

# AutoVue Desktop Deployment

## ActiveX API Documentation

### Certified Environments

The Desktop Deployment ActiveX is intended to be hosted in a 32-bit application. It is certified with Visual Studio 2008: Visual C++/VB.NET/C#(.NET).

### Initialization

- [setValue \(method\)](#)
- [initActiveX \(method\)](#)
- [doneInit \(Event\)](#)

### Document Management

- [setFile \(method\)](#)
- [getFile \(method\)](#)
- [closeDocument \(method\)](#)
- [setCompareFile \(method\)](#)
- [addOverlay \(method\)](#)
- [removeOverlays \(method\)](#)
- [onFileEvent \(Event\)](#)

### Document Viewing

- [setPage \(method\)](#)
- [getPage \(method\)](#)
- [getPagesNumber \(method\)](#)
- [getPageSizeMinX \(method\)](#)
- [getPageSizeMinY \(method\)](#)

- [getPageSizeMaxX \(method\)](#)
- [getPageSizeMaxY \(method\)](#)
- [setViewExtents \(method\)](#)
- [getViewExtentsMinX \(method\)](#)
- [getViewExtentsMinY \(method\)](#)
- [getViewExtentsMaxX \(method\)](#)
- [getViewExtentsMaxY \(method\)](#)
- [setZoom \(method\)](#)
- [getImageEx \(method\)](#)
- [onViewEvent \(method\)](#)

## Output

- [setPrintOptions \(method\)](#)
- [printIt \(method\)](#)
- [setConvertOptions \(method\)](#)
- [convertIt \(method\)](#)
- [exportEdaBom \(method\)](#)
- [export3DBom \(method\)](#)

## Markups

- [isMarkupModeEnabled \(method\)](#)
- [openMarkup \(method\)](#)
- [saveActiveMarkup \(method\)](#)
- [saveModifiedMarkups \(method\)](#)
- [importMarkup \(method\)](#)
- [exportMarkup \(method\)](#)
- [onMarkupEvent \(Event\)](#)

## General

- [invokeAction \(method\)](#)
- [invokeSubAction \(method\)](#)
- [onStateEvent \(Event\)](#)

# Methods

---

## addOverlay

Add an overlay.

*void* addOverlay(*BSTR* fileName)

### Parameters:

fileName – File to be added as an overlay

## closeDocument

Close the current document.

*void* closeDocument()

## convertIt

Perform file conversion with options specified by *setConvertOptions*.

*void* convertIt(*boolean* is2DConvert)

### Parameters:

Is2DConvert – Whether performing 2D conversion or 3D conversion

## export3DBom

Export 3D BOM to a file.

*void* export3DBom(*BSTR* fileName)

### Parameters:

fileName – Filename for the exported BOM result. Specify full path to the output file name. Path and name must satisfy the following:

- Path should point to either current directory, user home directory or a temporary directory
- File extension should be .csv
- File should not already exist

## exportEdaBom

Export EDA BOM to a file.

*void exportEdaBom(BSTR fileName, BSTR format, BSTR attributes, boolean entireDesign)*

### Parameters:

fileName – Filename for the exported BOM result. Specify full path to the output file name. Path and name must satisfy the following:

- Path should point to either current directory, user home directory or a temporary directory
- File extension should be .csv or .pdx
- File should not already exist

format – The export format, could be “csv” and “pdx”

attributes – The attributes used for generating BOM

entireDesign – Whether the exporting is for all pages. If false, only the current page will be used for generating BOM

## exportMarkup

Export a markup to the specified path and filename.

*void exportMarkup(BSTR filename)*

### Parameters:

filename – full path (path and filename) of markup file. Path and name must satisfy the following:

- Path should point to either current directory, user home directory or a temporary directory
- File should not already exist

## importMarkup

Import a markup from the specified path and filename.

```
void importMarkup(BSTR filename)
```

### Parameters:

filename – full path (path and filename) of markup file

## initActiveX

Initialize AutoVue Desktop Deployment ActiveX and make it visible.

```
void initActiveX()
```

## invokeAction

Directly invoke an action that has no sub-actions (child actions).

```
void invokeAction(BSTR actionClassStr)
```

### Parameters:

actionClassStr – The name of the action class to invoke, e.g. “VueActionViewAntiAlias”

Refer to the Desktop Deployment *Installation and Configuration Guide*, section *Customizing the GUI* for a list of VueAction control names.

