Oracle® AutoVue Desktop Deployment

Installation and Configuration Guide

Release 21.0.2

F10648-01

October 2018

ORACLE®

Oracle® AutoVue Desktop Deployment, Release 21.0.2

F10648-01

Copyright © 1999, 2018, Oracle and/or its affiliates. All rights reserved.

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 is software or related documentation that 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 America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware 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 that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware 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.

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

Contents

Preface v

Audience v
Documentation Accessibility v
Related Documents v
Conventions vi

1 Introduction

2 Upgrading AutoVue

2.1 Markups 2-1

3 AutoVue Hardware and System Requirements

3.1 Hardware Requirements 3-1
3.2 System Requirements 3-1
3.2.1 Oracle AutoVue Certified Operating System 3-1

4 AutoVue Installation

4.1 Planning 4-1
4.2 Pre-requisites 4-1
4.3 Installing AutoVue 4-1
4.4 Starting AutoVue 4-3
4.5 File Associations 4-4
4.6 Symbol Markup Entities on Multiple AutoVue Instances 4-4
4.7 Command-Line Options for AutoVue 4-5

A Configuring AutoVue

A.1 Configuration Parameters A-1
A.1.1 Changing the Locale of AutoVue A-4
A.2 Command-Line Parameters A-5
A.2.1 Setting the -PARAM Flag A-5

B Configuring Memory Settings

B.1 Configuring Memory Settings B-1

C Non-Interactive Installations

C.1 Installation C-1
C.2 Uninstallation C-1

D Debugging AutoVue

D.1 Logging for AutoVue D-1
D.1.1 Logger Information D-1
D.1.2 Configuring log4j D-2

E Customizing GUI

E.1 Customizing the GUI E-1
E.1.1 Choosing the GUI File E-1
E.1.2 Modifying the GUI E-1
E.1.2.1 Structure and Syntax of GUI Files E-1
E.1.2.2 GUI Configuration Syntax E-2
E.1.2.3 Control Names E-3

F Samples and API Examples Included with AutoVue

F.1 API Examples F-1
F.2 Sample Files F-2
F.3 Adding Custom Tools Menu to AutoVue F-2

G Feedback

G.1 General AutoVue Information G-1
G.2 Oracle Customer Support G-1
G.3 My Oracle Support AutoVue Community G-1
G.4 Sales Inquiries G-1

Preface

The Oracle AutoVue Installation and Configuration Guide describes how to install and configure Oracle AutoVue Desktop Deployment. For the most up-to-date version of this document, go to the AutoVue Documentation Web site on the Oracle Technology Network at https://www.oracle.com/technetwork/documentation/autovue-091442.html.

Audience

The Oracle AutoVue Installation and Configuration Guide is directed at any user of Oracle AutoVue Desktop Deployment.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Related Documents

For more information, see the following documents in the Oracle AutoVue documentation library:

- Oracle AutoVue Desktop Deployment User's Manual
- Oracle AutoVue Desktop Deployment Viewing and Configuration Guide
- Oracle AutoVue Release Notes
- Oracle AutoVue Acknowledgments
- Oracle AutoVue Supported File Formats
- Oracle AutoVue Product Variations - Feature Matrix
- Oracle AutoVue API Guide

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Introduction

AutoVue is Oracle's suite of Enterprise Visualization solutions, which are designed to view, digitally annotate and collaborate on any digital information from any system. AutoVue delivers visualization capabilities for many document types, including business documents such as Office and Graphics, as well as technical document types such as 2-D/3-D Computer Aided Design (CAD) and Electronic Design Automation (EDA).

AutoVue Desktop Deployment is a solution for users that want to run AutoVue locally as a personal productivity tool on their individual desktops.

This document provides instructions for installing and configuring AutoVue Desktop Deployment. In the remainder of this document, AutoVue Desktop Deployment will be referred to as AutoVue.

2

Upgrading AutoVue

If you are upgrading from AutoVue 20.2.x/21.0.x, you do not need to run the uninstaller before you install AutoVue 21.0.2. You can just run the installer for AutoVue 21.0.2. The installer detects if AutoVue 20.2.x/21.0.x is installed on your machine. If it is installed, the installer backs up required data, uninstalls version 20.2.x/21.0.x and then installs 21.0.2 to the same location. Below is a list of configuration settings that are migrated:

- User INI file, default.ini and allusers.ini files.
- If you have Oracle AutoVue 20.2.x/21.0.x installed previously, the installer upgrades the previous configuration settings from allusers.ini and autovue.properties when you install the new version.
- Customized GUI files (located in the <AutoVue Installation Directory>\bin\Profiles directory).
- Global user settings from the allusers.ini file and default user settings from the default.ini file (located in the <AutoVue Installation Directory>\bin directory).
- Custom logging settings from the log4j.xml file (located in the <AutoVue Installation Directory>\bin directory).
- Custom markup symbol libraries (located in the <AutoVue Installation Directory>\bin\Symbols directory).
- Markup files (located in the <AutoVue Installation Directory>\bin\Markups directory).

2.1 Markups

Markups are stored in an *avred* folder at the location of the base file. Alternatively, you can change the location of markups by setting option autovue.markups.local.directory. When upgrading from version 20.2.x/21.0.x, the installer will set option autovue.markups.directory to reflect the directory where the 20.2.x/21.0.x markups were saved (for example, <AutoVue installation directory>\bin\Markups). AutoVue will only read markups from this location. When you modify or resave these markups, they will be saved with the current version behavior (that is, they are saved either into the *avred* folder or at the location specified by autovue.markups.local.directory).

Note: Do not modify option autovue.markups.directory.

3

AutoVue Hardware and System Requirements

This chapter discusses the hardware and system requirements for installing Oracle AutoVue.

