

Oracle® AutoVue Client/Server Deployment

Installation and Configuration Guide

Release 21.0.1.4

E84697-03

January 2018

E84697-03

Copyright © 2010, 2018, Oracle and/or its affiliates. All rights reserved.

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

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

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

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

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

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

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

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

Contents

Preface	ix
Audience	ix
Documentation Accessibility	ix
Related Documents	ix
Conventions	ix
 1 Introduction	
 2 AutoVue Hardware and System Requirements	
2.1 Hardware Requirements	2-1
2.2 System Requirements	2-2
2.2.1 Server Platforms	2-2
2.2.2 Client Platforms	2-3
2.2.3 Application Servers	2-4
2.2.4 Integrations	2-4
 3 Prerequisites	
3.1 Pre-requisites for Installing AutoVue	3-1
3.1.1 Windows Prerequisites	3-2
3.1.2 Linux Prerequisites	3-2
3.2 Additional Guidelines for Security	3-3
 4 Installation Checklist	
4.1 Deployment Scenarios	4-1
4.2 Additional Recommendations	4-1
 5 Installing and Upgrading AutoVue	
5.1 Upgrading from earlier versions	5-1
5.1.1 Upgrading from AutoVue Version 20.2.x/21.0.0	5-1
5.1.2 Upgrading from AutoVue Version 20.1.x or Earlier	5-2
5.2 Secure Installation of AutoVue	5-3
5.3 Verifying AutoVue Server Installation	5-6
5.3.1 Verifying AutoVue Server Startup	5-6
5.3.2 Troubleshooting AutoVue Server Startup Issues	5-7

5.4	Post-Installation Instructions	5-8
5.4.1	Configuring SSL	5-9
5.4.2	Configuring User Authentication.....	5-9
5.4.3	Deploying VueServlet	5-9
5.4.3.1	Creating a WAR for the VueServlet.....	5-10
5.4.3.2	Deploying the VueServlet	5-11
5.4.3.3	Verifying VueServlet Deployment.....	5-11
5.4.3.4	Troubleshooting VueServlet Deployment	5-12
5.4.4	Deploying JNLP Components	5-12
5.4.4.1	Deploying VueJNLPServlet	5-13
5.4.4.2	Deploying VueKeyPairServlet.....	5-15
5.4.5	Installing AutoVue Client Components	5-15
5.4.5.1	Configuring AutoVue Client for Secure Socket Layer (SSL).....	5-18
5.4.5.2	Verifying AutoVue Client	5-19
5.4.5.3	Troubleshooting AutoVue Client.....	5-19
5.4.5.4	Configuring Online Help	5-20
5.5	Linux-Specific Post-Installation Instructions	5-20
5.6	Verifying Communication with AutoVue.....	5-20
5.6.1	Troubleshooting Communication Issues	5-21
5.7	Deploying AutoVue in Virtualized Environments.....	5-22

A Installing in an Integrated Environment

A.1	Installing AutoVue Client Components in an Integrated Environment.....	A-1
A.2	Installing the VueServlet in an Integrated Environment	A-2
A.3	Verifying your Integration.....	A-2

B Configuring AutoVue Server Farm

B.1	Setting Up AutoVue Server Load Balancing.....	B-2
B.1.1	Symbol Libraries	B-2
B.2	Verifying AutoVue Server Load Balancing.....	B-2
B.2.1	Troubleshooting AutoVue Server Load Balancing.....	B-3
B.3	Configuring VueServlet Load Balancing.....	B-3

C Failover and Disaster Recovery

C.1	AutoVue Server Configuration for Failover	C-1
C.2	AutoVue Failover Configuration on the VueServlet	C-1
C.3	Failover for the VueServlet.....	C-2
C.4	Failover for AutoVue client components	C-2
C.5	Verifying Failover Configuration	C-2

D Integrating With a DMS

D.1	Multiple Document Repositories.....	D-1
D.2	Creating a Stamp Template	D-2
D.2.1	Choosing a Background Image.....	D-2
D.2.2	Determining Stamp Attributes and Permissions	D-3
D.2.3	Interacting with Stamps.....	D-5

D.2.4	Identifying the Stamp Layout	D-5
D.2.5	Designing a Stamp Layout	D-5
D.2.5.1	Designing the Stamp	D-5
D.2.5.2	Setting the Stamp Attributes	D-7
D.2.6	Configuring Stamp Templates	D-8
D.2.7	Configuring Stamp with Your Integration	D-11
D.3	Verifying Your Integration	D-12

E Configuring for Real-Time Collaboration

E.1	Default Collaboration Configuration	E-1
E.2	Distributed Geographies Configuration	E-1
E.3	Distributed DMS Configuration	E-1

F Starting the AutoVue Server

F.1	Starting AutoVue on Windows	F-1
F.2	Starting AutoVue on Linux	F-2
F.3	Shutting Down the AutoVue Server	F-3
F.4	Running the AutoVue Server as a Service	F-3
F.4.1	On Windows OSes	F-3
F.4.1.1	Starting and Stopping the Service	F-4
F.4.2	On Linux OSes	F-4

G Monitoring the AutoVue Server

G.1	AutoVue Server Console	G-1
G.2	Usage Monitoring	G-2
G.3	Logging for the AutoVue Server	G-3
G.3.1	Log4j Appenders	G-3
G.3.1.1	Output Layout	G-4
G.3.2	Logger Information	G-5

H Customizing the AutoVue Client

H.1	AutoVue Client Parameters	H-1
H.1.1	COLLABORATION Client Parameter Examples	H-8
H.1.2	Scripting AutoVue Client	H-9
H.1.2.1	Basic Client	H-9
H.1.2.2	Advanced Scripting Functionality	H-9
H.2	Customizing the GUI	H-10
H.2.1	Choosing the GUI File	H-10
H.2.2	Modifying the GUI	H-10
H.2.2.1	Role-Based GUI	H-10
H.2.2.2	Structure and Syntax of GUI Files	H-11
H.2.2.3	GUI Configuration Syntax	H-11
H.2.2.4	Control Names	H-12
H.3	Customizing the Example AutoVue Client Pages	H-16

I AutoVue Server Configuration Options

I.1	Authentication Option	I-1
I.2	AutoVue Host Name Option	I-1
I.3	RMI and Socket Ports Options	I-1
I.4	Process Pool Size Option.....	I-2
I.5	Proxy Connection Options	I-3
I.6	Streaming Files Options	I-3
I.7	DMS Options	I-5
I.8	Collaboration Options.....	I-5
I.9	log4j and Diagnostics Options	I-6
I.10	Modified With JavaScript Option.....	I-7
I.11	Reboot Option.....	I-7
I.12	DocServer Timeout Option.....	I-7
I.13	Recovery Attempt Option.....	I-8
I.14	DLL Version Option	I-8
I.15	File Format Information Option	I-8
I.16	Global User Options	I-8
I.17	Markup Options.....	I-9
I.18	Online Help Options	I-10
I.19	Memory Optimization.....	I-10
I.20	Linux-Specific Options.....	I-11
I.20.1	Preload Java Class Option	I-12
I.20.2	Xvfb Options.....	I-12
I.20.3	WINE Options	I-12
I.20.4	OEM Copyright Notice	I-13
I.21	VueServlet Configuration Options.....	I-13

J Deploying the VueServlet on Application Servers

J.1	Generic Steps to Deploy the WAR File	J-1
J.2	Deploying the WAR File with WebLogic.....	J-1
J.3	Deploying the VueServlet with Tomcat	J-2
J.4	Deploying the VueServlet on non-J2EE Application Servers.....	J-2
J.4.1	Setting up the VueServlet	J-2
J.4.2	Deploying on Jetty	J-3

K Non-Interactive Installations

K.1	Installation.....	K-1
K.1.1	Sample Silent Installation for Windows OSes	K-1
K.1.2	Sample Silent Installation for Linux OSes	K-2
K.1.3	Silent Installation Parameters	K-2
K.2	Uninstallation	K-4

L Configuring AutoVue Plug-in for Enterprise Manager

L.1	Prerequisites.....	L-1
L.2	Installing the Plug-in	L-1

M Samples and API Examples Included with AutoVue

M.1	API Examples	M-1
M.2	Sample Files	M-2

N FAQ

N.1	Linux	N-1
-----	-------------	-----

O Feedback

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

Preface

The Oracle AutoVue Client/Server Deployment Installation and Configuration Guide describes how to install and configure Oracle AutoVue and its associated components.

For the most up-to-date version of this document, go to the AutoVue Documentation Web site on the Oracle Technology Network (OTN) at

<http://www.oracle.com/technetwork/documentation/autovue-091442.html>.

Audience

The Oracle AutoVue Client/Server Deployment Installation and Configuration Guide is directed at any user whose task is the installation and administration of Oracle AutoVue.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit

<http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

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

- *Oracle AutoVue Client/Server Deployment Planning Guide*
- *Oracle AutoVue Client/Server Deployment Viewing Configuration Guide*
- *Oracle AutoVue Client/Server Deployment Security Guide*
- *Oracle AutoVue Client/Server Deployment User's Manual*

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.
[root directory] \ [sub directory]	In Windows and Linux OSes, directory hierarchy is written with backward slashes (\) and forward slashes (/), respectively. In this document, unless mentioned otherwise, directory hierarchy for Windows and Linux OSes are written with the backward slash.
<angular brackets>	Indicates required entries but are not to be included in the entered information.
{curly braces}	Indicates mandatory information.
[square brackets]	Indicates optional syntactical elements.
	Indicates an either-or type of choice.
...	Indicates that information may be repeated.

Introduction

AutoVue is Oracle's suite of Enterprise Visualization solutions, which are designed to view, digitally annotate and collaborate on any digital information in an organization. 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).

The Client/Server Deployment of AutoVue has AutoVue installed on a server, to which client machines connect to access and view documents. The Client/Server deployment provides a complete, open and standards-based set of integration tools that allows customers to tie AutoVue to any enterprise applications to provide users with a consistent view of data and business objects and expand workflow automation to document-based processes.

This document provides instructions for installing and configuring AutoVue Client/Server deployment. Refer to the Planning Guide for information on how to plan your AutoVue deployments.

AutoVue Hardware and System Requirements

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

2.1 Hardware Requirements

Table 2–1 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 2–1 (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.

2.2 System Requirements

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

2.2.1 Server Platforms

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

Table 2–2 Oracle-Certified Operating Systems

Platform	AutoVue for Agile PLM ¹	AutoVue Client/Server Deployment ¹	AutoVue Desktop Deployment
Windows Server 2012 R2, 2012 —64-bit (AutoVue running in 32-bit mode)	x	x	
Windows Server 2008 R2—64-bit (AutoVue running in 32-bit mode)	x	x	
Windows 10, 8.1, 7—64-bit (AutoVue running in 32-bit mode)			x
Oracle Linux 6.X (x86_64) ² and 7.X (x86_64) ² 64-bit (AutoVue running in 32-bit mode)	x	x	

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

Platform	AutoVue for Agile PLM ¹	AutoVue Client/Server Deployment ¹	AutoVue Desktop Deployment
Red Hat Enterprise Linux 6.X (x86_64) ² , and 7.X (x86_64) ² 64-bit (AutoVue running in 32-bit mode)	x	x	

¹ The installation requires about 400MB of free space. Additional space will be required by AutoVue for storing other data such as streaming files and markups.

² 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.

2.2.2 Client Platforms

The following platforms are certified for the AutoVue client.

Table 2–3 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 ¹ — 64-bit <ul style="list-style-type: none"> ■ Internet Explorer 11 ■ Chrome 46 and up ■ Firefox ESR 52 and up Note: To open the applet version of AutoVue, either IE11 or Firefox ESR 52 is required.	x	x
Apple OS X 10.11 <ul style="list-style-type: none"> ■ Safari 10.0²³ ■ Safari 9.0²³ ■ Safari 8.0²³ ■ Firefox ESR 52 and up 	x	x
Java Virtual Machine		
<ul style="list-style-type: none"> ■ Java 7 update 45 and higher - 32-bit and 64-bit 	x	x
<ul style="list-style-type: none"> ■ Java 8 update 11 and higher - 32-bit and 64-bit 	x	x

¹ Microsoft Edge is not supported at this time due to known issues with the browser, which impacts AutoVue.

² 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>

³ 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>

2.2.3 Application Servers

The VueServlet has been certified on the following application servers:

Table 2–4 Oracle-Certified Application Servers

Platform	AutoVue for Agile PLM	AutoVue Client/Server Deployment
WebLogic 11g.x and up		x
WebSphere 7.1 and up		x
Tomcat 7.x and up	x	x
Jetty 7.0 and up		x

2.2.4 Integrations

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

Table 2–5 Oracle-Certified Integrations

Integrations	AutoVue for Agile PLM	AutoVue Client/Server Deployment
Agile PLM 9.3.6, 9.3.5, 9.3.4, and 9.3.3	x	
VueLink 19.3.2 for Documentum ¹		x
VueLink 20.1 for UCM ²		x

¹ New VueLink for Documentum certifications made after this release of AutoVue are listed in Oracle Support Document 1425804.1 (VueLink for Documentum Certification Matrix) found at: <https://support.oracle.com/epmos/faces/DocumentDisplay?id=1425804.1>

² New VueLink for UCM certifications made after this release of AutoVue are listed in Oracle Support Document 1383416.1 (Oracle AutoVue VueLink for UCM Certification Matrix) found at: <https://support.oracle.com/epmos/faces/DocumentDisplay?id=1383416.1>

Prerequisites

This chapter discusses the prerequisites of installing AutoVue.

