

Oracle® AutoVue for Agile PLM

Installation and Configuration Guide

Release 21.0.2

F12811-02

November 2022

AutoVue for Agile PLM Installation Guide, Release 21.0.2

F12811-02

Copyright © 2010, 2022, 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" Release 21.0.2 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	vii
Audience	vii
Documentation Accessibility	vii
Related Documents	vii
Conventions	viii
 1 Introduction	
 2 Architecture	
Components of Agile PLM	2-1
Agile PLM Application Server	2-1
File Manager	2-2
AutoVue Components (Server and Client Components)	2-2
 3 Prerequisites	
Pre-requisites for Installing AutoVue for Agile PLM	3-1
Windows Prerequisites	3-2
Linux Prerequisites	3-2
Additional Guidelines for Security	3-3
Default Locations	3-4
 4 AutoVue Hardware and System Requirements	
Hardware Requirements	4-1
System Requirements	4-2
Server Platforms	4-2
Client Platforms	4-3
Integrations	4-3
AutoVue for Agile PLM Integration Components	4-4
 5 Installation Checklist	
Deployment Scenarios	5-1
Installation Checklist	5-1
Additional Recommendations	5-2
 6 Installing and Upgrading AutoVue for Agile PLM	
Upgrading from earlier versions	6-1
Upgrading from AutoVue for Agile PLM Version 20.2.x/21.0.0/21.0.1	6-1
Upgrading from AutoVue for Agile PLM Version 20.1.x or Earlier	6-2
Installing on Windows platform	6-3
Running the Installation Program	6-4

Installing on Linux platform	6-5
Installing on other platforms	6-6
Installing AutoVue Client Libraries and Integration Components on Different Platforms ...	6-6
Installing AutoVue for Agile in Console Mode	6-7
Running the Installation Program in Console Mode	6-7
Configuring the File Manager	6-8
Upgrading from Earlier Version with MetaFilesRemover Utility	6-8
Verifying your Integration	6-9
Post-Installation Instructions	6-9
Configuring SSL	6-9
Configuring User Authentication.....	6-9
Deploying VueServlet.....	6-10
Creating a WAR for the VueServlet	6-10
Deploying the VueServlet.....	6-12
Verifying VueServlet Deployment	6-12
Troubleshooting VueServlet Deployment.....	6-12
Deploying JNLP Components.....	6-13
Deploying VueJNLPServlet.....	6-15
Deploying VueKeyPairServlet	6-16
Deploying VueRDVServlet.....	6-16
Installing AutoVueClient Components	6-17
Configuring AutoVue Client for Secure Socket Layer (SSL)	6-20
Verifying AutoVue Client.....	6-21
Troubleshooting AutoVue Client	6-21
Configuring Online Help.....	6-22

A Configuring AutoVue for Agile PLM

Security	A-1
Starting AutoVue Server on Windows	A-1
Starting AutoVue Server on Linux	A-2
Shutting Down the AutoVue Server	A-3
Running the AutoVue Server as a Service	A-3
On Windows OSes	A-3
Starting and Stopping the Service	A-3
On Linux OSes	A-4
Testing the VueServlet	A-5
Managing Multiple Viewers for Load Balancing	A-5
Client-Server Communications	A-6
Removing AutoVue for Agile PLM.....	A-6

B AutoVue Server Configuration Options

Authentication Option	B-1
AutoVue Host Name Option	B-1
RMI and Socket Ports Options	B-1
Process Pool Size Option	B-2
Proxy Connection Options	B-3
Streaming Files Options	B-3

DMS Options	B-5
Collaboration Options.....	B-5
log4j and Diagnostics Options	B-6
Modified With JavaScript Option.....	B-7
Reboot Option.....	B-7
DocServer Timeout Option	B-7
Recovery Attempt Option.....	B-8
DLL Version Option	B-8
File Format Information Option.....	B-8
Global User Options	B-8
Markup Options	B-9
Online Help Options	B-10
Memory Optimization.....	B-10
Vuelink Encryption Options.....	B-11
Linux-Specific Options	B-13
Preload Java Class Option	B-13
Xvfb Options	B-13
WINE Options	B-14
OEM Copyright Notice	B-14
VueServlet Configuration Options.....	B-14
 C ABV Configuration	
Configuring the ABV Framework.....	C-1
 D Migration from Loopback to Rendezvous	
Loopback Connection.....	D-1
Rendezvous Communication.....	D-2
Configuration.....	D-3
Cookies	D-4
Example	D-5
Migration from loopback connection to Rendezvous communication.....	D-7
 E Troubleshooting	
Troubleshooting AutoVue for Agile PLM	E-1
Enabling Client Verbosity	E-1
Detailed WebLogic Logs	E-1
File Manager (Tomcat logs)	E-1
File Manager VueServlet Logs	E-1
AutoVue Detailed Client/Server Logs	E-3
Errors Which May Occur While Using the AutoVue Server	E-3
Changing Port Numbers	E-4
Verify AutoVue Server.....	E-5
 F FAQ	
Linux.....	F-1

G Feedback

General AutoVue for Agile PLM Information	G-1
Oracle Customer Support	G-1
My Oracle Support AutoVue for Agile PLM Community	G-1
Sales Inquiries	G-1

Preface

Oracle Agile PLM is a comprehensive enterprise PLM solution for managing your product value chain. Oracle AutoVue for Agile PLM 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). The Oracle AutoVue for Agile PLM Installation and Configuration Guide describes how to install and configure Oracle AutoVue for Agile PLM.

Audience

This document is directed at any user whose task is the installation and administration of Oracle AutoVue for Agile PLM.

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

Oracle's Agile PLM documentation set includes Adobe® Acrobat PDF files. The Oracle Technology Network (OTN) Web site
<https://www.oracle.com/technetwork/documentation/agile-085940.html> contains the latest versions of the Agile PLM PDF files. You can view or download these manuals from the Web site, or you can ask your Agile administrator if there is an Agile PLM Documentation folder available on your network from which you can access the Agile PLM documentation (PDF) files.

Oracle's AutoVue for Agile PLM documentation set is available at
<https://www.oracle.com/technetwork/documentation/agile-085940.html#autovue>

For more information on Oracle AutoVue products, refer to the Oracle AutoVue documentation set available at
<https://www.oracle.com/technetwork/documentation/autovue-091442.html>

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.
<i>italic</i>	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.

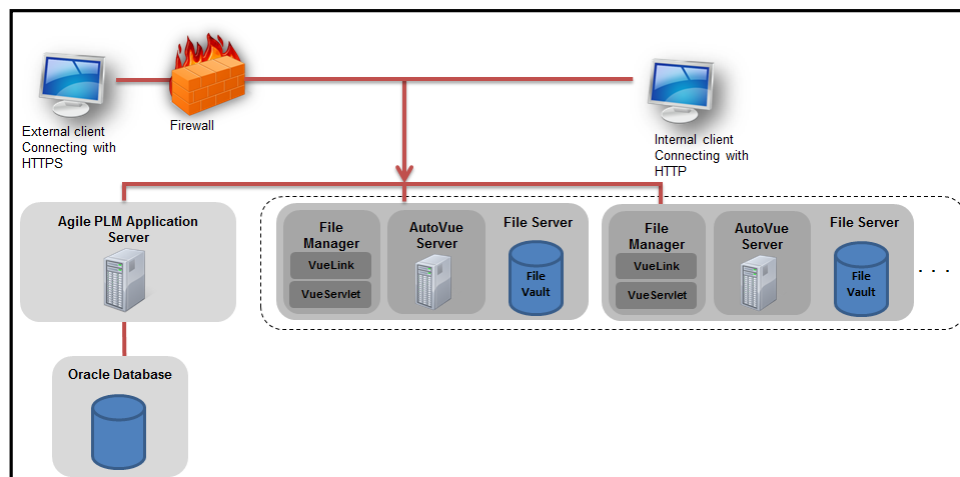
Introduction

Oracle AutoVue for Agile PLM is an Enterprise Visualization solution designed to provide viewing, digital annotation and collaboration capabilities to Agile PLM. Oracle 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). Hence, one of the main benefits of AutoVue for Agile PLM is that users can view numerous file types stored in Agile PLM without having the native application that created the file installed on their desktop machines. Another advantage of AutoVue for Agile PLM is that it decreases bandwidth requirements and improves response time through the use of streaming files. Streaming files contain file data and provide the benefit that AutoVue for Agile PLM can access the streaming files much faster than it can native files.

The VueServlet provides a connection between the AutoVue server and Agile PLM using standard HTTP/HTTPS protocols.

The following diagram and steps describe a typical configuration of how AutoVue for Agile PLM integrates with Agile PLM.

Figure 1-1 Architecture Diagram



1. Log onto the Agile PLM with web browser.
2. A link appears next to each file stored in Agile PLM.
3. When you click this link, the AutoVue client is launched and you may view that file inside the web browser window.

-
4. In Agile PLM, the AutoVue client communicates with the AutoVue server through servlet tunneling for HTTP or HTTPS connection. The Vuelink can communicate with Agile PLM to handle any request made by the AutoVue server like getting the files that the user requested for.

Architecture

This chapter discusses the architecture of AutoVue for Agile PLM.

AutoVue for Agile PLM is designed to provide server scalability to support increasing demand for file viewing and markup on intranets and the Internet. Adding capacity is as simple as adding one more "server". The servers automatically divide up the load in a "peer-to-peer" fashion to avoid any bottlenecks. In Agile PLM, there is a 1-to-1 relationship between the File Manager and AutoVue server. As a result, you can add another server (under a load-balancer with the File Manager and AutoVue client libraries installed) to increase the capacity. However, with sufficient CPU and memory resources on the server, you can also increase the number of processes in the AutoVue server to improve scalability.

AutoVue for Agile PLM embeds several different rendering schemes to accommodate multiple environments. These schemes allow the server to take into account issues such as network bandwidth, load balancing between server and client, server configuration and performance, intranet/extranet/Internet use, and file size and format to be viewed. However, in all rendering options, AutoVue for Agile PLM keeps all necessary information for querying purposes; the "intelligence" of drawings is preserved. For example, users can perform text searches on text and CAD files, or perform queries based on drawing attributes in a CAD file.

Note: Agile PLM is a Web application server with a browser-based client and configures access through the VueServlet, exclusively.

2.1 Components of Agile PLM

The components of AutoVue for Agile PLM include the:

- Agile Application Server
- File Manager
- AutoVue server
- AutoVue client libraries

2.1.1 Agile PLM Application Server

The Agile PLM Application Server is the center of the Agile PLM system, and manages data stored in the Agile PLM database. The Agile PLM Application Server runs on industry-leading J2EE application servers. The application server connects to the components in a persistence layer where product content is stored. All Agile PLM data is contained or organized in business objects that are set up by the administrator, and specified and used by the enterprise's Agile PLM users. Business objects is a general

term that implies objects created from the classes available to the enterprise, but other entities in Agile are also objects, such as workflows, searches, reports, and so forth.

2.1.2 File Manager

The File Manager stores all documents, drawings, and other files within the Agile PLM system. A File Manager provides a place to store and retrieve files locally or remotely. Due to the geographically dispersed nature of the global enterprise, multiple Agile PLM File Managers can be deployed in a distributed configuration for efficient distribution of product content. Agile PLM File Manager is made up of two main components: the File Manager Web application and the file vault. The file vault is the file system where the actual files reside. The file vault can be located on the same server as the Web application or on a dedicated storage system. The File Manager can be installed on the same machine as the Agile PLM Application Server but is, typically, installed on a separate machine. The File Manager can also be installed in a cluster and distributed across geographic regions.

2.1.3 AutoVue Components (Server and Client Components)

The installation involves AutoVue for Agile PLM installed on a server, to which client machines are connected to access and view documents. An AutoVue for Agile PLM solution has several components: the AutoVue server, an application server hosting the VueServlet, a Web server or an application server hosting AutoVue client libraries, and the AutoVue client. In the case of AutoVue for Agile PLM installation, the AutoVue for Agile PLM Servlet (VueServlet) is packaged and deployed as part of the File Manager installation. The VueServlet is also deployed on the File Manager and tunnels the requests to the Agile PLM Application Server using HTTP protocol.

The AutoVue client is a JAVA-based application that is launched through Java Web Start framework. It is fully customizable: you can modify the graphical user interface (GUI), setup a collaboration session, modify the menu options and toolbars, and so on. The AutoVue client is the main entry point to view AutoVue's capabilities. The AutoVue client libraries are delivered by Agile PLM (typically, the File Manager).

This chapter discusses the prerequisites of installing AutoVue.

3.1 Pre-requisites for Installing AutoVue for Agile PLM

Prior to installing AutoVue for Agile PLM, there are certain prerequisites that must be met. The following prerequisites are common to Windows and Linux OSes.

- It is recommended that you install the AutoVue server component with each File Manager.
- The machine that is hosting the AutoVue server 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.
- Ensure that the AutoVue server has permission to write to the operating system's temporary directory. To ensure temporary files are created in protected locations on Linux, the scripts `javueserver` and `javueserver_debug` have been changed to set a new temporary directory. In order to create a new temporary directory, set the environment variable `TMPDIR` (by default, `TMPDIR="/tmp/autovue"`), and the new temporary folder to the permissions 700 (owner can read, write and execute).

The AutoVue server administrator is responsible for setting this environment variable for security.

1. If setting `TMPDIR="/tmp"`, AutoVue will create temporary files in non-secure location `/tmp`, and log a security warning, such as "SECURITY WARNING: Incorrect or insecure temp folder (`/tmp`), using global temp directory".
 2. If a temporary folder is set to one owned by another user, AutoVue will prompt "AutoVue temporary folder `$TMPDIR` is owned by a different user", and exit.
 3. If a temporary folder on Linux is set to one without 700 permission, AutoVue will prompt "AutoVue temporary folder `$TMPDIR` has incorrect permissions", and exit.
 4. When setting temp folder `$TMPDIR` to a non-default location it must be ensured it is defined in the wine config file. The wine config file is at `<Av Install Dir>/config/jvuew_config`.
- If you are using a load balancer, ensure that the load balancer is configured to enable session stickiness (also referred to as session persistence, that is when requests come in from the client, the load balancer directs them to the same server that handled the initial request for the session). Session stickiness is normally achieved through the use of browser cookies.

- For performance reasons, it is recommended that the applications' streaming file cache directory is excluded from real-time virus scans. If virus scans are enabled, there is a performance impact on it and the impact is tied to the kind of operations done by the virus scan. It is also recommended that you run scheduled virus scans at a time when the server is not heavily in use.
- All server machines should obey the following conditions:
 - Have the same fonts installed.
 - If SSL is enabled, have identical certificates in their JVM repositories
 - If an AutoVue server farm is configured, then ensure that the `javueserver.rmi.host.*` is set to the same entries for all the servers in the farm. If a new server is added to the farm and the entries of `javueserver.properties` of the existing servers are not updated, this new server will not be allowed to connect to the farm. When this happens, a warning is logged in the server logs.
- It is recommended to install only one instance of the AutoVue server per physical machine.

The following sections describe Windows-specific and Linux-specific prerequisites.

3.1.1 Windows Prerequisites

- AutoVue for Agile PLM installs the AutoVue Document Converter print driver on Windows operating systems. Ensure that the print spooler service is enabled and that you have the permissions to install print drivers on the AutoVue server machine.

3.1.2 Linux Prerequisites

To correctly install AutoVue for Agile PLM on a Linux OS, it is recommended that you have basic knowledge of Linux and its administration. Note that you can install AutoVue for Agile PLM on Linux only in the GUI mode.

Note: To install packages on a Linux system you must have appropriate administrative privileges

- It is recommended to install AutoVue for Agile PLM using the same user account used to install Agile PLM.

If the drawings you are accessing from AutoVue (or any of the external references) reside on a mounted drive, ensure that the drive is mounted with the `noserverino` parameter.
- Install the TrueType core fonts package from <http://corefonts.sourceforge.net/>. Note that the AutoVue server must be restarted for the changes to take effect. These fonts are required for viewing files that use some specific fonts.

Verify that the fonts package is installed and configured properly by checking that the directory `/usr/share/fonts/msttcorefonts` exists and contains a collection of TrueType (TTF) files.

Note: The following step needs to be performed after the AutoVue server has been installed:

Create symbolic links from the TTF files in /usr/share/fonts/msttcorefonts to the <AutoVue Install Dir>/jvview_c/windows/fonts directory.

```
-cd <AutoVue Install Dir>/jvview_c/windows/fonts
-ln -s /usr/share/fonts/msttcorefonts/*.ttf
```

Note: Fonts are governed by certain licensing restrictions. Ensure that you verify the licensing for fonts before copying them over to different machines.

- With release 21.0.2, a new wine RPM package has been provided that is enhanced with dependency information. This can significantly simplify the AutoVue install process. If the "yum" utility is available on the installation machine, wine and all the required pre-requisite packages can be installed by running:

```
yum localinstall wine-av-20040914-25.i386.rpm
```

If yum is not available, the pre-requisite packages for Linux can be installed individually. For the most up-to-date list of libraries, refer to the Oracle Support Document 967083.1 (Prerequisite Library Names for AutoVue Client/Server Deployment on Linux) at:

<https://support.oracle.com/epmos/faces/DocumentDisplay?id=967083.1>

Note: The wine-av-20040914-25.i386.rpm is available at <https://oss.oracle.com/AutoVue>.