3.1 Hardware Requirements

The recommended requirements for a machine running AutoVue Desktop Deployment is as follows:

- 2 GB of RAM.

  Note: AutoVue is a 32-bit application. Generally, 32-bit processes are limited to 2 GB of RAM. However, the AutoVue Desktop Deployment executable is large address aware. This means that if AutoVue is running on a 64-bit OS that has 4 GB of RAM, AutoVue can use up to 4 GB of RAM. If AutoVue is running on a 32-bit machine that is booted up with the 3 GB switch, and the machine has 3 GB of RAM, AutoVue can use up to 3 GB of RAM.

- 2 GHz dual-core CPU.
- Display driver that supports OpenGL and has dedicated memory (required for 3D support).
- 400 MB of disk space for installation.
- 6 GB of free disk space:
  - 4 GB needed for storing cached files
  - Remainder needed by AutoVue to store temporary files (at the %TEMP% path).

3.2 System Requirements

This section details system requirements for the Oracle AutoVue line of products and integrations.

3.2.1 Oracle AutoVue Certified Operating System

The following platforms are certified for AutoVue installation and/or hosting.

Table 3–1 Oracle-Certified Operating Systems

Platform	AutoVue for Agile PLM	AutoVue Client/Server Deployment	AutoVue Desktop Deployment
Windows 10, 8.1, 7—64-bit (AutoVue running in 32-bit mode)			x

4

AutoVue Installation

This chapter discusses the planning, and pre-requisites required for installing Oracle AutoVue, and how to start AutoVue.

4.1 Planning

AutoVue, by default, installs at *C:\Oracle\AutoVue Desktop Deployment*. AutoVue must have write permissions to the installation path in order to start up successfully since it stores markup symbols, markups, profiles, and cache data in sub-folders at the installation path. On operating systems that have User Access Control (UAC) enabled, ensure that AutoVue is installed in a directory where the user has write permissions.

If UAC is enabled on the system, the installer notifies you that UAC is enabled and reminds you to install into a location where the user has write permissions.

4.2 Pre-requisites

- The installer for AutoVue also installs the AutoVue Document Converter print driver. Ensure that the print spooler service is enabled and that you have permissions to install print drivers on the machine where you will install AutoVue.

  Note: When installing on a Windows 8 or Windows 10 OS, right-click the AutoVue installer and select Run as administrator to properly install the AutoVue Document Converter print driver.

- The machine that is running AutoVue must have a color depth of at least 16-bits. If the machine has a lower color depth, you may run into discrepancies in color or filling when viewing, printing or converting from AutoVue.
- AutoVue stores temporary files at the user's %TEMP% directory. These files are generated by AutoVue while processing files and are generally removed once the processing is complete. Ensure that the user running AutoVue has a temporary directory defined and is able to write to the temporary directory.

4.3 Installing AutoVue

Note: If you want to install AutoVue in non-interactive mode, refer to "Non-Interactive Installations".

Important:

- Shutdown all applications (including AutoVue) before you run the installer for the AutoVue Desktop Deployment.
- Make sure the AutoVue installer is run as an Administrator. If User Access Control (UAC) is enabled, select *Yes* for the prompt that asks you if you want to run as Administrator. If UAC is disabled, right-click the installer, and then select **Run as Administrator**.
- When upgrading, if the installer prompts you to reboot the machine before or after the uninstallation, you must reboot the machine in order to get a successful installation.
- When upgrading your installation of AutoVue, do not cancel the installation process once it has begun. Cancelling the installation may leave your current installation of AutoVue unusable.

To install AutoVue, do the following:

1. Download the Oracle AutoVue Media Pack and extract its contents.
2. Close all applications before you start the installation for AutoVue.
3. Run the AutoVue installer executable *InstallDesktopDeployment.exe* that is located in folder *DesktopDeployment*.
4. Select a language from the installation dialog and then click **OK**.
5. Click **Next** to begin installation.
6. If the installer detects version of 20.2.x/21.0.x installed on the machine, it prompts if you want to upgrade your installation. If you choose to not upgrade, the installer will exit. You must backup required data manually, uninstall 20.2.x/21.0.x and then re-run the installer. If you choose to upgrade, the installation will continue. It will install to the same location as 20.2.x/21.0.x.
7. Specify the installation directory and then click **Next**.

   ```
   Example: C:\Oracle\AutoVue Desktop Deployment
   ```

8. Select an installation set and then click **Next**:

Table 4–1 Installation Sets

Installation Set	Description
Standard	Installs the most common AutoVue features. Note that this set does not install the sample drawing files or API examples.
Custom	You can select the features to install. Select this installation set to install the sample drawing files and API examples.

If you selected the **Custom** install set continue to step 9, otherwise proceed to step 10.

9. Select which of the following features to install and then click **Next**:

Table 4–2 Features for Installation

Option	Description
Program Files	Installs Oracle AutoVue. The option is selected by default.

Table 4–2 (Cont.) Features for Installation

Option	Description
Administration Documentation	Installs Oracle AutoVue system administration documentation. The option is selected by default.
User Documentation	Installs AutoVue end-user documentation. The option is selected by default.
Sample Files	Installs drawing sample files.
API Examples	Installs examples of how Oracle AutoVue features can be added to third-party applications using APIs.

10. Select one of the following locations to create shortcuts and then click **Next**.

Table 4–3 Installation Locations

Options	Description
In a new Program Group	Creates a shortcut in the Program group of the *Start* menu. For example, Oracle AutoVue. This is the default option.
In an existing Program Group	Adds a shortcut to an existing Program group. For example, Accessories.
In the Start Menu	Adds a shortcut in the Start menu.
On the Desktop	Adds a shortcut on the Desktop.
In the Quick Launch Bar	Adds a shortcut to the Quick Launch bar.
Other	Adds a shortcut to the specified location.
Don't create icons	Shortcuts are not created.