3.1 Pre-requisites for Installing AutoVue

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

- 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 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/jvuw_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). 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 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 on a Linux OS, it is recommended that you have basic knowledge of Linux and its administration.

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

- Install the core fonts from <http://corefonts.sourceforge.net/>. These fonts are required for viewing files that use some specific fonts.
- Ensure that the pre-requisite libraries for Linux are installed. 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>
- The following must be installed on the Linux machine:
 - The Xvfb, libXfont and Mesa packages. The Mesa packages must be 32-bit.
 - The following 32-bit libraries must be installed on the 64-bit machine:
 - * `libGL.so.1` (yum install mesa-libGL.i686)
 - * `libGLU.so.1` (yum install mesa-libGLU.i686)
 - * `libz.so.1` (yum install zlib.i686)
 - * `libbz2.so.1` (yum install bzip2-libs.i686)
 - * `libXt.so` (yum install libXt.i686)
 - * `libXp.so` (yum install libXp.i686)
 - * `libXmu.i686`
 - * `libXtst.i686`
 - * `libXrender.i686`

Note: The libXrender.i686 lib is required in order to get the Graphical interface when installing.

- Make sure to run the YUM update command for a package before installing it.

```
# yum update <package_name/s>
```

- Install the latest Xvfb by running the following command:

```
# yum install Xvfb
```

- Install the latest Mesa package (recommended version is 6.5.1 or later) by running the following command:

```
# yum install mesa-libGLU.i686
```

Note: In the event you want to use an earlier version of Mesa, it is acceptable to use the version that is included in the repository of the supported Linux distribution.

- Install the 32-bit libXp packages, if not already installed.
- Install the WINE RPM package. To do so, install wine-av-20040914-24.i386.rpm from <http://oss.oracle.com/AutoVue>.

- If you have an older version of WINE, you need to uninstall it and then install the package that is certified with your version of Oracle AutoVue.

- Install WINE as a root user by running the following:

```
#rpm -i wine-av-20040914-24.i386.rpm
```

Note: This version of WINE is installed in the */usr/av* directory.

Note: Refer to Doc ID 2166707.1 if you get a conflict error when installing wine on Linux 7 machine.

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 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.

Installation Checklist

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

AutoVue 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.

4.1 Deployment Scenarios

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

Table 4–1 *Deployment Scenarios*

Deployment Scenario	Jump to Chapter
Planning for failover and disaster recovery	See Failover and Disaster Recovery
Integrating with a Document Management System (DMS) ¹ .	See Integrating With a DMS .
Real-Time Collaboration across Firewalls	See Configuring for Real-Time Collaboration .
Customizing GUI/ AutoVue Client	See Customizing the AutoVue Client .
Usage Logging/Server Logging	See Monitoring the AutoVue Server .
AutoVue Plug-in for Oracle Enterprise Manager	See Configuring AutoVue Plug-in for Enterprise Manager .

¹ In this document, the terms document repository and DMS are used interchangeably to refer to Document Management Systems (DMS), Product Lifecycle Management (PLM) and Enterprise Resource Planning (ERP).

4.2 Additional Recommendations

Following are recommendations to ensure that the AutoVue 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.

Installing and Upgrading AutoVue

This chapter describes how to install AutoVue on Windows and Linux OSes.

Important:

- On Windows operating systems, make sure AutoVue installer is run as an Administrator. If User Access Control (UAC) is enabled, select yes for the prompt that asks you if you want to run as Administrator. If UAC is disabled, right-click the installer, then select **Run as Administrator** from the RMB.
 - If you want to install AutoVue in non-interactive mode, refer to [Non-Interactive Installations](#).
-

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

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

5.1 Upgrading from earlier versions

The following sections discuss upgrading AutoVue from earlier versions.

5.1.1 Upgrading from AutoVue Version 20.2.x/21.0.0

If you are upgrading from AutoVue 20.2.x/21.0.0, you do not need to run the uninstaller before you install AutoVue 21.0.1. You can just run the installer for AutoVue 21.0.1. The installer detects if AutoVue 20.2.x/21.0.0 is installed on your machine. If it is installed, the installer backs up required data, uninstalls version 20.2.x/21.0.0 and then installs 21.0.1 to the same location. If you already have 20.1.x/20.2.x, with codebase (client JAR files) set in Jetty, the installer will upgrade to AutoVue 21.0.1.

- If installing on a Linux OS, make sure you have the correct version of WINE installed. The version of WINE compatible with AutoVue 21.0.1 is wine-av-20040914-24.i386.rpm. You can download this version of WINE from <http://oss.oracle.com/AutoVue>.

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

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 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

Note: If AutoVue is not integrated with a DMS/PLM, the stamp icon will be grayed out, and stamps cannot be used.

- Intellistamps and definitions
- On Linux, changes made to <AutoVue Install Root>/config/jvuw_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.
- Changes made to markup policy file – markuppolicy.xml are migrated to the newer version of AutoVue.
- 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 21.0.1 GUI file.
- The AutoVue installer does not install client components with the Web server. You must manually install updates for the client-side components. You must manually copy all mandatory JAR files to the docroot folder referenced by the CODEBASE parameter of the Web pages that launches the AutoVue client. For information on which files to copy, refer to [Installing AutoVue Client Components](#).

5.1.2 Upgrading from AutoVue Version 20.1.x or Earlier

If you are upgrading AutoVue from version 20.1.x or earlier, you must manually move your configuration settings from your version to AutoVue 21.0.1. You must first uninstall any service packs that are installed for that version of AutoVue and then uninstall your previous version before installing the new version of AutoVue. Before you uninstall, you must backup all required data. After you install 21.0.1, you must

migrate your past data to 21.0.1. 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 21.0.1 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 Install Root>\bin directory) should be copied over manually to the new version

Note: If AutoVue is not integrated with a DMS/PLM, the stamp icon will be grayed out, and stamps cannot be used.

- Markup files, if markups are being managed by the AutoVue server (located in the <AutoVue 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 Install Root>\bin\Symbols directory) should be backed up and copied over to the new version of AutoVue.
- If `MarkupPolicy.xml` located in <AutoVue 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, backup the `javuew_config` file (located in the <AutoVue 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.

5.2 Secure Installation of AutoVue

The installation process of AutoVue consists of running the installer as well as performing manual post-installation steps. Note that the AutoVue installer provides a secure installation of the AutoVue server by default.

Important: Keep note of the following:

- Shutdown all applications (including AutoVue) before you run the installer for the AutoVue server.
 - When upgrading, if the installer prompts you to reboot the machine before or after the uninstallation, you must reboot the machine in order to get a successful installation. If you fail to reboot when prompted, the AutoVue 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. In addition, the operating systems should be the same on each server.
-

To install the AutoVue server, do the following:

1. Download the Oracle AutoVue Media Pack and extract its contents.
2. Run the AutoVue installer executable:



Windows OS: The installer is *InstallClientServer.exe*.

Linux OS: The installer is *InstallClientServer_lin.bin*.

Note: You might need to grant execute permissions to the installer binary on Linux. To do so, run *chmod +x InstallClientServer_lin.bin*.

3. Select a language from the installation dialog and then click **OK**.
4. Click **Next** to begin installation.
5. Click an installation set icon and then click **Next**:

Table 5–1 Installation Sets

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

If you have selected the **Custom** install set, do the following:

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

Table 5–2 Installation Feature Options

Option	Description
Program Files	Installs Oracle AutoVue. The option is selected by default.
User Documentation	Installs AutoVue end-user documentation. The option is selected by default.
Example Client Application	Installs the demo Web site pages.

Table 5–2 (Cont.) Installation Feature Options

Option	Description
Sample Files	Installs drawing sample files.
API Examples	Installs examples of how Oracle AutoVue features can be added to third-party applications using APIs.

If you have selected the **Standard** install set, proceed to Step 6.

6. For Windows OS installations, select one of the following locations to create shortcuts and then click **Next**.

Table 5–3 Installation Locations

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

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

7. Specify a host name or IP address for the AutoVue server and then click **Next**.

Note: The hostname cannot include an underscore (_) character. You must set the correct static or resolvable IP address or hostname or fully-qualified hostname (FQDN) in the file `jvueserver.properties`.

Example: `hostname1.domain.com`

8. Specify the authentication mechanism between the AutoVue server and the client and then click **Next**.

Table 5–4 Authentication Mechanism

Options	Description
Kerberos (JAAS)	Configures AutoVue with Kerberos authentication protocol.
Configure Later (Manual Configuration)	Authentication will not be configured. If needed, you have the option to configure after you complete the installation. For more information, refer to Post-Installation Instructions .

If you selected **Kerberos (JAAS)**, perform the following:

- a. Specify the security realm (`java.security.krb5.realm`) and Kerberos Key Distribution Center (`java.security.krb5.kdc`) for the Kerberos protocol and then click **Next**.

If you selected **Configure Later (Manual Configuration)**, proceed to the next step.

9. Specify how to configure Secure Socket Layer (SSL) and then click **Next**.

Table 5–5 Configure Secure Socket Layer

Options	Description
Configure SSL with a CA Certificate	Configures SSL with a Certificate Authority (CA) certificate.
Configure SSL with a generated self-signed certificate	Configure SSL with an automatically generated self-signed certificate (non-CA certificate).
Configure Later (Manual Configuration)	SSL will not be configured. If needed, you must configure after you complete the installation. For more information, refer to Post-Installation Instructions .

If you selected **Configure SSL with a CA Certificate**, perform the following steps:

- a. Select the CA certificate file and then click **Next**.
- b. Select the Identity JKS Keystore file and then click **Next**.
- c. Specify the Identity JKS Keystore password (minimum six characters) and then click **Next**.

If you selected **Configure SSL with a generated self-signed certificate**, perform the following steps:

- a. Specify the Identity JKS Keystore password (minimum six characters) and then click **Next**.

If you selected **Configure Later (Manual Configuration)**, proceed to the next step.

10. Review the pre-installation summary and then click **Install**.

The AutoVue server is installed in the specified directory. If there are any warnings or errors, refer to the installation log file, `Oracle_AutoVue_installLog_<Date>_<Time>.log`, located in the `<AutoVue Install Root>\uninstall\Logs`.

11. Follow the steps discussed in the [Post-Installation Instructions](#).

Note: For information on registering and running AutoVue as a service, refer to [Running the AutoVue Server as a Service](#).

5.3 Verifying AutoVue Server Installation

This section discusses how to verify AutoVue Server startup, and troubleshooting the issues that come up while starting AutoVue Server.

5.3.1 Verifying AutoVue Server Startup

Start the AutoVue server:

- On Windows Operating systems, run *Start AutoVue Server* from the *Oracle AutoVue* programs shortcut.
- On Linux Operating systems, go to `<AutoVue Install Root>/bin` and run `./jvueserver`. The AutoVue server console should start up and the P, 1, 2, 3, 4 and M buttons should turn green.

Note: If you have a firewall enabled, a prompt may appear asking you to block or unblock the AutoVue server executable. Select **Unblock**.

If any of the P, 1, 2, 3, 4 and M buttons stay red or yellow, refer to the troubleshooting steps in [Troubleshooting AutoVue Server Startup Issues](#).

5.3.2 Troubleshooting AutoVue Server Startup Issues

If the AutoVue console does not start up or if any of the buttons on the server console do not turn green, review the following trouble-shooting pointers. Error messages are written to the log4j-roll*.log file in <AutoVue Install Root>\bin\Logs. Refer to the logs to determine the specific cause for the issue.

- On Linux operating systems:
 - Ensure that your Linux terminal is properly configured for graphics display. When the terminal does not support graphics, AutoVue server will start up, but the console will not appear. The message you would see on the console in this case will be something like "No display defined; console will not be started". To start up the console separately, you can run ./jvueserver_debug -u.
 - If there is a fatal error message for Xvfb make sure that xvfb.new=1 is set in jvueserver.properties (for Linux 6 or later).
 - If you see a fatal error message for Xvfb in the log, it indicates that there are issues starting up Xvfb. Possible reasons are that Xvfb is not installed correctly or the user account running AutoVue does not have permissions to start Xvfb or there is a port conflict for the Xvfb port. To resolve this issue, ensure that Xvfb is correctly installed and that the user account running AutoVue has permissions to start Xvfb. If there is a port conflict, try modifying the port by modifying the port in the xvfb.display parameter in jvueserver.properties.
 - If you have an incorrect version of WINE or if you do not have WINE installed, AutoVue server will not start up. The following message appears in the log file: ". /jvueserver: could not locate WINE server should be at /usr/av/bin/wineserver...". Uninstall any previous version of WINE and install the version that is compatible with the version of AutoVue server.
 - If any of the pre-requisite libraries are missing from your installation, AutoVue server will not start up. If you see a message "Failed to initialize preloader class, aborting..." in the log, you must ensure that all pre-requisite libraries are installed on the machine.
- If the user account running AutoVue server does not have permissions to write into the AutoVue installation directory, AutoVue server will not start up. You will see error messages that say "Permission denied" in the AutoVue server logs. Ensure that the user account running AutoVue has write permissions into the AutoVue installation directory.