- If you have an older version of wine, you have to uninstall it and then install the package that is certified with your version of Oracle AutoVue.
 - To uninstall wine, find the package by running the following command:


```
rpm -qa | grep -i wine
```
 - Remove the package by running the following command:


```
rpm -e <package_name>
```

3.2 Additional Guidelines for Security

The following prerequisites are common to Windows and Linux OSes, and are essential for security purposes.

- Shutdown all applications (including the AutoVue server) before you run the installer for the AutoVue for Agile PLM installation.
- When upgrading, if the installer prompts you to reboot the machine before or after the un-installation, you must reboot the machine in order to get a successful installation. If you fail to reboot when prompted, the application installation may be left in an unusable state.
- In a multi-AutoVue server deployment, it is recommended to install AutoVue in the exact same folder path on each server.

3.3 Default Locations

The first two components – Agile PLM application server and File Manager, are installed using the Agile PLM installer, while the third component, AutoVue Server, is installed using the AutoVue for Agile PLM installer.

The default location of the components of AutoVue for Agile PLM is provided in the [Table 3–1, "Default Location of AutoVue for Agile PLM Components"](#).

Table 3–1 *Default Location of AutoVue for Agile PLM Components*

Component	Select this while installing	Default Location
Agile PLM application server	In the Choose Install Component(s) window, select Application Server check box.	AGILE_HOME/agileDomain
File Manager	In the Choose Install Component(s) window, select File Manager check box.	AGILE_HOME/FileManager
AutoVue Server	Select the AutoVue Server component.	AGILE_HOME/AVS

AutoVue Hardware and System Requirements

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

4.1 Hardware Requirements

Table 4–1 Hardware Requirements

Component	Oracle-Certified Hardware Requirements
Server	<p>Note: The AutoVue server is very CPU-, I/O-, memory-, and graphics-intensive. Ensure that the machine hosting AutoVue Server has sufficient resources for it and any other applications running on the machine.</p> <ul style="list-style-type: none">■ 8GB of RAM minimum, 12 GB or more recommended depending on complexity of documents being served■ Quad-core processor■ 400 MB of disk space for installation■ At least 30 GB of free disk space:<ul style="list-style-type: none">■ 20 GB for streaming files (if you configure a larger size for the AutoVue cache directory, ensure that the additional disk space is available).■ Additional space required for managing markup symbols, user profiles, and markups.■ AutoVue also stores temporary files. It stores files at %TEMP% path on Windows operating systems. On the Linux operating system, the temporary files are stored at the default location ie the /tmp/autovue folder. These files are generally deleted after processing is complete. Ensure that there is available disk space for AutoVue temporary files.

Table 4–1 (Cont.) Hardware Requirements

Component	Oracle-Certified Hardware Requirements
Client	<ul style="list-style-type: none"> Minimum requirements for your operating system (OS): <ul style="list-style-type: none"> 1 GHz CPU 1 GB of RAM <p>It is recommended that the Java Virtual Machine (JVM) used for the AutoVue client is configured for a maximum memory of 256 MB. If loading larger documents, you may need to increase this memory to a higher value (for example, 512 MB).</p> The AutoVue client works on most operating systems and browsers that support Java WebStart. To see what is certified by Oracle, refer to System Requirements. When running the AutoVue client on machines with non-Windows operating systems (OSes), ensure that these machines have a graphics card that supports OpenGL. This is necessary for loading 3D models. On Windows machines, it is recommended to have a graphics card with OpenGL support. In the absence of an adequate graphics card driver, Windows uses its OpenGL capability which is slower as compared to having a graphics card that supports OpenGL.

Important: It is recommended to allocate 2GB per document server if the documents are of low complexity. For 3D files, it is recommended to allocate 4 GB of memory per doc server.

4.2 System Requirements

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

4.2.1 Server Platforms

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

Table 4–2 Oracle-Certified Operating Systems

Platform	AutoVue for Agile PLM	AutoVue Client/Server Deployment	AutoVue Desktop Deployment
Windows Server 2016 —64-bit (AutoVue running in 32-bit mode)	x	x	
Windows Server 2012 R2 —64-bit (AutoVue running in 32-bit mode)	x	x	
Windows Server 2008 R2—64-bit (AutoVue running in 32-bit mode)	x	x	
Oracle Linux 6.X (x86_64) ¹ and 7.X (x86_64) ¹ 64-bit (AutoVue running in 32-bit mode)	x	x	

Table 4–2 (Cont.) Oracle-Certified Operating Systems

Platform	AutoVue for Agile PLM	AutoVue Client/Server Deployment	AutoVue Desktop Deployment
Red Hat Enterprise Linux 6.X (x86_64) ¹ , and 7.X (x86_64) ¹ 64-bit (AutoVue running in 32-bit mode)	x	x	

¹ AutoVue is supported on a 64-bit Linux operating systems (OS), but will run in 32-bit mode. All prerequisites libraries should be in 32 bit, with the exception of the X Windows base package (xorg-x11-server-Xorg) along with the Xvfb package xorg-x11-server-Xvfb. These should be the same architecture (64-bit) as the OS.

4.2.2 Client Platforms

The following platforms are certified for the AutoVue client.

Table 4–3 Oracle-Certified Client Platforms

Platform	AutoVue for Agile PLM	AutoVue Client/Server Deployment
Windows OSes: Windows 7— 64-bit Windows 8.1—64-bit Windows 10 — 64-bit <ul style="list-style-type: none"> ■ Internet Explorer 11 ■ Chrome 46 and up ■ Microsoft Edge ■ Firefox ESR 60.x and up Note: The AutoVue applet client is supported with IE 11 and Safari only.	x	x
Apple OS X 10.11 <ul style="list-style-type: none"> ■ Safari 10.0¹² ■ Safari 9.0¹² ■ Safari 8.0¹² ■ Firefox ESR 60.x and up 	x	x
Java Virtual Machine		
<ul style="list-style-type: none"> ■ Java SE 8 update 172 and higher - 32-bit and 64-bit 	x	x

¹ To know how to automatically launch JNLP files on Safari MAC after download, refer to <https://support.oracle.com/epmos/faces/DocumentDisplay?id=2257841.1>

² When launching AutoVue from a Mac client, certain configurations may be required. For more information, refer to Oracle Support Document 1662405.1 (Getting Message "Select a temporary directory" When Printing With AutoVue) which can be found at: <https://support.oracle.com/epmos/faces/DocumentDisplay?id=1662405.1>

4.2.3 Integrations

The following integrations have been certified for this release of AutoVue.

Table 4–4 Oracle-Certified Integrations

Integrations	AutoVue for Agile PLM	AutoVue Client/Server Deployment
Agile PLM 9.3.6 (RUP 8 and later), and 9.3.5 (RUP 8 and later)	x	

4.2.4 AutoVue for Agile PLM Integration Components

AutoVue for Agile PLM integration components can be installed on the following OSes:

- Oracle Solaris (SPARC and x64)
- Linux (Oracle, Red Hat Enterprise, SUSE)
- Microsoft Windows
- HP-UX Itanium
- IBM AIX (POWER)

For supported versions, refer to the "Agile Product Lifecycle Management Capacity Planning Guide" on OTN

<https://www.oracle.com/technetwork/documentation/agile-085940.html>

Installation Checklist

This chapter discusses the installation checklist that needs to be followed before installing AutoVue for Agile PLM solution.

AutoVue for Agile PLM can be deployed in a number of scenarios. An AutoVue installation consists of installing the an application server hosting VueServlet, JNLP components, and AutoVue client components.

5.1 Deployment Scenarios

The different components that have to be selected for deployment scenarios of AutoVue for Agile PLM are provided in the following table.

Table 5–1 *Deployment Scenarios*

Deployment	Configuration Steps during installation
Agile PLM application server and File Manager on one machine	Select the AutoVue server and AutoVue client libraries check box in the Choose Install Component(s) window.
Agile PLM application server and File Manager on different machines	Install AutoVue server and AutoVue client libraries separately. Client libraries should be installed on any machine that has a File Manager as well as on the standalone application server or the admin server if in a cluster. The AutoVue Server should be installed on all machines having a File Manager.

5.2 Installation Checklist

To install the AutoVue for Agile PLM solution:

Note: Stop all the Agile Application Servers and File Managers before installing the AutoVue Client Libraries.

- On a machine with only the Agile PLM application server installed in AGILE_HOME, run the AutoVue for Agile PLM installer and then select **AutoVue Client Libraries** check box from the Choose Install Component (s) window. The AutoVue Server is always installed with all File Managers only.

Note: If running the AutoVue for Agile PLM installer on a machine with the Agile PLM application server and/or File Manager installed then the "AutoVue client libraries" component will be displayed and is selected by default. The "AutoVue Server" component is always displayed and selected by default.

- On a machine with only the File Manager installed in AGILE_HOME or with both the application server and File Manager installed in AGILE_HOME, run the AutoVue for Agile PLM installer and then select **AutoVue Server and AutoVue Client Libraries** check box from the Choose Install Component (s) window.
- On a machine designated for only the AutoVue server (that is, there is no AGILE_HOME folder installed), run the AutoVue for Agile PLM installer and install only the AutoVue server.
- After installation, remember to redeploy your File Manager and run the MetaFilesRemover utility.

5.3 Additional Recommendations

Following are recommendations to ensure that the AutoVue for Agile PLM integration works correctly:

- Install the AutoVue client libraries on the application server machine.

Note: Only required on the Admin server in an application server cluster.

- Re-deploy the File Manager after installing the AutoVue client libraries.
- Re-deploy the application server after installing the AutoVue client libraries.
- Run the MetaFilesRemover tool after upgrading to a different AutoVue for Agile PLM version.

Installing and Upgrading AutoVue for Agile PLM

This chapter describes how to install AutoVue for Agile PLM on Windows and Linux OSes. The AutoVue for Agile PLM installer updates the Agile PLM Application Server and File Manager components for Agile PLM version 9.3.6 (RUP 8 and later), and 9.3.5 (RUP 8 and later) on all platforms. It also installs the AutoVue server component on Windows or Linux.

Note: When upgrading your installation of AutoVue for Agile PLM, do not cancel the installation process once it has begun. Canceling the installation may leave your current installation of AutoVue for Agile PLM unusable.

Note: Upgrade to the latest AutoVue for Agile PLM patch by installing the new Agile PLM RUPs and AutoVue RUPs.

6.1 Upgrading from earlier versions

You must run the AutoVue for Agile PLM installer on the application server and all File Managers as well as the machine where you wish to install the AutoVue server. Select the AutoVue Server option in the installer if you wish to install the AutoVue server, and select the AutoVue Client Libraries option if you are running the installer on an application server or File Manager. If it is required to install the AutoVue server on the same machine with the application server and File Manager, then both the AutoVue Server and the AutoVue Client Libraries options must be selected in the installer.

Note: Before updating the AutoVue client libraries, ensure that you stop the file server and any additional Agile processes.

The following sections discuss upgrading AutoVue from earlier versions.

6.1.1 Upgrading from AutoVue for Agile PLM Version 20.2.x/21.0.0/21.0.1

If you are upgrading from AutoVue for Agile PLM 20.2.x/21.0.0/21.0.1, you do not need to run the uninstaller before you install AutoVue 21.0.2 for Agile PLM. You can just run the installer for AutoVue 21.0.2 for Agile PLM. The installer detects if AutoVue for Agile PLM 20.2.x/21.0.0/21.0.1 is installed on your machine. If it is installed, the

installer backs up required data, uninstalls version 20.2.x/21.0.0/21.0.1 and then installs 21.0.2 to the same location. If you already have 20.1.x/20.2.x/21.0.1, with codebase (client JAR files) set in Jetty, the installer will upgrade to AutoVue 21.0.2 for Agile PLM.

- If installing on a Linux OS, make sure you have the correct version of WINE installed. The version of WINE compatible with AutoVue 21.0.2 for Agile PLM is wine-av-20040914-25.i386.rpm. You can download this version of WINE from <https://oss.oracle.com/AutoVue>.

Note: It is recommended to delete all cached and streaming files when updating AutoVue for Agile PLM.

Below is the list of data that is migrated if you are upgrading:

- Settings in jvueserver.properties are migrated to the new version.
- The installer upgrades the previous configuration settings from allusers.ini and jvueserver.properties when you install the new version.
- Any changes made to default.ini and allusers.ini are migrated to the new version.
- AutoVue for Agile PLM user profiles are left as is.
- The following are left as is since they are read by the new version of AutoVue:
 - Server-managed markups
 - Any stamps and stamp libraries
 - Intellistamps and definitions
- On Linux, changes made to <AutoVue for Agile PLM Install Root>/config/jvuew_config are left as is since this will be read by the newer version of AutoVue.
- Custom log settings: If you had custom log settings, these are migrated to the newer version of AutoVue for Agile PLM.
- Changes made to markup policy file – markuppolicy.xml are migrated to the newer version of AutoVue for Agile PLM.
- Any changes made to format-specific files such as color maps, font maps, fonts are migrated to the newer version.

The following are not migrated and must be migrated manually:

- GUI files: If you created custom GUI files, you must make sure to migrate the GUI settings. In order to migrate GUI, it is recommended that you run a diff utility between your current version's default.gui and your custom GUI. Identify what GUI components have been updated. Manually apply these settings to 21.0.2 GUI file.

6.1.2 Upgrading from AutoVue for Agile PLM Version 20.1.x or Earlier

If you are upgrading AutoVue for Agile PLM from version 20.1.x or earlier, you must manually move your configuration settings from your version to AutoVue 21.0.2 for Agile PLM. You must first uninstall any service packs that are installed for that version of AutoVue for Agile PLM and then uninstall your previous version before installing the new version of AutoVue for Agile PLM. Before you uninstall, you must backup all

required data. After you install AutoVue 21.0.2 for Agile PLM, you must migrate your past data to 21.0.2. Below is what you need to backup and migrate manually:

- Custom settings in `javueserver.properties`
- Custom settings in `VueServer.ini`. As of version 20.1, settings in `VueServer.ini` have been migrated to `javueserver.properties`. Refer to the *Oracle AutoVue 20.1 Release Notes* for a mapping of `VueServer.ini` options to `javueserver.properties` parameters.
- Custom settings in `default.ini` and `allusers.ini` should be backed up and migrated to the new version.
- User-specific INI files should be backed up and copied over to the Profiles folder of the new installation.
- GUI files: If you created custom GUI files, you must make sure to migrate the GUI settings. In order to migrate GUI, it is recommended that you run a diff utility between your current version's `default.gui` and your custom GUI. Identify what GUI components have been updated. Manually apply these settings to 21.0.2 GUI file.
- Custom log settings: If you had custom log settings, apply them manually on the new version of AutoVue.
- Stamp attributes and settings from the `dmstamps.ini` file (located in the <AutoVue for Agile PLM Install Root>\bin directory) should be copied over manually to the new version.
- Markup files, if markups are being managed by the AutoVue server (located in the <AutoVue for Agile PLM Install Root>\bin\Markups directory), should be backed up and copied over to the new version of AutoVue.
- Custom markup symbol libraries (located in the <AutoVue for Agile PLM Install Root>\bin\Symbols directory) should be backed up and copied over to the new version of AutoVue.
- If `MarkupPolicy.xml` located in <AutoVue for Agile PLM Install Root>\bin was modified, it should be backed up and changes to the policy should be manually applied to the new version.
- On Linux installations of AutoVue for Agile PLM, backup the `javuew_config` file (located in the <AutoVue for Agile PLM Install Root>/config directory) if it was modified and apply the changes manually to the new version.
- Any changes made to format-specific files such as color maps, font maps, fonts must be backed up and these changes should manually be applied to the new version.

6.2 Installing on Windows platform

To install AutoVue for Agile PLM on Windows Platform:

1. Download Oracle AutoVue 21.0.2 for Agile PLM Media Pack listed under the product Oracle Agile Applications product pack from Oracle Software Delivery Cloud or edelivery (<https://edelivery.oracle.com>) to a temporary directory and extract its contents.
2. For Windows run the `setup_av_win.exe` from the temporary directory.
3. Proceed to [Section 6.2.1, "Running the Installation Program"](#).

6.2.1 Running the Installation Program

The installation program prompts you to enter specific information about the system and configuration. See the [Table 6–1, "Installation Instructions"](#) for what you have to do during the installation.

