbortext® IsoView™ 7.0

Copyright © 2007 Parametric Technology Corporation. All Rights Reserved.

User and training guides and related documentation from Parametric Technology Corporation and its subsidiary companies (collectively “PTC”) is subject to the copyright laws of the United States and other countries and is 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.

For Important Copyright, Trademark, Patent, and Licensing Information: For Windchill products, select About Windchill at the bottom of the product page. For InterComm products, on the Help main page, click the link for Copyright 2007. For other products, select Help > About on the main menu for the product.

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 252.2277013 (OCT’88) or Commercial Computer Software-Restricted Rights at FAR 52.22719(c)(1)-(2) (JUN’87), as applicable. 02202007

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

Arbortext IsoView 7.0 Functional Reference	9
Embedding into HTML documents	9
General Functions	11
OpenFile	11
Iso3OpenFile	12
CloseFile	13
AboutBox	13
GetVersion	13
Iso3GetLicensee	14
SetPassword	14
SetZoomRange	14
ShutDownServer	15
Configuration Functions	17
ConfigEvents	17
ConfigTools	18
ConfigTools2	18
Iso7ConfigTools	19
Iso4SetPreference	20
Iso4GetPreference	21
Iso4EditPreferences	21
Print Functions	23
ConfigPrinting	23
DoPrint	24
SetPrintStrings	25
SetPrintStringStyle	25
Redlining Functions	27
Iso4StartRedlining	27
Iso4StopRedlining	27
Iso4ExportRedlining	27
Iso5ImportRedlining	28
Viewport Functions	29
GetFirstViewPort	29
GetNextViewPort	29
DisposeViewPortList	29
GetViewPortName	30
GoHome	31
SetHome	31
Iso3SetHome	32
GetObjectViewPortFile	32
GetObjectViewPortName	32

Layer Functions	35
GetLayerCount	35
GetLayerName	35
GetLayerVisibility	36
LayerHasObjects	36
SetLayerVisibility	36
DisposeLayerList	37
Context Menu Functions	39
CreateContextMenu	39
AddContextMenuItem	39
EnableContextMenuItem	40
DoPopupMenu	41
DisposeContextMenu	41
Rotation Functions	43
Iso3GetRotAngle	43
Iso3SetRotAngle	43
Animation Functions	45
Iso7StartAnimation	45
Iso7StopAnimation	46
Iso7ConvertAnimationMetadata	46
Magnifier Functions	47
Iso4ConfigMagnifier	47
Iso4OpenMagnifier	48
Iso4CloseMagnifier	48
Iso4GetMagnifierState	48
Iso4GetMagnifierScale	49
Iso4SetMagnifierScale	49
GlobalView Functions	51
Iso4ConfigGlobalView	51
Iso4OpenGlobalView	52
Iso4CloseGlobalView	52
Iso4GetGlobalViewState	52
Object Functions	53
General Functions	53
GetObjectCount	53
Iso3GetObjectCount	53

Iso3ShowObjects	53
GetObjectID	54
Iso3GetObjectID	55
Iso3GetObjectIDByIndex	55
GetObjectName	55
Iso3GetObjectName	56
Iso3GetObjectNameByIndex	57
GetObjectDataString	57
Iso3GetObjectLayer	58
HighlightHotspot	58
Iso3HighlightObject	59
DisposeObjectList	60
Object Tip	61
GetObjectTip	61
Iso3GetObjectTip	61
SetObjectTip	62
Iso3SetObjectTip	62
GetObjectTipColor	62
SetObjectTipColor	63
SetObjectTipStyle	63
Iso3SetObjectTipStyle	64
Object Attributes	66
Iso3GetAttrCount	66
Iso3GetAttrByIndex	66
Iso3GetAttrType	67
Iso3GetAttrValue	67
Iso3GetFloatAttrValue	68
Iso3SetFloatAttrValue	68
Iso3GetIntegerAttrValue	69
Iso3SetIntegerAttrValue	69
Iso3GetStringAttrValue	70
Iso3SetStringAttrValue	70
Iso3SetGrAttribute	71
Traversing of Object Tree	73
Iso3GetRootObject	73
Iso3GetChildCount	73
Iso3GetFirstChild	73
Iso3GetParent	74
Iso3GetNextSibling	74
Iso3GetPreviousSibling	74
Object Links	76
Iso3CountLinks	76
Iso3GetLinkByIndex	76
Iso3SelectLink	77
Properties	79

AllowJumps	79
FileName	79
HsVisible	79
Preview	80
PropertyChangeMode	80
ResizeToFit	80
ViewPort	81
ViewPortCount	81
ViewSize	82
AcceptDroppedFiles	82
ActiveTool	83
ExtentX	83
ExtentY	83
HomeFileName	84
HomeViewPort	84
ObjectTipState	84
OffsetX	85
OffsetY	85
Events	87
ContextMenuItemHit	87
FileChanged	87
HsHit	88
InitFinished	89
KeyDown	89
KeyPress	89
KeyUp	90
MouseDown	90
MouseMove	90
MouseUp	91
PropertyChanged	91
ObjectHit	91
ViewWindowChanged	92
Appendix	95
Parameter	95
Object identifier	95
objExpr	95
Extended Object Expression	96
WebCGM 2.0 DOM	98
Toolbar	99
Redlining	100
Finding Elements on the Illustration	100
Constants	101
tool set/selection constants – <i>ConfigTools2</i> , <i>ActiveTool</i>	101
tool set/selection constants – <i>Iso7ConfigTools</i> , <i>ActiveTool</i>	102
Tip constants - <i>Object Tip</i>	103
Object highlighting constants - <i>Iso3HighlightObject</i>	104
Event configuration constants - <i>ConfigEvents</i>	104
Print configuration constants – <i>ConfigPrinting</i> , <i>DoPrint</i>	104

Overview of preference identifiers - <i>Iso4SetPreference</i>	105
Overview of the error numbers	106
Unicode support	106
List of viewable file formats	106
List of required files for use	107
How to prepare IsoView for distribution	107
Annotations	108

Embedding into HTML documents

When embedding the ActiveX Control (e.g. in HTML documents) it is necessary to indicate the class ID or the Object Type.

```
classid= "CLSID:865B2280-2B71-11D1-BC01-  
006097AC382A"  
or  
type="application/x-isoview"
```

For use in Internet Explorer we strongly recommend the use of the OBJECT-tag, otherwise formatting and scaling problems might occur when printing HTML-documents with embedded IsoView objects.

```
<object id="ivx1" classid="CLSID:865B2280-2B71-  
11D1-BC01-006097AC382A" width="400"  
height="300">  
<param name="src" value="sample.iso">  
</object>
```

or

```
<object id="ivx1" type="application/x-isoview"  
width="400" height="300">  
<param name="src" value="sample.iso">  
</object>
```

The following parameters may be used:

mandatory:

width: *width of the illustration in pixels*

height: *height of the illustration in pixels*

optional:

classid or type: *enforces usage of IsoView, mandatory if no src file is specified*

id: *id of the element (for use with JavaScript)*

border: *border (1 or 0)*

src: *filename of the illustration*

view: *name of the home view*

tools: *toolbar (1 or 0)*

events: *enable/disable events (see ConfigEvents)*

Microsoft released a patch to the OS and MS-IE which affect the handling of ActiveX controls. If an ActiveX Control is embedded into an HTML page utilizing the tags APPLET, EMBED, OBJECT as is usual, it will now 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’ the ActiveX control needs to be inserted dynamically.

```
<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’

```
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>');
```

OpenFile

Note: This method is deprecated and supported for backward compatibility purposes only. Please use `Iso3OpenFile` instead.

This 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*). To set a new viewport in the current open illustration use an empty string “” as a `FileName`.