Note: On Linux, if you accidentally run AutoVue as a super-user, AutoVue will create/update files and set permissions as this super-user. If you later run AutoVue as a normal user, AutoVue server will not start up since the user does not have permissions to the files created by the super-user.

- If the host name of the AutoVue server changed since you installed AutoVue, the AutoVue server will not be able to start up. The message you will see in the log in this scenario is "Connection refused to host". Update `javueserver.properties` and set the correct server name in the parameter `javueserver.hostname`. Similarly, if you installed AutoVue server using the IP address and the IP address changed after you installed AutoVue, you must set the correct IP address in parameter `javueserver.hostname` in `javueserver.properties`.

Note: You must set the correct (specify static or resolvable) IP address or hostname or fully-qualified hostname (FQDN) in the file `javueserver.properties`.

- AutoVue server needs RMI ports in order to run correctly. If the RMI ports required by AutoVue are used by other applications, the server will not start up. In this instance, you will see "java.rmi.connection" exceptions in the log. AutoVue needs the RMI port specified in property `javueserver.rmi.port` and `n` consecutive ports following this port, where `n` is the AutoVue process pool size. If the RMI ports required by AutoVue are not available, change the parameter `javueserver.rmi.port` to point to a port that is available. Ensure that this port and `n` consecutive parts following this port are available to AutoVue.

Refer to [AutoVue Server Configuration Options](#) for a list of all AutoVue server configuration options in `javueserver.properties`.

If you verified all the above and AutoVue server still does not start up, contact Oracle customer support for help with trouble-shooting your AutoVue server startup issues.

5.4 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 Installation Folder>\jre\lib\security\cacerts`.
- 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>`
- The following points are only relevant if you plan to use Jetty for testing purposes or actual deployment.
 - In `<AutoVue Installation Folder>\bin\jetty\etc\webdefault.xml`, `EnableSSL` is set to `TRUE`.
 - In `<AutoVue Installation Folder>\bin\jetty\etc\jetty-ssl.xml`, `keystore` and `password` parameters have been updated.
 - In `<AutoVue Installation Folder>\bin\jetty\bin\startJetty.bat` file, the `CLASSPATH` parameter is updated to include `"%JETTY_DIR%\etc\jetty-ssl.xml"`.

5.4.1 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*.

5.4.2 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:

```
javueserver.authenticator=com.cimmetry.javueserver.JAASAuthenticator
```

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*.

5.4.3 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. For information on deploying VueServlet in an integrated environment, refer to [Installing the VueServlet in an Integrated Environment](#).

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.

Refer to [Deploying the VueServlet on Application Servers](#) for instructions for deploying the VueServlet with WebLogic, WebSphere, Tomcat, and Jetty.

5.4.3.1 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:\csiwar

2. In the folder C:\csiwar, create a sub-directory WEB-INF.
3. In WEB-INF, create a directory lib: C:\csiwar\WEB-INF\lib
4. Copy vueservlet.jar from <AutoVue Install Root>\bin to C:\csiwar\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="ISO-8859-1"?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
2.2//EN" "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
```

- 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 DTD definition. 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*. Refer to [VueServlet Configuration Options](#) for more information.
9. 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:\csiwar), use the following command:

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

5.4.3.2 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.

Refer to the instructions available with your J2EE application server for WAR deployment information and [Deploying the VueServlet on Application Servers](#).

5.4.3.3 Verifying VueServlet Deployment

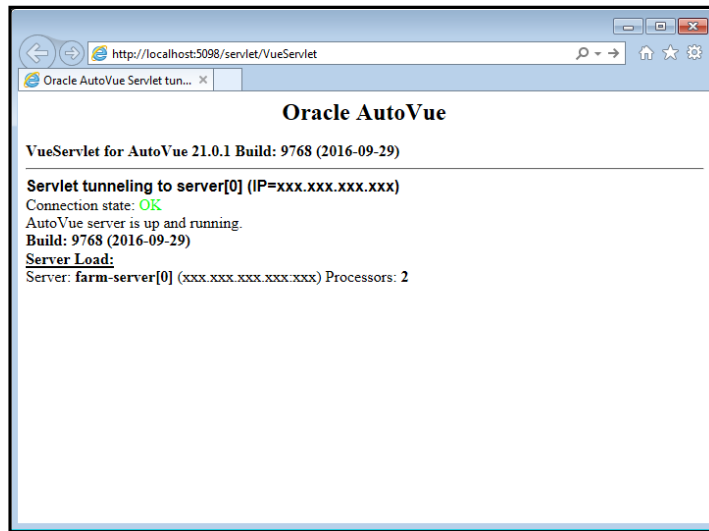
After the VueServlet has been deployed into your application server, test the VueServlet by accessing the URL to the VueServlet from a user machine that will be accessing AutoVue. If you are configuring for both internal and external access, you must deploy the VueServlet on a path that is externally 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 5–1 After successful deployment

5.4.3.4 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.

5.4.4 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. A sample of the required JNLP file is provided in [Example 5–1](#).

Example 5–1 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"/>
        </resources>
        <application-desc main-class="com.cimmetry.jvue.JVueApp">
                <!-- AutoVue Client Parameters -->
        [...]
        </application-desc>
</jnlp>

```

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

The implementation of `VueJNLPServlet` requires a JNLP template file – `autovue.jnlp` 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.jnlp`.

Some integration solution requires sending cookies to AutoVue client. 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. 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\jnlp\avjnlpservlet.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.

5.4.4.1 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 5–2 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.jnlp]</param-value>
  </init-param>
  <init-param>
    <param-name>Cookies</param-name>
    <param-value>[Cookie names]</param-value>
  </init-param>
  <init-param>
    <param-name>JVueServer</param-name>
    <param-value>https://AutoVueServer:ServletPort/servlet/VueServlet
    </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 5–6 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 autovue.jnlp with respect to root context (Make sure your root context is used consistently across your entire deployment). For example: /AutoVue
Cookies	Semicolon ';' separated list of cookie names identifying the cookies to transfer to AutoVue at the start-up. For example: JSessionID;... NOTE: This parameter supports also two special values not on the format described above: <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. It is essential that you keep the list of cookies short for security reasons.

Table 5–6 (Cont.) Descriptor Parameters of VueJNLPServlet

Parameter	Description
JVueServer	Specify servlet connection to AutoVue server (separate multiple values with a semi-colon). For example: <code>https://AutoVueServer:5098/servlet/VueServlet</code>

4. Add a block `<init-param>...</init-param>` for each AutoVue client parameter that you want to pass to AutoVue at the start-up. Refer to AutoVue Client Parameters for a complete list of the client parameters supported by AutoVue.

5.4.4.2 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>
```

5.4.5 Installing AutoVue Client 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. For information on deploying AutoVue in an integrated environment, refer to [Integrating With a DMS](#).

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 AutoVue) on your Web server docroot.
2. Copy all mandatory JAR files from the `<AutoVue Install Root>\html` directory to the directory you created on your Web server docroot. The files to copy are `jsvue.jar`, `jogl.jar`, `gluegen-rt.jar`, and [jsonrpc4j.jar](#).
3. AutoVue client supports a scripting API and starts a socket listening to XML HTTP requests invoking this API. 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="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>\html` 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 5–7 Parameters

Parameter	Description
JNLP_HOST	Specifies the URL on your Web/application server, to a host returning the JNLP File required by Java Web Start to run <code>AutoVue</code> client. For example, if you deploy the servlet <code>VueJNLPServlet</code> provided with <code>AutoVue</code> : http://AutoVueServer:5098/servlet/VueJNLPServlet
CODEBASE	Specify the URL to the <code>AutoVue</code> client files on your Web/application server (the folder created above). For example: http://AutoVueClient:5098/autovue

Table 5–7 (Cont.) Parameters

Parameter	Description
CL_PRTS	<p>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.</p> <p>For example:</p> <pre>[2345, [7500, 7510], [8500, 8510], 8888]</pre> <p>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.</p>
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': 'http://AutoVueServer:5098/servlet/VueServlet', '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>

Some of these parameters are defined into JavaScript variables (*CODEBASE*, *JNLP_HOST*, *CL_PRTS*, etc) within *av_jnlp.html* and passed to AutoVue Object constructor. If you use *av_jnlp.html*, update these variables and the line instantiating AutoVue Object with the appropriate values. Add more optional parameters if needed.

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.

- 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 5–8 with the appropriate value:

Table 5–8 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>

5.4.5.1 Configuring AutoVue Client for Secure Socket Layer (SSL)

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:

- 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.
- Copy the complete certificate file (`av_cert.pem`) to the directory you created to your Web server root.
- 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>
```

- 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 localhost 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.

5.4.5.2 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>:5098 /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.

5.4.5.3 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 `javueserver.authentication.enable` to `FALSE` so that users can connect to the server.

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

5.4.5.4 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 `javueserver.properties` file. For more information, refer to [Installing AutoVue Client Components](#).

5.5 Linux-Specific Post-Installation Instructions

To ensure that temporary files are created in a secure location on Linux, the default value setting of the AutoVue temporary directory – `TMPDIR` – has been set to `"/tmp/autovue"`, and its permission is set to 700. Unix permission 700 only allows the owner of the folder to have read/write/execute access.

To modify the temporary folder, the system administrator must edit the `javueserver` and `javueserver_debug` scripts. The AutoVue server administrator is responsible for ensuring that the environment variable follows security guidelines. Take note of the following when modifying `TMPDIR`:

- If the `TMPDIR` value is set to `"/tmp"`, then a security warning will be logged as AutoVue will be creating temporary files in the unsecure `/tmp` directory.
- If the temporary folder is set to one owned by another user, AutoVue will display a warning and exit.
- If the temporary folder is set to one without 700 permission, AutoVue will display a warning and exit.
- `TMPDIR` supports both absolute and relative file paths, but it is recommended to use absolute paths.

5.6 Verifying Communication with AutoVue

Once your AutoVue server has started up correctly, you must verify if clients can connect to the AutoVue server and you can load files in AutoVue.

1. For verification purposes, start up Jetty that ships with AutoVue:
 - **Start VueServlet on Jetty** from the AutoVue programs shortcut on Windows OS
 - `<AutoVue Install Root>/bin/Jetty/bin/startJetty` on Linux OS

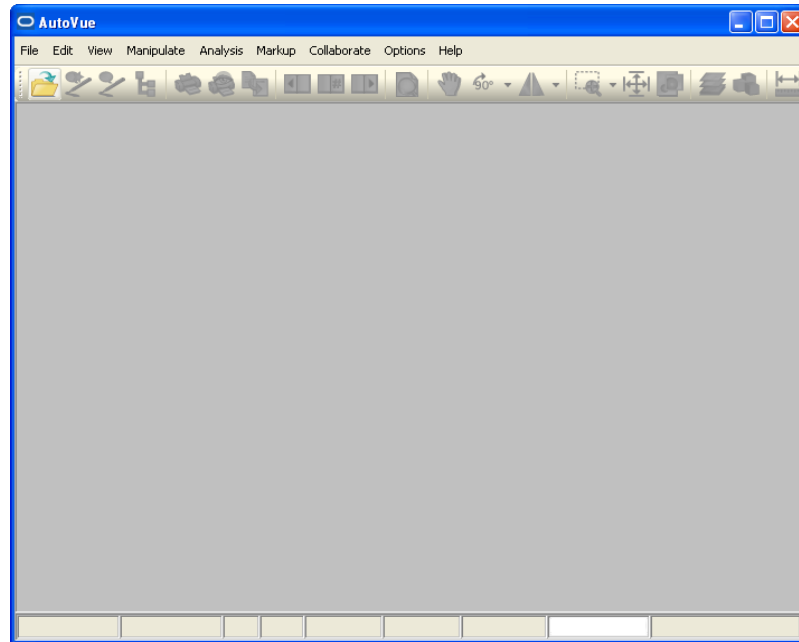
Note: Once you finish the verification, you can shut down Jetty if you do not plan to use Jetty as part of your deployment.

2. If installed, launch the example AutoVue client application by running:

- <AutoVue Install Root>\bin\jvue.bat on Windows OS
- <AutoVue Install Root>/bin/jvue on Linux OS

The AutoVue client should load successfully.