## invokeSubAction

Directly invoke a sub-action (child action) of the specified action.

```
Void invokeSubAction(BSTR actionClassStr, BSTR subActionStr)
```

### Parameters:

actionClassStr – The name of the action class, e.g. “VueActionViewZoom”

subActionStr – The name of its sub-action to be invoked, e.g. “FitBoth”

## isMarkupModeEnabled

Determines whether the ActiveX is currently in markup mode.

```
boolean isMarkupModeEnabled()
```

### Return Value:

True if currently in markup mode

## **getFile**

Get the full path filename of document currently opened

*BSTR* getFile()

### **Return Value:**

Full path filename of document currently opened

## **getImageEx**

Gets the raster image display attribute.

*long* getImageEx(*int* flag)

### **Parameters:**

flag – Display attribute to query, it could be one of the following:

0 – Image contrast

1 – Image anti-alias

2 – Image invert

### **Return Value:**

Depends on the given flag value, it could return the following:

0 – Image contrast:      -50 – Light, 0 – Normal, 66 – Dark, 100 - Darkest

1 – Image anti-alias:      0 – Off, 1 – On

2 – Image invert:      0 – Off, 1 – On

## **getPage**

Gets the current page index of the currently opened document.

*int* getPage()

### **Return Value:**

Page index of current page

## **getPageSizeMaxX**

Gets maximum X extents of the current page of the currently opened document.

*double* getPageSizeMinX()

### **Return Value:**

Maximum X of the extents

## **getPageSizeMaxY**

Gets maximum Y extents of the current page of the currently opened document.

*double* getPageSizeMinX()

### **Return Value:**

Maximum Y of the extents

## **getPageSizeMinX**

Gets minimum X extents of the current page of the currently opened document.

*double* getPageSizeMinX()

### **Return Value:**

Minimum X of the extents

## **getPageSizeMinY**

Gets minimum Y extents of the current page of the current open document.

*double* getPageSizeMinX()

### **Return Value:**

Minimum Y of the extents

## **getPagesNumber**

Gets total number of pages of the currently opened document.

*int* getPagesNumber()

### **Return Value:**

Total number of pages

## **getViewExtentsMaxX**

Gets current rectangular extents of the portion of the document being displayed.

*double* getViewExtentsMaxX()

### **Return Value:**

Maximum X of the extents

## **getViewExtentsMaxY**

Gets current rectangular extents of the portion of the document being displayed.

*double* getViewExtentsMaxY()

### **Return Value:**

Maximum Y of the extents

## **getViewExtentsMinX**

Gets current rectangular extents of the portion of the document being displayed.

*double* getViewExtentsMinX()

### **Return Value:**

Minimum X of the extents

## **getViewExtentsMinY**

Gets current rectangular extents of the portion of the document being displayed.

*double* getViewExtentsMinY()

### **Return Value:**

Minimum Y of the extents

## **openMarkup**

Open a markup given its property.

*void* openMarkup(*BSTR* markup)

### **Parameters:**



markup – A property string separated by semicolons specifying all the markup attributes (name1=value1;name2=value2).

NOTE:

- To open a markup specify the markup property as "CSI\_DocName=markupName". markupName must follow the convention *<base file name without extension>.<custom name>*. For example, if base file name is acad12.dwg, markup file name would be acad12.<custom name>. `openMarkup("CSI_DocName=acad12.review");`
- To open markup in read-only mode add markup property "CSI\_DocReadOnly=true".
- To open all markups associated with the currently loaded document specify the markup property as "\*".

## printIt

Perform printing with options specified by *setPrintOptions*.

`void printIt()`

## removeOverlays

Remove all overlays.

`void removeOverlays()`

## saveActiveMarkup

Save current active markup with the given property.

`void saveActiveMarkup(BSTR markup)`

### Parameters:

markup – A property string separated by semicolons specifying all the markup attributes (name1=value1;name2=value2). To save a local markup specify the markup property as "CSI\_DocName=markupName". markupName must follow the convention *<base file name without extension>.<custom name>*.

For example, if base file name is acad12.dwg, markup file name would be acad12.<custom name>.