```
OpenFile (lpszFileName, lpszView)
```

return value `BOOL`

The `lpszFileName` parameter defines the file name resp. URL of the illustration to be opened as `BSTR`.

The `lpszView` parameter defines the name of the view (ViewPort) as `BSTR`. `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 `$OBJ_` resp. `$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 “`$OBJ_HS1`” resp. “`$NAME_HS`”.*

*`$ID_ObjectID`
where `$ID_` precedes the ID of the object you want to show and `ObjectID` is the object’s ID. If the name of the object is `2006` the string would be “`$ID_2006`”.*

*`$EXTENT`
use this name to instruct `IsoView` to build a viewport that fits around the extent of the file. You can use the `OpenFile` command to only change the viewport. In this case pass an empty string for the filename.*

```
document.ixv1.OpenFile("http://www.ptc.com/  
demo/test.iso", "View1");
```

First introduced with IsoView 1.0

Iso3OpenFile

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

```
Iso3OpenFile (url)
```

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

The url parameter defines the url which should be opened as BSTR.
Allowed URLs are:

http.../abc.iso (absolute URL)
abc.iso (rel URL, if a file is open only)
abc.iso#fragment (fragment specifies object or viewport)
#fragment (fragment points to current file)

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

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

Fragment samples using the ID of the object:

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

Fragment samples using the name of the object:

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

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

First introduced with IsoView 3.0

CloseFile

This method closes the current illustration.

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

First introduced with IsoView 2.0

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.

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

First introduced with IsoView 1.0

GetVersion

This method retrieves the version number of the plugin.

```
GetVersion ()
```

This method returns the version number as a BSTR.

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

First introduced with IsoView 2.0

Iso3GetLicensee

returns the string identifying the licensee of IsoView

```
Iso3GetLicensee ()
```

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

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

First introduced with IsoView 3.0

SetPassword

Use this method to enable IsoView to read password protected IsoDraw-Files.

```
SetPassword (Password)
```

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

The *Password* parameter defines the password to open protected files as BSTR.

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

First introduced with IsoView 2.02

SetZoomRange

Use this method to limit the view size (set using *ViewSize*)

```
SetZoomRange (minVal, maxVal)
```

The *minVal* parameter defines the lower range for zoom factor (min. 1%) as double.

The *maxVal* parameter defines upper range for zoom factor (max. 6400%) as double.

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

First introduced with IsoView 2.02

ShutDownServer

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

```
ShutDownServer ()  
  
document.ivx1.ShutDownServer()
```

Please note that NO further processing is possible after this command. Thus use this method only when your container process is completed!

First introduced with IsoView 1.0

ConfigEvents

This method allows you to select which events (see *Events*) will be generated by IsoView. Disabling unneeded events will increase performance. See appendix for constant values (*Event configuration constants - ConfigEvents*).

```
ConfigEvents (events)
```

The *events* parameter defines which events are enabled as long. Allowed values are:

```
1: HSHIT
2: FILECHANGED
4: CONTEXTMENUITEMHIT
8: PROPERTYCHANGED
16: INITFINISHED
32: OBJECTHIT
64: VIEWWINDOWCHANGED
1024: MOUSEMOVE
2048: MOUSEDOWN
4096: MOUSEUP
8192: DONTHANDLECLICKS
which prevents IsoView from internal click handling (ie. from
bringing up a context menu), must always be combined with
"MOUSEMOVE" + "MOUSEDOWN"
```

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

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

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

First introduced with IsoView 2.0

ConfigTools

Note: This method is deprecated and supported for backward compatibility purposes only. Please use `Iso7ConfigTools` instead.

This method allows you to configure the *Toolbar*.

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

The *bWithTools* parameter defines if the Tool bar is available as boolean (default: true).

The *bAsPopup* parameter defines if the Tool bar is displayed as popup menu as boolean (not supported at this time).

The *bTop* parameter defines if the Tool bar is displayed on the top window border as boolean (default: false).

The *bRight* parameter defines if the Tool bar is displayed right aligned as boolean (default: false).

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

First introduced with IsoView 1.0

ConfigTools2

Note: This method is deprecated and supported for backward compatibility purposes only. Please use `Iso7ConfigTools` instead.

This method allows you to configure the *Toolbar*.

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

The *ToolMode* parameter sets the mode of the IsoView tools as long. Allowed values are:

0: tools disabled
1: default tools
2: force all tools as popup
4: toolbar invisible, select using *ActiveTools*
8: auto hide Redlining tools if not in Redlining mode

Use *ActiveTool* to switch between tools.

The *ToolSet* parameter configures the available tools as long. See appendix (*tool set/selection constants – ConfigTools2, ActiveTool*) for allowed values.

The *SelectedTool* parameter selects the active tool as long. See appendix (*tool set/selection constants – ConfigTools2, ActiveTool*) for allowed values.

The *Top* parameter defines if the toolbar is anchored on the top of the window as boolean. The default value is false.

The *Right* parameter defines if the toolbar is right aligned as boolean. The default value is false.

```
// 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  
document.ivx1.ConfigTools2(1, 1+2+4+16, 1, true,  
false);
```

Introduced first with IsoView 2.0

Iso7ConfigTools

This method allows you to configure the *Toolbar*. This is the enhanced version of the *ConfigTools* and *ConfigTools2* methods.

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

The *nToolMode* parameter sets the mode of the IsoView tools as long. Allowed values are:

0: *tools disabled*
1: *default tools*
2: *force all tools as popup*
4: *toolbar invisible, select using ActiveTools*
8: *auto hide Redlining tools if not in Redlining mode*

Use ActiveTool to switch between tools.

The `nBtnSet` parameter configures the available tools as long. See appendix (*tool set/selection constants – Iso7ConfigTools, ActiveTool*) for allowed values.

The `nPopSet` parameter configures the available tools in the popup menu as long. See appendix (*tool set/selection constants – Iso7ConfigTools, ActiveTool*) for allowed values.

The `nSelectedTool` parameter selects the active tool as long. See appendix (*tool set/selection constants – Iso7ConfigTools, ActiveTool*) for allowed values.

The `nTop` parameter defines if the toolbar is anchored on the top of the window as boolean. The default value is false.

The `nRight` parameter defines if the toolbar is right aligned as boolean. The default value is false.

```
// 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 tool  
document.ivx1.Iso7ConfigTools(1, 1+2+4+16,  
32+64+128, 1, true, false);
```

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

Introduced first with IsoView 7.0

Iso4SetPreference

Set the data of a given preference. See appendix for constant values (*Overview of preference identifiers - Iso4SetPreference*).

```
Iso4SetPreference (prefName, prefValue)
```

The *prefName* parameter defines the name of the preference item as BSTR. (see appendix: *Overview of preference identifiers - Iso4SetPreference*)

The *prefValue* parameter defines the new data of the given preference as BSTR.

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

First introduced with IsoView 4.0

Iso4GetPreference

retrieve the data of a given preference item

```
Iso4GetPreference (prefName)
```

This method returns the preference data as BSTR. (see appendix: *Overview of preference identifiers - Iso4SetPreference*)

```
pref1 =  
document.ivx1.Iso4GetPreferences("default_tools  
et");
```

First introduced with IsoView 4.0

Iso4EditPreferences

opens the IsoView preferences dialog

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

First introduced with IsoView 4.0

ConfigPrinting

This method allows you to set preferences for printing. See appendix for constant values (*Print configuration constants – ConfigPrinting, DoPrint*).

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

return value BOOL

The *OnlyVisContent* parameter defines if only the visible content will be printed as short. Allowed values are:

- 1: *KEEPPRINTERSETTINGS*
- 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))

The *SplitPages* parameter defines if the illustration will be splitted as short. Allowed values are:

- 1: *KEEPPRINTERSETTINGS*
- 0: *FITONPAGE* (the illustration will be fitted onto one page (default))
- 1: *SPLITPAGES* (the illustration will be split across multiple pages if necessary)

The *Orientation* parameter selects the orientation of printed page as short. Allowed values are:

- 1: *KEEPPRINTERSETTINGS*
- 1: *PORTRAIT*
- 2: *LANDSCAPE*

The *Scale* parameter defines the scale of the printed illustration as double. Value must be greater than zero or -1 for “KEEPPRINTERSETTINGS”.

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

Introduced first with IsoView 2.0

DoPrint

This method prints the open illustration to the default printer. You can set additional information using *SetPrintStrings*. See appendix for constant values (*Print configuration constants – ConfigPrinting, DoPrint*).

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

The *OnlyVisContent* parameter defines if only the visible content will be printed as short. Allowed values are:

- 1: *KEEPPRINTERSETTINGS*
- 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))

The *SplitPages* parameter defines if the illustration will be splitted as short. Allowed values are:

- 1: *KEEPPRINTERSETTINGS*
- 0: *FITONPAGE* (the illustration will be fitted onto one page (default))
- 1: *SPLITPAGES* (the illustration will be split across multiple pages if necessary)

The *Orientation* parameter selects the orientation of printed page as short. Allowed values are:

- 1: *KEEPPRINTERSETTINGS*
- 1: *PORTRAIT*
- 2: *LANDSCAPE*

The *Scale* parameter defines the scale of the printed illustration as double. Value must be greater than zero or -1 for “KEEPPRINTERSETTINGS”.

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

Introduced first with IsoView 2.0

SetPrintStrings

Use this method to set the text for the header and footer strings placed on a printed page. Use *SetPrintStringStyle* to configure the style and position of these strings.