Figure 5–2 AutoVue Client



3. After the AutoVue client starts, verify if you can load files. Load a few files belonging to various format groups that you intend to load using AutoVue.

If you have issues starting the client or loading files in AutoVue, refer to the following section for trouble-shooting pointers.

5.6.1 Troubleshooting Communication Issues

If the AutoVue server has started up fine, but you are unable to open the client or load files, you can review the following pointers to identify other potential issues:

- If the AutoVue client does not start up, verify that the VueServlet is working properly with Jetty. Open the URL to the VueServlet in a Web browser:
<http://<AutoVue server hostname>:5098/servlet/VueServlet> or
<https://<AutoVue server hostname>:8443/servlet/VueServlet> when SSL is configured. As displayed in the following Oracle AutoVue Servlet tunneling page, *Connection State: OK* states that the VueServlet is configured correctly:

Figure 5–3 Oracle AutoVue Servlet Tunneling Page

- If the above verification fails, ensure that Jetty is working properly. If another application is using the same port as Jetty, change the port for Jetty by updating the `jetty.port` parameter in `jetty.xml` or the `port` parameter in `jetty-ssl.xml` when SSL is configured. This parameter is located in the `<AutoVue Install Root>\bin\jetty\etc` directory.
- Verify that the example client application is pointing to the correct URL for the VueServlet. If you modified the port for Jetty, ensure that the example client application is using the right port.
- Verify that the socket port required by AutoVue is not in use by other applications. If the socket port is not available for AutoVue, modify the socket port by updating the `javueserver.socket.port` parameter in `javueserver.properties`.

If you modify the socket port, ensure that the VueServlet points to the correct socket port. Update the `JVueServer` parameter in `webdefault.xml` to point to the correct AutoVue server name and socket port.

5.7 Deploying AutoVue in Virtualized Environments

To set up additional AutoVue servers on virtualized environments, copy an image of the virtual machine where AutoVue server is installed. Once you modify the name of the machine, change the name of the AutoVue server host in `javueserver.properties`. Refer to [AutoVue Host Name Option](#) for information on changing the host name.

Installing in an Integrated Environment

This section describes the components that must be installed and configured when AutoVue is integrated with a DMS. Whether it is an Oracle VueLink integration or a custom integration with AutoVue, you must install and configure AutoVue client components and the VueServlet. The following are generic instructions for deploying AutoVue components in an integrated environment. For specific instructions refer to your VueLink/integration documentation.

Note: If the VueServlet and VueLink are on the same application server, then each will consume one worker thread per file open. As a result, depending on the load, the number of worker threads may need to be increased. At a minimum, two worker threads are required per concurrent file open.

A.1 Installing AutoVue Client Components in an Integrated Environment

In most cases, when AutoVue is integrated with a DMS, the AutoVue client components are deployed with an application server that hosts the integration components and/or the DMS.

In order to deploy the AutoVue client components in these environments, follow these steps:

1. Identify where the AutoVue client components are located in your integration/VueLink deployment.
2. If the AutoVue client components are deployed in a WAR file, extract the contents of the file.
3. Replace the following files in your integration environment (or extracted files) with files from the <AutoVue Install Root>\html directory:
 - jvue.jar
 - jogl.jar
 - gluegen-rt.jar
 - jsonrpc4.j.jar
 - Online help files
4. Re-create the WAR file (if you had extracted it in step 3).

Note: For information on creating a WAR file, refer to your VueLink/integration documentation.

5. Redeploy the WAR file.
6. Update `juvserver.properties` and set the URL to the online help files to the URL specified by the application server.

A.2 Installing the VueServlet in an Integrated Environment

In most cases, when AutoVue is integrated with a DMS, the VueServlet is deployed on the application server that hosts the integration servlet and/or the DMS.

In order to deploy the VueServlet in these environments, follow these steps:

1. Identify where the `VueServlet.jar` is located in your integration/VueLink deployment.
2. If the VueServlet is deployed in a WAR file, extract the contents of the WAR file.
3. Replace `VueServlet.jar` from the extract with the `VueServlet.jar` file from the current release.
4. Modify any of the configuration parameters for the VueServlet as needed. Refer to [VueServlet Configuration Options](#) for more information.
5. Re-create the WAR file (if you had to extract it in step 2).
6. Redeploy the WAR file.

A.3 Verifying your Integration

After installing the AutoVue client components and the VueServlet, you must verify that your integration works correctly with this version of AutoVue:

- Verify that you can load your DMS files in AutoVue.
- Verify Help-About from the AutoVue client to make sure you have the JAR files of the right version and build.
- Access the VueServlet page from a client machine and verify that it reports the correct build number.
- Verify that you can launch the AutoVue online help files successfully.
- Verify the other features supported by your integration to make sure all features work as expected.

Configuring AutoVue Server Farm

In order to meet your concurrent usage requirements, it may be necessary to setup more than one AutoVue server and balance requests to AutoVue across these servers. Configuring multiple AutoVue servers to communicate with each other to handle the load is referred to as an AutoVue server farm. Each AutoVue server has a primary server and multiple document servers. The primary server accepts all requests to AutoVue and is responsible for distributing document requests across the document servers. When AutoVue is configured in a server farm, the primary servers across the servers in the farm communicate with each other in order to distribute load across all the document servers in the server farm.

Take note of the following when setting up an AutoVue server farm:

- If AutoVue is running in standalone mode, Symbols and Markups are not shared between the servers.
- If AutoVue is integrated with a DMS:
 - Markups are saved in the DMS and are shared between the servers.
- Cache should not be shared between the servers.
- Symbols, Symbols folder, and Profiles should not be shared between the servers. Symbol sharing may result in race condition and the profile may become corrupt (that is, more than one thread may be writing to a file at the same time). For more information, refer to [Symbol Libraries](#).
- All server machines should obey the following conditions:
 - Run the same AutoVue server version (up to the patch version)
 - Install the AutoVue server in the same directory paths. Otherwise, streaming file validation fails.
 - Have the same fonts installed.
 - Have identical certificates in their JVM repositories.
 - Have the same proxy settings.
 - Have the same operating system.
 - Have identical firewall settings.
 - Must be configured with one VueServlet per AutoVue server.
 - 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.

- Configure the VueServlet for load balancing. For more information, refer to [Configuring VueServlet Load Balancing](#).
- Do not use Real-Time Collaboration on local files. Real-Time Collaboration is only supported for AutoVue servers configured in a server farm.
- For high security, use HTTPS protocol from the AutoVue server to the VueLink.

B.1 Setting Up AutoVue Server Load Balancing

This section describes how to set up AutoVue server load balancing.

1. Add a new machine to the same network as the original AutoVue server.
2. Install and configure the AutoVue server on the new server, going through the same steps as in the original installation.

Note: In a multi-AutoVue server deployment, it is recommended to install the same version of AutoVue in the exact same folder path on each server.

3. Once installed, edit the `jvueserver.properties` file located in the `<AutoVue Install Root>\bin` directory on the machines hosting the servers in the server farm and add the following parameters:

```
jvueserver.rmi.host.1=jvueserver1.company.com:1099
jvueserver.rmi.host.2=jvueserver2.company.com:1099
```

Where

`jvueserver.rmi.host.1` is set to the name and the RMI port of the one of the AutoVue servers in the farm,

`jvueserver.rmi.host.2` is set to another AutoVue server in the farm and so on.

Note: Ensure that the RMI host entries are specified in the same order on all the servers in the server farm.

B.1.1 Symbol Libraries

If you are using Symbol markup entities, we recommend that you do not share the Symbols folder between the servers in the farm. Instead, replicate the Symbols folder across all AutoVue servers in the server farm at regular intervals. You should plan the replication for a time when AutoVue is not in use.

Sharing the Symbols folder between servers causes reliability and stability issues. Make sure that the replication is done when AutoVue is not in use to ensure that Symbols are not being used when replication occurs.

B.2 Verifying AutoVue Server Load Balancing

In order to verify AutoVue server load balancing, you must:

- Open a few different connections to the AutoVue server. Monitor the AutoVue server console on one of the servers in the farm. You should see that user sessions should be balanced across the servers in the server farm.

- From these connections, open multiple documents. On the AutoVue server console, for each session, click the **Documents** column and verify the server where the document is being opened from. The documents will be balanced across the document servers on the server where the users session is located. Refer to [AutoVue Server Console](#) for more information on how to see session and document information from the AutoVue server console.
- If a firewall is enabled, you must add java.exe and javaw.exe to the firewall exceptions.
- RMI and socket ports that are defined in jvueserver.properties must be open. For example, when the following is set in jvueserver.properties:

```
jvueserver.socket.port=5099
jvueserver.rmi.port=2099
jvueserver.processPoolSize=4
```

Ports 5099, 2099, 2100, 2101, 2102, 2103, 2104 should be open.

For information on configuring RMI and sockets ports refer to [RMI and Socket Ports Options](#)

Note:

- When opening a file using the upload:// protocol, the document is opened on the same server as the user's session.
 - When opening a file, the DocServer with the least number of documents is selected. However, if two or more DocServers have the same load, then the DocServer that is on the same server as the session is selected.
 - User sessions will be balanced across all document servers attached to the primary server if session is established. User (sessions) will be balanced across the entire cluster and will be attached to the primary servers with the smallest load (documents open).
-
-

B.2.1 Troubleshooting AutoVue Server Load Balancing

If you see that requests are not load balanced across the servers in the farm, verify the following:

- jvueserver.rmi.host.X entries are in the same order across all the servers in the farm.
- All the servers in the farm are up and running.

Note: In some instances when you open connections simultaneously, it is possible that requests are not load balanced. This is as expected. When there is some lag between the connections, requests are load balanced.

B.3 Configuring VueServlet Load Balancing

The VueServlet needs to be deployed within an application server. You must rely on the load balancing capabilities of the application server or rely on an external load balancer that is configured to distribute load across all your application server

(VueServlet) instances. You must also ensure that the load balancer is configured to enable session stickiness (also referred to as session persistence). Session stickiness is normally achieved through the use of browser cookies.

Ensure that each VueServlet instance has the same entry for the JVUESERVER parameter.

For example:

```
<param-name>JVueServer</param-name>  
<param-value>AutoVueServer1:5099;AutoVueServer2:5099</param-value>
```

If the AutoVue server is also configured for load balancing, then you must ensure that each of the VueServlet instances also specify the AutoVue servers in the same order.

Failover and Disaster Recovery

This chapter describes how to configure AutoVue for fail-over and disaster recovery.

C.1 AutoVue Server Configuration for Failover

For failover, AutoVue server should be deployed in a horizontal cluster or server farm. In a horizontal cluster, servers are spread over multiple machines.

You can have these AutoVue servers configured for load balancing so that if one AutoVue server in the server farm goes down requests are re-directed to other servers in the farm. Refer to [Configuring AutoVue Server Farm](#) for instructions on setting up a server farm. All AutoVue servers should be identified as peer servers acting as multiple entry points for all VueServlets communications (that is, there is no primary AutoVue server handling all the VueServlet communication). Each server in the farm acts as a backup server so that if one server goes down, another server is available to continue serving clients.

In certain situations, when you want a backup server without setting up load balancing, you must configure the VueServlet to communicate to the backup server if the production server is unavailable.

Note: When a server goes down, the users on that machine, along with all their open documents, are moved over to another machine. Any markups not stored in a DMS, or any user specific settings, are not moved over to the backup machine.

C.2 AutoVue Failover Configuration on the VueServlet

In the event of a failure of an AutoVue server, either when using a cluster or when using a standalone server, you can configure the VueServlet so that it directs requests to another AutoVue server. When using a cluster, the failover server can be another server in the cluster. When using standalone installation, you must install another instance of the AutoVue server.

To configure VueServlet for failover, update the JVUESERVER parameter of the VueServlet to add multiple AutoVue servers. Separate values using a ','.

Note: Each VueServlet must have the same list of servers for the JVUESERVER parameter, and this list must be in the same order for all VueServlets.

```
<param-name>JVueServer</param-name>
<param-value>AutoVueServer1:5099;AutoVueServer2:5099;AutoVueServer3:5099</param-value>
```

C.3 Failover for the VueServlet

Since the VueServlet is hosted within an application server, you can rely on the application server's load balancing and high availability features. You must ensure that there are multiple VueServlet instances on separate machines so that if one instance of the VueServlet is not accessible, users are automatically redirected to another VueServlet instance.

Similarly, you must plan for backups for the AutoVue client components and online help within your application server so that if one instance is not available, users are redirected to another instance.

C.4 Failover for AutoVue client components

If the AutoVue client components are deployed with an application server, you can rely on the application server's load balancing and high availability features. Ensure that the JAR files and the online help files are served through the load balancer.

C.5 Verifying Failover Configuration

To verify that AutoVue is configured fully for failover, you must bring down certain nodes and verify that users are still able to connect and use AutoVue:

- Shutdown an AutoVue server. Preferably the server that is the entry point (the first AutoVue server in the VueServlet configuration) for AutoVue requests. Open connections to AutoVue and verify that users are connected to the backup servers.
- Bring down an application server instance hosting the VueServlet and the client components. Ensure that users are still able to launch the AutoVue client, load files and load online help files.