Table 6–1 Installation Instructions

In this window...	Perform the following action...
Welcome to AutoVue for Agile PLM Installation	Click Next to proceed with the installation.
Customer Information	Enter the User Name and Company Name . Click Next .
Choose Install Component (s)	<p>If running the AutoVue for Agile PLM installer on a machine with the Agile PLM application server and/or File Manager installed then the "AutoVue client libraries" component will be displayed and is selected by default. The "AutoVue Server" component is always displayed and selected by default.</p> <p>Click Install.</p> <p>Note: Refer to Deployment Scenarios to determine what components are required for your installation.</p>
Installation Location	<p>Enter the location or folder where you want to install the selected AutoVue components in the Specify where to install the selected AutoVue component(s) box. Click Next.</p> <p>If a supported AGILE_HOME exists then the default Installation Location is the directory AGILE_HOME/AVS and this would be the AutoVue Server home directory.</p>
AutoVue Server	<p>Enter the host name and the port number of the AutoVue server that Agile File Manager will connect to. Click Next.</p> <p>Note: Ensure that you enter fully qualified machine/domain name for the Host Name. Example: host.domain.com (machinename.domainname).</p>
AutoVue Product	<p>Choose the AutoVue product that you have licensed. The options are:</p> <ul style="list-style-type: none"> ■ AutoVue 2D Professional for Agile ■ AutoVue Electro-Mechanical Professional for Agile <p>Click Next.</p>
Choose Shortcut Folder	<p>Here you can choose where you want to create the product icons. Following are the options:</p> <ul style="list-style-type: none"> ■ In a new Program Group ■ In an existing Program Group ■ In the Start Menu ■ On the Desktop ■ In the Quick Launch Bar ■ Other ■ Don't create icons <p>Note: In case you select the Create Icons for All Users, then the product icon appears as a short cut for all users.</p> <p>Click Next.</p>

Table 6–1 (Cont.) Installation Instructions

In this window...	Perform the following action...
Pre-Installation Summary	Review the Product Name, Install Folder, Shortcut Folder, Product Features, Disk Space Information (for Installation) and additional information before continuing with the installation. After the review, click Install .
Install Complete	Click Finish to complete the installation.

Note: Before installing the AutoVue client libraries, remember to stop the file server and any additional Agile processes.

After installing on the Application Server: After installing the AutoVue client libraries, redeploy the application.ear file and restart the application server.

For instructions on restarting the application server, refer to either the *Installing Agile PLM on Oracle Application Server* or the *Installing Agile PLM on Oracle WebLogic Server* manuals.

After installing on the File Manager: After installing the AutoVue client libraries, redeploy the webfs.war file (Filemgr web application) on the File Manager.

Note: The webfs.war file and application.ear files are available at "AGILE_HOME\agiledomain\applications".

6.3 Installing on Linux platform

To install AutoVue for Agile PLM on Linux platforms:

1. Uninstall any previous versions of AutoVue.
2. Run the Red Hat Update Agent, yum up2date, to download the latest Xvfb and Mesa files.

Note: Both the AutoVue installer and the AutoVue for Agile PLM installer do not detect the installation of Mesa or Xvfb.

3. It is recommended that AutoVue is installed on the same user account used to install Agile PLM to avoid any library conflicts between AutoVue and Agile or Oracle database.
4. Install the pre-requisites as described in [Linux Prerequisites](#).
5. Download Oracle AutoVue for Agile PLM Media Pack listed under the product Oracle Agile Applications product pack from Oracle Software Delivery Cloud or edelivery (<https://edelivery.oracle.com>) and extract its contents.
6. Launch the AutoVue for Agile PLM installer (Linux: setup_av_lin.bin).

Note: It is recommended to run the AutoVue for Agile PLM installer UI on the server itself either through the primary display or, if installing remotely, by using vncserver.

7. Follow the instructions to proceed with the installation. Select the components you would like to install.
8. For information about any screen in the installer, click **Help**.
9. Click **Next** to proceed through the installer.

Note: If you are installing on an Application Server or File Manager, select the **AutoVue Client Libraries** check box on the **Choose Install Components** panel.

10. Click **Done** to finish the installation.

6.4 Installing on other platforms

The AutoVue server can be installed on only Windows and Linux. However, Agile PLM application server and File Manager components can be installed on several other platforms including Solaris (SPARC), Solaris (x64), AIX (POWER) and HP-UX (Itanium). You can install AutoVue Client Libraries on any of these platforms.

6.4.1 Installing AutoVue Client Libraries and Integration Components on Different Platforms

Oracle AutoVue Client Libraries and integration components can be installed on the following platforms:

Note: VueLink/VueServlet are delivered as part of the File Manager web application, and do not have to be installed separately. See Note in [Testing the VueServlet](#) for details.

Table 6–2 *Platforms and the corresponding executable file*

Platform	Executable File
Solaris (SPARC)	setup_av_sol.bin
Solaris (x64)	setup_av_solx86.bin
AIX (POWER)	setup_av_aix.bin
HP-UX (Itanium)	setup_av_hpux.bin

1. Download Oracle AutoVue 21.0.2 for Agile PLM Media Pack listed under the product Oracle Agile Applications product pack from Oracle Software Delivery Cloud or edelivery (<https://edelivery.oracle.com>).
2. Extract the media pack and run the executable file.
3. Follow the instructions to proceed with the installation.
4. For information about any screen in the installer, click **Help**.
5. Click **Next** to proceed through the installer.

Note: If you are installing on an Application Server or File Manager, select the **AutoVue Client Libraries** check box on the **Choose Install Components** panel.

6. Click **Done** to finish the installation.

6.5 Installing AutoVue for Agile in Console Mode

To install AutoVue for Agile PLM in console mode:

1. Download Oracle AutoVue 21.0.2 for Agile PLM Media Pack listed under the product Oracle Agile Applications product pack from Oracle Software Delivery Cloud or edelivery (<https://edelivery.oracle.com>) to a temporary directory and extract its contents.
2. Run the `./setup_av_<platform>.bin -i console`.
3. Proceed to [Section 6.5.1, "Running the Installation Program in Console Mode"](#).

6.5.1 Running the Installation Program in Console Mode

Table 6–3 *Installation Instructions*

In this window...	Perform the following action...
Welcome to AutoVue for Agile PLM Installation	Press <Enter> to proceed with the installation.
Customer Information	Enter the User Name and Company Name . Press <Enter>.
Choose Install Component (s)	<p>Enter a comma_separated list of numbers representing the features you would like to select, or deselect. To view a feature's Description, enter <NUMBER>.</p> <p>1- AutoVue Server</p> <p>2- AutoVue Client Libraries</p> <p>Press <Enter> after you have entered the number.</p> <p>Note: Refer to Section 5.1, "Deployment Scenarios" to determine what components are required for your installation.</p>
Installation Location	<p>Enter the location or folder where you want to install the selected AutoVue components in the Specify where to install the selected AutoVue for Agile PLM component(s).</p> <p>Enter an absolute path, or press <Enter> to accept the default.</p>
AutoVue Server	<p>Enter the Host Name and Port Number of the AutoVue Server that Agile File Manager will connect to. Press <Enter>.</p> <p>Note: Ensure that you enter fully qualified machine/domain name for the Host Name. Example: host.domain.com (machinename.domainname).</p>
AutoVue Product	<p>Choose the AutoVue product that you have licensed.</p> <p>1- AutoVue 2D Professional for Agile</p> <p>2- AutoVue Electro-Mechanical Professional for Agile</p> <p>Enter the number of the desired choice.</p> <p>After entering your choice, press <Enter>.</p>

Table 6–3 (Cont.) Installation Instructions

In this window...	Perform the following action...
Install Complete	After the installation is complete, you get the following message: All the selected components for AutoVue for Agile PLM have been successfully installed to the selected folder.

6.6 Configuring the File Manager

For information on how to configure and re-deploy the File Manager, refer to the "Configuring the File Manager" chapter of the *Agile Product Lifecycle Management Application Installation Guide* located on the Agile OTN site

<https://www.oracle.com/technetwork/documentation/agile-085940.html>.

6.7 Upgrading from Earlier Version with MetaFilesRemover Utility

Note: If you are upgrading from an earlier version it is required to run the MetaFilesRemover utility on all File Manager machines.

If you are upgrading from an earlier version:

1. Remove the CMF files of the previous version on the file vault.
2. To remove the CMF files, unzip the MetaFilesRemover.zip file and run the MetaFilesRemover utility located at agile_home\agileDomain\tools.

Usage: java -jar MetaFilesRemover.jar [-delete] [-age <value>] [-size <value>] [-basedir <value>] [-prefix <value>] [-serverURL <value>] [-username <value>] [-password <value>] [-dburl <value>] [-dbuserid <value>] [-dbpassword <value>]

where

- **delete** deletes the metafiles.
- **age** specifies the access time (day).
- **size** specifies file size (KB).
- **basedir** is the file vault location where the metafiles are removed.
- **prefix** is the file name prefix.
- **serverURL** is the location of the DMS service.

For example, Server URL should be -

<http://<ApplicationServerHostName>:7001/Agile/DmsService/DmsViewerAPIService>.

- **username** is the DMS service username.
- **password** is the DMS service password.
- **dburl** is the database location (ex: hostname.us.oracle.com:1521:agile9).
- **dbuserid** is the database user id.
- **dbpassword** is the database password for this user.

6.8 Verifying your Integration

After you have installed the AutoVue client libraries and File Manager, you must verify that your integration works correctly with this version of AutoVue for Agile PLM. The following details the verification steps:

- Start the AutoVue server, application server, and File Manager.
- For each File Manager, verify the status of the VueServlet by accessing the File Manager configuration page:
`http://fmhost:port/Filemgr/Configuration`

6.9 Post-Installation Instructions

If you choose option "Configure SSL with a CA certificate" or "Configure SSL with a generated self-signed certificate", the following is done by the AutoVue Installer:

- The certificate is imported to `<AutoVue Install Root> \jre\lib\security\cacerts`.
AutoVue expects a trusted certificate in the .cer format.
- The installer modifies the `javue.bat` sample standalone application and configures it to connect via SSL (HTTPS protocol and port 8443).
- The following options are added to `javueserver.properties` file:
 - `javueserver.ssl.enable=true`
 - The parameter, `javueserver.cmdline` has the following options added to it:
 - * `Djavax.net.ssl.keyStore=<full path to keystore>`
 - * `Djavax.net.ssl.keyStorePassword=<keystore password>`
- If you plan to use the War sample provided with AutoVue, ensure that `EnableSSL` is set to `TRUE` in `<AutoVue Install Root> examples\WarSample\AutoVue\WEB-INF\web.xml`

6.9.1 Configuring SSL

If you did not configure SSL during the AutoVue installation, you can do so post-installation. For more information, refer to the "Enabling SSL Communication" section of the *Oracle AutoVue Client/Server Deployment Security Guide*.

6.9.2 Configuring User Authentication

If you did not configure an authentication mechanism during the AutoVue installation, you may do so post-installation.

Note: User authentication should be configured only if required. If the DMS handles the authentication, then you do not have to configure the user authentication.

The administrator can disable the authentication if not needed.

To configure the server to use the JAAS authentication plug-in supplied with AutoVue, perform the following:

1. Edit `javueserver.properties` to specify the plug-in by removing the comment in the following line:

2. Create a text file called `jaas_authen.conf` in the `<AutoVue install root>\bin` directory. Add the following text in the file:

```
/**
** Example JAAS Login Configuration for the AutoVue server
**/
AVServer
{
com.sun.security.auth.module.Krb5LoginModule required storeKey=true;
};
```

3. Edit `javueserver.properties` and add the following highlighted lines after the `-Djava.security.policy` parameter of `javueserver.cmdline`:

```
javueserver.cmdline=-Xmx128M -
Djava.security.policy="C:\Oracle\AutoVue\bin\policy"
-Djava.security.krb5.realm=<realm> -Djava.security.krb5.kdc=<kdc>
-Djava.security.auth.login.config=<full path to jaas_authen.conf>
```

Replace `<realm>` with your security realm.

Replace `<kdc>` with your key distribution center.

4. Start up the AutoVue server.
5. Launch the AutoVue client.

An authentication dialog appears and prompts for login information. On logging in successfully, the AutoVue client launches.

For more information on user authentication, refer to the “User Authentication” section of the *Oracle AutoVue, Client/Server Deployment Security Guide*.

6.9.3 Deploying VueServlet

The VueServlet is the main entry point for communications between the AutoVue clients and the AutoVue server. The client makes requests using the HTTP/HTTPS protocol to the VueServlet and the VueServlet communicates with AutoVue using AutoVue's socket port. The instructions for deploying VueServlet vary based on whether or not you are integrating AutoVue with a DMS. This section discusses installing a single-instance of VueServlet in a non-integrated environment. In an integrated environment, the same instructions apply, except with the difference that the VueServlet may be deployed in a different context. .

The first step to deploy the VueServlet is to create a WAR file for the VueServlet. Once the WAR file has been successfully created, you can deploy the WAR file with your J2EE-enabled application server.

6.9.3.1 Creating a WAR for the VueServlet

To deploy the VueServlet with your J2EE-enabled application server, you must first create a WAR file. The following steps explain how to do this:

1. Create a directory.
For Example: `C:\avwar`
2. In the folder `C:\avwar`, create a sub-directory `WEB-INF`.
3. In `WEB-INF`, create a directory `lib`: `C:\avwar\WEB-INF\lib`
4. Copy `vueservlet.jar` from `<AutoVue Install Root>\bin` to `C:\avwar\WEB-INF\lib`.

5. Create a deployment descriptor file named *web.xml* in the WEB-INF directory.

- The following is the mandatory header for the *web.xml* document. It defines the document as an XML file and relates the file syntax to the DOCTYPE resource specified.

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="3.1"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd">
```

- Use the following code to specify the deployment descriptor needed to deploy the VueServlet.

```
<web-app>
  <servlet>
    <servlet-name>com.cimmetry.servlet.VueServlet</servlet-name>
    <servlet-class>com.cimmetry.servlet.VueServlet</servlet-class>
    <init-param>
      <param-name>JVueServer</param-name>
      <param-value>hostname:socketport</param-value>
    </init-param>
    <init-param>
      <param-name>EnableSSL</param-name>
      <param-value>TRUE</param-value>
    </init-param>
  </servlet>
  <servlet-mapping>
    <servlet-name>com.cimmetry.servlet.VueServlet</servlet-name>
    <url-pattern>/servlet/VueServlet</url-pattern>
  </servlet-mapping>
</web-app>
```

The *<servlet-name>* parameter is how the servlet is known within the XML file.

The *<servlet-class>* parameter is the fully qualified Java programming language class name of the Servlet.

The *<url-pattern>* parameter is how the servlet is referenced from a Universal Resource Indicator (URI).

Note: The parameter structure must follow the order in the Document Type Definition (DTD). For example, all *<servlet>* tags must be defined before any *<servlet-mapping>*s can be specified.

6. Update hostname in *web.xml* with the name of the AutoVue server machine.
7. Update socketport in *web.xml* with the socket port for the AutoVue server.
8. To enable secure communication between VueServlet and AutoVue, set *EnableSSL* parameter to *TRUE*.
9. Such WAR file folder example is shipped with AutoVue under <AutoVue Install Root>/examples/WarSample.
10. To create the WAR file, use the jar utility from the Java Development Kit distribution. If you are in the root directory you created for the WAR contents (C:\avwar), use the following command:

```
jar cvf VueServlet.war WEB-INF
```

6.9.3.2 Deploying the VueServlet

Deploy VueServlet.war into your J2EE compliant application server.

Note: If the VueServlet is to be configured for internal and external access, then you must deploy it to file path that is externally accessible. You should verify deployment by accessing the VueServlet from an external user machine.

6.9.3.3 Verifying VueServlet Deployment

After the VueServlet has been deployed into your application server, test the VueServlet by accessing the URL to the VueServlet from a user machine that will be accessing AutoVue. If you are configuring for both internal and external access, you must deploy the VueServlet on a path that is externally accessible and verify the connection from an external user machine. To verify the VueServlet deployment, enter the following in your Web browser:

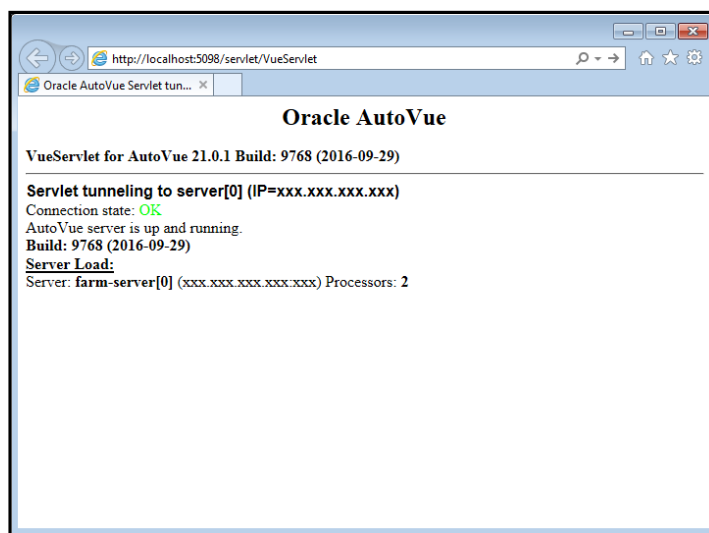
`http://hostname:socketport/<context>/servlet/VueServlet`

where <context> is the context you specified when deploying the VueServlet into your application server.

Note: Some application servers allow you to specify the context name, but generally the WAR file name is used as the context.

On successful deployment, the VueServlet should display a Web page as follows:

Figure 6–1 After successful deployment



6.9.3.4 Troubleshooting VueServlet Deployment

If you are not successful with the VueServlet verification, below are some pointers to help you troubleshoot your VueServlet deployment:

- If you get a page not found error, ensure that the application server hosting the VueServlet is up and running. Check for correct syntax and verify that you are accessing the correct port for the Application server.
- If you are able to access the VueServlet page and it indicates an error connecting to the AutoVue server:
 - Verify that the AutoVue server is running.
 - Verify that you specified the correct connection parameters to the AutoVue server in the VueServlet's JVUEServer parameter.
 - Verify that you can ping the AutoVue server machine from the VueServlet machine and vice-versa.

6.9.4 Deploying JNLP Components

AutoVue Client is a Java application that can be started through Java Web Start framework. This framework requires a Java Network Launching Protocol (JNLP) file to start the application. The JNLP protocol, defined with an XML schema, specifies how to launch Java Web Start applications. JNLP consists of a set of rules defining how exactly to implement the launching mechanism. JNLP files include information such as the location of the JAR package file and the name of the main class for the application, in addition to any other parameters for the program. An integration solution requires that the server generate a JNLP file to be used by Java Web Start framework to launch AutoVue client.

Note: If you have a proxy server running on the client machine, you have to configure your network correctly for a proper communication between the AutoVue JNLP client and the AutoVue server. It is important that you remove the local host from the proxy list.

A sample of the required JNLP file is provided in [Example 6-1](#).

Example 6-1 Required JNLP File