```
SetPrintStrings (Header, Footer);
```

The *Header* parameter defines the text for page header as BSTR.

The *Footer* parameter defines the text for the page footer as BSTR. Within this text, the following macros can be used:

\$PATH, filename and path of the illustration

\$NAME, filename of the illustration

\$DATE, current date

\$TIME, current time

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

First introduced with IsoView 2.0

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.

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

The *Target* parameter defines the target for the text on the printed page as short. Allowed values are:

-
- 1: *HEADER*
 - 2: *FOOTER*
 - 3: *HEADER AND FOOTER*

The *Font* parameter defines the display name of the font as BSTR.

The *Size* parameter defines the size of the text in pt as short.

The *Style* parameter defines the style of the text as short. Allowed values are:

- 0: *NORMAL*
- 1: *BOLD*
- 2: *ITALIC*
- 3: *BOLDITALIC*

Add 256 if you want IsoView to keep the tip window outside the object's area!

The *Alignment* parameter defines the alignment of the text as short. Allowed values are:

- 0: *LEFT*
- 1: *CENTER*
- 2: *RIGHT*

The *TopMargin* parameter defines the margin (in mm) between top of the illustration and the upper border of the printable area (KEEPSETTINGS: -1) as double.

The *BotMargin* parameter defines the margin (in mm) between bottom of the illustration and the lower border of the printable area (KEEPSETTINGS: -1) as double.

The *TopGap* parameter defines the gap (in mm) between top of the illustration and the lower border of the header (KEEPSETTINGS: -1) as double.

The *BotGap* parameter defines the gap (in mm) between bottom of the illustration and the upper border of the footer (KEEPSETTINGS: -1) as double.

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

First introduced with IsoView 2.0

Iso4StartRedlining

starts the Redlining mode - Redlining tools can only be selected if Redlining mode is enabled.

```
Iso4StartRedlining ()
```

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

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

This is controlled through the *Iso7ConfigTools* settings. If the Redlining tools are disabled the Redlining mode cannot be started through this method.

First introduced with IsoView 4.0

Iso4StopRedlining

stops the Redlining mode

```
Iso4StopRedlining ()
```

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

First introduced with IsoView 4.0

Iso4ExportRedlining

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

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

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

The *illuFile* parameter defines the destination path and name of the illustration as BSTR.

The *rlFile* parameter defines the destination path and name of the redlining file as BSTR.

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

First introduced with IsoView 4.0

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(...)

```
Iso5ImportRedlining (rlFile)
```

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

The *rlFile* parameter defines the destination path and name of the redlining file as BSTR.

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

First introduced with IsoView 5.0

GetFirstViewPort

This method provides the name of the first viewport in the graphic (all further names can be determined by *GetNextViewPort*).

```
GetFirstViewPort ()
```

The method returns the name of the first viewport as BSTR.

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

First introduced with IsoView 1.0

GetNextViewPort

This method provides the name of the next viewport in the graphic (the first name can be obtained by selecting *GetFirstViewPort*).

```
GetNextViewPort ()
```

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

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

First introduced with IsoView 1.0

DisposeViewPortList

This method disposes the viewport list. This list generated by calls to *GetViewPortName*. Once you have retrieved your preferred viewport information it is recommended to call *DisposeViewPortList* to free the used memory.

```
DisposeViewPortList ()
```

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

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

This method is obsolete since version 3.0.

First introduced with IsoView 1.0

GetViewPortName

This method retrieves the name of the viewport with the given index. Use *ViewPortCount* to get the number of viewports in the illustration. After you have retrieved your preferred viewport information it is recommended to call *DisposeViewPortList* to free the memory used by IsoView to keep the layer information for all viewports. This list will be rebuilt after the next call to *GetViewPortName*.

```
GetViewPortName (Index)
```

This method returns the name of this viewport as BSTR.

The *Index* parameter defines the index of the viewport as long (0 based).

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

First introduced with IsoView 2.0

GoHome

This method instructs IsoView to reset the view to the home viewport. This is the first viewport set with *OpenFile*, or the one set by *SetHome*.

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

First introduced with IsoView 2.0

SetHome

Note: This method is deprecated and supported for backward compatibility purposes only. Please use Iso3SetHome instead.

Sets the file and the name of the home viewport.

```
SetHome (FileName, View)
```

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

The *FileName* parameter defines the file name resp. URL of the illustration as BSTR.

The *View* parameter defines the name of the home view port as BSTR.

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

First introduced with IsoView 2.0

Iso3SetHome

Set the home location. May use a fragment.

```
Iso3SetHome (url)
```

The url parameter defines the location as BSTR.

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

First introduced with IsoView 3.0

GetObjectViewPortFile

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

```
GetObjectViewPortFile (ID, Name)
```

The method returns the Object Viewport Filename as BSTR.

The *ID* parameter defines the id of the object as long.

The *Name* parameter defines the name of the object as BSTR.

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

First introduced with IsoView 2.02

GetObjectViewPortName

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

```
GetObjectViewPortName (ID, Name)
```

The method returns the Object Viewport Name as BSTR.

The *ID* parameter defines the id of the object as long.

The *Name* parameter defines the name of the object as BSTR.

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

First introduced with IsoView 2.02

GetLayerCount

Use this method to retrieve the number of layers in the current illustration. To retrieve further layer information you can use *GetLayerName*, *GetLayerVisibility*.

```
GetLayerCount ()
```

This method returns the number of layers as long.

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

First introduced with IsoView 2.0

GetLayerName

This method retrieves the name of the layer with the given index. Use *GetLayerCount* to get the number of layers in the illustration. After you have retrieved your preferred layer information it is recommended to call *DisposeLayerList* to free the memory used by IsoView to keep the layer information for all layers. This list will be rebuilt after the next call to *GetLayerName*.

```
GetLayerName (index)
```

This method returns the name of the layer as BSTR.

The *index* parameter defines the index of the layer as long (0 based).

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

First introduced with IsoView 2.0

GetLayerVisibility

This method retrieves the visibility of the layer with the given name. Use *SetLayerVisibility* to modify the visibility of a layer. Use *GetLayerName* to get the name of an layer.

```
GetLayerVisibility (Name)
```

The method returns true if layer is visible, false if not.

The *Name* parameter defines the name of the layer retrieve visibility from as BSTR.

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

First introduced with IsoView 2.0

LayerHasObjects

Use this method to retrieve if the given layer has objects (see *Object Functions*).

```
LayerHasObjects (Name)
```

This method returns true if the given layer contains objects.

The *Name* parameter defines the layer name as BSTR.

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

First introduced with IsoView 2.0

SetLayerVisibility

Use this method to modify the visibility of a layer in the illustration.

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

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

The *Name* parameter defines the name of the layer as BSTR.

To define if the layer should be visible select true for the *IsVisible* parameter, false otherwise

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

First introduced with IsoView 2.0

DisposeLayerList

This method disposes the layer list. This list generated by calls to *GetLayerName*. Once you have retrieved your preferred layer information it is recommended to call *DisposeLayerList* to free the used memory.

```
DisposeLayerList()
```

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

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

This method is obsolete since version 3.0.

Introduced first with IsoView 2.0

CreateContextMenu

This method activates a context menu which has been created previously using *AddContextMenuItem*. This menu is from then on opened and evaluated upon a right mouse button click. The menu item selected by the user is reported back by *ContextMenuHit*.