To create icons for all users of AutoVue, select **Create Icons for All Users**.

11. Review the pre-installation summary and then click **Install**.
12. Click **Done** to quit the installer.

AutoVue is installed in the specified directory. Start AutoVue to verify that your installation went through successfully. Refer to Starting AutoVue for more information.

Note: If there are any warnings or errors, refer to the installation log file, Oracle_AutoVue_Desktop_Deployment_Install_<Date>.log, located in the <AutoVue Installation Folder>\uninstall\Logs directory.

4.4 Starting AutoVue

To verify your AutoVue installation start up AutoVue by doing one of the following:

- From the program shortcuts, select **Oracle AutoVue Desktop Deployment** and then **Start AutoVue**.

  or

- Run autovue.exe located in the <AutoVue Installation Directory>\bin directory.

AutoVue loads and the AutoVue icon appears in the system tray.

> **Note:** On machines that have firewall enabled, a shortcut menu appears asking if you want to block or unblock the program. AutoVue will work fine no matter what option you choose.

If you close the AutoVue window, it continues to run in the background. To close AutoVue completely, right-click the AutoVue icon in the system tray and select **Close**.

> **Note:** If AutoVue does not load, enable logging and contact Oracle Customer Support. For information on enabling logging, refer to Logging for AutoVue.

4.5 File Associations

You can associate AutoVue with certain file types so that when you double-click on a file of this type, it opens in AutoVue.

Below are the ways to associate file types with AutoVue:

1. From Windows explorer:
   - When you double-click on a file that is not associated with another application, you will be prompted with the **Open With** dialog where you can specify the application with which to open the file. Select **AutoVue Desktop Deployment** from the list of programs and then select **Always use the selected program to open this kind of file**.
   - If a file type is associated with an application, right click on the file and select **Open With** and then select **AutoVue Desktop Deployment** from the list of programs. Select the **Always use the selected program to open this kind of file** check box.
2. Programmatically:

   You can create a batch file to associate file types with AutoVue. Create a patch file in the <AutoVue Install Directory>\bin directory with the following content:

   ```
   @assoc .ext=AVFile
   @ftype AVFile="%~dp0autovue.exe" "%1"
   ```

   Replace .ext with the file extension that you want to associate with AutoVue. Repeat this line for all extensions that you want to associate with AutoVue.

   For example:

   ```
   @assoc .dwg=AVFile
   @assoc .dgn=AVFile
   @assoc .pdf=AVFile
   @ftype AVFile="%~dp0autovue.exe" "%1"
   ```

   Run this batch file as an administrator to associate the specified file extensions with AutoVue.

4.6 Symbol Markup Entities on Multiple AutoVue Instances

If you are using Symbol markup entities, it is recommended to avoid sharing the Symbols folder between different AutoVue instances as this may lead to reliability and stability issues. It is good practice to replicate the Symbols folder across all AutoVue

instances at regular intervals. You should plan the replication for a time when AutoVue is not in use to ensure that Symbols are not being used.

4.7 Command-Line Options for AutoVue

AutoVue supports the following command line options:

- You can specify an alternative configuration file for AutoVue, by specifying the -config parameter.

  ```
  Syntax: autovue.exe -config <fullpath to config file>
  For example: autovue.exe -config
  "C:\users\avuser\appdata\autovue\autovue.properties"
  ```

- You can specify a filename to load when starting AutoVue by specifying the full path to the file on the command line.

  ```
  For example: autovue.exe C:\samples\2D\AutoCAD.dwg
  ```

  Note: You can specify multiple files as command line options. AutoVue opens the specified files in new child windows.

  Note: By default, the file opens in a new window. To change this behavior, you must set NEWWINDOW parameter to FALSE in autovue.properties. Refer to Configuring AutoVue for more information.

```
For example: autovue.exe C:\samples\2D\AutoCAD.dwg C:\samples\3D\Solidworks.sldprt
```

A

Configuring AutoVue

You can configure AutoVue by modifying *autovue.properties* located in the <AutoVue Installation Directory>\bin directory. Configuration parameters include configuring the maximum memory available to AutoVue, location of markups, cache and profiles, GUI file to use on startup, new window behavior, and so on. Changes to autovue.properties will take effect when AutoVue is restarted. Refer to the following table for the list of configuration parameters:

A.1 Configuration Parameters

The following table lists the parameters that can be configured.

Table A–1 Configuration Parameters

Property	Description	Default
Main Options		
autovue.inifile = *file name*	Specify the INI file where AutoVue stores information on the file formats supported by AutoVue. We recommend that you do not change this option value. Note: AutoVue also saves certain memory management settings in this file. We recommend that you do not modify these options or the file (VueCore.ini).	VueCore.ini
User Profile Options		
autovue.users.directory = *file path*	Contains the directory in which user information is stored (initialization files and GUI files). The user must have write permissions to the folder. AutoVue creates an <*username*> .ini file for every user that accesses AutoVue. This file is stored at the location specified by the autovue.users.directory parameter.	<AutoVue installation directory>\bin\Profiles
autovue.users.defaultini = *file name*	AutoVue provides a way to force certain INI settings to the user INI the first time the user accesses AutoVue. This is done by setting the required options in the default.ini file or in the file specified by autovue.users.defaultini parameter. This file should be located at <AutoVue Installation Directory>\bin directory.	default.ini