Integrating With a DMS

This chapter describes the additional configuration to consider when AutoVue is integrated with a DMS. Configuring AutoVue for multiple backend systems and creating a Stamp template is described in this section.

Note: Backward compatibility (*dms.vuelink.version=[19.3]*) has been desupported. You must upgrade your integration to be compatible with AutoVue 21.x.x

D.1 Multiple Document Repositories

If AutoVue is integrated with multiple DMS, AutoVue's Universal File Chooser (File Open) dialog allows you to browse and search them. You can browse/search the DMS even if your client has not already established a connection to the DMS. To enable AutoVue to browse/search when you have not yet established a connection with the DMS, you must create a file named *vuelinks.xml* in the <AutoVue Install Root>\bin directory with the following format:

```
<DMSList>
  <vuelink url="vuelink_url">
    <name>your_DMS_name</name>
    <DMSArgs>
      <DMSArg name="your_argument" value="your_value" />
    </DMSArgs>
    <seed>seed_url</seed>
  </vuelink>
</DMSList>
```

- The <vuelink> tag defines the URL location of the backend DMS system. Replace *vuelink_url* with the URL to your VueLink/integration servlet.
- The <name> tag defines the DMS button name to appear in the File Open dialog. Replace *your_DMS_name* with the name of your DMS.
- The <DMSArgs> tag defines arguments for the specified integration. Replace *your_argument* and *your_value* with any DMSArgs you may use with DMS integration.
- The <seed> tag defines the URL format for retrieving a file from the DMS. Replace *seed_url* with a URL to a file from your DMS. This is generally the FILENAME URL that is passed to the AutoVue client when you view your DMS file in AutoVue.

Note: In your seed URL, you must replace any special characters with its character entity reference. For example: Replace **&** in a URL with **&**;

For more information on the File Open dialog, refer to the *Oracle AutoVue Client/Server Deployment User's Manual*.

D.2 Creating a Stamp Template

Stamps are dynamic stamp entities that can retrieve user and document metadata from the DMS. Stamps attributes can be used to update the document in the DMS. They are only available to end-users after they have been configured by the system administrator.

Note: If AutoVue is not integrated with a DMS/PLM, the stamp icon will be grayed out, and stamps cannot be used.

The following steps describe the procedure for creating a Stamp template:

1. Identify what images to use as a background image. For more information, refer to [Choosing a Background Image](#).
2. Determine Stamp attributes and permissions. For more information, refer to [Determining Stamp Attributes and Permissions](#).
3. Identify the Stamp layout. For more information, refer to [Identifying the Stamp Layout](#).
4. Design the Stamp. For more information, refer to [Designing a Stamp Layout](#).
5. Additional configurations for the Stamp. For more information, refer to [Configuring Stamp Templates](#).
6. Configure the Stamp with your integration. For more information, refer to [Configuring Stamp with Your Integration](#).

D.2.1 Choosing a Background Image

To design a Stamp, you must first choose an image that best fits your stamping needs. For example, you may choose an image that indicates stamp approval. Additionally, you should place attributes on the Stamp that indicate approval status, approver name, department, approval date, and so on.

AutoVue supports adding Windows Metafile (WMF), Enhanced Metafile (EMF), and Bitmap (BMP) files as background images for your Stamp. It is recommended to use EMF as the background image for Stamps. WMF and BMP are currently supported for backward compatibility.

Recommendation: It is recommended to use EMF as the Stamp image for the following reasons:

- Supports transparency
 - Supports resizing of Stamp text when the Stamp image is resized
-

For information on how to create EMF, refer to

<https://support.us.oracle.com/oip/faces/secure/km/DocumentDisplay.jspx?id=1404486.1>.

D.2.2 Determining Stamp Attributes and Permissions

You must determine what attributes to display on your Stamp:

- Do you want to display a value that is from the backend system (DMS)?
- Do you want to display a local attribute (that is, an attribute determined by the user's local machine) such as \$date and \$user?

You must also determine the permissions you want set for the attributes:

- If you want to read values from the DMS, you must create an attribute with the same name as the DMS attribute.
- If you want to assign local values, you may assign the attribute a value that can be one of the following:
 - \$date: current date
 - \$user: user name
- Do you want users to be able to modify attributes?
- Do you want to transfer these modifications to the DMS?

You can specify specific permissions for each attribute.

Note that an attribute may also be a drop-down list (single-value list, constrained list, non-constrained list, or a multi-valued list that maps to a DMS attribute that is a drop-down list). Users are presented with a list in the Value from DMS column of the DMS Attributes dialog:

- Single-Value List: Can only select one value from the list.
- Constrained List: Can only select values from the list.
- Non-Constrained List: Can add values that are not in the list.
- Multi-Valued List: Can select multiple values from the list. AutoVue uses a semi-colon (;) as a separator for multiple values.

Note: The local attributes cannot have multiple values. Hence, a drop-down list is not available for local attributes. The drop-down list is only available for multi-value DMS attributes.

The following table lists the available attribute permissions and their behavior.

Table D–1 Attribute Permissions

Attribute Permissions	Description	Read from DMS	Reset Value to DMS	Assign Local Value (\$date, \$user)	Modify value in edit dialog	Write value back to DMS
ReadOnce	The attribute value is read only once from the DMS when the Stamp is the first created. The attribute value will not be updated later.	Yes ¹	No	Yes ²	N/A	N/A
Read	The attribute queries its value from the DMS every time the Stamp is opened.	Yes	Yes	Yes ²	N/A	N/A
Edit	The user is allowed to modify the attribute value in the Stamp.	N/A	N/A	N/A	Yes	N/A
Write	Allows the user to write the attribute value back to the value of the attribute in the DMS.	N/A	N/A	N/A	N/A	Yes

¹ Only reads DMS at creation.

² If defined, retrieves the default value (for example, \$user or \$date) when the DMS cannot be read (a no read permission) or when there is a read permission but the DMS returns an empty value or NULL.

The following table illustrates how the attribute permissions can be combined.

Table D–2 Combination of Attribute Permissions

Combination	Read from DMS at creation	Read from DMS every time	Reset Value to DMS	Assign Local Value (\$date, \$user)	Modify value in edit dialog	Write value back to DMS
ReadOnce	Yes	No	No	Yes ¹	No	No
ReadOnce / Edit	Yes	No	No	Yes ¹	Yes	No
ReadOnce / Write	Yes	No	No	Yes ¹	No	Yes
ReadOnce / Edit / Write	Yes	No	No	Yes ¹	Yes	Yes
Read	Yes	Yes	Yes	Yes ¹	No	No
Read / Edit	Yes	Yes	Yes	Yes ¹	Yes	No
Read / Write	Yes	Yes	Yes	Yes ¹	No	Yes
Read / Edit / Write	Yes	Yes	Yes	Yes ¹	Yes	Yes
Edit	No	No	No	Yes ¹	Yes	No
Edit/Write	No	No	No	Yes	Yes	Yes

¹ If defined, retrieves the default value (for example, \$user or \$date) when the DMS cannot be read (a no read permission) or when there is a read permission but the DMS returns an empty value or NULL.

D.2.3 Interacting with Stamps

After a Stamp has been created, users can double-click the entity to open the DMS Attributes dialog. The dialog lists all of the values included in the Stamp as well as the values from DMS for the configured attributes.

From the DMS Attributes dialog, users can do the following:

- Modify Stamp values (if the attribute has Edit permission)
- Reset the attribute to the current value from DMS
- Write attribute values to the DMS

Note the following about the Value from DMS column:

- If an attribute has ReadOnce permission, then the Value from DMS column displays the value read from the DMS only when the stamp is created. Once the markup is saved, it does not populate the Value from DMS column.
- If an attribute has Read permission, then the Value from DMS column displays the current value of the attribute in the DMS.
- If an attribute does not have Read or ReadOnce permission, then the Value from DMS column does not display a DMS value.

D.2.4 Identifying the Stamp Layout

After the background image and Stamp attributes and permissions are determined, you must identify the Stamp layout. The following points have to be considered when identifying the layout:

- Within the Stamp, it is possible to position attributes at various locations with respect to the background image.
- Text can be aligned within a box (left, center, right, top, bottom, and so on).
- You can define the bounding size of the text box. When text runs beyond an attribute box, you have the option to either decrease the text font size in order to keep it within the box or to ignore the bounding box.
- You can specify whether or not the stamps can be resized by users.
- You can specify whether or not the stamp should be a fixed size.
- You can hide certain attributes on the stamp and display them only in the Stamp edit dialog.

D.2.5 Designing a Stamp Layout

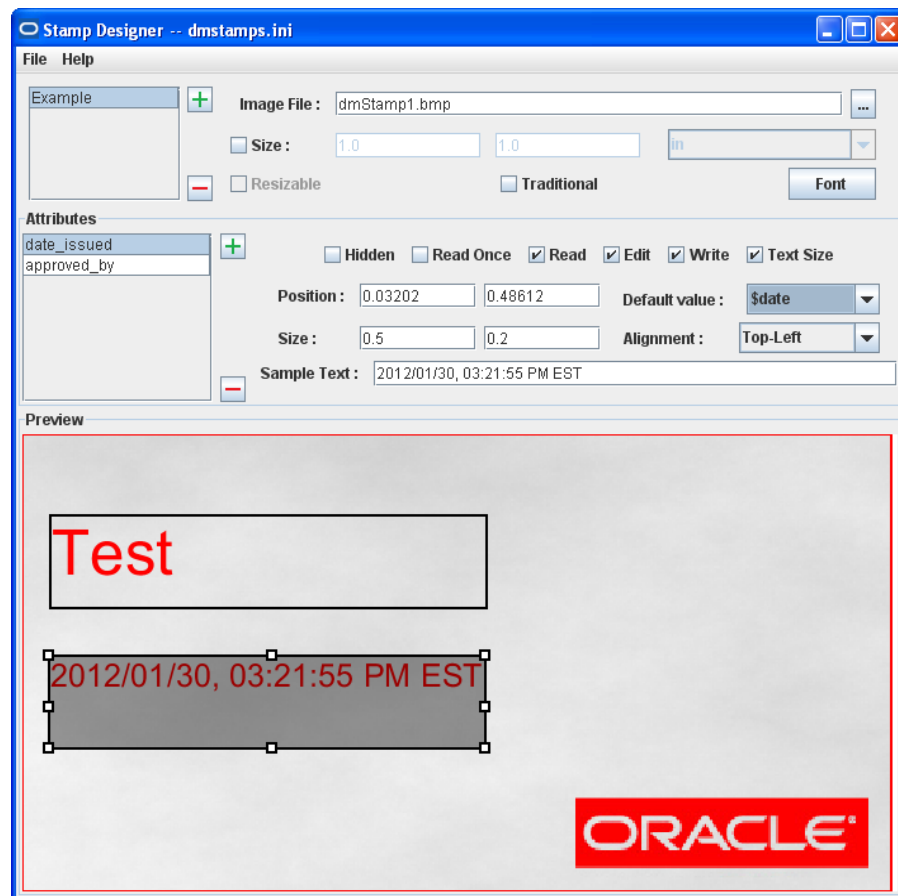
The following section describe how to design the look-and-feel of the Stamp template and how to set the Stamp attributes.


D.2.5.1 Designing the Stamp

To design an Stamp template, do the following:


1. Go to the <AutoVue Install Root>\bin directory and run *designer.bat*.


The Stamp Designer appears.

Figure D–1 Stamp Designer

2. Click  symbol to add a Stamp name.

The Input dialog appears.

Note: You can delete a Stamp by clicking the minus symbol  .

3. Enter a stamp name and then click **OK**.
4. In the Image Field, click  to select an image file.
The Open dialog appears.
5. Browse to the WMF/EMF/BMP file that you want to set as your Stamp background and then click **Open**. The image appears in the Preview section.

Note: Make sure that the image is a valid WMF, EMF, or BMP file.

6. You can set the size of the image by selecting the **Size** check box, entering the XY dimensions, and then specifying the units.

Note: The size feature is supported for formats which have valid page size (i.e., unit Unknown is not supported). For formats that use pixels as the unit and don't have a dpi, AutoVue defaults to a dpi of 200.

7. Selecting the **Resizable** check box allows the Stamp to be resizable
8. Selecting the **Traditional** check box converts the Stamp to a static stamp entity when it is created. That is, attribute values of the Stamp cannot be modified once created.
9. Specify the default font that you want to use for the Stamp. To do so, click **Font** and specify the font details. This is the font that is used when you create a Stamp. The font changes when you resize the Stamp or when you change the font from the Markup toolbar.


Note: When a Stamp is created in the AutoVue workspace, AutoVue scales the font up/down depending on whether the Stamp is drawn larger/smaller in relation to the underlying image.

The end-user must modify the font from the Markup toolbar as necessary.