`saveActiveMarkup("CSI_DocName=acad12.review");`

## saveModifiedMarkups

Save all modified markups.

```
void saveModifiedMarkups()
```

## setCompareFile

Sets the file to compare against the currently opened document. This initiates compare mode with the currently opened document in the first window and the file specified by *fileName* in the second window.

```
void setCompareFile(BSTR fileName)
```

### Parameters:

fileName – Full path filename of document to be compared against

## setConvertOptions

Specify options for file conversion.

```
void setConvertOptions(BSTR key, BSTR value)
```

### Parameters:

key – Option key, it could be one of the following:

“FORMAT” – Required option that specify output file format, *value* could be one of the following:

“PCRS\_BMP” – Windows Bitmap

“PCRS\_TIF” – TIFF

“PCVC\_PDF” – PDF

“PC3D\_STL” – Stereolithography

“SUBFORMAT” – Specify sub-format of the output file format, *value* is an integer.

“PCRS\_TIF” sub-formats:

“0” – Uncompressed

“1” – Packbits compressed

“2” – Group III compressed

“3” – Group IV compressed

“PC3D\_STL” sub-formats:

“0” – ASCII STL

“1” – BINARY STL

“OUTPUT” – Required option that specify filename of the output file. Specify full path to the output file name. Path and name must satisfy the following:

- Path should point to either current directory, user home directory or a temporary directory
- File should not already exist

“AREA” – Print area type, *value* could be “DISPLAY”, “EXTENTS” (Default) or “SELECTED” (3D only)

“SIZEX” – Specify raster output’s x dimension; required to specify either SIZEX and SIZEY or SCALE for all output formats except PDF

“SIZEY” – Specify raster output’s y dimension

“SCALE” – Specify scale factor, *value* is an integer specifying scale factor

“COLORDEPTH” – Specify raster output’s color depth, *value* could be “ORIGINAL” (Default) for obtaining color depth from base document or an integer specifying color depth e.g. Value “8” for 256 color.

“PAGERANGE” – Specify page range, *value* could be “ALL”, “CURRENT” (Default) or a string “<fromPage>-<toPage>” e.g. “1-3”

“STL\_POSITIVE\_TRIANGLES” – Specify positive triangulation is on for STL output, *value* could be “ON” and “OFF” (Default)

*value* – Depends on value of *key* parameter. See above descriptions.

## setFile

Open a document in the ActiveX.

*void* setFile(*BSTR* fileName)

### Parameters:

fileName – Full path (path and filename) of document to be opened; Prefix the full path with “upload://”.

For example: setFile(“upload://C:\files\autocad.dwg”);

## setPage

Sets the specified page on the currently opened document.

*void setPage(int page)*

### Parameters:

page – The page index

## setPrintOptions

Specify options for printing.

*void setPrintOptions(BSTR key, BSTR value)*

### Parameters:

key, – Option key, it could be one of the following:

“ORIENTATION” – Page orientation, *value* could be “LANDSCAPE”, “PORTRAIT” or “AUTO” (Default)

“AREA” – Print area type, *value* could be “DISPLAY” or “EXTENTS” (Default)

“SCALE” – Scaling type, *value* could be “FIT” (Default), a string “<percentage>%” indicates a scale or a string “<factor>” indicates scaling to a factor

“UNITS” – Set scaling factor units, *value* could be “INCH” (Default) or “MM”

“FORCETOBLACK” – Set force to black value, *value* could be “TRUE” or “FALSE” (Default)

“ALIGNMENT” – Set print alignment, *value* could be “CUSTOM”, “TOPLEFT” (Default), “TOPCENTER”, “TOPRIGHT”, “MIDDLELEFT”, “CENTER”, “MIDDLERIGHT”, “BOTTOMLEFT”, “BOTTOMCENTER” or “BOTTOMRIGHT”

“PRINTOFFSETX” – Set x-offset for custom print alignment (i.e. “ALIGNMENT=CUSTOM”)

“PRINTOFFSETY” – Set y-offset for custom print alignment (i.e. “ALIGNMENT=CUSTOM”)

“PRINTROWHEADERS” – Print row header for spreadsheet documents, *value* could be “TRUE” or “FALSE” (Default)

“PRINTCOLHEADERS” – Print column header for spreadsheet documents, *value* could be “TRUE” or “FALSE” (Default)

“LH” – Specify left header text

“CH” – Specify center header text

“RH” – Specify right header text

“LF” – Specify left footer text

“CF” – Specify center footer text

“RF” – Specify right footer text

“PRINTDRIVER” – Specify printer name (Default: Last used printer)

“PAPERSIZE” – Specify paper size (Default: “1” - Letter)

“PAGERANGE” – Specify page range, *value* could be “ALL”, “CURRENT” (default) or a string “<fromPage>-<toPage>” e.g. “1-3”

*value* – Depends on value of *key* parameter. See above descriptions.