```
<?xml version="1.0" encoding="utf-8"?>
  <jnlp codebase="[URL of AutoVue Client code]" spec="1.0+">
    <information>
      <title>AutoVue</title>
      <vendor>Oracle</vendor>
      <homepage href="http%3A%2F%2Fwww.oracle.com"/>
      <description>A Leading CAD Visualization
Tool</description>
      <offline-allowed/>
    </information>
    <security>
      <all-permissions/>
    </security>
    <resources>
      <j2se version="1.5+"/>
      <jar href="jvue.jar" main="true"/>
      <jar href="jogl.jar"/>
      <jar href="gluegen-rt.jar"/>
      <jar href="jsonrpc4j.jar"/>
      <property name="jnlp.secure.argument.*" value="true"/>
    </resources>
  </application-desc main-class="com.cimmetry.jvue.JVueApp">
```

```
        <!-- AutoVue Client Parameters -->
        [...]
    </application-desc>
</jnl>
```

The servlet `VueJNLPServlet` provided with AutoVue (included by the JAR `<AutoVue Install Root>\examples\jnl\avjnlpservlet.jar`) is designed to generate the required JNLP file.

The implementation of `VueJNLPServlet` requires a JNLP template file – `autovue.jnl` that it loads at the runtime, as a resource file. It updates it with runtime parameters and generates dynamically a JNLP file to launch AutoVue client through Java Web Start, with the appropriate runtime parameters. So, if you use `VueJNLPServlet`, you must deploy its resource file `autovue.jnl`.

Integration solutions need usually to send commands to AutoVue from an HTML client. AutoVue supports a limited set of scripting APIs allowing that and provide two methods to the HTML client to invoke them:

- **Loopback connection:** The HTML client could provide a free localhost port to AutoVue in order to communicate directly with AutoVue client through this port. AutoVue Client launches an embedded server and start listening to the given port for scripting commands coming from the HTML client.
- **Rendezvous Communication:** In this case, the communication is done through the server. We need to deploy a "Rendezvous" servlet called `VueRDVServlet` provided within AutoVue components. Both the HTML client and AutoVue client exchange messages through a shared mailbox managed by this dedicated servlet, in order to interact together. In this case the integration solution needs to send session authentication cookies to AutoVue client in order to be authenticated properly when it communicates with `VueRDVServlet`.

Note:

- The loopback connection was released in 21.0.1.
 - The Rendezvous communication was released in 21.0.2 to replace the loopback connection.
 - The loopback connection was deprecated in 21.0.2 and may be removed in the future.
-
-

Some integration solutions require sending cookies to AutoVue client. For example, Session authentication cookies are required by AutoVue in order to be authenticated by the Rendezvous servlet if the integration solution chooses to use the Rendezvous communication protocol described above. In this case, the browser specifies to the server the list of cookies to forward to AutoVue. The server collects the required cookies and includes them in the JNLP file as parameters of AutoVue. The servlet `VueJNLPServlet` provided with AutoVue collects the required cookies and write them on the generated JNLP file.

On the other side, the cookies may hold sensitive information, hence writing them on the JNLP file "in-clear" may raise security concerns (refer to the Oracle AutoVue Client/Server Security Guide for more information and guidelines related to a secure installation of AutoVue). AutoVue client supports the cookies encrypted using RSA encryption and "RSA/ECB/PKCS1Padding" ciphers, as far as it receives the private encryption key required to decrypt them. However, this encryption is supported under loopback connection only. So, do not encrypt cookies if you use Rendezvous

communication protocol. An integration solution that needs to send encrypted cookies must generate an RSA encryption key-pair for this purpose. The servlet `VueKeyPairServlet` provided with `AutoVue` (included by the JAR `<AutoVue Install Root>\examples\jnlpservlet.jar`) is designed to generate the required key-pair. `VueKeyPairServlet` encodes it into Base64 format and serialize it into a HEX format. `AutoVue` client and `VueJNLPServlet` have the code to de-serialize and decode the key-pair.

The source code for `VueJNLPServlet` and `VueKeyPairServlet` is provided with `AutoVue` as well.

6.9.4.1 Deploying VueJNLPServlet

`VueJNLPServlet` is provided as a reference implementation for validation and testing purposes. To use it, you need to do the following:

1. Deploy `VueJNLPServlet` on a J2EE-enabled application server within a JAR or a WAR file, following the approach described for `VueServlet` in *Deploying the VueServlet*.
2. Add the code provided in Example 6–2 to the configuration file `web.xml` of the J2EE-enabled application server used to deploy `VueJNLPServlet` in order to specify the deployment descriptor needed for it.

Example 6–2 Code to be included in web.xml

```
<servlet id="VueJNLPServlet">
  <servlet-name>VueJNLPServlet</servlet-name>
  <servlet-class>com.cimmetry.servlet.VueJNLPServlet</servlet-class>
  <init-param>
    <param-name>URL-Dir</param-name>
    <param-value>[URI of the folder containing autovue.jnlpservlet.jar]</param-value>
  </init-param>
  <init-param>
    <param-name>Cookies</param-name>
    <param-value>[Cookie names]</param-value>
  </init-param>
</servlet>
<servlet-mapping>
  <servlet-name>VueJNLPServlet</servlet-name>
  <url-pattern>/servlet/VueJNLPServlet</url-pattern>
  <url-pattern>/servlet/VueJNLPServlet/*</url-pattern>
</servlet-mapping>
```

3. Update `VueJNLPServlet` descriptor parameters with the appropriate values as provided in Table 6–4.

Table 6–4 Descriptor Parameters of VueJNLPServlet

Parameter	Description
URL-Dir	Specify the relative path on your J2EE-enabled application server, of the folder containing the resource file <code>autovue.jnlpservlet.jar</code> with respect to root context (Make sure your root context is used consistently across your entire deployment). For example: <code>/AutoVueClient</code>

Table 6–4 (Cont.) Descriptor Parameters of VueJNLPServlet

Parameter	Description
Cookies	<p>Semicolon ';' separated list of cookie names identifying the cookies to transfer to AutoVue at the start-up.</p> <p>For example: JSessionID;...</p> <p>NOTE:</p> <p>This parameter supports also two special values not on the format described above:</p> <ul style="list-style-type: none"> ■ true: Pass all the browser cookies of the domain to AutoVue (this is to be used carefully. It must be combined with filtering in the implementation of VueJNLPServlet and encryption to ensure a secure deployment). ■ false: Do not pass any cookie to AutoVue. <p>It is essential that you keep the list of cookies short for security reasons.</p>

6.9.4.2 Deploying VueKeyPairServlet

VueKeyPairServlet is provided as a reference implementation for validation and testing purposes. It is a demonstration sample and integrators are strongly advised to review their deployment from a security point of view and chose the best way to generate these encryption keys. To use it, you need to do the following:

1. Deploy VueKeyPairServlet on a J2EE-enabled application server within a JAR or a WAR file, following the approach described for VueServlet in Deploying the VueServlet.
2. Add the code provided in Code to be included in web.xml to the configuration file web.xml of the J2EE-enabled application server used to deploy VueKeyPairServlet in order to specify the deployment descriptor needed for it.

Code to be included in web.xml

```
<servlet id="VueKeyPairServlet">
  <servlet-name>VueKeyPairServlet</servlet-name>
  <servlet-class>com.cimmetry.servlet.VueKeyPairServlet</servlet-class>
  <load-on-startup>0</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>VueKeyPairServlet</servlet-name>
  <url-pattern>/servlet/VueKeyPairServlet</url-pattern>
  <url-pattern>/servlet/VueKeyPairServlet/*</url-pattern>
</servlet-mapping>
```

6.9.4.3 Deploying VueRDVServlet

VueRDVServlet is required by the Rendezvous communication described above, between Autovue and an HTML client. To use this servlet, you need to do the following:

1. Deploy VueRDVServlet on a J2EE-enabled application server within a JAR or a WAR file, following the approach described for VueServlet in "Deploying the VueServlet" section.
2. Add the code provided in Code to be included in web.xml to the configuration file web.xml of the J2EE-enabled application server used to deploy VueRDVServlet in order to specify the deployment descriptor needed for it.

Code to be included in web.xml is provided in the following example:

Example 6-3 Code to be included in web.xml

```

<servletid="VueRDVServlet">
  <servlet-name>VueRDVServlet</servlet-name>
  <servlet-class>com.cimmetry.servlet.VueRDVServlet</servlet-class>
  <load-on-startup>0</load-on-startup>
</servlet>
<servlet-mapping>
  <servlet-name>VueRDVServlet</servlet-name>
  <url-pattern>/servlet/VueRDVServlet</url-pattern>
  <url-pattern>/servlet/VueRDVServlet/*</url-pattern>
</servlet-mapping>

```

6.9.5 Installing AutoVueClient Components

The AutoVue client is the main entry point to AutoVue's capabilities. The AutoVue client components need to be made accessible to end-users at an application server or Web server location. The instructions for deploying the client components vary depending on whether you have AutoVue integrated with a DMS or if you are using a non-integrated environment.

This section discusses deploying AutoVue client components in a non-integrated environment. In an integrated environment, the same instructions apply, except with the difference that the client components may be deployed in a different location.

1. Create a folder (for example, named AutoVueClient) on your Web server docroot.
2. Copy all mandatory JAR files from the <AutoVue Install Root>\examples\WarSample\AutoVueClient directory to the directory you created on your Web server docroot. The files to copy are jvue.jar, jogl.jar, gluegen-rt.jar, and jsonrpc4j.jar.
3. AutoVue client supports a scripting API and is invoked through XML HTTP requests sent by a custom HTML client. These requests are wrapped into a JavaScript Object named AutoVue and implemented in the file autovue.js. This object is designed to simplify the integration of AutoVue and provides a JavaScript method for each scripting API supported by AutoVue. If you want to send XML HTTP requests using your own approach, then look at the methods `initScriptService` and `processScriptMethod` of the file `autovue.js` to see how to build such requests. However, an integration solution could use this AutoVue JavaScript object to communicate with AutoVue client in a simpler way. To do so, you must include the JavaScript source in your HTML code integrating AutoVue:

```
<script type="text/javascript" src="AutoVueClient/autovue.js"></script>
```

Then, you must instantiate an AutoVue Object into a JavaScript block within your HTML code as shown in the following code:

JavaScript block to instantiate an AutoVue Object

```

<script language="JavaScript">
  var myAvApp = new AutoVue(JNLP_HOST, CODEBASE_HOST, CLIENT_PORTS,
                           INIT_PARAMS, ...)
</script>

```

Note: For a complete description of AutoVue JavaScript Object Constructor and its parameters, refer to the "JavaScript API" section of the API Guide.

Options like startup delay can be modified in Java Client: Admin > Server Settings > Preferences > General Information tab > AutoVue Client Startup Delay (in seconds).

Additional information about AutoVue Client Startup Delay is present in the Agile Product Lifecycle Management Administrator Guide.

To use the sample HTML pages provided with AutoVue (typically, only used for testing purposes, or a baseline example to create your own custom AutoVue client) copy them with the file `autovue.js` from `<AutoVue Install Root>\examples\WarSample` to the directory you created to your Web server root.

4. Edit the files that launch the AutoVue client (for example, `av_jnlp.html`) and update AutoVue Object Constructor parameters with appropriate values.

Table 6–5 Parameters

Parameter	Description
CODEBASE	Specify the URL to the AutoVue client files on your Web/application server (the folder created above). For example: http://AutoVueClient:5098/autovue
JVUESERVER	Specify the servlet connection to the AutoVue server. Separate multiple values with a semi-colon. For example: http://AutoVueServer:5098/servlet/VueRDVServlet
JNLP_HOST	Specifies the URL on your Web/application server, to a host returning the JNLP File required by Java Web Start to run AutoVue client. For example, if you deploy the servlet <code>VueJNLPServlet</code> provided with AutoVue: http://AutoVueServer:5098/servlet/VueJNLPServlet
RDVSERVLET_HOST	Specify the servlet connection to the Rendezvous servlet (<code>VueRDVServlet</code> host). For example: http://AutoVueServer:5098/servlet/VueRDVServlet
CL_PRTS	Specify a list of <i>localhost</i> ports to be used by the browser to communicate with AutoVue client. The expected format is a vector of port values or port intervals. For example: <code>[2345, [7500, 7510], [8500, 8510], 8888]</code> The parameter value above means that the browser and AutoVue client should try to communicate through the ports 2345, then 7500, 7501, ..., 7510, 8500, 8501, ..., 8510, 8888, and use the first one available.

Table 6–5 (Cont.) Parameters

Parameter	Description
INIT_PARAMS	<p>Specify the client parameters to pass at the start-up of AutoVue client. The expected format is a <i>JSON</i> format of an object where the object param/value fields are the names/values pairs of AutoVue client parameters. The only parameter that is required by the client to start is <i>JVUESERVER</i>, which specifies servlet connection to AutoVue server (separate multiple values with a semi-colon).</p> <p>Example:</p> <pre>{ 'JVUESERVER':JVUESERVER, 'RDVSRVLET':RDVSRVLET_HOST, 'VERBOSE': 'Debug' }</pre> <p>NOTE: If you use the servlet <i>VueJNLPServlet</i> distributed with AutoVue, then these parameters could also be provided as initialization parameters within the descriptor of <i>VueJNLPServlet</i> as described in the previous section about JNLP servlets deployment. Furthermore, setting them in the descriptor of <i>VueJNLPServlet</i> takes precedence on setting them here.</p>
ENCRYPT_COOKIES	<p>Whether to encrypt the cookies shared with AutoVue by the JNLP servlet (See section 5.4.4 – <i>Deploying JNLP Components</i>). Never encrypt the cookies under the RendezVous communication protocol though otherwise you will hit an exception in <i>autovue.js</i>. Cookies encryption is only supported under loopback connection protocol.</p>

5. Launch AutoVue client by invoking the following start API:

```
myAvApp.start(onInit, onFail, user_data);
```

The parameter `<onInit>` is a JavaScript Callback method invoked when the browser connects to AutoVue and the scripting API is ready for use. This callback does not carry any argument.

The parameter `<onFail>` is a JavaScript Callback method invoked when the browser fails to connect to AutoVue. An example that prompts the user for "Retry" three times and suggests notifying AutoVue administrator through e-mail is provided in `av_jnlp.html`. This callback must follow the following prototype:

```
function onFail(xmlhttp_request, error_msg, user_data),
```

where:

- `xmlhttp_request`: The last XMLHttpRequest object used by the browser to communicate with AutoVue.
- `error_msg`: A text string describing the error preventing the connection to AutoVue.
- `user_data`: Custom object sent among the argument of this start API.

The parameter `<user_data>` is a custom object that will be sent within the arguments of the `onFail` callback.

6. If you installed the optional sample files that ship with AutoVue, edit `av_jnlp.html` and replace the values for the variable identified in Table 6–6 with the appropriate value:

Table 6–6 Variables

Variable	Description
AVSAMPLES	<p>Note: This feature is optional. During AutoVue installation, you must select the Sample Files check box to include the sample files.</p> <p>Specify the URL to the AutoVue sample files on your Web Server.</p> <p>For example:</p> <p>http://AutoVueClient:5098/AutoVue/samples</p>

6.9.5.1 Configuring AutoVue Client for Secure Socket Layer (SSL)

Note: If you use the Rendezvous communication under HTTPS protocol instead of the loopback connection then skip this section (No need for a localhost SSL certificate in the Rendezvous solution).

When used in an https-based environment, web browsers enforce a "mixed active content" restriction which requires the communication with AutoVue Client be performed through a secure socket. In order to open this socket, the client needs to possess a server certificate. The AutoVue Client detects when this configuration occurs, and retrieves the server certificate. The certificate can be either self-signed, or signed with an existing local certificate authority.

Note: AutoVue supports SHA-2 (Secure Hash Algorithm 2) certificates.

A new tool (makeAvCert) is distributed with AutoVue to provision this certificate for an installation. It can be implemented in an installer, or handled as a manual configuration. To run AutoVue under HTTPS Protocol, you have to do the following additional steps:

1. Generate a security certificate for "localhost". This certificate will only be used to enable SSL communication between the users' browsers and the AutoVue Client, so it should be as restricted as possible. The administrative tool provided with AutoVue (makeAvCert) produces suitable certificates. Running the utility will generate two files: av_cert.pem which contains the complete certificate with key-pair, and localhost.cer which contains the public information. The utility need the package tools.jar to run.
2. Copy the complete certificate file (av_cert.pem) to the directory you created to your Web server root.
3. Add the URL of the SSL certificate file (av_cert.pem) as an AutoVue client parameter, to the JNLP file required to start AutoVue. For example, if you use the servlet VueJNLPServlet provided with AutoVue, then you should add this URL in the application-desc portion of the autovue.jnlp template used by this servlet:

```
<application-desc main-class="com.cimmetry.jvue.JVueApp">
  <argument>
    -paramsslcert_url= https://AutoVueServer:ServletPort/AutoVue/av_cert.pem
  </argument>
  ...
</application-desc>
```

4. Note that using self-signed localhost certificates requires them to be loaded as a certificate exception in each user's browser (The localhost.cer file generated by makeAvCert is intended for this purpose). For the best experience, this should be done before users attempt to use AutoVue. In Microsoft Windows environments, administrators can automate this operation by using the Group Policy Management facility. Mozilla Firefox maintains its own independent certificate store, normally stored in a file named cert8.db. A variety of third party tools have been developed for administrators to manage the certificates pushed to users. Any of these tools may be used to import the local host certificate for AutoVue.

In Apple and Linux environments, there are fewer standard distributed administration tools. Importing the certificate may be handled manually if necessary by exporting the certificate from the keystore to DER or PEM format and importing to certificate stores by opening the files in the browser, or by using the command line tools provided in each system (keychain application (MacOS), certutil (Linux)).

In Enterprise environments where a local certificate authority (CA) has been set up, a localhost certificate can be generated that is signed by the local CA. Since users will already have the local CA configured in their browsers, the importing of a self-signed certificate can be omitted. This option is only suitable for environments where the infrastructure work has already been configured.

6.9.5.2 Verifying AutoVue Client

Once you have the AutoVue client components installed, you can verify if the installation is successful by opening the URL to the AutoVue client files. For example: `http://<AutoVue server hostname>:<Port#>/AutoVue/av_jnlp.html` (case-sensitive).

The AutoVue client should start and the AutoVue user interface should be displayed.

You should be able to launch the online help files from the AutoVue client by selecting **Help > Online Help** from the AutoVue user interface.

By default, depending on the connection protocol, av_jnlp uses the following approach to interact with AutoVue client:

- HTTP protocol: Loopback connection
- HTTPS protocol: Rendezvous communication.

Protocol command processing can be seen in both the browser and application server console outputs.

However, you can force using any of the two approaches by appending a parameter to the URL above:

- `http://<AutoVue server hostname>:<Port#>/AutoVue/av_jnlp.html?protocol=loopback` (or `lbk`) to force a loopback connection
- `http://<AutoVue server hostname>:<Port#>/AutoVue/av_jnlp.html?protocol=rendezvous` (or `rdv`) to force a Rendezvous communication

6.9.5.3 Troubleshooting AutoVue Client

If you are unable to load the AutoVue client or the client starts up but is unable to connect to the AutoVue server, refer to following trouble-shooting pointers:

- Verify that all the proper JAR files are copied over to the Web server/application server, and that they match with the ones on the AutoVue server. This step is very important when upgrading your version of AutoVue.

- If you cannot load the HTML/JSP pages, ensure that you have the correct URL to the AutoVue client in the Web server/application server.
- If you are able to load the HTML/JSP pages, but the AutoVue client does not load:
 - Ensure that a JRE is installed on the client machine. If using a browser, ensure that the Java webstart is installed.
 - Verify that the CODEBASE_HOST parameter (second parameter in the AutoVue object constructor called in av_jnlp.html or batchprint_jnlp.html files) set in the AutoVue client pages points to the correct URL.
- If you get an error message that indicates that there was a problem communicating with the server:
 - Verify that the JVUESERVER parameter in the Web server/application server point to the right VueServlet URL.
 - Verify that the application server hosting the VueServlet is running.
 - Verify that the AutoVue server is running.
- If you get an error message stating “Failed to connect as an authenticated user”:
 - Verify that there is an authentication mechanism (for example, authentication plug-in or DMS authentication) configured for the AutoVue server.
 - If AutoVue is installed as a stand-alone server (that is, without an authentication plug-in or DMS authentication), then you must set jvueserver.authentication.enable to FALSE so that users can connect to the server.

Note: Oracle recommends to prohibit all unauthenticated connections to the AutoVue server.

6.9.5.4 Configuring Online Help

You must copy the AutoVue Online help files to the docroot location and then verify the URL of the online help to make sure you are able to load the online help files. Update the URL to the help files in AutoVue server's jvueserver.properties file. For more information, refer to [Installing AutoVueClient Components](#).

Configuring AutoVue for Agile PLM

This section details additional configurations that can be made to your installation of AutoVue for Agile PLM.

A.1 Security

During the viewing process, documents available for viewing are rendered on AutoVue for Agile PLM, and an intelligent and displayable data stream is delivered to the client. In this way, the AutoVue for Agile PLM implementation eliminates any concerns about security since the original document is not transferred to the client computer. While the viewable data is being transmitted, AutoVue for Agile PLM uses compression and streaming algorithms to achieve a quick and responsive feedback to the client side for large and multi-page documents.

The server has complete control over the functions available to clients. For example, users may be granted or denied rights to perform printing, redlining, or any other functions depending on their access permissions.

AutoVue for Agile PLM uses cookies to track the number of users. The cookie is a unique number assigned to each browser that defines a user as a single session, regardless of the number of files a user consecutively or simultaneously opens.

Note: If you disable or refuse cookies in your browser, AutoVue for Agile PLM will continue to work. However, each new instance of the browser creates a new session in AutoVue for Agile PLM.

HTTPS can be used for secure communications if you tunnel all communications between the viewer client and the server through HTTPS, which uses SSL. The applet must communicate with the AutoVue server through the AutoVue for Agile PLM servlet and any Agile PLM communications that are tunneled are encrypted.

A.2 Starting AutoVue Server on Windows

Following is the procedure to start the AutoVue server on Windows:

1. Start the AutoVue server in the **Oracle AutoVue** Program Manager group.

By default, when the server is started, the console is displayed and the server appears in the system tray.

When you start the server as a service, you may not see the server console. To display the console, run the following command from the <AutoVue Install Root>\bin directory:

```
jvueserver_debug -u
```

Note: The AutoVue server starts up with a default ProcessPoolSize of 4. To modify the ProcessPoolSize, set the `jvueserver.processPoolSize` parameter in `jvueserver.properties`.

2. Start the application server and then start the File Manager.

Note: For information on starting AutoVue as a service, refer to [Section A.5, "Running the AutoVue Server as a Service"](#).

A.3 Starting AutoVue Server on Linux

Following is the procedure to start the AutoVue server on Linux:

1. Start the AutoVue server by entering the following:

```
./jvueserver
```

This starts up the server console as long as the `DISPLAY` environment variable is properly set.

When you start the server as a service, or when the `DISPLAY` environment variable is not set properly, you will not see the server console. To display the console, run the following command from the `<AutoVue Install Root>/bin` directory:

```
./jvueserver_debug -u
```

Note: The AutoVue server starts up with a default ProcessPoolSize of 4. To modify the ProcessPoolSize, set the `jvueserver.processPoolSize` parameter in `jvueserver.properties`. Refer to [Chapter A, "Configuring AutoVue for Agile PLM"](#) for more information.

2. Start the application server and then start the File Manager.
3. The startup script for the AutoVue server on Linux OSes also starts up the Xvfb server. Xvfb is an X11 virtual frame-buffer that helps the AutoVue server render files. The Xvfb server runs on port 909 by default. To modify this port and configure other Xvfb properties, open `jvueserver.properties` (located in the `<AutoVue Install Root>/bin` directory) and locate property names containing "xvfb".

If you want the AutoVue server to continue running after you close the terminal window, or after you log out of the Linux machine, you must exit the shell (console window) used to start the AutoVue server before logging out of Linux. The server continues running even after you log off. To exit the shell, you must enter `exit` (do not exit by clicking the **Close** button).

Note: For information on starting AutoVue as a service, refer to [Section A.5, "Running the AutoVue Server as a Service"](#).

A.4 Shutting Down the AutoVue Server

To shut down the AutoVue server, click **Shutdown** on the AutoVue server console. You can also shut down the AutoVue server through the command-line interface by entering the following command:

```
jvueserver_debug -u -shutdown
```

If you are running the AutoVue server as a service, you must shut it down as you would any service.

A.5 Running the AutoVue Server as a Service

When running the AutoVue server as a service, you must run it as a named user and not as Local System Account, as the local system account has more privileges than a named account.

A.5.1 On Windows OSes

The AutoVue server can be run as a Windows Service. The advantage of this is that it continues to run even after you log off of Windows. Before running the AutoVue server, first verify that it runs properly in "non-service" mode (for example, run by clicking the **Start AutoVue Server** button in the *Start* menu).

To install the service, go to the \bin folder of the directory where you installed the AutoVue server and enter the following:

```
jvueserverX.exe -install<user information>
```

where <user information> is in the form "domain\username password". This ensures that the AutoVue server service runs as a named user instead of the local system account.

Note that the user account information entered must be registered in the *Log on as a service* local security policy. If the user information is not there, the Oracle AutoVue service will not start. To check or edit the security policy, perform the following steps:

1. Open the Windows Control Panel.
2. Double-click **Administrative Tools**.
3. Double-click **Local Security Policy**. The Local Security Settings window appears.
4. Expand **Local Policies** and then select **User Rights Assignment**.
5. Double-click **Log on as service**. The Log on as a service Properties window opens.
6. Verify that the user account is listed under the Local Security Setting tab. If not, click **Add User or Group** to add the user account information.

To remove the service, go to the <AutoVue Install Root>\bin directory and enter the following: Starting and Stopping the Service

```
jvueserverX.exe -remove
```

A.5.1.1 Starting and Stopping the Service

1. In the Control Panel select **Administrative Tools > Services**.

2. Select the **Oracle AutoVue Server** service.
3. Click **Startup**.
4. Select whether you want the service started automatically on re-boot or manually. The default is **Manual**.
5. If you select **Manual**, you can start the service by doing one of the following:
 - Click **Start** in the Services dialog
 - or*
 - Use the **sc.exe** utility.
For example: `SC start "Oracle AutoVue Server"`
 - or*
 - Use the NET program.
For example: `NET start "Oracle AutoVue Server"`

To shutdown the service, select the **Oracle AutoVue Server** service in the Services dialog and click **Shutdown**.

A.5.2 On Linux OSes

Oracle provides an *RC-Script* to manage the AutoVue server on Linux. The AutoVue server can be configured to startup automatically when the machine is restarted by following these steps:

1. Edit file `<AutoVue Install Root>/etc/jvueserver_rc` and locate the following lines:

```
AUTOVUEDIR=$USER_INSTALL_DIR$
AUTOVUEUSER=__JVUEUSER__
```

1. Replace `$USER_INSTALL_DIR$` with the path to AutoVue installation and `__JVUEUSER__` with the name of the user that will be running the AutoVue server.
2. Rename `jvueserver_rc` to `autovue`.
3. Login as a root and copy `autovue` to `/etc/init.d` folder.
4. As root, go to the `/etc/init.d` folder and add AutoVue as a service:

```
chkconfig --add autovue
```

5. Configure `autovue` to startup automatically:

```
chkconfig autovue on
```

The AutoVue server now starts up automatically when the machine starts up.

To start the AutoVue service, manually, run

```
service autovue start
```

To stop the AutoVue service manually, run

```
service autovue stop
```

To remove the AutoVue service, run

```
chkconfig -del AutoVue
```


A.6 Testing the VueServlet

Note: The Agile View servlet (VueServlet) is packaged and deployed as part of the File Manager installation. The VueServlet is deployed on the File Manager and tunnels the requests to the Agile PLM Application Server using HTTP protocol. No additional configuration is required.

The VueServlet is used to access the AutoVue server across firewalls from external clients on standard HTTP/HTTPS ports.

You must ensure that the VueServlet has been loaded successfully and that the VueServlet communicates with the AutoVue server. Open a browser and type the following URL in the address bar:

```
http://FileManager_hostname:port/Filemgr/Configuration
```

If this page loads, it means that the File Manager was successfully started and initialized. To view the status of the VueServlet from this page, enter the following in the address bar of the browser:

```
http://FileManager_hostname:port/Filemgr/VueServlet
```

- If you specify this URL in the browser and receive the message "Page not found," check the following:
 - Ensure that the AutoVue server and File Manager were started successfully.
 - Check the URLs to make sure that you are specifying the correct File Manager hostname, port and path (for example, /Filemgr/Configuration and /Filemgr/VueServlet)

A.7 Managing Multiple Viewers for Load Balancing

The AutoVue server is designed to provide server scalability to support increasing demand for file viewing and markup on intranets and the Internet. It can run in a process pool on a single computer that improves the responsiveness for simultaneous connections and balances the load across a multi-CPU computer.

The default process pools four concurrent services. It is specified in the `jvueserver.properties` file located in the AutoVue server home folder at `\\AGILE_HOME\AVS\bin`:

```
jvueserverx.nt.processPoolSize=4
```

As a general rule, you should allow approximately 50 MB for each process in the pool. A process pool size of four services requires approximately 200 MB of RAM. On a computer with 256 MB of RAM, this allows the remaining RAM to run the operating system and other services. Increasing the RAM on a computer allows you to increase the number of services using the process pool.

The system allows you to add servers to support more users while the servers are online, without restarting or disconnecting existing servers. Adding capacity is as simple as adding one more server. When using multiple viewers, the load can be efficiently balanced across the servers by manually directing the viewer client requests to different viewers.

A.7.1 Client-Server Communications

For the AutoVue server the issue of using IP addresses for client-server communication at runtime (rather than hostnames or DNS names) is resolved because the File Manager also hosts the VueServlet. The VueServlet acts as a proxy for the AutoVue server, and it is placed on the intranet with other Agile server components. In addition, since the VueServlet is running on the File Manager with an Agile Web client, the Web proxies also forward external requests to the intranet. The VueServlet comes as part of the File Manager installation.

A.7.2 Removing AutoVue for Agile PLM

Follow these instructions to remove AutoVue for Agile PLM.

To remove AutoVue for Agile PLM on Windows:

1. Stop the AutoVue service.
 - a. Open the Services dialog box.
 - b. Select **AutoVue Server** and stop the service.
 - c. Close the Services dialog box.
2. Choose to **Start > Settings > Control Panel**.
3. Double-click **Add/Remove Programs**.

The Add/Remove Programs Properties dialog appears.
4. Scroll down the window and select **AutoVue for Agile PLM**.
5. Click **Change/Remove**.
6. Click **Uninstall**, and click **OK** to confirm the deletion.
7. Click **Exit**.

To remove AutoVue for Agile PLM from the Start menu:

1. Choose **Start > All Programs > AutoVue for Agile PLM > Setup > Uninstall AutoVue**.
2. Select and delete the installation folders from the location of Agile home directory.

To remove AutoVue for Agile PLM on Linux:

1. Stop the AutoVue server by clicking Shutdown on the View Server applet or killing the jvueserver process.
2. Go to the VIEWER_HOME/AVS/_uninst directory.
3. Run the uninstall_linux2.bin file to uninstall AutoVue for Agile PLM.

AutoVue Server Configuration Options

You can configure the AutoVue server by modifying `javueserver.properties` located in the `<AutoVue for Agile PLM Install Root>\bin` directory. For example, the following sections describe options that can be configured if you wish to modify the ports that the AutoVue server is running on, or if you wish to set up a server farm or perform any other server configuration.

Note: You must restart the AutoVue server for the changes in `javueserver.properties` to take effect.

B.1 Authentication Option

The option allows you to indicate that the AutoVue server has authentication enable and that users must enter login credentials in order to gain access to the server.

Table B–1 Authentication Option

Parameter	Description	Default
<code>javueserver.authentication.enable</code> =[TRUE FALSE]	Specify whether authentication is enabled for the AutoVue server. Oracle recommends to prohibit all unauthenticated connections to the AutoVue server.	FALSE

B.2 AutoVue Host Name Option

If you rename your server machine name after you install AutoVue, you must update this parameter:

```
javueserver.hostname = [host name]
```

Note: This new server hostname must be properly reflected in the `JVUESERVER` parameter specified in the `VueServlet` descriptors that point to this server. You must set the correct static or resolvable IP address or hostname or fully-qualified hostname (FQDN) in the file `javueserver.properties`.

B.3 RMI and Socket Ports Options

This section provides RMI and socket port parameters that may be configured. For example, the RMI port may need to be configured when setting up an AutoVue server

farm, and the socket port may need to be modified to meet company policy requirements on the usage of ports within a certain range.

Table B–2 RMI and Socket Port Options

Parameter	Description	Default
<code>javueserver.rmi.objectPorts = [2020-2029]</code>	Specify a range of ports to use, or leave commented for automatic allocation.	
<code>javueserver.socket.timeout= <integer></code>	Specify the inactive time in seconds after which socket times out. When 0, there is no timeout.	0
<code>javueserver.rmi.port = <port value></code>	<p>The RMI port can be used to communicate with other servers when AutoVue is set up in a server farm.</p> <p>In certain situations you may need to modify the RMI port. For example, you must modify the port when the default port is used by other applications or when a company policy requires the usage of ports within a certain range.</p> <p>Note: These port numbers are not related to the HTTP port used by the Web server.</p> <p>AutoVue uses n+1 consecutive ports starting from the base RMI port, where n is the processPoolSize value specified in <code>javueserver.properties</code>. You should verify that the required port is open and not in use by any other process. The <code>netstat -a</code> program displays which ports are in use.</p>	1099
<code>javueserver.socket.port = <port value></code>	<p>In certain situations you may need to modify the socket port. For example, you must modify the port when the default port is used by other applications or when a company policy requires the usage of ports within a certain range.</p> <p>This new socket port needs to be properly reflected in the <code>JVUESERVER</code> parameter specified in the <code>VueServlet</code> descriptors that point to this server.</p> <p>Note: This port number is not related to the HTTP port used by the Web server.</p> <p>AutoVue uses n+1 consecutive ports starting from the base socket ports, where n is the processPoolSize value specified in <code>javueserver.properties</code>. You should verify that the required port is open and not in use by any other process. The <code>netstat -a</code> program displays which ports are in use.</p>	5099
<code>javueserver.ssl.enable= [TRUE FALSE]</code>	<p>Specify whether to enable/disable secure socket (SSL) connections for the server. This property is required when SSL connection is enabled for the <code>VueServlet</code>.</p> <p>Set to FALSE to disable SSL connections for the server.</p> <p>Set to TRUE to enable SSL connections for the server.</p>	TRUE

B.4 Process Pool Size Option

The AutoVue server can run in a process pool on a single machine. The default process pool (DocServer) size is 4, and is set in the `javueserver.properties` file.

Table B-3 Process Pool Size Option

Parameter	Description	Default
<code>lvueserver.processPoolSize</code> = [integer]	Set the process pool size to specify the number of DocServers to run when the AutoVue server starts up.	4

Creating a process pool helps improve the responsiveness when handling simultaneous connections and also helps balance the load across processors in a multi-CPU machine. As a general guideline, you should allow for a minimum of 200MB for each process in a pool, of which approximately 50MB is for the JVM and 128MB for the Java heap. As a result, a process pool size of 4 requires at least 1GB of RAM on the machine to run comfortably. The load is balanced across the pool on the single machine.

File viewing requires memory on top of the amount for each process in the pool. Depending on the number of users and files loaded at any given time, the recommended minimum is 2GB per DocServer. Provisioning for 2GB of memory per process in the pool should be expected.

Note: Do not modify the DocServer memory settings in `lvueserver.properties`.

B.5 Proxy Connection Options

If the machine hosting the AutoVue server 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 must connect to the Internet to retrieve required resources if missing from a file. To do so, the proxy server name must be specified in `lvueserver.properties`.

Note: If AutoVue is integrated with a DMS, you must also set the proxy setting to allow requests to go through as all machines on the internal network must pass through the proxy server.

```
lvueserver.http.proxyhost=my.proxyserver.com:80
lvueserver.ftp.proxyhost=my.proxyserver.com:80
```

Replace `my.proxyserver.com` with the name of the proxy server running on the server and the port with the appropriate port number.

B.6 Streaming Files Options

This section provides streaming files parameters that may be configured. By setting these parameters, you can specify whether to allow streaming file generation, the maximum lifetime of streaming files, and much more.

Table B–4 Streaming File Options

Parameter	Description	Default
jvueserver.metacache. enable = [TRUE FALSE]	<p>Specifies whether to generate streaming files.</p> <p>When set to TRUE, streaming files are stored in the location specified by the <code>jvueserver.cache.directory</code> parameter.</p> <p>When set to FALSE, streaming files are not generated.</p> <p>Note: Streaming file generation is not supported for PDF Portfolio files.</p>	TRUE
Note: The following options can be set if <code>jvueserver.metacache.enable=TRUE</code> .		
jvueserver.metacache. pdf.enable = [TRUE FALSE]	<p>Set to FALSE: Streaming file is not generated for PDF.</p> <p>Set to TRUE: Streaming file is generated for PDF files.</p> <p>This configuration parameter should be set manually.</p> <p>It is recommended to set this option to FALSE as there is no benefit to enabling streaming files for PDF.</p>	FALSE
jvueserver.dms.save. metafile = [TRUE FALSE]	<p>Specifies whether or not streaming files are saved in the DMS.</p> <p>Set to TRUE to save streaming files in DMS.</p> <p>Set to FALSE so that streaming files will not be saved in DMS.</p>	TRUE
jvueserver.metacache. process = [TRUE FALSE]	<p>Flag for using a separate process for streaming file generation.</p> <p>If set to FALSE, the DocServers handle streaming file generation and the dedicated streaming file process does not start.</p>	TRUE
jvueserver.metacache. threshold = [non-negative integer]	<p>Specifies the DocServer threshold at which the streaming file DocServer handles the generation of streaming files. <code>jvueserver.metacache.process</code> must be TRUE for this option to take effect.</p> <p>Increasing this value allows the DocServer that loads a file to generate the streaming file.</p> <p>By default, the threshold is set to 0. That is, the streaming file DocServer generates the streaming files for all documents.</p>	0
jvueserver.cache. directory=[directory path]	<p>Specifies in which directory the cached files should be saved. A central cache information file named <i>cache.map</i> is stored in the same directory. By default, the directory is the <i>Cache</i> subdirectory of the AutoVue server program directory.</p>	<AutoVue Install Root>\bin\Ca che
jvueserver.cache. forceascii=[0 1]	<p>Set to 1 to force the use of ASCII characters in cached files names.</p> <p>Set to 0 to leave characters as is.</p> <p>For example, you may want to use force ASCII characters if the server does not support file names with Unicode characters.</p>	0

Table B–4 (Cont.) Streaming File Options

Parameter	Description	Default
jvueserver.cache.size= [value in MB]	Specifies, in Megabytes, the maximum size of the file cache. The default value is 20GB. If not specified, or if value specified is less than 50 MB, a value of 4GB will be used.	20480
jvueserver.cache. maxlifetime = [number of days]	Specifies the maximum number of days a file is kept in the AutoVue cache directory. When the maximum life time is reached, the file is deleted from the cache directory. Note: The minimum value is 1.	30
jvueserver.metacache. zipfile.enable = [TRUE FALSE]	Specifies whether to generate streaming files from ZIP files. If set to TRUE, then streaming files are generated. If set to FALSE, no streaming files are generated. Note: This option will only take effect when the jvueserver.metacache.enable option is set to TRUE. Only one streaming file is allowed for a ZIP file. It is recommended that this setting is left unmodified when the deployment is integrated with a DMS.	FALSE
jvueserver.cache. maxnumfiles= [value]	Specifies the maximum number of files allowed in the AutoVue cache directory. When the threshold is reached, the least recently used files are deleted. Note: The minimum value is 1000.	64000

B.7 DMS Options

This section provides DMS parameters than may be configured. However, we recommend that you do not modify these parameters.

Table B–5 DMS Options

Parameter	Description	Default
dms.save.compress=[TRUE FALSE]	Set to TRUE to compress save data transmitted to the DMS. Note: We recommend that you do not modify this parameter.	TRUE

Note: These options will be deprecated in the next release of AutoVue.

B.8 Collaboration Options

When using the collaboration feature in AutoVue for Agile PLM, you can configure the following parameters.

For example: You may choose to enable the collaboration feature on the server, and/or you can specify the protocol to use for collaboration.

Table B–6 Collaboration Options

Parameter	Description	Default
jvueserver.collaboration. .dmsargsfilter=<regex>	This option filters out sensitive DMS arguments from the DocID that is passed to collaboration session guests. <regex>: The Java-style regular expression matching the entire DMS argument name. It is not case sensitive.	.*(SESSION USERNAME PASSWORD).*
jvueserver.collaboration. .enable = [TRUE FALSE]	Set to TRUE to enable collaboration mode on the server. Set to FALSE to disable collaboration mode. Note: If not using Real-Time Collaboration, set this parameter to FALSE.	FALSE
jvueserver.collaboration. .tcp.port=[integer]	BaseTCP port to be used. Note: The configuration parameters below need to be changed when using more than one server cluster in a server farm.	9700
jvueserver.collaboration. .id.min=[integer]	Minimum ID given to users and collaboration sessions by this server. Change this ID when you are running many AutoVue servers that must communicate together for collaboration. The second server must have a minimum ID of at least jvueserver.collaboration.id.min + jvueserver.collaboration.id.range of the first server. Otherwise, an ID overlap may occur.	0
jvueserver.collaboration. .id.range=[integer]	Range of IDs given to users and collaboration sessions by this server. This will limit the number of simultaneous connections.	100000

B.9 log4j and Diagnostics Options

This section provides log4j and diagnostics parameters that may be configured. These parameters can be set to configure the logging level and time interval for detecting log4j configuration changes, and the output diagnostics information.

Table B–7 Log4j and Diagnostics Options

Parameter	Description	Default
jvueserver.log4j.configurea ndwatch = [TRUE FALSE]	Set this to <i>TRUE</i> to be able to dynamically change the log4j logging level.	FALSE
jvueserver.log4j.configurea ndwatch.delay =[integer] seconds	Time interval for waking up and detecting log4j configuration change.	60
jvueserver.diagnostics.for mat=[xml text]	Specify the output format for the AutoVue server diagnostics.	xml
jvueserver.diagnostics.peri od=[interval in minutes]	Specify the interval in minutes at which the AutoVue server diagnostics are generated.	No default; diagnostics are generated on demand.

B.10 Modified With JavaScript Option

This option allows administrators to define which INI options cannot be modified using JavaScript.

Table B–8 Modified with JavaScript Option

Parameter	Description	Default
jvueserver.profile.options. filter= <INI option>	This option allows administrators to define which INI options under the [OPTIONS] section cannot be modified using JavaScript. The option takes a regular expression string. Example: The default value for the option is "(XREFPATHS XFONTPATHS BKIMAGES)", which means that both XREFPATHS, XFONTPATHS and BKIMAGES cannot be modified using JavaScript.	(XREFPATHS XFONTPATHS BKIMAGES)
jvueserver.profile.markups. .filter= <INI option>	This option allows administrators to define which INI options under the [MARKUPS] sections cannot be modified using JavaScript. The option takes a regular expression string. Example: The default value for the option is "SIGNOFFFILE", which means that SIGNOFFFILE cannot be modified using JavaScript.	SIGNOFFFILE

B.11 Reboot Option

The following table describes the reboot parameter for DocServers in jvueserver.properties. By setting this parameter, you can control the reboot time interval for DocServers.

Table B–9 Reboot Option

Parameter	Description	Default
jvueserver.reboot.timeout =<interval in minutes>	If a DocServer is idle for the time specified by this parameter, the DocServer is rebooted. The default time out is 30 minutes.	30

B.12 DocServer Timeout Option

The following table describes the DocServer timeout parameter for DocServers in jvueserver.properties. By setting this parameter, you can control the hang logs generation time interval for DocServers.

Table B–10 DocServer Timeout Option

Parameter	Description	Default
docserver.timeout=N (minutes), ClassName	Specify the timeout in minutes after which the AutoVue server hang logs are generated. It is possible to implement a custom action to invoke a hang situation. When implemented, replace ClassName with the name of the class implementing the custom action.	10

B.13 Recovery Attempt Option

The following table describes the recovery attempt parameter in `juveserver.properties`. By setting this parameter, you can control the number of recovery attempts for the DocServer.

Table B–11 *Recovery Attempt Option*

Parameter	Description	Default
<code>juveserver.recovery.attempts=[integer]</code>	Specify the number of recovery attempts for the DocServer when an exception is thrown. After this number of failed recovery attempts, the DocServer restarts.	5

B.14 DLL Version Option

The following table describes the extra libraries parameter that allows you to specify user-defined DLLs:

Table B–12 *DLL Version Option*

Parameter	Description	Default
<code>juveserver.version.extraLibraries=DLL_1;DLL_2;DLL_3;...</code>	Specify a semi-colon separated list of user-defined DLLs. AutoVue will list the versions of these DLLs in the Help > About dialog.	

B.15 File Format Information Option

On start up, AutoVue registers all of its components into a `VueServer.ini` file. You can specify an alternate path for `VueServer.ini` using this option.

Table B–13 *File Format Information Option*

Parameter	Description	Default
<code>juveserver.inifile=[file name]</code>	Specify the INI file where AutoVue stores information on the file formats supported by AutoVue. By default, it is <code>VueServer.ini</code> located in the <AutoVue Install Root>\bin directory. Note: AutoVue saves certain memory management settings in this file. We recommend that you do not modify these options or the file.	<code>VueServer.ini</code>

B.16 Global User Options

The following global user settings may be configured. These parameters specify the directory in which user information is stored, and the names for global configuration files.

Table B–14 *Global User Options*

Option	Description	Default
<code>juveserver.users.directory</code>	Contains the directory in which user information is stored (initialization files and GUI files).	<AutoVue Install Root>\bin\Profiles

Table B–14 (Cont.) Global User Options

Option	Description	Default
juvueserver.users.defaultini	AutoVue provides a way to push 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 juvueserver.users.defaultini parameter. This file should be located at <AutoVue Install Root>\bin directory.	default.ini
juvueserver.users.allusersini	AutoVue provides a way to push 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 juvueserver.users.allusersini). This file should be at <AutoVue Install Root>\bin directory.	allusers.ini
juvueserver.users.timeout=[interval in seconds]	Specify the user session timeout in seconds. If the user session is idle for the specified time period, the session is closed.	1800

B.17 Markup Options

You can configure the Markup Files dialog, Markup Files directory, permissions, and markup symbols library by setting the following options.

Table B–15 Markup Options

Option	Description	Default
juvueserver.markup.nativegui.type	Add Author, Date, and Markup Info columns to the Markup Files dialog. 0: Name column displays 1: Enable Author 2: Enable Date 4: Enable Markup Info Note: These are ORed flags. For example: Enter 7 to enable all three columns.	0
juvueserver.markups.directory	Specifies in which directory the Markup files should be saved. Markups are saved with random names in this directory, and the mapping between Markup files and their base file is held in a central map file named markups.map , stored in the same directory. Note that multiple servers should not share the same location for storing markups. Note: This option is for server-managed markups.	<AutoVue Install Root>\bin\Markups
juvueserver.markups.permissions	By default, all users can see the Markups of a file but only the owner of a Markup can modify it. The Permissions key can be used to change that behavior. Setting it to 0 allows all users to see and change Markup files. Setting it to 1 prevents all users from modifying the Markups.	1

Table B–15 (Cont.) Markup Options

Option	Description	Default
<code>javueserver.markups.symbols.directory</code>	Specifies in which directory the Markups symbol libraries are stored. By default, the directory is the symbols subdirectory of the AutoVue server program directory.	<AutoVue Install Root>\bin\Symbo ls

B.18 Online Help Options

You can specify the entry points for language-specific Online help by setting the following options.

Table B–16 Online Help Options

Option	Description
<code>javueserver.help.file_en</code>	Entry specifies the URL to the English Help file. If Online Help does not exist for a language, AutoVue loads the English help file by default.
<code>javueserver.help.file_xx</code>	Entry specifies the URL to the Help file for the language "xx."

B.19 Memory Optimization

AutoVue performs memory management when loading large files. If AutoVue memory hits a pre-defined threshold, AutoVue dumps the least recently used data from memory to the disk. This memory management scheme helps load larger models in AutoVue. Memory management is enabled by default. To disable it, you must set `javueserver.memory.managed=FALSE` in `javueserver.properties`. Refer to the following table for all memory management-related configurations.

In `javueserver.properties`, you can set the following parameters to optimize memory or performance speed.

Table B–17 Memory Optimization

Parameter	Description	Default
<code>javueserver.memory.managed = [TRUE FALSE]</code>	<p>This option orients the optimization in the product towards speed or memory.</p> <p>If set to FALSE, speed is optimized.</p> <p>If set to TRUE, memory is optimized. Setting this option to TRUE does not impact the loading of Office and Raster formats.</p> <p>Note: When this option is set to TRUE, AutoVue's memory manager dumps the least recently used components from memory onto disk when the process memory hits the threshold specified in <code>javueserver.memory.threshold</code>.</p> <p>Effect on Performance:</p> <p>Performance speed is improved if value is set to FALSE.</p> <p>If loading large files, or files that require a lot of memory, we recommend that you set the option to TRUE to optimize memory usage.</p>	TRUE

Table B–17 (Cont.) Memory Optimization

Parameter	Description	Default
jvueserver.memory.threshold= <i>value</i>	<p>Specifies the process memory threshold for AutoVue after which the memory manager dumps data. Specify value in MB.</p> <p>When set to 0, AutoVue calculates the memory threshold based on the following formula:</p> <p>[(Total memory on the machine)/(n+1 where n is processpoolsize)]*1.2</p> <p>The computed value does not exceed 1GB or the maximum memory size addressable for the process on the system multiplied by 0.8, whichever is less.</p> <p>Note: Minimum value is 256MB</p> <p>Effect on Performance:</p> <p>Performance speed is improved when the threshold is a larger value.</p> <p>To optimize memory usage, set a lower threshold.</p>	0

AutoVue programmatically writes certain memory management options in VueServer.ini. These options should not be modified:

Table B–18 Memory Management Options (Not to be Modified)

Parameter	Description	Default
MNGMEMPAGE SIZE=[num]	<p>When memory management is enabled, specify the size of pages (memory) to allocate when storing the managed data. Each memory page is predefined.</p> <p>num = number of bytes used to allocate pages in memory.</p> <p>Minimum value: 8192 (8KB)</p> <p>Maximum value: 1048576 (1MB)</p> <p>Note: The memory pages are dumped to temporary dumping files located in the path defined in MNGTEMPFNAME.</p>	131072 (128KB)
MNGTEMPFNAME=[<i>folder location</i>]	<p>When memory management is enabled, specifies the location and name of the temporary dumping folder.</p> <p>If the temporary dumping folder does not exist, the folder is created and marked for deletion.</p>	<AutoVue Install Root>\bin\avdump

B.20 Vuelink Encryption Options

The authorization block containing the user credentials is encrypted between Vuelink and AutoVue using the Key-Agreement encryption approach. The algorithms used for this encryption are configurable to allow the customer keeping them up-to-date with respect to the state of art practices in terms of security. The configuration parameters that you can configure on the AutoVue Server side and the Vuelink side are as follows:

- On the Server side: Define the following two parameters in the jvueserver.properties configuration file:
 - jvueserver.encryption.algorithm

- `javax.server.encryption.keyagreement.algorithm`
- On the Vuelink side: Define the following two initialization parameters in the Vuelink servlet descriptor in the web configuration file (`web.xml`):
 - `DecryptionAlgorithm`
 - `KeyAgreementAlgorithm`

All the parameters are required by an integration solution because AutoVue does not define default values for them. So, the integration solution will not work if they are not provided. AutoVue installer adds them currently to `javax.server.properties` and the Vuelink web configuration file. It uses the following values for them:

- Encryption / Decryption Algorithm ie `javax.server.encryption.algorithm= AES`
- Key Agreement algorithm (Secret key)
`javax.server.encryption.keyagreement.algorithm=EC`

Code Example:

```
<servlet-name>DMS</servlet-name>

<servlet-class>com.mycompany.autovueconnector.DMS</servlet-class>

<!-- Encryption Algorithms -->
  <init-param>
    <param-name>DecryptionAlgorithm</param-name>
    <param-value>AES</param-value>
  </init-param>

  <init-param>
    <param-name>KeyAgreementAlgorithm</param-name>
    <param-value>EC</param-value>
  </init-param>
</servlet>
```

Table B–19 Vuelink Encryption Parameters

Parameter	Description	Default
<code>javax.server.encryption.keyagreement.algorithm=value</code>	Key Agreement algorithm	Insecure legacy encryption is used by default for backward compatibility with 21.0.1 (highly not recommended)
<code>javax.server.encryption.algorithm=value</code>	Key-Pair Generator algorithm used for the data encryption	Insecure legacy encryption is used by default for backward compatibility with 21.0.1 (highly not recommended)

B.21 Linux-Specific Options

The following section lists Linux-specific parameters that can be configured in `javueserver.properties`.

Note: It is not recommended that you change these options. The AutoVue installer will configure these options to values that is appropriate for your system. These options should not be modified unless there are particular constraints for your deployment.

B.21.1 Preload Java Class Option

The following table describes java class preload parameter in `javueserver.properties`.

Table B–20 Preload Java Class Option

Parameter	Description	Default
<code>javueserver.preload=[preloader class name]</code>	Enables loading of specified java class prior to the AutoVue server startup.	<code>com.cimmetry.javueserver.util.UnixPreloader</code>

B.21.2 Xvfb Options

AutoVue provides the following options to initialize Xvfb parameters. It is not recommended that you change these options.

Table B–21 Xvfb Options

Option	Description	Default
<code>xvfb.display=<port number></code>	Specifies the initial port to use for Xvfb. In the event of a port conflict, modifying <code>xvfb.display</code> may resolve the issue.	909
<code>xvfb.process</code>	Specifies the name of process to start or kill Xvfb process.	Xvfb
<code>xvfb.policy</code>	If set, this contains path name to security policy file, used with Xvfb <code>-sp</code> option when the Xvfb process is started. Note: This is applicable for Linux 5.x.	
<code>xvfb.colormap</code>	If set, this contains path name to color map file, used with Xvfb <code>-co</code> option when the Xvfb process is started. Note: This is applicable for Linux 5.x.	
<code>xvfb.args</code>	These are arguments to be passed to Xvfb when it is started. Note that the default setting runs Xvfb on a non-listening port.	<code>-nolisten tcp -cc 4 -screen 0 1280x1024x24 -pixdepths 1 8 15 16 24 32</code>
<code>xvfb.new=[0 1]</code>	Startup parameter. Set to 1 on Oracle Enterprise Linux (OEL) 6 and Red Hat Enterprise Linux (RHEL) 6 and above. Set to 0 on Oracle Enterprise Linux (OEL) 5 and Red Hat Enterprise Linux (RHEL) 5.	0 for OEL 5 and RHEL 5 1 for OEL 6 and RHEL 6

B.21.3 WINE Options

AutoVue provides the following options to configure WINE parameters. It is recommended that you do not modify these options.

Table B–22 WINE Options

Option	Description
wine.dir	Specifies the directory where WINE is installed.
wine.config.dir	Specifies the directory where the WINE configuration file is located.
wine.config.file	Specifies the WINE configuration file name.

B.21.4 OEM Copyright Notice

AutoVue provides an option to append a custom copyright to the copyright in the Help About dialog:

Table B–23 OEM Copyright Notice Option

Parameter	Description	Default
oem.copyright.notice=	Specify a notice to add to the default Oracle copyright in the Help About dialog.	

B.22 VueServlet Configuration Options

The following table describes VueServlet initialization parameters that can be set in a Web configuration file web.xml. If a context has a webdefault.xml descriptor, it is applied before the context's own web.xml file. For more information, refer to section [Deploying VueServlet](#).

Table B–24 VueServlet Initialization Parameters

Parameter	Description	Default
DebugLevel=[0-100]	Set the debug output category. Values: 0: no logging 10: basic logging 20: HTTP connection information 100: debug	0
EnableSSL=[TRUE FALSE]	Set to TRUE to enable secure socket connection to the AutoVue servers.	TRUE
EnableEM=[TRUE FALSE]	Specify whether or not to retrieve Oracle Enterprise Management information. Set to TRUE to retrieve information. Set to FALSE to disable information retrieving.	FALSE

Table B–24 (Cont.) VueServlet Initialization Parameters

Parameter	Description	Default
JVueServer=[server hostnames]	<p>A semicolon separated list of the AutoVue server host names. This parameter is used by the VueServlet to connect to the AutoVue servers through a socket connection.</p> <p>The JVueServer parameter needs to be set to the hostname:port value used when starting the AutoVue server. This port value must match the port set in jvueserver.properties. You can specify more than one hostname:port separated by semi-colons (;) for fail-over. In other words, if one machine is down the servlet will try the next machine.</p> <p>If JVueServer is not specified, it defaults to localhost:5099. The servlet assumes that the AutoVue server is running on the same machine as the Web server and communicates through port 5099.</p> <p>Note: The port listed in this option should match the port listed in the jvueserver.socket.port option in the jvueserver.properties file.</p>	local hostname:5099
InvokerCount=[value]	<p>Set the number of simultaneous connections from the VueServlet to the AutoVue server.</p> <p>If the number of pending requests at any given time exceeds this set value, then the remaining requests wait in a queue until a connection is free.</p> <p>Note: If this parameter is set too high, then it will cause an overload of requests and result in no connections being made.</p>	100
ServerInfo=[TRUE FALSE]	<p>Set to TRUE to include the AutoVue server information on VueServlet status page.</p> <p>Set to FALSE to hide the AutoVue server information.</p>	FALSE

ABV Configuration

Oracle also delivers an Augmented Business Visualization (ABV) framework, which enables the connecting of portions of documents back to business data in enterprise applications. When integrated with enterprise applications such as Agile PLM, it helps organizations create rich and actionable visual decision-making environments, making it easier for users to consume and understand data in the specific context of business. The solution enables the creation of change objects in Agile from AutoVue PLM via an ABV integration. When a user right-clicks a part in a 3D model in Agile from AutoVue PLM, options are listed to create various change objects (which the user has privileges to create) such as Create Engineering Change Request (ECR) or Create Engineering Change Order (ECO).

C.1 Configuring the ABV Framework

In order to configure the ABV framework, do the following:

1. Copy folder "abvpx" to what ever location from Agile installation location "apcm\integration\sdk\samples\".
2. Modify build.bat to set ANT_HOME to your location.
3. Modify custom.property file to set "wls.deploy.agilelib.dir" and "sdk.deploy.loc"; In installation case, use the installation proper location.
4. Run "build.bat" in command line.

Migration from Loopback to Rendezvous

AutoVue client supports a scripting API. It also provides the following two mechanisms for Web clients to invoke this scripting API:

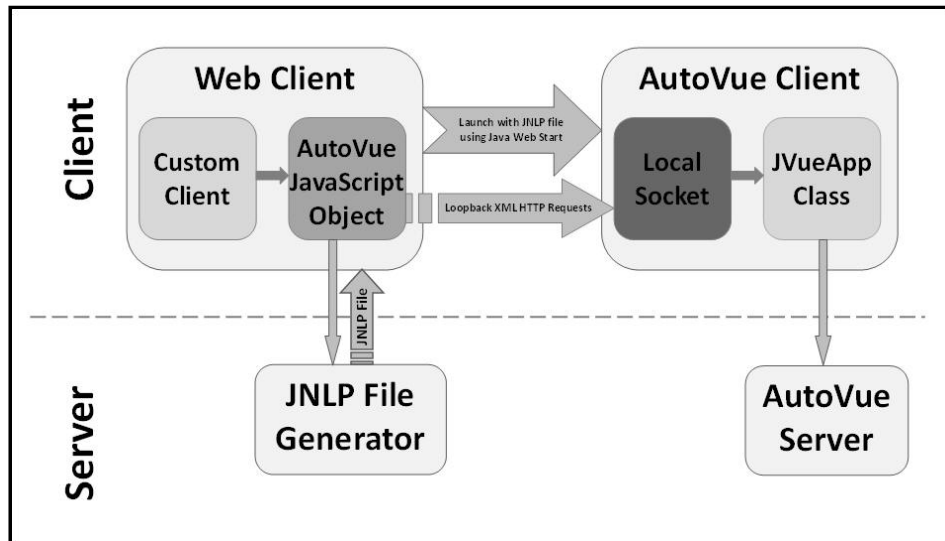
- Loopback connection through a free localhost port
- Rendezvous communication through a Rendezvous servlet.

For convenience, AutoVue includes a JavaScript Object named AutoVue which is implemented in the file `autovue.js`. Currently, this object supports both communication approaches and simplifies the integration of AutoVue into a Web context. The AutoVue object also provides a JavaScript method for each scripting API supported by AutoVue. This appendix provides a summary of both communication solutions and how to configure each solution. Then it provides a migration path from the loopback connection to the Rendezvous communication solution.

D.1 Loopback Connection

Note: This mechanism has been deprecated.

AutoVue client starts a socket listening to XML HTTP requests invoking its scripting API. The socket is opened by an embedded application server and listens to a localhost port suggested by the integration solution.

Figure D–1 Architecture - Loopback Connection**Advantage**

Direct localhost connection provides efficiency and privacy.

Disadvantages

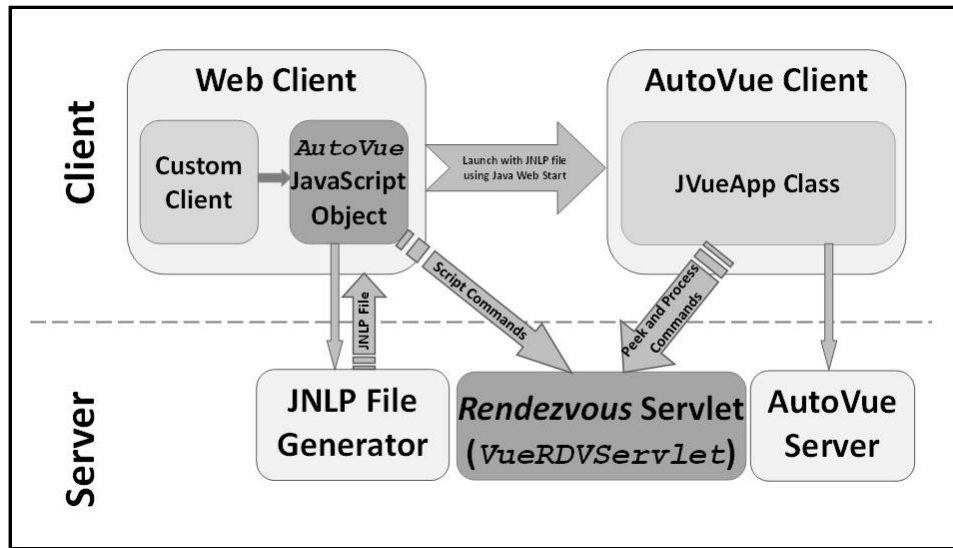
The loopback connection requires some setup that could make it heavy to use in some customers' contexts:

- The integration solution must provide a set of candidate localhost port ranges to try by AutoVue in order to find a free one for the direct connection between AutoVue and the HTML client.
- As described in the section "Integrating in an SSL Environment" of the Oracle AutoVue Security Guide, users need to import a localhost SSL certificate to run this solution under HTTPS protocol.

Note: The loopback connection implemented in Oracle AutoVue Release 21.0.1 is deprecated in Release 21.0.2. As an alternative, the loopback connection is replaced by the Rendezvous communication.

D.2 Rendezvous Communication

The HTML client and AutoVue client communicate together through a remote "Rendez-Vous" servlet named `VueRDVServlet`, deployed on server side. They share a common "Rendez-Vous" ID used by the Rendezvous servlet to link them together.

Figure D-2 Architecture - Rendezvous Communication**Advantages**

After you deploy the Rendezvous servlet on the server side, the solution works out of the box and does not require any additional setup on the client side.

Disadvantages

This approach introduces client/server traffic, which results in the following drawbacks:

- Potentially, less efficient than a loopback connection approach. For example, a saturated Rendezvous server may slowdown all the clients scripting. However, the client load on the Rendezvous servlet is so small that it is most unlikely to overload a protected Rendezvous servlet.
- The solution is slightly more vulnerable in terms of security than the loopback approach.

D.2.1 Configuration

The HTML client has to configure AutoVue JavaScript Object according to the communication protocol that it wants to setup with AutoVue. Following are the parameters to be used in order to achieve this configuration:

- **Localhost Port Ranges:** A list of port ranges to try on the client for the loopback connection. The communication will be established on the first free port available within the provided list. This parameter is passed to AutoVue JavaScript Object within the constructor argument list (3rd argument):

```
AutoVue(JNLP_HOST, CODEBASE_HOST, CLIENT_PORTS, INIT_PARAMS,
        ENCRYPT_COOKIES, VERBOSITY, STARTUP_DELAY)
```

- **Rendezvous Servlet Host:** The URL of VueRDVServlet used by the Rendezvous communication. It needs to be deployed on a J2EE-enabled application server and must be protected to prevent denial-of-service attacks. This parameter (RDVServlet) is provided among AutoVue Client parameters (INIT_PARAMS), which are passed to AutoVue JavaScript Object within the constructor argument list (4th argument):