Note: You must ensure that the font used for the Stamp Designer is available on all the machines that invoke the AutoVue client. If the font specified in the designer is not available, then the AutoVue client uses another available font for the Stamp. This may cause differences in display between the Stamp Designer and the client and potentially cause differences in display between different clients.

The following [Section D.2.5.2, "Setting the Stamp Attributes"](#), describes how to set the Stamp attributes from the Stamp Designer.

D.2.5.2 Setting the Stamp Attributes

1. In the Attributes section of the Stamp Designer, click  to add the attributes that you want to be displayed on the Stamp.
2. You can add an attribute that is mapped to a DMS attribute or you can add a custom attribute that does not correspond to a DMS attribute. When mapping to a DMS attribute, you must specify the name of the attribute as it appears in the DMS system.
3. The following check boxes allow you to specify certain restrictions to the selected attribute:
 - **Hidden:** The attribute is not displayed in Stamp. However, it is accessible from the Stamp Edit dialog.
 - **ReadOnce:** The attribute value is read only once from the DMS when the Stamp is the first created. The attribute value will not be updated later.
 - **Read:** The attribute queries its value from the DMS every time the Stamp is opened.
 - **Edit:** The user is allowed to modify the attribute value in the Stamp.

- **Write:** Updates the attribute value back to the value of the attribute in the DMS.
 - **Text Size:** Set the text box size for the attribute text. All text that appears in the box is resized to fit the specified dimensions. If the text box size is not specified, a default text size is used.
4. Set the position of text box by specify the X and Y coordinates in regards to the dimension of the background image in the Position fields. For example, setting the X and Y values to [0.5,0.5] places the top-left corner of the text box exactly at the center location of the background image. Note that values for X and Y can be decimal value [0.0-1.0].
 5. When the Text Size check box is selected, you can set the size of the text box with regards to the dimensions of the background image. For example, setting the X and Y values to [1,1] creates a text box the size of the background image. Note that values for X and Y can be a decimal value [0.0-1.0]
 6. To map an attribute to the pre-defined variables, **\$user** or **\$date**, select the value from the **Default value** list. If the DMS attribute does not have a value, these variables may be used to set the default value for the attribute.
 - *\$user* is the name of the current user (either the DMS user name or the operating system user name if AutoVue is not integrated with a DMS).
 - *\$date* is the system date. The date format can be changed by the setting the *Format[n]* INI option. Refer to [Configuring Stamp Templates](#) for more information.
 7. From the Alignment list, you can specify the location of text inside the text box.
 8. To preview how attribute text will appear on the Stamp, you can enter text in the Sample Text field.
 9. In the Preview section, select and place each attribute as required in the Stamp. Resize the attributes as necessary.

D.2.6 Configuring Stamp Templates

When Stamp templates are designed, the Stamp INI file, dmstamps.ini, is updated with information regarding the Stamp. The default location of this INI file is <AutoVue Install Root>\bin.

The [Table D-3, "INI Options"](#) contains a description of the available INI options in dmstamps.ini.

Note that all of these options, with the exception of *Format[n]* and *Isotropic*, may be set from the Stamp Designer. As a result, administrators may not need to update the INI options directly.

Note: Options in the [Stamps] section of the file apply to all Stamps.

Table D–3 *INI Options*

INI Option	Description
NumStamps=<integer>	<p>This option indicates the total number of Stamps that have been designed.</p> <p>Important: Do not update this option value.</p> <p>Example:</p> <p>NumStamps=12</p> <p>Indicates that there are 12 Stamps that have been configured.</p>
AttributesNames=attribute1;attribute2;...	<p>This option lists the attribute names that should appear in the Stamp Designer list. Separate multiple attributes using a semi-colon (;). Ensure that the last attribute has a semi-colon after it.</p> <p>By default, the Stamp Designer has 2 drop-down items: approved_by and date_issued.</p> <p>Example:</p> <p>AttributesNames=approved_by;date_issued;dm_approval_status;</p>

Each Stamp must have a [Stamp[*n*]] section in the INI file, where [*n*] is an integer starting from 0 and is an index for each defined Stamp. For example, when two Stamps are designed, there should be two sections [Stamp0] and [Stamp1] in the dmstamps.ini file.

Each section contains information pertaining to the Stamp represented by the section:

[Stamp[*n*]]

Table D–4 *INI Options*

INI Options	Description	Default
Name=name of Stamp	<p>Specifies the name of the Stamp.</p> <p>Example:</p> <p>Name=ReviewerStamp</p>	
ImageFile=full path to image	<p>Sets the full path to the background image for the Stamp.</p> <p>Example:</p> <p>ImageFile=C:\stamps\reviewbg.wmf</p>	
ReadOnly=[0 1]	<p>Once the stamp is created and the markup is saved, the stamp can no longer be edited.</p> <p>Set to 1 so that the Stamp is read-only.</p> <p>Example:</p> <p>ReadOnly=1</p>	0
NumAttributes=<integer>	<p>Indicates the number of attributes associated with this Stamp</p> <p>Example:</p> <p>NumAttributes=3</p> <p>This indicates that the Stamp has 3 attributes.</p>	

Table D–4 (Cont.) INI Options

INI Options	Description	Default
Font=[font style, font size, font weight]	Specifies the default font for the Stamp. This is the font that is used by the Stamp at creation time. The end-user can also change the font and font size when creating a Stamp. Example: Font=Arial,16,400	
Isotropic	When Isotropic is set to 1, the Stamp resizes uniformly in all directions. Example: Isotropic=0	1
Attribute[n]	Specifies the name of the [n]th attribute Example: Attribute1=dm_approval_status	
Value[n] =[\$user \$date]	Specifies Value of the [n]th attribute	
Format[n]	If value[n] is \$date, specify the format for the date. Example: Format1=EEE, MMM d, ' 'yy Any date format specified by java.text.SimpleDateFormat can be specified: http://docs.oracle.com/javase/8/docs/api/java/text/SimpleDateFormat.html	
Hidden[n]	Specifies whether or not Attribute[n] is hidden. If set to 1, Attribute[n] is a hidden attribute Example: Hidden=0	
permission[n]	Specifies the attribute permissions. Example: Permission1=Read,Edit,Write	
PosX[n] PosY[n]	Specifies the position of Attribute[n] relative to the background image. The top-left corner is 0,0. Example: PosX1=0.621622 PosY1=0.029661	

Table D–4 (Cont.) INI Options

INI Options	Description	Default
Size[n]=Width, Height, Unit, Resizable	<p>Specifies the width, height, unit and resizability of the Stamp.</p> <p>Parameters:</p> <p>Width: Set the width.</p> <p>Height: Set the height.</p> <p>Units: Supported units are as follows:</p> <ul style="list-style-type: none"> ■ 1 - Inch. ■ 2 - Millimeters ■ 4 - Twips ■ 5 - Centimeter ■ 6 - Decimeter ■ 7 - Meter ■ 8 - Kilometer ■ 9 - Feet ■ 10 - Yard ■ 11 - Mile ■ 12 - Thousandth of an inch ■ 13 -Ten Thousandth of an inch ■ 14 - Micron ■ 15 - Micro inch <p>Resizable: Set whether the Stamp can be resizable. (0=no, 1=yes)</p> <p>Example:</p> <p>The following setting for a non-resizable Stamp with a width=1, height=3, and the units set to centimeters.</p> <p>Size1=1, 3, 5, 0</p>	

D.2.7 Configuring Stamp with Your Integration

The default location of the Stamp INI file, dmstamps.ini, is located in the <AutoVue Install Root>\bin directory and must be made available to your integration/VueLink. Additionally, when Stamp templates are created, they refer to paths for the background images. These images must be made available to the VueLink. Perform the following steps to ensure that the VueLink can access Stamp templates:

1. Identify where you want to store dmstamps.ini. This INI file can be placed in a folder path that is accessible to the VueLink or can be checked into the DMS.
2. If you want to check-in dmstamps.ini and the Stamp background images into your DMS, do the following:
 - a. Ensure that all users have read permissions to the folder in the repository where you will check-in the INI and the Stamp background images.
 - b. Check-in dmstamps.ini into the repository.
 - c. Configure your VueLink so that it is aware of where dmstamps.ini is located. Typically, you can do this by updating the web.xml that contains the VueLink

configuration. Edit web.xml in a text editor such as Notepad. Update the value for CSI_IntelliStampDefLocation to point to the full path to dmstamps.ini.

Example: CSI_IntelliStampDefLocation=/System/dmstamps.ini

- d. Check-in all the WMFs into the repository.
- e. Edit dmstamps.ini and update the ImageFile option value for all Stamps templates to point to the Stamp background images in the repository.

Example: ImageFile=/System/StampImage1.wmf

- f. Repeat steps c through e for all repositories. Ensure that dmstamps.ini and the WMFs are available at the same path in all the repositories.

3. To have the Stamp templates in Windows folder paths, follow these steps:

- a. Copy dmstamps.ini to a folder path accessible to the VueLink.
- b. Update web.xml to point to the path to the dmstamps.ini file. Edit web.xml in a text editor such as Notepad. Update the value for CSI_IntelliStampDefLocation to point to the full path to dmstamps.ini.

Example: CSI_IntelliStampDefLocation=C:/stamps/dmstamps.ini

Note: You must use forward slash (/) for the path to the INI file.

- c. Copy the background images to the same location as the INI file.
- d. Edit dmstamps.ini and update the path to the Stamp background images to reflect the location where you placed them.

Example: ImageFile=C:\stamps\StampImage1.wmf

- e. Repeat steps c through e for all VueLink instances. It is not recommended to use a shared network location for dmstamps.ini and/or the WMFs.

D.3 Verifying Your Integration

- Verify that you can load your DMS files in AutoVue. Verify other functionality that is supported by your integration to ensure that your integration will work with the latest release of AutoVue.
- If you are using Stamps, verify that users can create Stamps. Verify that the attributes in the Stamp reflect backend system attributes.
- If you are working with multiple DMSes and you have vuelinks.xml configured, launch AutoVue client and verify that you can browse or search through the DMSes.

Configuring for Real-Time Collaboration

This chapter describes how to configure AutoVue for Real-Time Collaboration. All configurations are performed in `javueserver.properties`. For a complete list of AutoVue server configuration options including collaboration options, refer to [AutoVue Server Configuration Options](#).

E.1 Default Collaboration Configuration

When configuring AutoVue for a real-time collaboration deployment, you must set the following basic parameters in `javueserver.properties`.

- `javueserver.collaboration.enable`
This parameter must be set to `TRUE` to enable collaboration mode on the AutoVue server. By default, this parameter is set to `TRUE`.
- `javueserver.collaboration.id.min`
When running an AutoVue server farm, you can specify the minimum ID to use for collaboration sessions and users by this AutoVue server. The second server in the farm should have a minimum ID of at least the first server's minimum ID + the first server's ID range.
- `javueserver.collaboration.id.range`
Specify the range of IDs given to collaboration and users by this AutoVue server.

Refer to [Collaboration Options](#) for more information.

E.2 Distributed Geographies Configuration

Configuring for distributed geographies consists of the same steps as described in [Default Collaboration Configuration](#).

E.3 Distributed DMS Configuration

Configuring for distributed DMS consists of the same steps as described in [Default Collaboration Configuration](#), except that there must also be an internal virtual private network (VPN) setup among the AutoVue servers. The VPN is required so that the AutoVue servers in distributed geographies are able to communicate securely.


Starting the AutoVue Server

This chapter discusses how to start and stop the AutoVue server on Windows and Linux.

Note: If you have a firewall enabled, a prompt may appear asking you to block or unblock the AutoVue server executable. Select **Unblock**.

F.1 Starting AutoVue on Windows

Following is the procedure to start AutoVue on Windows:

1. Start the AutoVue server by clicking **Start 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`. Refer to [AutoVue Server Configuration Options](#) for more information.

2. Start the application server on which VueServlet is deployed.

Note: If you are using Jetty, you must start it up by running the **Start VueServlet on Jetty** shortcut in the AutoVue programs group.

3. Make sure to start the Web server if you are using it for the AutoVue client components.

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

F.2 Starting AutoVue on Linux

Following is the procedure to start AutoVue 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 [AutoVue Server Configuration Options](#) for more information.

2. Start the application server on which VueServlet is deployed.

Note: If you are using Jetty, you must start it up by running `startJetty` from the <AutoVue Install Root>/bin/jetty/bin directory.

3. Make sure to start the Web server, if you are using it, for AutoVue client components.

The startup script for the AutoVue server on Linux OSes also starts up the Xvfb server. Xvfb is an X11 virtual framebuffer 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 [Running the AutoVue Server as a Service](#).

F.3 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.

F.4 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.