Table A–1 (Cont.) Configuration Parameters

Property	Description	Default
autovue.users.allusersini = *file name*	AutoVue provides a way to force INI settings to the user profile every time a user accesses AutoVue. This is done by setting required options in allusers.ini (or the files specified by autovue.users.allusersini). This file should be at <AutoVue Installation Directory>\bin directory.	allusers.ini
Proxy Options		

autovue.http.proxyhost
autovue.ftp.proxyhost
	If a file requires resources from the Internet, AutoVue must connect to the Internet to retrieve the required resources. If the machine that AutoVue is installed on uses a proxy server to connect to the Internet, you must set the proxy setting to allow the request to go through.
	For example:
	`autovue.http.proxyhost=my.proxyserver.com:80`
	`autovue.ftp.proxyhost=my.proxyserver.com:80`	

Startup Options		
autovue.classpath	Specifies the classpath needed for AutoVue to startup successfully. We recommend that you do not change this parameter.	
autovue.jre	Specifies the path to the java executable used by AutoVue. We recommend that you do not change this value.	<AutoVue installation directory>\jre\bin\java.exe
autovue.jre.path	Specifies the path to the JRE used by AutoVue. We recommend that you do not change this value.	<AutoVue installation directory>\jre
autovue.cmdline	Specifies the command line parameters for AutoVue. You can modify memory settings for AutoVue, modify the log4j configuration and specify additional commands here using the *-Param* flag. Do not modify other values. Refer to Command-Line Parameters for more information.	
autovue.splash=*full path of PNG/JPEG*	Specify the splash screen to display while AutoVue starts up. If the full path of the PNG or JPEG is not specified, the splash screen does not display.	<AutoVue installation directory>\bin\AutovueSplash.png

Table A–1 (Cont.) Configuration Parameters

Property	Description	Default
autovue.help.file.<*language*> = *file path*	Entry specifies the file path to the English (en)/French (fr)/German (de) Help file.	<AutoVue installation directory>\html\help\<*language*>\AutoVueOnlineHelp.html *For example:* <AutoVue installation directory>\html\help\en\AutoVueOnlineHelp.html
Markup Options		
autovue.markup.nativegui.type = [0|1|2|3|4|5|6|7]	Add Author, Date, and Markup Info columns to the Markup Files dialog box. **0**: Name column displays **1**: Enable Author **2**: Enable Date **4**: Enable Markup Info **5**: 1 + 4 (Author + Markup Info) **6**: 2 +4 (Date + Markup Info) **Note**: These are ORed flags. For example: Enter 7 to enable all three columns. The Markup Info consists of 5 columns: ■ USER ■ DEPARTMENT ■ COMPANY ■ LOCATION ■ TELEPHONE	0
autovue.markups.local.directory = *file path*	In version 20.1 and up, markup files are saved with the name <base file name>.<end-user specified extension>. **Note**: The end-user specified extension is identified as "Markup ID" in the "Save Markup File AS" dialog. By default, AutoVue Desktop Deployment saves markups in the avred directory at the base file location. If the user does not have permission to write into the base file location, they can export the markup into an alternate directory. Alternatively, user can set this property autovue.markups.local.directory to a location where all markup files should be stored. **Note**: If you change the markup save location, you will need to import or open and then save the markups from the old location so that they are saved at the new location.	No default value; markups are stored at: <*Base file location*> \avred

Table A–1 (Cont.) Configuration Parameters

Property	Description	Default
autovue.markups.symbols.directory=*file path*	Specifies the directory where AutoVue stores all symbol libraries and symbol files.	<AutoVue Installation Directory>\bin\Symbols.

Table A–2 Cache Options

Property	Description	Default
autovue.cache.directory = *file path*	Specifies in which directory the cached files should be saved. A central cache information file named *cache.map* is stored in the same directory. The user must have write permissions to the folder.	<AutoVue installation folder>\bin*Cache*
autovue.cache.size = *int*	Specifies, in Megabytes, the maximum size of the file cache. If not specified, or if value specified is less than 50 MB, the default value of 4GB is used.	4096
autovue.cache.maxnumfiles=*integer*	Specifies the maximum number of files allowed in the cache directory.	64000
autovue.cache.maxlifetime=*integer*	Specifies the maximum number of days a file is kept in the AutoVue cache directory.	30

A.1.1 Changing the Locale of AutoVue

AutoVue supports the following languages for the user interface:

- English (EN)
- Dutch (NL)
- French (FR)
- German (DE)
- Japanese (JA)
- Traditional Chinese (TW)
- Simplified Chinese (ZH)
- Korean (KO)
- Swedish (SV)

On startup, AutoVue detects if the language of the OS is one of its supported locales and automatically switches the user interface to the locale of the operating system. When the OS language is not supported by AutoVue, the AutoVue user interface defaults to English.

It is possible to force the AutoVue user interface to a supported language by specifying the LOCALE parameter. Refer to Command-Line Parameters for the structure and syntax. In cases where the operating system locale and AutoVue locale are different, you might notice that some dialogs or buttons in AutoVue use the locale from the OS. This is because AutoVue relies on the OS for UI items such as, Open, Close, OK, Cancel,.... You can set the Java parameter user.locale=<locale> to ensure that these items also appear in the right locale.

This option must be set in the autovue.cmdline property in autovue.properties.

You can also set the Java parameter user.country=<country> for country-specific localization.

A.2 Command-Line Parameters

You can configure AutoVue by modifying certain settings in autovue.cmdline property in autovue.properties.

A.2.1 Setting the -PARAM Flag

You can specify command line parameters to AutoVue by setting the -Param flag in the *autovue.cmdline* property in autovue.properties.

Syntax:

```
-PARAM <name>=<value>
```

The Table A–3 provides a list of parameters that can be configured.