```
INIT_PARAMS['RDVServlet'] = <Server
Host>:<Port>/<Contexte>/servlet/VueRDVServlet
AutoVue(JNLP_HOST, CODEBASE_HOST, CLIENT_PORTS, INIT_PARAMS, ENCRYPT_COOKIES,
VERBOSITY, STARTUP_DELAY)
```

- **Rendezvous Communication Format:** The request content type used by VueRDVServlet. The supported formats are:
 - 'application/json': This is the default format.
 - 'application/x-www-form-urlencoded': This is another format with larger content but useful for deployments using filters that consume the requests streams. For example, by calling `GetParameter()`.

This parameter `RDVFormat` is also provided among AutoVue Client parameters (`INIT_PARAMS`):

```
INIT_PARAMS['RDVFormat'] = ['application/json' |
                             'application/x-www-form-urlencoded']
```

The [Table D–1](#) summarizes how AutoVue JavaScript Object selects the communication protocol based on the parameters passed by the HTML client to its constructor.

Table D–1 Communication Protocol

Rendezvous Servlet Host			
		Provided	Not Provided
localhost port ranges	Provided	Use loopback connection under HTTP protocol, but use the Rendezvous approach under HTTPS protocol	Always loopback connection
	Not Provided	Always use the Rendezvous communication	In this case, you cannot communicate with AutoVue, hence no scripting is needed. Inform the user and start AutoVue in standalone mode (do not send a ticket).

D.2.2 Cookies

Integration solutions often need to pass some of the browser cookies to AutoVue to pass authentication and/or context information.

- More specifically, since the Rendezvous servlet is protected, the Rendezvous solution requires sending the Rendezvous servlet authentication cookies to AutoVue at the start-up in order to allow it communicating with the Rendezvous servlet.
- These cookies must be sent within the JNLP file. So, the JNLP generator must handle them and store them in the JNLP file within the "COOKIES" client parameter, designed for this purpose.
- `VueJNLPServlet` supports a servlet initialization parameter listing the names of the selected cookies and adds them within the "COOKIES" Client parameter to the JNLP file. Therefore, a solution that uses `VueJNLPServlet` must list the Rendezvous servlet authentication cookies within the "Cookies" servlet initialization parameter of `VueJNLPServlet`, in order to use the Rendezvous communication protocol.

- Since cookies carry usually sensitive private information, VueJNLPServlet provides an additional level of security by encrypting them before writing them down on the JNLP file. This encryption is enabled by default but it is configurable. An HTML client can specify to the AutoVue Java Script Object whether to request cookies encryption to VueJNLPServlet through a Boolean parameter passed within the constructor argument list (5th argument):

```
AutoVue(JNLP_HOST, CODEBASE_HOST, CLIENT_PORTS, INIT_PARAMS,
        ENCRYPT_COOKIES, VERBOSITY, STARTUP_DELAY)
```

- When cookies encryption is requested, the HTML Client must provide an RSA Encryption Key-Pair to the AutoVue Java Script Object, using the API provided for this purpose:

```
AutoVue.setEncryptionKeyPair(public_key, private_key)
```

- Oracle distributes VueKeyPairServlet within AutoVue components that can be deployed to generate an RSA Key-Pair. It is provided as a reference implementation for validation and testing purpose. Refer to the "Deploying JNLP Components" section of the Oracle AutoVue Client/Server Installation and Configuration Guide for more information about this component.
 - When cookies encryption is enabled, the HTML client must pass the encryption private encryption key to AutoVue as a client parameter (EncryptionKey):
- ```
INIT_PARAMS['EncryptionKey'] = <Encoded private RSA Key, HEX serialized>
```
- The private encryption key can be safely transmitted to AutoVue within the Client parameters through a loopback connection. However, under the Rendezvous communication protocol, the client parameters are sent to AutoVue through the Rendezvous servlet. Therefore, the encryption private key cannot be safely transmitted to AutoVue using the Rendezvous communication. For that reason, the cookies encryption is only supported under loopback connection protocol. It must be disabled under Rendezvous Communication. The security risk is low though since the session ID has a limited validity period and the JNLP file is automatically deleted by Java after usage.

## D.2.3 Example

This section presents an example achieving the following functionality:

- Use the Rendezvous communication.
- Use the URL form encoding content type for the Rendezvous HTTP requests.
- Do not encrypt cookies since we are using the Rendezvous communication.
- Set a Debug logging level on client side and provide a logging file for it.
- Open a given sample file in AutoVue at start-up.
- Set a custom User Interface.

To use the given VueJNLPServlet, provide the two required servlet initialization parameters within its WEB descriptor, in the WEB configuration file web.xml:

```
<servlet>
 <servlet-name>VueJNLPServlet</servlet-name>
 <servlet-class>com.cimmetry.servlet.VueJNLPServlet</servlet-class>

 <init-param>
 <!-- relative path of autovue.jnlp with respect to server context -->
```

```
 <param-name>URL-Dir</param-name>
 <param-value>/AutoVueClient</param-value>
 </init-param>

 <init-param>
 <!-- Authentication cookies required by the Rendezvous servlet -->
 <param-name>Cookies</param-name>
 <param-value>JSESSIONID;</param-value>
 </init-param>
</servlet>
```

**HTML Client Sample Code:**

```
<script type="text/javascript" src="AutoVueClient/autovue.js"></script>

<script>

 // Deployment URL of the sample
 var basedir = window.location.toString().substring(0, loc.lastIndexOf('/') + 1)

 // JNLP Servlet HOST
 var JNLP_HOST = basedir + 'servlet/VueJNLPServlet'

 // AutoVue Codebase Location
 var CODEBASE = basedir + 'AutoVueClient'

 // Set Client port ranges to null to use Rendezvous communication under HTTP as
 // well
 var CLIENT_PORTS = NULL

 // AutoVue Client Parameters
 var INIT_PARAMS = {}
 INIT_PARAMS['JVUESERVER'] = basedir + 'servlet/VueServlet'
 INIT_PARAMS['RDVServlet'] = basedir + 'servlet/VueJNLPServlet'
 INIT_PARAMS['RDVFORMAT'] = 'application/x-www-form-urlencoded'
 INIT_PARAMS['FILENAME'] = basedir + 'Samples\Desktop-
 Office/Basell_Autovue_Case_Study.pdf'
 INIT_PARAMS['GUIFILE'] = custom.gui
 INIT_PARAMS['VERBOSE'] = 'Debug'
 INIT_PARAMS['LOGFILE'] = C:/TEMP/av_client.log

 // Encrypt the cookies only under HTTP loopback connection
 var ENCRYPT_COOKIES = false

 // Instantiate an AutoVue JavaScript Object
 var myAvApp = new AutoVue(JNLP_HOST, CODEBASE, CL_PRTS, INIT_PARAMS, ENCRYPT_
 COOKIES)

 // Start AutoVue through Java Web Start
 myAvApp = myAvApp.start(onInit, onInitError)
 /* onInit and onInitError are Custom JavaScript callbacks that AutoVue
 JavaScript Object will invoke on initialization success or failure,
 respectively */

</script>
```

## D.2.4 Migration from loopback connection to Rendezvous communication

Moving an HTML client using VueJNLPServlet from loopback connection protocol to a Rendezvous communication one, requires the following changes:

1. Set "CLIENT\_PORTS" parameter to null (not needed), in order to disable completely the loopback connection.
2. Add the Rendezvous authentication cookies to the "Cookies" servlet initialization parameter of VueJNLPServlet. This is required to authenticate AutoVue requests to the Rendezvous servlet.
3. Add the Rendezvous servlet descriptor to the WEB configuration file web.xml.
4. Deploy the Rendezvous servlet on server side.
5. Specify the Rendezvous servlet host on the HTML code side using the RDVSERVLET client parameter within the INIT\_PARAMS argument.
6. If required, specify the Rendezvous requests content-type using the RDVFORMAT client parameter within the INIT\_PARAMS argument as well. JSON format is used by default.
7. Disable cookies encryption setting the parameter "ENCRYPT\_COOKIES" to false (not supported).
8. Remove the servlet VueKeyPairServlet (not needed).



# Troubleshooting

In this chapter we discuss the common troubleshooting issues with AutoVue for Agile PLM.

## E.1 Troubleshooting AutoVue for Agile PLM

This section provides information for troubleshooting your AutoVue for Agile PLM installation.

### E.1.1 Enabling Client Verbosity

To enable detailed logging, you need to set certain parameters, and set the folder where the logs are stored.

#### E.1.1.1 Detailed WebLogic Logs

To enable detailed logging, set STARTMODE to false in \$AGILE\_HOME/agileDomain/bin/startAgile.sh.

Logs will be in \$AGILE\_HOME/agileDomain/servers/\$HOSTNAME-AgileServer.

#### E.1.1.2 File Manager (Tomcat logs)

Edit \$AGILE\_HOME/FileManager/conf/server.xml and uncomment the following valve:

```
<Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs"
prefix="localhost_access_log." suffix=".txt" pattern="%h %l %u %t "%r"%s %b"/>
```

The log will be in \$AGILE\_HOME/FileManager/logs

#### E.1.1.3 File Manager VueServlet Logs

To enable logging for VueServlet, do the following:

- Stop Tomcat <Agile Install>/Filemgr/FileManager/bin/shutdown.sh
- Backup <Agile Install>/Filemgr/agileDomain/applications/webfs/WEB-INF/classes/log.xml
- Replace contents of <Agile Install>/Filemgr/agileDomain/applications/webfs/WEB-INF/classes/log.xml with

```
<?xml version="1.0" encoding="UTF-8" ?>
<!--!DOCTYPE log4j:configuration SYSTEM "log4j.dtd"-->
```

```

<log4j:configuration xmlns:log4j='http://jakarta.apache.org/log4j/'>

 <appender name="webfs" class="org.apache.log4j.ConsoleAppender">
 <layout class="org.apache.log4j.PatternLayout">
 <param name="ConversionPattern" value="%d%> %p%> %m\n"/>
 </layout>
 </appender>

 <appender name="viewer" class="org.apache.log4j.ConsoleAppender">
 <layout class="org.apache.log4j.PatternLayout">
 <param name="ConversionPattern" value="%d%> %p%> %m\n"/>
 </layout>
 </appender>

 <appender name="metafile" class="org.apache.log4j.ConsoleAppender">
 <layout class="org.apache.log4j.PatternLayout">
 <param name="ConversionPattern" value="%d%> %p%> %m\n"/>
 </layout>
 </appender>

 <appender name="cimmetry" class="org.apache.log4j.ConsoleAppender">
 <layout class="org.apache.log4j.PatternLayout">
 <param name="ConversionPattern" value="%d%> %p%> %m\n"/>
 </layout>
 </appender>

 <category name="com.agile.webfs" additivity="false">
 <priority value="debug" />
 <appender-ref ref="webfs" />
 </category>

 <category name="com.agile.viewer" additivity="false">
 <priority value="debug" />
 <appender-ref ref="viewer" />
 </category>

 <category name="com.agile.metafile" additivity="false">
 <priority value="debug" />
 <appender-ref ref="metafile" />
 </category>

 <category name="com.cimmetry" additivity="false">
 <priority value="debug" />
 <appender-ref ref="cimmetry" />
 </category>
</log4j:configuration>

```

- To make the change permanent, update webfs.war:
  - cd <Agile Install>/Filemgr/agileDomain/applications
  - Extract the log.xml from webfs.war:
 

```
<Agile Install>/Filemgr/jdk/bin/jar -xvf webfs.war
WEB-INF/classes/log.xml
```
  - Update log.xml
  - Update webfs.war: <Agile Install>/Filemgr/jdk/bin/jar -uvf webfs.war
 WEB-INF/classes/log.xml

- Restart Tomcat <Agile Install>/Filemgr/FileManager/bin/start\_up.sh.

The output log with VueServlet data is in <Agile Install>/Filemgr/FileManager/logs/catalina.out

#### E.1.1.4 AutoVue Detailed Client/Server Logs

The AutoVue Client/Server Deployment allows verbose logging. This is applicable when integrated with Agile PLM. Following are the two stages to enable client logging when using the AutoVue Client/Server Deployment:

1. Modify user's AutoVue profile for VERBOSE debugging:

- Edit <Agile Install>/avs/bin/profiles/<UserID>.ini by adding entry:

[Logging]

VERBOSE=DEBUG

2. Set up Java logging:

- a. Access the Java Control Panel:

On Windows: From **Start**, select **Control Panel**, then **Java**.

On Linux: Navigate to the JRE installation directory (meaning the JRE used by the web browser on the particular client machine), then into the 'bin' folder, then type ./ControlPanel.

- b. In the Java Control Panel that appears, click **Advanced** tab. In the Debugging option, select the check boxes-**Enable tracing** and **Enable logging**.

- c. In the Java Control Panel, click **Java** tab, and then select **View**. In the Java Runtime Environment Settings window that appears, in the "Runtime Parameters" text field enter the following line:

```
-Djavapugin.trace=true-Djavapugin.trace.option=basic|net|cache|security|ext|liveconnect|temp-Djavapugin.logging=true
```

- d. Click **OK** in the Java Runtime Environment Settings window, and click **Apply** in the Java tab.

- e. After launching the Applet and doing the required tests the tracing and logging files will be dumped in the windows user profile. For instance, if the client is using:

Windows XP: The files will be dumped in c:\Documents and Settings\<User Name>\Application Data\Sun\Java\Deployment\log

Windows 7: The files will be dumped in C:\Users\User\_Name\AppData\LocalLow\Sun\Java\Deployment

---

**Note:** For more information, refer to the KM Note-Doc ID [752589.1](#)

---

### E.1.2 Errors Which May Occur While Using the AutoVue Server

Common errors that occur while using the AutoVue server are listed in the table below along with solutions.

While the Web or Java client tries to start the AutoVue server:

**Table E-1 Problem and solution while trying to start the AutoVue server**

Error	Reasons	Solution
jVue not found	Viewer Content URL in Admin > Server Settings > Locations > File Manager is incorrect.  Incorrect client applet alias specified instead of mandatory "jVue."	Confirm that the Viewer Content URL is accessible and the jVue.jar file is listed.
An error occurred while connecting to the server. Restart the applet?	The AutoVue server or File Manager is down.  Incorrect View Server URL in Admin > Server Settings > Locations > File Manager.  The AutoVue server is configured incorrectly, or is unavailable	Confirm that the Viewer Server URL is accessible and returns a successful connection status.

When the Web Client or the Java Client tries to connect to a File Manager:

**Table E-2 Problem and solution while connecting to a File Manager**

Error	Reason	Solution
File not found	The File Manager is down.  The VueServer.ini file points to incorrect File Manager.	Confirm that the File Manager is up and running.
File not found	File not present in the File Vault	No solution

If an attachment uses special fonts:

**Table E-3 Problem and Solution if attachment used special fonts**

Error	Reason	Solution
Fonts may not appear correctly or a file error may occur or an incorrect font substitution may occur with viewing or printing the attachment.	The computer where the Viewer is installed does not have the fonts installed.	The computer where Viewer is installed must have those special fonts installed locally.  For fonts other than the ones used by the operating system, the user's profile should include a reference to the location of the fonts in the XFONTPATHS parameter.

### E.1.3 Changing Port Numbers

Some AutoVue server port numbers can be changed in the event of a port conflict. It is important to change the port number in all referenced locations.

- Port 5099 is used for communication between the AutoVue server and its clients.

For the AutoVue server, go to the \AVS\bin folder and open the jvueserver.properties file. Locate the following port information:

# \* Socket port number

jvueserver.socket.port=5099



You can change port 5099 to another available port number.

If you are changing the AutoVue server port, update the port number in the `VueServlet` section of the `web.xml` file located at `AgileHome\agileDomain\applications\webfs.war\WEB-INF\` on all viewer clients. Extract `webfs.war` file from `AgileHome\agileDomain\applications`, then open the `WEB-INF` folder for `web.xml` file.

- Port 2001 is used for RMI.

For the AutoVue server, go to the `\AVS\bin` folder and open the `javueserver.properties` file. Locate the following port information:

# \* RMI Registry port number

`javueserver.rmi.port=2001`

You can change port 2001 to another available port number.

---

**Note:** Refer to the AutoVue Installation and Configuration Guide for more information on port numbers.

---

## E.2 Verify AutoVue Server

To verify the installation, do the following:

- Check the File Manager configuration. Refer to the "Testing File Manager Configuration Page" chapter of the *Oracle AutoVue for Agile PLM Testing Guide*.



This appendix includes frequently asked questions pertaining to the installing and configuring of AutoVue for Agile PLM.

### F.1 Linux

**Q:** When installing AutoVue on a Linux 64-bit OS, the AutoVue installer does not show up and I receive the following notice:

Installer User Interface Mode Not Supported The installer cannot run in this UI mode. To specify the interface mode, use the `-i` command-line option, followed by the UI mode identifier. The valid UI modes identifiers are GUI, Console, and Silent.

**A:** This issue may be attributed to the fact that you are missing the following 32-bit packages:

- libXmu.i686
- libXp.i686
- libXtst.i686
- libXrender.i686

To resolve this issue, you must install these packages.



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

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

## **G.1 General AutoVue for Agile PLM 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 for Agile PLM Community**

---

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

---

## **G.4 Sales Inquiries**

---

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

---