F.4.1 On Windows OSes

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 service, 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: It is essential that AutoVue server service use it's own user 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 does 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
```

F.4.1.1 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**.

F.4.2 On Linux OSes

Oracle provides an *RC-Script* to manage the AutoVue server on Linux. The AutoVue server can be configured to start up 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 start up automatically:

```
chkconfig autovue on
```

AutoVue 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
```

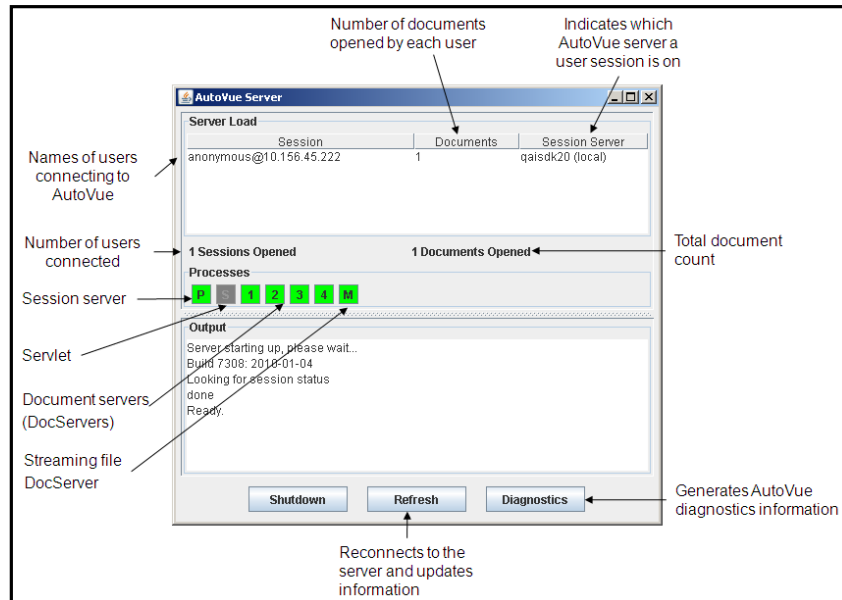

Monitoring the AutoVue Server

You can monitor the AutoVue server from the AutoVue Server Console. The console displays information on the number of clients connected to AutoVue, the document opened by each connection, the server that a user session is connected to. Additional information such as AutoVue usage history, server diagnostics are also available. The following sections describe how to go about monitoring your AutoVue server.

G.1 AutoVue Server Console

The AutoVue server console displays the user connection state (process, username, client IP and number of open documents) and the process pool state. Upon starting the server, the console launches and the connection and process pool states are queried.

Figure G–1 AutoVue Server Console















- Click **Refresh** to update the console display to regenerate server information.
- To stop running the AutoVue server and all attached processes, click **Shutdown**.
- The **Diagnostics** feature of the AutoVue server console generates a report, *JVueServerDiagnostics.out*, to the <AutoVue Install Root>\bin\Logs directory and contains pertinent troubleshooting information. An Oracle Global Customer

Support representative may require you to generate the report to identify problems you may have with your deployment of AutoVue. Any errors that occur during initialization are listed under *Output*. If you require more information regarding the diagnostics feature, contact an Oracle Global Customer Support representative.

- The Processes section of the console lists the servers and their status:

Table G–1 Servers

Pool State	Description
 Red	Process is not running.
 Green	Process is running.
 Yellow	Process is initializing.
 Grey	Process is disabled by the user (applies only to servlet process).
 Black	Process is not responding.

-  indicates the session server. You can view AutoVue server information (build number and server load) by clicking  button.
-  indicates the servlet engine. This only represents the Jetty servlet engine that ships with AutoVue and is only relevant if customer decides to use Jetty as their application server to host the VueServlet and/or client components.
- , ,  ... represent document servers (DocServers). You can restart a DocServer by clicking on the button and selecting **Restart**. The number of document servers is set in the `juveserver.processpoolsize` parameter in `juveserver.properties`. For more information, refer to [Process Pool Size Option](#).
-  represents an additional DocServer reserved for generating streaming files (only visible when `juveserver.metacache.enable` is set to **TRUE**, which is the default value in `juveserver.properties`). You can restart this server by clicking on the button and selecting **Restart**.
- Session information is displayed on the Console:
 - The names of the users connecting to AutoVue.
 - The number of documents opened by each user.
 - The AutoVue server that the user session is on.
- Double-click on the session listed in the Console to view the following information regarding the session:
 - Which document(s) are currently opened by the user.
 - Which DocServer(s) are loading the user's documents.
 - Which AutoVue server(s) are loading the user's documents.

G.2 Usage Monitoring

AutoVue has usage monitoring to enable system administrators to track how many files of a format group are opened at any given time. For example, you can use this feature to track the number of licenses for the different product variations of a single deployment of AutoVue. Usage data is written to *licusage.out* file in the <AutoVue Install Root>\bin\Log directory.

AutoVue ships a utility to parse the usage log and present meaningful information to the system administrator.

The following is the format of the command line to run this utility:

```
usagestat [-c] <path to the input file>
```

where `usagestat` is the command to run this utility.

`[-c]` is an optional parameter and indicates that the utility should run in continuous mode (that is, the output displays continuously).

`<path to the input file>` specifies the full path to the input file (for example, the log file on which the statistics is based). This argument is mandatory.

The following is an example command line that runs the utility in a standard mode.

```
usagestat c:\AutoVue\bin\Log\LicUsage.out
```

G.3 Logging for the AutoVue Server

The configuration file *log4j.xml* (located in the `<AutoVue Install Root>\bin` directory) lets you configure the logging for the AutoVue server. By default, the logs are saved to the `<AutoVue Install Root>\bin\Log` directory. The configuration file defines several appenders (Log4j output destinations) and output layouts. Review the logging information in log4j to troubleshoot any issues you experience with the AutoVue server. If you are unable to resolve the issue yourself, provide the logging information in *log4j-rol1XX.log* to an Oracle Global Customer Support representative.

To set the logging level and time interval for detecting log4j configuration change, you must set log4j parameters in *javueserver.properties*. For more information, refer to [log4j and Diagnostics Options](#).

The following sections provide information on the available appenders, output layouts, and logger information.

G.3.1 Log4j Appenders

The following table lists the appenders that are defined in *log4j.xml*. However, you may use any other appender as you see fit.

Table G–2 Log4j Appenders

Appender	Description	Parameters
File Appender	Appender for logging to file.	<p><code>fileName</code>: The name of the log file.</p> <p><code>Append</code>: If set to TRUE, the logs are appended to the file after the process restarts. If set to FALSE, the old files are discarded after the process restarts.</p>

Table G–2 (Cont.) Log4j Appenders

Appender	Description	Parameters
Rolling File Appender	Backs up (rolls) previous files when the maximum files size is reached. Note: This appender is enabled by default.	filePattern: The pattern of the file name of the archived log file produced by a roll over.
		SizeBasedTriggeringPolicy / size: The maximum size that the output file is allowed to reach before being rolled over to the backup files.
		DefaultRolloverStrategy / max: The maximum number of backup files to keep.
		<code>\${jvueserver.processIndex}</code> : The AutoVue server-specific parameter used to log AutoVue's processes into a separate file.
Daily Rolling File Appender	Defines the frequency for rolling over a file.	filePattern: Name pattern to use for the files created on a periodic basis. TimeBasedTriggeringPolicy / interval: How often a rollover should occur based on the most specific time unit in the date pattern. For example if the Date Pattern is in MMM-dd-yyyy-HH format, and the interval is set to '1', it means that the roll over will be scheduled on an hourly basis.
Console Appender	Used for logging the program console window.	

G.3.1.1 Output Layout

The [Table G–3, "Output Layout"](#) defines the available output layouts in the log4j.xml configuration file.

Note: The output XML file can be viewed in a GUI-based log viewer such as Apache Chainsaw.

Table G–3 Output Layout

Output Layout	Description	Parameters
XML Layout	When this layout is enabled, log4j outputs the logs in XML format.	Properties: Set this value to TRUE to force log4j to record Mapped Diagnostic Complex (MDC) values. AutoVue-specific MDC values: User: Outputs source username with logging event. Document: Outputs current document name with logging event.

Table G–3 (Cont.) Output Layout

Output Layout	Description	Parameters
Pattern Layout	When this layout is enabled, log4j outputs the logs in textual format allowing for flexible string format configuration.	<p>complete: Set this value to TRUE in order to force the inclusion of the XML header and footer in the logs.</p> <p>ConversionPattern: The string that controls formatting. For the list of formatting characters, refer to log4j documentation.</p> <p>AutoVue-specific formatting characters in conversion pattern:</p> <p>%s: Outputs current document server index or "0" in the case of a session server.</p>

Note: To enable logging output to the console, you must uncomment the *AUTOVUE-CONS* line in the log4j.xml configuration file as shown in the following figure:

```
<!-- Root logger -->
<Root level="debug">
<AppenderRef ref="AUTOVUE-ROLL"/>
<!-- <AppenderRef ref="AUTOVUE-FILE"/> -->
<!-- <AppenderRef ref="AUTOVUE-DAILY"/> -->
<AppenderRef ref="AUTOVUE-CONS"/>
</root>
```

G.3.2 Logger Information

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

Table G–4 Logger Information

Class	Description
com.cimmetry.jvueserver.management	Displays information relating the start-up of the AutoVue server, communications between the AutoVue server clusters and connections from the console, and other server management-related reports.
com.cimmetry.jvueserver.licusage	Displays information related to the usage of the AutoVue server (opening and closing sessions and documents).
com.cimmetry.jvueserver.configuration	Displays reports on loading errors of the server's configuration.
com.cimmetry.jvueserver.event	Displays information concerning posting and handling of different server events (opened and closed sessions, opened and closed documents, and so on).
com.cimmetry.jvueserver.cache	Displays information concerning the server's cache. Reports messages and errors related to loading the cache, locking, saving, deleting cached files as well as searching for archive and XRef files.
log4j.category.com.cimmetry.connection	Displays information concerning downloading files from the network.

Table G–4 (Cont.) Logger Information

Class	Description
com.cimmetry.jvueserver.session	Displays reports on sessions opening, closing and being restored, and the loading and saving of session profiles.
com.cimmetry.jvueserver.document	Displays document-related information (open, information, properties, and so on).
com.cimmetry.jvueserver.document.native	Displays messages and error reporting for document related native code execution.
com.cimmetry.jvueserver.dms	Displays DMS-related operations (open, download, save, properties, and so on).
com.cimmetry.jvueserver.streamingfile	Displays information concerning generation and usage of streaming files.
com.cimmetry.jvueserver.collaboration	Displays all server-side collaboration activity.
com.cimmetry.jvueserver.collaboration.rmi	Displays all messages regarding RMI connection handling for real-time collaboration.
com.cimmetry.jvueserver.console	Displays messages on server console loading, connecting information and Server Console Frame errors.
com.cimmetry.jvueserver.symbols	Displays mapping between the symbol library name in the UI and the actual SMB file name on the AutoVue server.

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

Table G–5 Information Levels

Information Level	Description
OFF	Turn off all logging.
FATAL	Logs severe events that could cause the application to abort.
ERROR	Logs error events that might still allow the application to continue running.
WARN	Logs potentially harmful situations. This is the default logging level.
INFO	Logs informational messages that highlight the progress of the application at coarse-grained level.
DEBUG	Logs fine-grained informational events that are most useful to debug an application.

Note: If you need more specific error messages, you can turn on verbosity for specific classes.

For Example:

```
<logger name="com.cimmetry.jvueserver.management">
<level value="info"/>
<logger name="com.cimmetry.jvueserver.session">
<level value="warn"/>
<logger name="com.cimmetry.jvueserver.document">
<level value="error"/>
<logger name="com.cimmetry.jvueserver.dms">
<level value="fatal"/>
```

These four lines mean that informational messages will be logged for the management class, warning messages will display for the session class, error messages pertaining to document requests will display for the document class. For the `com.cimmetry.jvueserver.dms` package, fatal messages will be reported.

Refer to the Apache Web site and log4j documentation for more information.

Customizing the AutoVue Client

AutoVue allows you to customize the client application and graphical user interface (GUI). For example, you can change the locale for the AutoVue client or you can customize the AutoVue user interface by modifying the menu and the toolbars.

Note: AutoVue client components (jvue.jar, jogl.jar, gluegen-rt.jar and jsonrpc4j.jar) include the Trusted-Library manifest attribute to satisfy the changes made in Java 7 Update 21 (7u21). As a result, when deploying on Java 7u21 or later, you must set the Trusted-Library manifest attribute in your customized JAR files.

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

H.1 AutoVue Client Parameters

AutoVue allows you to customize the client. For example, with the *GUIFILE* parameter you can configure its user interface (GUI), or with the *DMS* parameter you can specify the DMS servlet that the AutoVue server uses to interface with a DMS.