Table A–3 Command Line Parameters

Name	Type	Value
FILENAME	Note: When specifying a local file, ensure that the filepath is preceded by an upload:// protocol. For example: `upload://c:\samples\2d\acad.dwg`	
	http:*//host/file*	Specify a HTTP URL for file open.
	ftp:*//host/file* or... ftp:*//<user>:<password>@ <ftpserver>/file*	Specify a FTP URL for file open.
EXTRABUNDLES	*name of the bundle file*	If you are adding custom actions to AutoVue, you can specify the name of the custom resources file using this parameter. Names of the custom resource files are expected to follow: filename_ XX.properties, where XX – is an ISO639 two-letter code representation of a language. When specifying the custom resources using this parameter, do not specify the language and the extension. For example: -PARAM EXTRABUNDLES=/CustomActions

Table A–3 (Cont.) Command Line Parameters

Name	Type	Value
GUIFILE	String	The Graphical User Interface (GUI) definition file used. GUI files are stored in subdirectories of the root directory specified in the autovue.users.directory parameter of the autovue.properties file. The specification can also specify a local file using the **"file://"** convention. Default for the autovue.users.directory parameter is **<bin dir>\Profiles**. Refer to Customizing GUI for more information.
HEAVYWEIGHT	[TRUE | FALSE | AUTO]	Specify if you would like to use JOGL's heavyweight or lightweight widget to render 3D Models. When heavyweight is on, AutoVue uses hardware acceleration to render 3D. Default is AUTO.
LOCALE	[DE | EN | NL | FR | JA | KO | TW | ZH | SV]	The Locale to be used in the user interface, specified as an ISO639 two-letter code. Using this parameter, you can force the AutoVue GUI to be displayed in one of the supported languages. If not set, the Locale is determined using the system properties. Syntax: -PARAM LOCALE="FR" Refer to Changing the Locale of AutoVue for additional configuration that maybe necessary for forcing AutoVue user interface to a specific LOCALE.
NEWWINDOW	[TRUE | FALSE]	Specifies whether to launch a new window when running an additional instance of AutoVue. The default value is **TRUE**. If set to **TRUE**, a new window is launched without replacing the existing AutoVue window when there is already an instance of AutoVue running. If set to **FALSE**, a new window is not launched when there is already an instance of AutoVue running. **Note**: If you set NEWWINDOW to FALSE and you specify 2 or more files to open in the command line for AutoVue, AutoVue will open these files in separate windows.

B

Configuring Memory Settings

This chapter discusses how to configure the memory settings.

B.1 Configuring Memory Settings

The maximum memory available to the Java component of the Desktop Deployment is determined by the *-Xmx* flag in the *autovue.cmdline* property. By default, this is set to 256MB. If you encounter a situation where AutoVue indicates that it is running low on Java Virtual Memory (JVM), you must set the maximum memory to a higher value. It is recommended that you increase the maximum memory for the Java component to 512MB.

AutoVue Desktop Deployment has both Java and native components. The *-Xmx* flag only controls the memory available to the Java components. It is recommended that you do not set the memory for the Java component to a very large value as it results in less memory for the native components. The native components use the remaining memory on the machine after the specified Java memory used by the Desktop Deployment.

You can tell the JVM how to allocate/deallocate heap memory with the *-XX:MinHeapFreeRatio* and*-XXMaxHeapFreeRatio* options. By setting these options to a reasonably small value (for example, *XX:MinHeapFreeRatio=10* and *-XXMaxHeapFreeRatio=10)* the memory allocated is reduced and the unused memory may be utilized elsewhere. Ultimately, it is up to the garbage collection settings of the JVM whether or not to use these options. As a result, these options are not critical and removing them does not cause the Desktop Deployment to stop working, the Desktop Deployment may just use more memory.

For more information on these options, refer to the documentation at `https://www.oracle.com/technetwork/java/javase/vmoptions-jsp-140102.html`.

C

Non-Interactive Installations

This chapter discusses the non-interactive installations of the application, and covers both installation, and uninstalling the components.

C.1 Installation

To install AutoVue in non-interactive mode, you need to specify a configuration file that contains the required installation parameters. To do so, you must generate the configuration file manually following this syntax.

```
#Specify Installation Directory
#------------------------------
USER_INSTALL_DIR=C:\\Oracle\\AutoVue

#Select Shortcut Folder
#----------------------
USER_SHORTCUTS=C:\\Documents and Settings\\Administrator\\Start
Menu\\Programs\\Oracle AutoVue
```

Following are the installation parameters that you can specify in the configuration file:

Table C–1 Installation Parameters

Parameter	Description
USER_INSTALL_DIR={*file path*}	Specify the path where you want to install AutoVue.
USER_SHORTCUTS={*file path*}	Specify the shortcut path.

After you specify the parameters for the configuration file, you can run the installation in non-interactive mode. Enter the following command lines:

```
InstallDesktopDeployment.exe -i silent -f <full path to configuration file>
```

C.2 Uninstallation

If AutoVue is installed in non-interactive mode, the uninstallation is automatically in non-interactive mode. Simply invoke the uninstaller for AutoVue:

```
<AutoVue Installation Folder>\uninstall\uninstall.exe
```

D

Debugging AutoVue

This chapter discusses what package AutoVue uses for debugging.

D.1 Logging for AutoVue

AutoVue uses the log4j package to generate debug information. The configuration file *log4j.xml* (located in the <AutoVue Install Root>\bin directory) makes it possible to display debugging information for AutoVue. Log files are created in the <AutoVue Install Directory>\bin\logs folder by default. In situations where trouble-shooting information is necessary, an Oracle Global Customer Support representative will ask you to update the log4j.xml and enable logging for AutoVue.

Note: By default, logging is enabled at the WARN level. Any warning messages generated while you use AutoVue are written to the log4j output file.

For additional information regarding log4j, go to Apache's log4j documentation.

The following section describes the classes for which you can enable debugging information.