```
CreateContextMenu()
```

The method returns the number of menu items as short.

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

First introduced with IsoView 1.0

AddContextMenuItem

This method adds a new context menu item to the list. A context menu created in this way is enabled by the method *CreateContextMenu*. No additional menu items can be added to an existing context menu. It has to be disposed first by *DisposeContextMenu*.

```
addContextMenuitem (ippszTitle, bEnabled)
```

return value short

The *lpszTitle* parameter sets the Title (text) of the menu item as BSTR.

The *bEnabled* parameter defines if the Menu item is enabled as boolean.

```
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();
```

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

First introduced with IsoView 1.0

EnableContextMenuItem

This method enables resp. disables menu items added by *AddContextMenuItem*. (This is also possible after the menu has been created by *CreateContextMenu*.)

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

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

The *item* parameter sets the number of the menu item (base = 1) as short.

The *enabled* parameter defines if the menu item is enabled as boolean.

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

First introduced with IsoView 1.0

DoPopupMenu

This method opens an existing context menu at the specified position. You can set the context menu items using *AddContextMenuItems* / *CreateContextMenu*.

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

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

The *x* parameter defines the horizontal position for the upper left menu corner in pixels as long.

The *y* parameter defines the vertical position for the upper left menu corner in pixels as long.

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

First introduced with IsoView 2.0

DisposeContextMenu

This method disposes a context menu. All menu items previously inserted by *AddContextMenuItems* will be removed.

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

First introduced with IsoView 1.0

Iso3GetRotAngle

get 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.

```
Iso3GetRotAngle ()
```

This method returns the rotation angle in degrees as double.

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

First introduced with IsoView 3.0

Iso3SetRotAngle

set 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.

```
Iso3SetRotAngle (newValue)
```

The *newValue* parameter defines the rotation angle in degrees as double.

```
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.

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

First introduced with IsoView 3.0

Iso7StartAnimation

Used to start a named animation sequence of an object.

```
BOOL Iso7StartAnimation (object, sequence,  
offset);
```

The *object* parameter defines the the object identifier as BSTR.

The *sequence* parameter defines the name of the sequence which should be started (or stopped) as string. By using the keyword “\$STOP” instead of the name of a sequence you can stop all running animations of the defined object.

The *offset* parameter defines the offset in seconds after which the animation will be started.

```
// starts SEQ_1 on object with ID "Z_01"  
result = ivx1.Iso7StartAnimation ("Z_01",  
"SEQ_1", 0);  
  
// starts SEQ_1 on object with ID "Z_02"  
result = ivx1.Iso7StartAnimation ("id(Z_02)",  
"SEQ_1", 0);  
  
// starts SEQ_1 on all objects with name "blue"  
result = ivx1.Iso7StartAnimation ("name(blue)",  
"SEQ_1", 0);  
  
// stops all animations on object with ID "Z_01"  
result = ivx1.Iso7StartAnimation ("Z_01",  
"$STOP", 0);  
  
// stops all animations on objects with name  
"blue"  
result = ivx1.Iso7StartAnimation ("name(blue)",  
"$STOP", 0);
```

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

First introduced with IsoView 7.0

Iso7StopAnimation

Used to stop all running animations and optionally restore them to the state before animation.

```
Iso7StopAnimations (bRestore);
```

The *bRestore* parameter defines if the animation should be restored to the state before starting it as Boolean.

```
ivx1.Iso7StopAnimations (false)
```

First introduced with IsoView 7.0

Iso7ConvertAnimationMetadata

Used to convert animation metadata added using DOM calls into iso animation format (required before running the animation).

```
Iso7ConvertAnimationMetadata ();
```

```
ivx1.Iso7ConvertAnimationMetadata ();
```

First introduced with IsoView 7.0

Iso4ConfigMagnifier

configure the display and position of the global view window

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

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

The *styleFlags* parameter defines the magnifier style as long.
Allowed values are:

- 1: *docked*
- 2: *show slider*
- 4: *show in b/w*
- 256: *top left*
- 512: *top right*
- 768: *bottom left*
- 1024: *bottom right*

add values to combine different attributes

The *width* parameter defines the width of the magnifier window in pixels as long.

The *height* parameter defines the height of the magnifier window in pixels as long.

The *scale* defines the initial scale used in the magnifier window as double.

The *minScale* parameter defines the minimum scale (used for slider) as double.

The *maxScale* parameter defines the maximum scale (used for slider) as double.

```
document.ixv1.Iso4ConfigGlobalView(257,100,100,  
400,200,1200);
```

First introduced with IsoView 4.0

Iso4OpenMagnifier

opens the magnifier window

```
Iso4OpenMagnifier ()
```

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

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

see Iso4ConfigMagnifier for configuration options

First introduced with IsoView 4.0

Iso4CloseMagnifier

closes the magnifier window

```
Iso4CloseMagnifier ()
```

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

First introduced with IsoView 4.0

Iso4GetMagnifierState

retrieves the current state of the magnifier window

```
Iso4GetMagnifierState ()
```

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

First introduced with IsoView 4.0

Iso4GetMagnifierScale

retrieves the current scale of the magnifier window

```
Iso4GetMagnifierScale ()
```

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

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

First introduced with IsoView 4.0

Iso4SetMagnifierScale

set the scale of magnifier window

```
Iso4SetMagnifierScale (Scale)
```

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

The *Scale* parameter defines the scale of the magnifier window as double. The scale scope lies between 1 and 6400%.

```
document.ivx1.Iso4SetMagnifierScale(206);
```

First introduced with IsoView 4.0

Iso4ConfigGlobalView

configure the display and position of the global view window

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

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

The *styleFlags* parameter defines the global view style as long.
Allowed values are:

1: *docked*
4: *show in b/w*
256: *top left*
512: *top right*
768: *bottom left*
1024: *bottom right*

add values to combine different attributes

The *width* parameter defines the width of the global view window in pixels as long.

The *height* parameter defines the height of the global view window in pixels as long.

The *useStroke* parameter is TRUE if stroke rect indicates current view position, otherwise a filled rect is used.

The *strokeWidth* defines the line width of the stroke rect (if chosen) as long.

The *color* parameter defines the color of the stroke/fill rect as OLE_COLOR.

```
document.ixv1.Iso4ConfigGlobalView(257,100,100,  
TRUE,2)
```

First introduced with IsoView 4.0

Iso4OpenGlobalView

opens the global view window

```
Iso4OpenGlobalView ()
```

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

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

see Iso4ConfigGlobalView for configuration options

First introduced with IsoView 4.0

Iso4CloseGlobalView

closes the global view window

```
Iso4CloseGlobalView ()
```

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

First introduced with IsoView 4.0

Iso4GetGlobalViewState

retrieves the current state of the global view window

```
Iso4GetGlobalViewState ()
```

This method returns a value indicating the settings of the global view window (see Iso4ConfigGlobalView) as long.

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

First introduced with IsoView 4.0

General Functions

GetObjectCount

Note: This method is deprecated and supported for backward compatibility purposes only. Please use `Iso3GetObjectCount` instead.

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

```
GetObjectCount ()
```

This method returns the number of objects as long.

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

First introduced with IsoView 2.0

Iso3GetObjectCount

get the number of objects in an illustration

```
Iso3GetObjectCount ()
```

This method returns the number of objects in an illustration as long.

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

First introduced with IsoView 3.0

Iso3ShowObjects

set the kind of object to be shown

```
Iso3ShowObjects (bShowObjects, bShowHotSpot)
```

Set the *bShowObjects* parameter to TRUE if objects without hotspots shall be highlighted.

Set the *bShowHotSpot* parameter to TRUE if objects with hotspots shall be highlighted.

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

First introduced with IsoView 3.0

GetObjectID

Note: This method is deprecated and supported for backward compatibility purposes only. Please use Iso3GetObjectID instead.

This method retrieves the ID of the object with the given index. 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 IsoView to keep the object information of all objects. This list will be rebuilt after the next call to *GetObjectID* resp. *GetObjectName*.

```
GetObjectID (Index)
```

This method returns the ID of the object as long.