## setValue

Specify client parameters. This should be done before *initActiveX* method is called.

*void* setValue(*BSTR* key, *BSTR* value)

### Parameters:

key – Key of parameter

value – Value of parameter

Refer to the *Desktop Deployment Installation and Configuration Guide*, section *Command-Line Parameters* for a list of client parameters that can be set using the setValue method.

## setViewExtents

Sets the rectangular extents of the portion of the document to be displayed.

*void* setViewExtents(*double* minX, *double* minY, *double* maxX, *double* maxY)

### Parameters:

minX – Minimum X of extents

minY – Minimum Y of extents

maxX – Maximum X of extents

maxY – Maximum Y of extents

## setZoom

Sets zoom factor.

```
void setZoom(double zoom)
```

### Parameters:

zoom – Zoom factor in percentage

# Events

---

## doneInit

This event is fired when the ActiveX is initialized.

```
void doneInit(JVueAXEvent e)
```

### Parameters:

e – Event object, not being used in this event.

## onFileEvent

A file event is fired when some file operation is performed.

```
void onFileEvent(JVueAXEvent e)
```

### Parameters:

e – Event object that holds the event ID and any parameters. Could invoke its *getEventID* method to obtain the ID (e.g. 4 – ONSETFILE) and invoke its *getEventParameter* method to obtain string representation of the event parameters if there's any. For more information on possible value for Event ID and its parameters, please refer to JavaDoc of the *com.cimmetry.vuebean.event.VueEvent* class in the AutoVue JavaBean API documentation.

## onMarkupEvent

A file event is fired when some markup operation is performed.

```
void onMarkupEvent(JVueAXEvent e)
```

### Parameters:

e – Event object that holds the event ID and any parameters. Could invoke its *getEventID* method to obtain the ID (e.g. 4 – ONSETFILE) and invoke its *getEventParameter* method to obtain string representation of the event parameters if there's any. For more information on possible value for Event ID and its parameters, please refer to JavaDoc of the *com.cimmetry.vuebean.event.VueEvent* class in the AutoVue JavaBean API documentation.

## onStateEvent

A file event is fired when the state of the VueBean inside the ActiveX is changing. This includes server reconnection, file recovery, etc.

*void onStateEvent(JVueAXEvent e)*

### Parameters:

e – Event object that holds the event ID and any parameters. Could invoke its *getEventID* method to obtain the ID (e.g. 4 – ONSETFILE) and invoke its *getEventParameter* method to obtain string representation of the event parameters if there's any. For more information on possible value for Event ID and its parameters, please refer to JavaDoc of the *com.cimmetry.vuebean.event.VueEvent* class in the AutoVue JavaBean API documentation.

## onViewEvent

A file event is fired when some view operation is performed.

*void onViewEvent(JVueAXEvent e)*

### Parameters:

e – Event object that holds the event ID and any parameters. Could invoke its *getEventID* method to obtain the ID (e.g. 4 – ONSETFILE) and invoke its *getEventParameter* method to obtain string representation of the event parameters if there's any. For more information on possible value for Event ID and its parameters, please refer to JavaDoc of the *com.cimmetry.vuebean.event.VueEvent* class in the AutoVue JavaBean API documentation.

Copyright © 1999, 2012, Oracle and/or its affiliates. All rights reserved.

Portions of this software Copyright 1996-2007 Glyph & Cog, LLC.

Portions of this software Copyright Unisearch Ltd, Australia.

Portions of this software are owned by Siemens PLM © 1986-2012. All rights reserved.

This software uses ACIS® software by Spatial Technology Inc. ACIS® Copyright © 1994-2008 Spatial Technology Inc. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

#### U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.