D.1.1 Logger Information

The following descriptions explain what kind of logger information will be seen for each class specified:

Table D–1 Logger Information

Class	Description
`com.cimmetry.jvueserver.configuration`	Displays reports on loading errors of AutoVue's configuration.
`com.cimmetry.jvueserver.event`	Displays information concerning posting and handling of different AutoVue events (opened and closed sessions, opened and closed documents, and so on).
`com.cimmetry.jvueserver.cache`	Displays information concerning the AutoVue cache. Reports messages and errors related to loading the cache, locking, saving, deleting cached files as well as searching for archive and XRef files.
`log4j.category.com.cimmetry.connection`	Displays information concerning downloading files from the network.
`com.cimmetry.jvueserver.document`	Displays document-related information (open, information, properties, and so on).
`com.cimmetry.jvueserver.document.native`	Displays messages and error reporting for document related native code execution.

Table D–1 (Cont.) Logger Information

Class	Description
`com.cimmetry.jvueserver.symbols`	Displays mapping between the stamp library name in the UI and the actual SMB file name on AutoVue.
`com.cimmetry.frontend`	Displays all messages and errors reported from AutoVue.

You can specify what kind of information to output by setting the classes to one of the following information levels:

Table D–2 Information Levels for Classes

Information	Description
`INFO`	Displays informative messages such as session information, document open requests.
`WARN`	Displays warning messages. This is the default value.
`ERROR`	Displays errors or exceptions.
`OFF`	Turn logging off.
`DEBUG`	Displays debug messages

D.1.2 Configuring log4j

The log4j configuration file to use is specified by the *-Dlog4j.configurationFile* parameter in *autovue.cmdline*. By default, this file is <AutoVue Installation directory>\bin\log4j.xml.

E

Customizing GUI

The following sections describe in detail how to configure AutoVue to your needs.

E.1 Customizing the GUI

This section discusses the GUI file to choose and modify.

E.1.1 Choosing the GUI File

AutoVue provides you the option of customizing your graphical user interface (GUI). By default, a GUI definition file is not set and AutoVue uses an internal GUI file for the menus and toolbars. The GUI file that AutoVue generates is the same as the default.gui file located in the <AutoVue Install Root>\bin directory.

If you wish to have a customized GUI for AutoVue, you must create a custom GUI file and specify this custom file using the *GUIFILE* parameter. GUI files are placed at the location specified by the autovue.users.directory parameter in autovue.properties. By default, the location is <AutoVue Install Root>\bin\Profiles.

E.1.2 Modifying the GUI

The GUI definition file describes which controls are added to which context (such as MenuBar, ToolBar, and so on).

If you are customizing your GUI file, it is recommended that you make a backup of the default.gui file and modify the controls in this file to meet your needs. The default.gui file is located in the <AutoVue Install Root>\bin folder.

If you have a previous version of AutoVue and you used a customized GUI in this previous version, we recommend that you use the diff utility to perform a comparison between the previous version's default.gui and your customized GUI. The delta between the two GUI files should be manually applied to the current version GUI.

Important: It is good practice to update your newer GUI file with the delta between the two GUI files. In order to avoid situations where some or all of the GUI elements fail to load, we recommend that you do not use the previous version's GUI file.

It is recommended to hide GUI items by changing PERM_READ to PERM_HIDE instead of removing them from the GUI file.

E.1.2.1 Structure and Syntax of GUI Files

The GUI definition file describes which controls (corresponding to available actions in AutoVue, like Rotate, Open, and so on) are to be added to which context (like

MenuBar, ToolBar, and so on), thus allowing users to have complete control over the functionality of the AutoVue interface.

AutoVue supports five modes: View, Compare, Markup, Collaboration, and Print Preview. A GUI file defines the graphical interface for each mode. Menu bars, toolbars, status bar and Right Mouse Button (RMB) menus are defined in this file. For some of these objects, location (north, south, west, east) may be specified. Toolbars are located in north, west or east. The status bar is always located at the bottom of the component (south).

> **Note:** Popup menus may be added to menu bars. Menu items, popup menus or separators may be added to popup menus. Toolbars only accept buttons. Buttons or panes may be defined for the status bar. The RMB popup is processed as any other popup menu.

The Table E–1, " GUI Keyword" lists each GUI keyword for each mode:

Table E–1 GUI Keyword

	3D	2D	EDA
View	SMVIEW	VIEW	ECADVIEW
Markup	MARKUP3D	MARKUP	ECADMARKUP
Collaboration	COLLABORATION3D	COLLABORATION	ECADCOLLABORATION
Compare	COMPARE3D	COMPARE	COMPARE
Print Preview	PRINTPREVIEW	PRINTPREVIEW	PRINTPREVIEW

E.1.2.2 GUI Configuration Syntax

The most generic definition of a GUI file can be described through the symbols below:

- Words with `CAPITAL LETTERS` should be entered literally.
- The character '| ' is used as "or" (for example, a|b means a or b)
- The character '* ' means "zero or more occurrences of."
- A GUI file can contain one or more "GUI configuration" blocks as shown in the following table:

Table E–2 GUI Configuration Blocks

GUI Configuration Blocks

`GUI_configuration =`
`BEGIN UI VIEW UI_mode_configuration END`
`{BEGIN UI COMPARE | MARKUP UI_mode_configuration END}`

`*UI_mode_configuration =` `{menu_bar_configuration | {toolbar_configuration}* |status_bar_configuration|RMB_popup_menu_configuration}`
`menu_bar_configuration =` `MENUBAR BEGIN {popup_menu_configuration}* END`
`toolbar_configuration =` `TOOLBAR NORTH|WEST|EAST BEGIN {button_control}* END`

Table E–2 (Cont.) GUI Configuration Blocks

GUI Configuration Blocks
`status_bar_configuration =` `STATUSBAR SOUTH BEGIN {button_control|pane_control} * END`
`RMB_popup_menu_configuration =` `RMB BEGIN {popup_menu_configuration|menu_item_control}* END`
`popup_menu_configuration =` `POPUP IDS_{FILE |EDIT |VIEW |OPTIONS |HELP |MANIPULATE |ANALYSIS |MARKUP |COLLABORATION} BEGIN {popup_menu_configuration|menu_item_control| SEPARATOR }* END`
`button_control =` `BUTTONaction_control`
`menu_item_control =` `MENUITEMaction_control`
`pane_control =` `PANEaction_control`
`action_control =` `control_name,control_key_list,permissions`
`control_name`: For list of available control names refer to Control Names.
`control_key_list`: For the control key list for different controls refer to Control Names.