The *Index* parameter defines the index of the object as long (0 based).

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

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

document.ivx1.DisposeObjectList();
```

First introduced with IsoView 2.0

Iso3GetObjectID

returns ID of an object

```
Iso3GetObjectID (object)
```

This method returns the ID of an object as BSTR.

The *object* parameter defines the object identifier as BSTR.

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

This function returns the plain ID of the object, not a complete object identifier including the "id(...)" part.

First introduced with IsoView 3.0

Iso3GetObjectIDByIndex

returns ID of an object

```
Iso3GetObjectIDByIndex (index)
```

This method returns the ID of the indexed object as BSTR.

The *index* parameter defines the index of the object as long (0-based).

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

This function returns the plain ID of the object, not a complete object identifier including the "id(...)" part.

First introduced with IsoView 3.0

GetObjectName

Note: This method is deprecated and supported for backward compatibility purposes only. Please use Iso3GetObjectName instead.

This method retrieves the name of the object with the given index. 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 IsoView to keep the object information of all objects. This list will be rebuilt after the next call to *GetObjectName* resp. *GetObjectID*.

```
GetObjectName (Index)
```

This method returns the name of the object as BSTR.

The *Index* parameter defines the index of the object as long (0 based).

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

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

document.ivx1.DisposeObjectList();
```

First introduced with IsoView 2.0

Iso3GetObjectName

returns name of an object

```
Iso3GetObjectName (object)
```

The method returns the name of an object as BSTR.

The *object* parameter defines the object identifier as BSTR.

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

This function returns the plain name of the object, not a complete object identifier including the "name(...)" part.

First introduced with IsoView 3.0

Iso3GetObjectNameByIndex

returns the name of an object

```
Iso3GetObjectNameByIndex (index)
```

This method returns the name of the indexed object as BSTR.

The *index* parameter defines the index of the object as long (0-based).

```
strName =  
document.ivx1.Iso3GetObjectNameByIndex(n);
```

This function returns the plain ID of the object, not a complete object identifier including the "name(...)" part.

First introduced with IsoView 3.0

GetObjectDataString

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

```
GetObjectDataString (ID, Name)
```

This method returns the Object data property as a BSTR.

The *ID* parameter defines the id of the object as long.

The *Name* parameter defines the name of the object as BSTR.

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

First introduced with IsoView 2.0

Iso3GetObjectLayer

returns the layer of this object (see *Layer Functions*)

```
Iso3GetObjectLayer (object)
```

The method returns the layer name as BSTR.

The *object* parameter defines the object identifier as BSTR.

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

First introduced with IsoView 3.0

HighlightHotspot

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

```
HighlightHotspot (lID, lpszName, lFlags,  
oleColor)
```

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

The *lID* parameter defines the ID of the hotspot (ID field in IsoDraw object info, usage not recommended) as long.

The *lpszName* parameter defines the name of the hotspot as string.

The *lFlags* parameter defines the display attributes as long. Allowed values are:

```
0: HIDE  
2: FRAME  
4: FILL  
8: FLASH  
16: CENTER (not yet supported)
```

To process ALL hotspots use ID = 32. To combine attributes add the different values.

The *olcColor* parameter 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 standing for the colors red, green and blue in a range between 0 and 255.

```
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>
```

First introduced with IsoView 1.0

Iso3HighlightObject

Highlight an object. See appendix for constant values (*Object highlighting constants - Iso3HighlightObject*).

```
Iso3HighlightObject (objExpr, flags, color)
```

The *objExpr* paramter defines the object expression identifying for one or multiple objects as BSTR.

The *flags* parameter defines the display attributes long. Allowed values are:

0: *HIDE*
2: *FRAME*
4: *FILL*
8: *FLASH*

Add values to combine attributes.

The color 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 standing for the colors red, green and blue in a range between 0 and 255.

```
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>
```

First introduced with IsoView 3.0

DisposeObjectList

This method disposes the object list. This list generated by calls to *GetObjectID/GetObjectName*. Once you have retrieved your preferred object information it is recommended to call *DisposeObjectList* to free the used memory.

```
DisposeObjectList ()
```

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

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

This method is obsolete since version 3.0.

Introduced first with IsoView 2.0.

Object Tip

See appendix for constant values (*Tip constants - Object Tip*).

GetObjectTip

Note: This method is deprecated and supported for backward compatibility purposes only. Please use `Iso3GetObjectTip` instead.

This method retrieves the text used in object tips for this object.

```
GetObjectTip (ID, Name)
```

This method returns the tip text of the object as BSTR.

The *ID* parameter defines the id of the object as long.

The *Name* parameter defines the name of the object as BSTR.

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

First introduced with IsoView 2.0

Iso3GetObjectTip

retrieve the object tip of this object

```
Iso3GetObjectTip (object)
```

This method returns the object tip text of the given object as BSTR.

The *object* parameter defines the object identifier as BSTR.

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

First introduced with IsoView 3.0

SetObjectTip

Note: This method is deprecated and supported for backward compatibility purposes only. Please use `Iso3SetObjectTip` instead.

Use this method to add tip information for an object in the illustration. The usage of these tips is controlled by *ObjectTipState*.

```
SetObjectTip (ID, Name, Tip)
```

The *ID* parameter defines the id of the object as long.

The *Name* parameter defines the name of the object as BSTR.

The *Tip* parameter defines the tip text of the object as BSTR.

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

First introduced with IsoView 2.0

Iso3SetObjectTip

set the object tip

```
Iso3SetObjectTip (object, tip)
```

The *object* parameter defines the object identifier as BSTR.

The *tip* parameter defines the text for the tip as BSTR.

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

First introduced with IsoView 3.0

GetObjectTipColor

This method retrieves the color used for the background or the text of object tips.

```
GetObjectTipColor (GetBackCol)
```

This method returns the RGB color as 32 bit unsigned integer.

The *GetBackCol* parameter defines which color is retrieved as BOOL. If true, the background color is retrieved; if false, the text color.

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

First introduced with IsoView 2.0

SetObjectTipColor

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

```
SetObjectTipColor (Background, Text)
```

The *Background* parameter defines the background color as 32 bit unsigned integer.

The *Text* parameter defines the text color as 32 bit unsigned integer.

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

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

First introduced with IsoView 2.0

SetObjectTipStyle

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

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

The *Font* parameter defines the display name of the font as BSTR.

The *Size* parameter defines the size of the text in pt as short.

The *Style* parameter defines the style of the text as short. Allowed values are:

- 0: *NORMAL*
- 1: *BOLD*
- 2: *ITALIC*
- 3: *BOLDITALIC*

Add 256 if you want IsoView to keep the tip window outside the object's area!

The *Alignment* parameter defines the alignment of the text as short. Allowed values are 0 for “LEFT”, 1 for “CENTER” and 2 for “RIGHT”.

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

First introduced with IsoView 2.0

Iso3SetObjectTipStyle

set the style of the object tips

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

The *Font* parameter defines the Display name of the font as BSTR.

The *Size* parameter defines the Size of the text in pt as short.

The *Style* parameter defines The style of the text as short. Allowed values are:

- 0: *NORMAL*
- 1: *BOLD*
- 2: *ITALIC*
- 3: *BOLDITALIC*

add 256 if you want IsoView to keep the tip window outside the object's area

The *Alignment* parameter defines the alignment of the Text as short. Allowed values are:

0: *LEFT*
1: *CENTER*
2: *RIGHT*

The *Delay* parameter defines the delay before a tip comes up (in 1/10 of a second) as short.

The *ShowTime* parameter defines the time before a tip disappears automatically (in 1/10 of a second) as short.

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

First introduced with IsoView 3.0

Object Attributes

Iso3GetAttrCount

returns the number of attributes of an object

```
Iso3GetAttrCount (object)
```

This method returns the number of attributes of an object as long.

The *object* parameter defines the object identifier as BSTR.

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

First introduced with IsoView 3.0

Iso3GetAttrByIndex

returns the name of an attribute

```
Iso3GetAttrByIndex(object, index)
```

The method returns the attribute name as BSTR.

The *object* parameter defines the object identifier as BSTR.

The *index* parameter defines the attribute index as long (1-based).

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

You can get the actual count of attributes on an object using *Iso3GetAttrCount*.

First introduced with IsoView 3.0

Iso3GetAttrType

returns the type of an attribute

```
Iso3GetAttrType (object, attrName)
```

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

The *object* parameter defines the object identifier as BSTR.

The *attrName* parameter defines the attribute name as BSTR.

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

First introduced with IsoView 3.0

Iso3GetAttrValue

returns the value of an attribute as a string

```
Iso3GetAttrValue (object, attrName)
```

The method returns the value of the attribute as BSTR. The string is empty if the attribute doesn't exist.

The *object* parameter defines the object identifier as BSTR.

The *attrName* parameter defines the attribute name as BSTR.

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

First introduced with IsoView 3.0

Iso3GetFloatAttrValue

returns the value of a floating number attribute

```
Iso3GetFloatAttrValue (object, attrName)
```

This method returns the value of a floating point number attribute.

The *object* parameter defines the object identifier as BSTR.

The *attrName* parameter defines the attribute name as BSTR.

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

First introduced with IsoView 3.0

Iso3SetFloatAttrValue

set the value of a floating number attribute. If the attribute does not exist it will be added to the object.

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

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

The *object* parameter defines the object identifier as BSTR.

The *attrName* parameter defines the attribute name as BSTR.

The value parameter defines the new float value.

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

First introduced with IsoView 3.0

Iso3GetIntegerAttrValue

returns the value of an integer attribute

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

This method returns the value of an integer attribute as long.

The *object* parameter defines the object identifier as BSTR.

The *attrName* parameter defines the attribute name as BSTR.

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

First introduced with IsoView 3.0

Iso3SetIntegerAttrValue

set the value of an integer attribute. If the attribute does not exist it will be added to the object.

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

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

The *object* parameter defines the object identifier as BSTR.

The *attrName* parameter defines the attribute name as BSTR.

The *value* parameter defines the new longinteger value.

```
document.ivx1.Iso3SetIntegerAttrValue ("id (myObj  
1) ", "Count", 5
```

First introduced with IsoView 3.0

Iso3GetStringAttrValue

returns the value of a string attribute

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

This method returns the value of a string attribute as BSTR.

The *object* parameter defines the object identifier as BSTR.

The *attrName* parameter defines the attribute name as BSTR.

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

First introduced with IsoView 3.0

Iso3SetStringAttrValue

set the value of a string attribute. If the attribute does not exist it will be added to the object.

```
Iso3SetStringAttrValue (object, attrName, value)
```

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

The *object* parameter defines the object identifier as BSTR.

The *attrName* parameter defines the attribute name as BSTR.

The *value* parameter defines the new String value as BSTR.

```
document.ivx1.Iso3SetStringAttrValue ("id(myObj1  
)", "Description", "Bracket");
```

First introduced with IsoView 3.0

Iso3SetGrAttribute

set a graphic attribute for an object

```
Iso3SetGrAttribute (objExpr, attrName, value)
```

The *objExpr* paramter defines the object expression identifying for one or multiple objects as BSTR.

The *attrName* parameter defines the attribute name as BSTR. There are several predefined attribute names:

- stroke-width: *lineweight of stroked elements*
- stroke: *stroke color of stroked elements*
- fill: *fill color of filled elements*
- text-color: *text color*
- color: *sets stroke, fill, and text colors*
- backColor: *sets background color*
- visibility: *visibility of the object*
- greeked-size: *min. text size before greeking is applied (negative value disables greeking)*
- click-radius: *selection proximity for line hotspots*

The *value* parameter defines a new value for the attribute, empty string to reset original values.

Allowed values for stroke-width: *positive value in mm or percentage*

Allowed values for colors: *RGB values separated by commas or percentage*

Allowed values for visibility: *"hidden" to hide, all other values will show the object*

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

This method lets you change the appearance of an object and its children on screen. The content of the graphics file is not changed.

First introduced with IsoView 3.0

Traversing of Object Tree

Iso3GetRootObject

returns object identifier of root node

```
Iso3GetRootObject ()
```

This method returns the object identifier of the root node as BSTR.

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

First introduced with IsoView 3.0

Iso3GetChildCount

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

```
Iso3GetChildCount (object)
```

This method returns the count of children as long.

The *object* parameter defines the object identifier as BSTR.

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

First introduced with IsoView 3.0

Iso3GetFirstChild

returns an object identifier of the first child of object, if any. If object is the root the first layer will be returned. The object identifier always uses the ID of the object. (see *Iso3GetObjectID*)

```
Iso3GetFirstChild (object);
```

This method returns the object identifier of first child as BSTR.

The *object* parameter defines the object identifier as BSTR.

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

First introduced with IsoView 3.0

Iso3GetParent

returns object identifier of parent

```
Iso3GetParent (object)
```

This method returns the object identifier of the parent object. An empty string is returned for the root node.

The *object* parameter defines the object identifier as BSTR.

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

First introduced with IsoView 3.0

Iso3GetNextSibling

returns an object identifier of next sibling, if any. If object is a layer the next layer will be returned. The object identifier always uses the ID of the object. (see *Iso3GetObjectID*)

```
Iso3GetNextSibling (object)
```

This method returns an object identifier of next sibling as BSTR.

The *object* parameter defines the object identifier as BSTR.

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

First introduced with IsoView 3.0

Iso3GetPreviousSibling

returns object identifier of previous sibling. If the object is a layer the previous layer will be returned.

```
Iso3GetPreviousSibling (object)
```

This method returns the object identifier of the previous sibling as BSTR.

The *object* parameter defines the object identifier as BSTR.

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

First introduced with IsoView 3.0

Object Links

Iso3CountLinks

get the number of links of an object

```
Iso3CountLinks (object)
```

This method returns the number of links of the object as long.

The *object* parameter defines the object identifier as BSTR.

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

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

First introduced with IsoView 3.0

Iso3GetLinkByIndex

get the specified link of an object. This method will only retrieve 'linkuri' attributes.

```
Iso3GetLinkByIndex (object index)
```

This method returns the link of an object as BSTR.

The *object* parameter defines the object identifier as BSTR.

The *index* parameter defines the index of the link as long (0 based).

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

You can retrieve the number of links on an object using Iso3CountLinks.

First introduced with IsoView 3.0

Iso3SelectLink

lets the user select one of the links of an object in a dialog.

```
Iso3SelectLink (object)
```

This method returns the index of the link chosen by the user as long.

The *object* parameter defines the object identifier as BSTR.

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

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.

First introduced with IsoView 3.0

AllowJumps

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

This property enable/disable internal jumping, default: true

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

First introduced with IsoView 1.0

FileName

This property defines the name resp. the URL of the current illustration

This property sets the current filename as BSTR.

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

First introduced with IsoView 1.0

HsVisible

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

This property sets the visible / invisible status of hotspots as boolean (default: true).

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

First introduced with IsoView 1.0

Preview

This property determines whether the illustration is displayed in preview mode or not. Preview has the same meaning here that it has inside IsoDraw.

This property sets if the preview mode is turned on or off as boolean (default: True).

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

This property is obsolete since version 3.0

First introduced with IsoView 1.0

PropertyChangeMode

With this property you start / end a sequence of property changes. 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.

This property sets the property change mode status as boolean.

```
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;
```

First introduced with IsoView 1.0

ResizeToFit

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.

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

This property is obsolete since version 3.0

First introduced with IsoView 1.0

Viewport

This property sets the name of the current viewport. If this property changes IsoView tries to establish this viewport based on the information found in the current illustration.

This property sets the name of the current viewport (case sensitive).

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

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_ resp. \$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" resp.
"\$NAME_HS1", \$ID_ObjectID*

First introduced with IsoView 1.0

ViewPortCount

This property returns the number of the viewports contained in the illustration.

This property is supported for compatibility reasons with IsoView 1.0 only.

This property returns the number of viewports contained in the illustration as long.

```
lvp = document.ivx1.ViewPortCount;
```

This property is read only.

First introduced with IsoView 1.0

ViewSize

This property sets the scale (zoom factor) of the representation. If you use this, we recommend to disable the *ResizeToFit* feature to avoid recursive effects.