`permissions`: All action names need "`PERM_READ`".
These are the exceptions to this rule:
`VueActionFilePrint` needs: `PERM_READ|PERM_HEADERS|PERM_WATERMARK`
`EcadActionSelect needs: PERM_HIDE`
`SMActionSelect needs: PERM_HIDE`
Example:
To define a very basic user interface that only allows users, through menu items, to open or print a file and get the file information without changing watermark/headers/footers:
`BEGIN UI VIEW`
`  MENUBAR BEGIN`
`  POPUP IDS_FILE BEGIN`
`  MENUITEM VueActionFileOpen, , PERM_READ`
`  MENUITEM VueActionFileProperties, , PERM_READ`
`  MENUITEM VueActionFilePrint, , PERM_READ`
`  END`
`  END`
`END`

E.1.2.3 Control Names

The following table lists available Control Names and their functionality.

The letters in the *UI* Modes* column of the table indicate:

V - **V** iew

C - **C** ompare

M - **M** arkup

Table E–3 Control Names

Control Name	UI* Mode	Functionality	Control Key List	Contexts			
				Popup Menu	Toolbar	Status Bar	RMB
VueAction FileOpen	VC	When INI option EnableUniversalFile Chooser is set to 0, invokes open URL dialog. When option is set to 1, the universal file chooser dialog (that supports URLs, local files, and DMS files) appears. Default for EnableUniversalFile Chooser is 1.		X			
VueAction FileUpload	VC	Upload local file when EnableUniversalFile Chooser=0. Not available when EnableUniversalFile Chooser=1.		X	X		
VueAction FileMarkup	V	Switch to Markup mode		X	X	X	X
VueAction FileCompare	V	Switch to compare mode		X			
VueAction FileOverlays	V	Launches the Overlays dialog to select and modify overlays		X			
VueAction FileProperties	VCM (M: status bar only)	Show file properties		X		X	
VueAction FilePrint	VCM	Launch the print dialog that lets you modify print options and print a file		X	X		
VueAction FileMRU	V	List most recently used documents		X			
VueAction EditSearch	VM	Launch the search dialog to perform search or repeat search		X	X		

Table E–3 (Cont.) Control Names

Control Name	UI* Mode	Functionality	Control Key List	Contexts			
				Popup Menu	Toolbar	Status Bar	RMB
VueAction ViewZoom	VCM	Apply zoom	In/ Out/ Previous/ FullRes/ FitBoth/	X	X		X
VueAction ViewFlip	VCM	Apply flip	Vertical/ Horizontal /Both	X	X		
VueAction ViewRotate	VCM	Apply rotation	0/ 90/ 180/ 270	X	X		
VueAction ViewContrast	VCM	Apply contrast		X			
VueAction ViewAntiAlias	VCM	Apply anti alias		X			
VueAction ViewInvert	VCM	Apply invert		X			
VueAction ViewPage	VCM	Go to next page, previous page or select page number.		X	X		
VueAction ViewViewPoi nt	VC	Launches the view point dialog that lets you define a view point.		X			
VueAction ViewXrefs	VCM	Launches the XRefs dialog that lets you toggle XRefs visibility on or off.		X	X		
VueActionVie wLayers	VCM	Launches the dialog that lets you toggle layer visibility on or off.		X	X		
VueAction ViewBlocks	VCM	Launches the Blocks dialog that lets you select a block to display.		X	X		
VueAction ViewViews	VCM	Launches the Views dialog that lets you select a view to display.		X	X		
VueAction ViewDrawing Info	VCM	Get the selected entity's drawing information		X			

Table E–3 (Cont.) Control Names

Control Name	UI* Mode	Functionality	Control Key List	Contexts			
				Popup Menu	Toolbar	Status Bar	RMB
VueAction ViewMeasure	VCM	Launches the Measurement dialog that lets you measure distance, cumulative distance, area, or calibrate		X			
VueAction ViewSpecialViewModes	VCM	Show special view modes	Pan and Zoom Window/ MagnifyWindow/MagnifyGlass	X	X		
VueAction ToolsDrawingInfo	VCM	Get drawing information for one entity, some entities or a block **Note**: This feature is equivalent to the Show Entitiy Properties option from the AutoVue UI. For more information, refer to the Oracle AutoVue User's Manual.		X			
VueAction OptionsBars	VCM	Hide or show toolbars or status bar		X			
VueActionFileBrowse	VM	Opens the File Browse dialog when browsing documents from different sources (local, DMS, server, URL). The document is open as soon as it is single clicked. Available only when EnableUniversalFileChooser=1 (default).			X		
VueActionFileConvert	VM	Launches the Convert dialog that lets you convert a file to different formats using convert options.			X		
VueActionFileOpenNewWindow	VM	Same as VueActionFileOpen, but opens file in a new window.			X		

The columns indicate:

- **Control Name**: Column shows the list of available control names.

- ***UI modes***: Column specifies in which modes (View, Markup, and Compare) the control names can be used.

  Example: VueActionFileOpen can be added to View and Compare Modes, but not Markup mode.

- **Functionality**: Column specifies which functionalities are provided when this control is added to a context.

  Example: Adding **VueActionFileMarkup** to any context enables you to switch to Markup mode.

- **Control key list:** Column provides the optional functionalities that can be added to a context.

  - If there is no entry for a control name in this list, it means that there is only one action to invoke. A list specifies sub-actions. For example, for **VueActionFileOverlays**, there is no entry in the control key list, so adding it to a popup menu will provide both select and modify functionalities for overlays. The entry will look like this:

    ```
    MENUITEM VueActionFileOverlays, , PERM_READ
    ```

  - If there is a list of strings separated by '/', you can specify which functionalities you want added. If you don't specify any of them, by default all functionalities will be added. For example the following entry adds two buttons to the toolbar: one for Zoom In and one for Zoom Out:

    ```
    BUTTON VueActionViewZoom, In/Out, PERM_READ
    ```

    Whereas

    ```
    BUTTON VueActionViewZoom, , PERM_READ
    ```

    is interpreted as

    ```
    BUTTON VueActionViewZoom, In/Out/Previous/FullRes/FitBoth, PERM_READ
    ```

- **Contexts:** Column provides the contexts to which you can add the control to.

  Example: You can add an entry in a popup menu of the menu bar, but not in an RMB configuration

F

Samples and API Examples Included with AutoVue

During the installation process, if you select Custom installation, AutoVue provides you options to install samples and API examples. This chapter provides an overview of the samples and API that are installed with AutoVue.

F.1 API Examples

The following API Example is installed at <AutoVue Install Root>\examples if you do a Custom installation and choose to install API example:

- VueActionSample

  This provides an example of how to implement hotspots using the VueAction. For more information on VueAction, refer to the AutoVue API Programmer's Guide and to the VueBean JavaDocs.

  To use this sample:

  - Hotspots.txt contains some hotspot definitions. You can use these definitions to test this sample or you can create your own definitions.
  - PartCatalogueAction.java and PartListAction.java demonstrate how to write custom actions.
  - A custom GUI file is the customized GUI for the custom actions. Copy this file to <AutoVue Install Root>\bin\Profiles.
  - Update the Java code as needed and compile the code
  - Bundle all the class files into VueActionSample.jar
  - Run this sample using the following command:

    ```
    java -cp <full path to jvue.jar>;<full path to VueActionSample.jar>
    com.cimmetry.jvue.JVue -param GUIFILE=<path>/custom.gui -param
    EXTRABUNDLES=/PartCatalogueAction
    ```

Note: The "Hotspots" section of the Oracle AutoVue API Guide provides information on how to implement AutoVue's hotspots API using JavaScript.

F.2 Sample Files

When you choose to install sample files during the AutoVue installation process, the following samples are installed:

- Sample 2D, 3D, EDA, Office and Graphics files are installed at <AutoVue Install Root>\html\samples

F.3 Adding Custom Tools Menu to AutoVue

You can add custom Tools menu items to AutoVue so that batch files can be triggered from the menu items. AutoVue provides some example code that demonstrates how to build custom Tools menu items. The example code is installed when you do a Custom installation of AutoVue and select the API Examples check box during installation. The steps for adding custom Tools menu are below. Note that this requires familiarity with programming in Java.

1. Copy your batch files to <AutoVue Installation Directory>\bin.
2. Edit <AutoVue Installation Directory>\examples\CustomActionTools\CustomActionTools.java and edit the following lines:

Table F–1 Code Lines

Code Lines to be edited
`private static final String resources[]={"TOOL1","TOOL2"};`
private static final String[] subActions={"tool1","tool2"};
private static final String batchfiles[]={"batch1.bat","batch2.bat"};

where:

- *resources* is the list of strings that will show up in the Tools menu. The first string should correspond to the first batch file, the second string to the second batch file, and so on.
- *subActions* is a list of strings that is used to associate a particular Tool menu item to a batch file.
- *batchfiles* is the list of batch files.

3. Compile the Java code.
4. Copy the generated CustomActionTools.class to <AutoVue Installation Directory>\bin.
5. Make a copy <AutoVue Installation Directory>\bin\default.gui to a file called tools.gui. Copy this GUI file to <AutoVue Installation Directory>\bin\Profiles.
6. Edit tools.gui and add the new CustomActionTools to an existing or new menu: *MENUITEM CustomActionTools, PERM_READ*
7. If you are referencing new resources, you will need to update file CustomResources_en.properties with the new resources.
8. You can translate the custom resources. The translations should be named CustomResources_*xx* .properties, where *xx* represents the two-letter language code. Refer to Changing the Locale of AutoVue for the list of language code.

9. Copy the resource files to <AutoVue Installation Directory>bin.
10. Add the following to the autovue.cmdline property in autovue.properties:*-param GUIFILE=tools.gui -param EXTRABUNDLES=/CustomResources*

When you startup AutoVue, you should see new Tools menu items corresponding to your definition in CustomActionTools.java.

G

Feedback

If you have any questions or require support for AutoVue please contact your system administrator.

If at any time you have questions or concerns regarding AutoVue, please contact us.

G.1 General AutoVue Information

Web Site	http://www.oracle.com/us/products/applications/autovue/index.html
Blog	http://blogs.oracle.com/enterprisevisualization/

G.2 Oracle Customer Support

Web Site	http://www.oracle.com/support/index.html

G.3 My Oracle Support AutoVue Community

Web Site	https://communities.oracle.com/portal/server.pt

G.4 Sales Inquiries

E-mail	https://www.oracle.com/corporate/contact/global.html