This property sets the scope factor as double (1..6400%).

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

First introduced with IsoView 1.0

AcceptDroppedFiles

With this property you instruct IsoView to handle or to ignore files dropped on the viewer window.

This property sets how files are handled when dropped on the viewer window as boolean(accept, default: true).

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

This property works with the IsoView ActiveX Control only.

First introduced with IsoView 2.0

ActiveTool

With this property you determine which tool is currently selected. Use *Iso7ConfigTools* to configure the behavior of the toolbar and the set of selectable tools.

This property returns the active tool. (see *tool set/selection constants* – *Iso7ConfigTools*, *ActiveTool* for allowed values)

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

First introduced with IsoView 2.0

ExtentX

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

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

This property is read only.

First introduced with IsoView 2.0

ExtentY

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

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

This property is read only.

First introduced with IsoView 2.0

HomeFileName

Use this property to retrieve the filename of the home illustration set by the first call to *Iso3OpenFile* or *SetHome*.

This property returns the filename of the home illustration as BSTR.

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

This property is read only.

First introduced with IsoView 2.0

HomeViewPort

Use this property to retrieve the home viewport set by the first call to *Iso3OpenFile* or *SetHome*.

This property returns the home viewport as BSTR.

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

This property is read only.

First introduced with IsoView 2.0

ObjectTipState

Use this property to determine the content of the object tips displayed when an object is 'touched' by the mouse pointer.

This property returns the content type of the object tip as long (the default value is 0). Allowed values are:

-
- 0: *NONE*
 - 1: *TIP*
 - 2: *ID*
 - 3: *TIP OR ID*
 - 4: *NAME*
 - 5: *TIP OR NAME*
 - 16: *HOTSPOTS ONLY*

The content of an object tip can be determined at runtime. If the state is set to *NONE* no object tips are shown. If the state is set to *TIP* only object tips will be shown which have been assigned previously by `SetObjectTip`. If the state is set to *ID* resp. *NAME* the ID or name of the object will be shown. The settings *TIP OR ID* resp. *TIP OR NAME* will show a pre-assigned tip if available or the ID/name otherwise. Add *HOTSPOTS ONLY* to force that object tips only are displayed on hot spots.

First introduced with IsoView 2.0

OffsetX

This property describes the horizontal position of the representation. If you use this, we recommend to disable the *ResizeToFit* feature to avoid recursive effects.

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

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

First introduced with IsoView 1.0

OffsetY

This property describes the vertical position of the representation. If you use this, we recommend to disable the *ResizeToFit* feature to avoid recursive effects.

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

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

First introduced with IsoView 1.0

ContextMenuItemHit

This event is generated when the user selects an item from a context menu. (see *Context Menu Functions*)

```
ContextMenuItemHit (item, Title)
```

The *item* parameter indicates the No. of the menu item (1 based) as short.

The *Title* parameter indicates the Title of the menu item as BSTR.

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

First introduced with IsoView 1.0

FileChanged

This event is generated when the file has been changed. This may occur either from outside (via *Iso3OpenFile* resp. *FileName*) or when IsoView has executed internal “jumps” to another file.

```
FileChanged (FileName)
```

The *FileName* parameter indicates the current filename as BSTR.

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

First introduced with IsoView 1.0

HsHit

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

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

The *MouseButton* parameter indicates which mouse button has been clicked as short. Allowed values are:

0: *no mouse button*
1: *left mouse button*
2: *right mouse button*

The *RefID* parameter indicates the ID of the hotspot (may be 0) as long.

The *Name* parameter indicates the Name of the hotspot (may be empty) as BSTR.

The *ViewPortName* parameter indicates the Name of the viewport connected to the hotspot (may be empty) as BSTR.

The *ViewPortFile* parameter indicates the Name of the file in which the viewport is located (may be empty) as BSTR.

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

As soon as the mouse leaves the hotspot area, another HsHit event is generated. All parameters are 0 resp. empty.

First introduced with IsoView 1.0

InitFinished

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

```
InitFinished ()

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

First introduced with IsoView 2.0

KeyDown

```
KeyDown(Char, ShiftState)
```

The *Char* parameter indicates which key has been pressed as short.

The *ShiftState* parameter indicates if the shift-key was pressed as short.

First introduced with IsoView 3.0

KeyPress

```
KeyPress (Char)
```

The *Char* parameter indicates which key has been pressed as short *.

First introduced with IsoView 3.0

KeyUp

```
KeyUp (Char, ShiftState)
```

The *Char* parameter indicates which key has just been released as short.

The *ShiftState* indicates if the shift-key was pressed as short.

First introduced with IsoView 3.0

MouseDown

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

The *Button* parameter indicates which mouse button has been clicked as short.

The *Shift* parameter indicates if the shift key has been pressed as short.

The *x* parameter indicates the horizontal position as OLE_XPOS_PIXELS.

The *y* parameter indicates the vertical position as OLE_YPOS_PIXELS.

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

First introduced with IsoView 2.0

MouseMove

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

The *Button* parameter indicates which mouse button has been clicked as short.

The *Shift* parameter indicates if the shift key has been pressed as short.

The *x* parameter indicates the horizontal position as OLE_XPOS_PIXELS.

The *y* parameter indicates the vertical position as OLE_YPOS_PIXELS.

This event is disabled by default. Use *ConfigEvents* to enable it.

First introduced with IsoView 2.0

MouseUp

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

The *Button* parameter indicates which mouse button has been clicked as short.

The *Shift* parameter indicates if the shift key has been pressed as short.

The *x* parameter indicates the horizontal position as OLE_XPOS_PIXELS.

The *y* parameter indicates the vertical position as OLE_YPOS_PIXELS.

This event is disabled by default. Use *ConfigEvents* to enable it.

First introduced with IsoView 2.0

PropertyChanged

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

```
PropertyChanged()
```

First introduced with IsoView 2.0

ObjectHit

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

```
ObjectHit (MouseButton, object)
```

The *MouseButton* parameter indicates which mouse button has been pressed as short. Allowed values are:

0: *no mouse button*
1: *left mouse button*
2: *right mouse button*

The object parameter indicates the object identifier as BSTR.

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

As soon as the mouse leaves the object's area, another ObjectHit event is generated. All parameters are 0 resp. empty.

First introduced with IsoView 3.0

ViewWindowChanged

this event occurs when one of the additional view windows (magnifier, global view) is opened or closed.

```
ViewWindowChanged (Window, Action)
```

The *Window* parameter indicates the type of the window which has changed as short. Allowed values are:

- 1: *MAGNIFIER*
- 2: *GLOBAL*

The *Action* indicates the action as short. Allowed values are:

- 1: *WINDOW OPENED*
- 2: *WINDOW CLOSED*

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

First introduced with IsoView 4.0

Parameter

Object identifier

The object identifier is used to identify a single object. The expression must uniquely identify the object. (see *Object Functions*)

Parameters

`id (objID)`

The expression consists of “id (”, the actual ID of the object, and “)”.

`name (objName)`

The expression consists of “name (”, the actual name of the object, and “)”.

If the name is used on multiple objects the first object with that name will be identified.

`objID`

The actual ID of the object is a shortcut for the identifier “id(objID)”.

`$ROOT`

special ID for the root node of the entire illustration.

`id(myID1)`

identifies the object with the ID “myID1”.

`name(myName1)`

identifies the object with the name “myName1”.

Use an object expression to identify multiple objects with a common name.

objExpr

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

parameters

```
name (myName1, all)
```

The expression consists of “name (“, the actual name of the object, and “, all)”.
This will identify all objects with this name.

```
name (myObj1, all)
```

identifies all objects with the name “myObj1”.

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

Extended Object Expression

The 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.

```
isoForm          ::= "isoeval("evalTerm(", " objbehavior)? ")"
evalTerm         ::= objInfoTerm
                  |   objIDTerm
                  |   objNameTerm
                  |   objTipTerm
                  |   objAttrTerm
                  |   textTerm
```

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.

```
objInfoTerm      ::= "obj_ex"
objIDTerm        ::= "id_cont=" objid
```

To address objects by their the name attribute you have 3 possible methods. The `objNameTerm` supports check for existing name attribute and the non-existence of a name attribute. Objects may be addressed by parts of the value of the name attribute.

```
objNameTerm      ::= objNameExTerm
                  |   objNameNExTerm
                  |   objNameContTerm
```

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

```
objNameExTerm    ::= "name_ex"
objNameNExTerm   ::= "name_nex"
objNameContTerm  ::= "name_cont=" objname
```

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

```
objTipTerm       ::= objTipEqTerm
                  |   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, is not equal to a value or just if it contains a certain value






















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

WebCGM 2.0 DOM

Arbortext IsoView 7.0 supports WebCGM 2.0 and the WebCGM2.0 DOM (document object model). For any details regarding the DOM, please refer to the “WebCGM 2.0 Standard“ at <http://www.oasis-open.org/specs/index.php#webcgm2.0> or the WebCGM 2.0 recommendation at <http://www.w3.org/Graphics/WebCGM/WG/>

Toolbar

The toolbar is optionally (see *Iso7ConfigTools*) shown in the IsoView window.


	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	Activate the redlining 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
	Popup	Opens the popup tools

Since the toolbar is configurable trough 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, IsoView 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.


While running in Redlining mode IsoView offers the possibility to draw freehand lines and other graphical primitives and to add textual annotations.

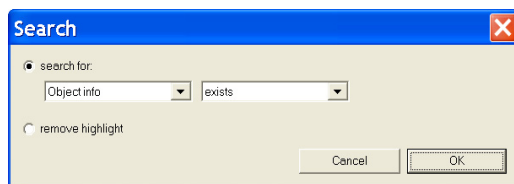
After adding redlining information to the illustration you have the possibility to export these for later use.

With IsoView 5 the *Iso5ImportRedlining* method has been introduced to reimport the redlining layers for later displaying.

Finding Elements on the Illustration

This functionality has been introduced with IsoView 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 dialog:



Elements can be highlighted similar to the IsoDraw search dialog. Here, it is not necessarily required to have object information assigned.

Constants

tool set/selection constants – *ConfigTools2, ActiveTool*

use these constants to enable tool and/or let tool appear in main toolbar:

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_RLNote	1024	0x00000400L
kIV4_Btn_RLDelete	2048	0x00000800L
kIV4_Btn_RLExport	4096	0x00001000L
kIV4_Btn_Prefs	8192	0x00002000L
kIV4_Btn_RLImport	16384	0x00004000L
kIV4_AllButtons	65535	0x000FFFFFL

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_RLNote	67108864	0x04000000L
kIV4_Pop_RLDelete	134217728	0x08000000L
kIV4_Pop_RLExport	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 IsoView toolbar with buttons
kIV4_ToolsPopup	2	all tools as popup menu
kIV4_ToolsRemote	4	remote control of tools (using
kIV4_ToolsRLAutoHi	8	ActiveTool)
de		automatically hide/show redlining tools

tool set/selection constants – *Iso7ConfigTools*, *ActiveTool*

use these constants to enable tool and/or let tool appear in main toolbar:

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_Animate*	2097152	0x00200000L
kIV7_Tool_Animate*	4194304	0x00400000L
kIV7_Tool_Animate*	8388608	0x00800000L
kIV7_Tool_Animate*	16777216	0x01000000L
kIV7_Tool_3DRotate**		

Use these constants also to enable tool and/or let tool appear in popup menu.

* only visible if file contains animation.

** only visible if file contains 3D data

Tip constants - *Object Tip*

kIV4_ToolsNone	0	no tools to interact with viewer
kIV4_ToolsButtons	1	default IsoView toolbar with buttons
kIV4_ToolsPopup	2	all tools as popup menu
kIV4_ToolsRemote	4	remote control of tools (using
kIV4_ToolsRLAutoHide	8	ActiveTool)
		automatically hide/show redlining tools

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_ShowTipOrID	3	use tip of object or ID if none
kIV_ObjTip_ShowName	4	use name of object in tip
kIV_ObjTip_ShowTipOrName	5	use tip of object or name if none
me	16	add to get objects tips on hotspots
kIV_ObjTip_HotspotsOnly		only

Object highlighting constants - *Iso3HighlightObject*

kIV_HotSpotHide	0	hide hot spot highlight
kIV_HotSpotFramed2		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	
kIV_HotSpotCircled64	64	process all hotspots with given ID a circle around will indicate hotspot

Event configuration constants - *ConfigEvents*

(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 IsoView from handling mouse clicks
s		internal

Print configuration constants – *ConfigPrinting, DoPrint*

kIV_Printer_KeepSettings	-1	do not change existing setting
kIV_Printer_PrintAll	0	print entire illustration
kIV_Printer_PrintVisible	1	print only visible portion of illustration
kIV_Printer_FitOnPage	0	fit printed image onto one page
kIV_Printer_SplitPages	1	split printed image across multiple pages if
kIV_Printer_Portrait	1	necessary
kIV_Printer_Landscape	2	print in portrait orientation print in landscape orientation

Overview of preference identifiers - *Iso4SetPreference*

anti_aliasing

switches the anti-aliasing mode on ("1") or off ("0").

default_toolset

Default set of toolbar buttons, see Iso7ConfigTools for values

default_toolflags

Default flags for toolbar configuration, see Iso7ConfigTools for values

redlining_linecolor

Color of RedLining line

redlining_linewidth

Line width (in pt) of RedLining line

redlining_fontsize

Font size (in pt) of RedLining annotations

redlining_fontname

Font name of RedLining annotations

use_default_printer

automatically use default printer ("1") or popup printer selection dialog ("0")

object_behavior

change the behavior of the ObjectHit event. Values: "0" (IsoView 4): left click on hotspots only, right click, move on all objects "1" (hotspots only): left click, right click, move on

hotspots only"2" (all objects): left click, right click, move on all objects

Overview of the error numbers

*5: invalid function call
7: insufficient free memory
53: file not found
75: error in file access
380: invalid parameter values
1000: unknown error
1001: error when starting IsoView Server
1002: invalid command line parameter
1003: prohibited extension of existing context menu
1004: error when creating context menu
1005: illegal try to open global view/magnifier twice
1006: selected tool/function not enabled for usage*

Unicode support

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

Due to the lack of UNICODE support on the operating systems:
Windows 95/98/Me
these platforms are not supported.

List of viewable file formats

IsoDraw Files
CGM
Product View (.ol, .ed, .edz, .pvs, .pvz)
SVG
TIFF
JPEG
PNG
BMP

PCX
CALS Raster

List of required files for use

For the use of the IsoView ActiveX Control you need the files listed below:

PTC
ISOVIEWX7.OCX (Version 7.0.0.x)
ISOENGINE7.EXE (Version 7.0.0.x)
OptCGM.prf
IsoView7License.iso
DefaultHatchings.iso
pvGlue.dll
PV2ISO.pvr

CGM Open
cgmopenbho.dll (Version 1.0.0.5)
cgmopenbho.cfg

Microsoft
MFC42U.DLL (Version 6.0.8665.0)
MSVCRT.DLL (Version 6.0.8797.0)

All Microsoft and CGM Open DLLs in the indicated version (or higher) should be located in the Windows System directory.

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

There are also following components of Microsoft Internet Explorer (Version 4.01 or higher) needed. It has therefore to be installed on the processing computer.

IE-DLLs
SHLWAPI.DLL (Version 4.70.1300)
URLMON.DLL (Version 4.70.1300)

How to prepare IsoView for distribution

IsoView ActiveX Control

1. Open the self-extracting archive "isoviewx7.exe" using WinZip and extract all files to a new folder.
2. Add the file "IsoView7License.iso" which you received with your 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" using WinZip Self-Extractor. Make sure you select "Self-extracting Zip file for Software Installation" as kind of the new archive.

Annotations

In principle, the Arbortext IsoView ActiveX Control may be used in all ActiveX capable environments. Among others we have tested the following applications: Microsoft Internet Explorer 5.01, 5.5 and 6.0, Microsoft Visual Basic 6.0 and Microsoft Access 2000.

When using Microsoft Internet Explorer, PTC strongly recommends version 5.01 or newer. Older versions do not support all kinds of navigation that IsoView can handle.