AutoVue provides two mechanisms to set these client parameters. Following are the two mechanisms to set the client parameters:

- At the startup stage, as command-line parameters of the client Java program
- At the initialization stage, with initialization parameters *INIT_PARAMS* passed to the constructor of AutoVue JavaScript object during its instantiation

Start-up Stage – In order to run AutoVue from the browser context, you need to generate a JNLP file for it as described in [Deploying JNLP Components](#). The program start-up parameters must be specified in the application-desc portion of the JNLP file as provided in the following code:

Code to include start-up parameters

```
<application-desc main-class="com.cimmetry.jvue.JVueApp">
  <argument>-paramName1=Value1</argument>
  <argument>-paramName2=Value2</argument>
  ...
</application-desc>
```

Where each parameter is specified by its name and its value, with the tag "-param" (not followed by a space) within a block `<argument>...</argument>`.

If you use *VueJNLPServlet* to generate the required JNLP file, add the code block provided in [Code block to be included in web.xml file](#) to the deployment descriptor *web.xml* file within the initialization parameters of *VueJNLPServlet*, for each AutoVue client parameter that you want to pass:

Code block to be included in web.xml file

```
<init-param>
  <param-name>[Parameter Name]</param-name>
  <param-value>[Parameter Value]</param-value>
</init-param>
```

Initialization stage – After you start AutoVue client with Java Web Start framework, it launches a socket on the first available port within the list of ports given at the start-up stage. The framework is triggered through file association protocol when the required JNLP file is downloaded. The socket waits for an initialization request to establish the connection with custom client. The initialization request must carry all the information required to build the client if not provided at the start-up stage (the Server URL, GUI specifications, etc). This information can be passed using the constructor argument *INIT_PARAMS* of AutoVue JavaScript Object (See [Installing AutoVue Client Components](#)). This must be provided in a the following *JSON* format of a JavaScript object:

```
INIT_PARAMS={'Name1':'Value1', 'Name2':'Value2', ..., 'NameN':'ValueN'}
```

Note: Some important points to consider are as follows:

- Client parameters values set at start-up stage take precedence on the ones set at initialization stage.
 - Before the initialization stage, AutoVue shows a small “*AutoVue Launcher*” frame informing the user that it is waiting for a browser connection.
-
-

The [Table H-1, “Customizable Parameters”](#) describes the customizable parameters in the AutoVue client.

Table H-1 Customizable Parameters

Name	Value	Description
ALLOWEDFORMAT CLASSES	[Office 2D 3D EDA]	<p>You can use this parameter to override the formats allowed by the server. For example, if the parameter is set to Office;2D, user will only be able to view Office and 2D formats. User will not be allowed to view EDA or 3D even if the server supports these formats.</p> <p>The value is a semicolon-separated list.</p> <p>For example: ALLOWEDFORMATCLASSES=Office;2D</p> <p>Note: This parameter can only be used to restrict the formats, not allow more formats than the server license allows.</p>

Table H–1 (Cont.) Customizable Parameters

Name	Value	Description
COOKIES	Semi-colon separated list of key-value pairs	<p>Specify a list of cookies to AutoVue. AutoVue client will apply the filter specified by the parameter DMS_PRERESERVE_COOKIES to select the ones it should pass the AutoVue server and integration components. The expected format is:</p> <pre>cookie1=value1;cookie2=value2,...;cookieN=valueN</pre> <p>In order to comply with security requirements, AutoVue supports encrypted cookies as well. AutoVue expects the encrypted cookies to be encoded with Base64 and serialized in HEX format.</p> <p>The maximum size of data which can be encrypted with RSA is 245 bytes. Some integration solutions may require a long list of cookies that needs more. The approach supported by AutoVue to encrypt the cookies is:</p> <ol style="list-style-type: none"> 1. Generate an AES symmetric key. 2. Encrypt the list of cookies with the AES, then encode the result with Base64 and serialize it into HEX format. 3. Generate an RSA key-pair. Only RSA encryption and "RSA/ECB/PKCS1Padding" ciphers are currently supported. 4. Encrypt the AES key with RSA, then encode the result with Base64 and serialize it into HEX format. 5. Append the encoded AES key to the cookies using the separator '-', in the form: cookies-key <p>If you send the cookies to AutoVue in encrypted format, then you must also send the RSA private key as a client parameter (See the parameter ECRYPTIONKEY).</p>
COLLABORATION		<p>The parameters and values described here are set automatically when initiating and joining collaboration sessions from the AutoVue client.</p> <p>Note: These parameters are read only after a file is set (SETFILE operation).</p>
	INIT:	Initiate collaboration session.
	CSI_ClbSessionID=987654321	DMS collaboration session ID.
	CSI_ClbDMS=dmsIndex	DMS index.
	CSI_ClbSessionData=123456789	DMS collaboration session data

Table H–1 (Cont.) Customizable Parameters

Name	Value	Description
	CSI_ClbSessionSubject=Subject	Collaboration session subject.
	CSI_ClbSessionType=public private	Collaboration session type.
	CSI_ClbUsers=user1, user2,...	Invited users.
	JOIN:	Join collaboration session in progress.
	CSI_ClbSessionID=987654321	DMS collaboration session ID.
	CSI_ClbDMS=dmsIndex	DMS index.
	CSI_ClbSessionData=123456789	DMS collaboration session data.
DMS	<i>http://name:port/dmsServlet</i>	Specifies the DMS servlet that the AutoVue server uses to interface with a DMS.
DMSARGS	<i>String</i>	<p>List of DMS arguments passed in as Client parameters. Specify semicolon separated list of client parameters. The value will be sent with every request to the DMS.</p> <p>Example:</p> <pre><PARAM NAME="DMSARGS" VALUE="ARG1;ARG2"> <PARAM NAME="ARG1" VALUE="value1"> <PARAM NAME="ARG2" VALUE="value2"></pre>
DMS_PRESERVE_COOKIES	[TRUE FALSE semi-colon separated list of cookies]	<p>Set this parameter to TRUE if you want AutoVue client to pass on all cookies to the AutoVue server and integration components. Set to FALSE if you do not want AutoVue client to pass on any cookies.</p> <p>Specify a semi-colon separated list of cookies that the AutoVue client should pass on to the AutoVue server and integration components. If your integration relies on a cookie, it is recommended that you set DMS_PRESERVE_COOKIES to the specific cookies needed for the integration instead of setting it to TRUE.</p> <p>Default: FALSE</p> <p>Example:</p> <pre><PARAM NAME="DMSARGS" VALUE="DMS_PRESERVE_COOKIES"> <PARAM NAME="DMS_PRESERVE_COOKIES" VALUE="TRUE"></pre>
ENCRYPTIONKEY	Base64 string HEX-serialized	Specify the private RSA key used to encrypt the cookies (See the parameter COOKIES). It must be encoded with Base64 and serialized in HEX format.

Table H-1 (Cont.) Customizable Parameters

Name	Value	Description
EXTRABUNDLES	<i>name of the bundle file</i>	<p>If you are adding custom actions to AutoVue, you can specify the name of the custom resources file using this parameter. Names of the custom resource files are expected to follow: filename_XX.properties, where XX is a two-character representation of a language.</p> <p>When specifying the custom resources using this parameter, do not specify the language and the extension.</p> <p>For example:</p> <pre><PARAM NAME="EXTRABUNDLES" VALUE="CustomActions"></pre>
FILENAME	URL	<p>Set it to the file to be opened at the client's start-up.</p> <p>Note: This parameter should not be set if opening a file through Javascript.</p> <p>Note: To open multiple files, specify a semi-colon separated listed of files.</p> <p>Will be understood as a client local file to be uploaded on the server to be viewed.</p> <p>For example:</p> <pre>upload://dir/.../file</pre> <p>Specify a HTTP/HTTPS URL for file open.</p> <p>Examples:</p> <pre>http://host/file ftp://host/file or ftp://<user>:<password>@ <ftpserver>/file</pre>
GUIFILE	<i>String</i>	<p>The Graphical User Interface (GUI) definition file used. Using this parameter, Web servers can customize the GUI of the AutoVue client according to client credentials. GUI files are stored in subdirectories of the root directory specified in the jvueserver.users.directory parameter of the jvueserver.properties file. The specification can also specify a local file using the "file://" convention.</p> <p>Note: If the GUIFILE parameter is not specified, the default AutoVue GUI is used.</p> <p>Default for the jvueserver.users.directory parameter is <i><bin dir>\Profiles</i>.</p> <p>For more information, refer to Customizing the GUI.</p>

Table H-1 (Cont.) Customizable Parameters

Name	Value	Description
HEAVYWEIGHT	[TRUE FALSE AUTO]	<p>Specify if you would like to use JOGL's heavyweight or lightweight widget to render 3D Models. When heavyweight is on, AutoVue uses hardware acceleration to render 3D.</p> <p>The default is AUTO which has the following behavior depending on the client platform:</p> <ul style="list-style-type: none"> ■ MAC: Use lightweight widget. ■ Other platforms: Check if the INI option with the same name (HEAVYWEIGHT) is set. If it is set, apply it, otherwise; use heavyweight rendering on all operating systems except Windows 7 and up.
JSONRPC_PORT	Semi-colon separated list of integers and integer intervals in the format min-max	<p>Specify a list of localhost ports to be used by the client to open a socket on which it can listen to XML HTTP requests in order to run scripting commands. It will use the first port in the list that is available.</p> <p>The expected format is a list of port numbers and/or port intervals.</p> <p>Example: 2345;7500-7510;8500-8510,8888</p> <p>The parameter value above asks AutoVue client to try to port 2345, then 7500, 7501, ..., 7510, 8500, 8501, ..., 8510, 8888, and use the first one available.</p> <p>NOTE: To be used this parameter must be sent at the start-up stage. Otherwise, AutoVue will assume that all the parameters required to build the client, as Server URL and GUI specifications, are already provided at start-up stage, and goes building the client. No communication or scripting will be possible with the client, in this case.</p>
JVUESERVER	Semicolon-separated list.	<p>Specify the list of VueServlet URLs to the AutoVue servers. Separate multiple values with a semi-colon.</p> <p>If multiple servers are listed, the client attempts them in a left-to-right order.</p> <p>Example:</p> <p>http://AutoVueServer:7001/servlet/VueServlet</p>
LISTUSERS	[TRUE FALSE]	<p>Show list of users connected to the AutoVue server when initiating a collaboration session or when inviting users to a collaboration session.</p> <p>The list of users is shown in the initiate session dialog and in the invite users dialog boxes. If this parameter is set to FALSE, then the list of users is not shown.</p> <p>Default: TRUE</p>

Table H-1 (Cont.) Customizable Parameters

Name	Value	Description
LOCALE	[DE EN FR JA KO NL TW ZH SV]	<p>The Locale to be used in the user interface, specified as an ISO639 two-letter code.</p> <p>Using this parameter, Web servers can force the AutoVue client GUI to be displayed in one of the supported languages. If not set, the Locale is determined using the client system properties.</p>
LOADHIDDEN	[TRUE FALSE]	Specify whether to open the main AutoVue frame hidden. Secondary frames will open visible though. The default value is "FALSE"
LOADMINIMIZED	[TRUE FALSE]	Specify whether to open the AutoVue frames that are minimized. The default value is "FALSE"
LOGFILE	String	<p>Specify full path to the log file for messages. <i>null</i> is for standard output.</p> <p>Example: C:\temp\clientlog.txt</p> <p>Default: null</p>
SWINGLAF	[String NULL]	<p>Specify a look and feel for Swing. For example, com.java.swing.plaf.motif.MotifLookAndFeel.</p> <p>If null, platform's default look and feel will be used, obtained by UIManager.getSystemLookAndFeelClassName().</p> <p>Default: null</p> <p>Note: There are several Look and Feels available for Swing. Make sure to test the Look And Feel that you plan to use before you deploy it to production.</p> <p>On some Linux clients, you may need to set the LAF to the "Metal" look and feel (javax.swing.plaf.metal.MetalLookAndFeel) to have the AutoVue client working correctly.</p>
TICKET	String	Arbitrary string used to authenticate the call to <i>init</i> API. To use it, the parameter, must be provided at the start-up, and then attached to init API call as an XML HTTP Request header (<i>AV-Session-Ticket</i>). It can be provided to prevent cross session connection between AutoVue client and a custom client.
VERBOSE	FALSE ERROR INFO DEBUG ALL	<p>Set to ERROR to output all error messages.</p> <p>Set to INFO to display all informative messages.</p> <p>Set to DEBUG to display all debug messages.</p> <p>Set to ALL to display all messages.</p> <p>Set to FALSE to turn off verbosity after client is initialized.</p> <p>Default: FALSE</p> <p>Note that during initialization, messages are logged as ERROR level by default.</p>

H.1.1 COLLABORATION Client Parameter Examples

The following Real-Time Collaboration (RTC) examples show how to set parameter and values for the COLLABORATION parameter:

The following information is needed for hosts to launch AutoVue to initiate a RTC:

- DMS is the URL for the DMS servlet (main class).
- MEETINGID is a number identifying the RTC meeting and holds the same value as CSI_ClbSessionID mentioned below.
- CSI_ClbSessionData can hold more information in addition to the CSI_ClbSessionID (MEETINGID), but your integration should know how to parse the CSI_ClbSessionData to retrieve the CSI_ClbSessionID.
- CLBUSERS are comma separated strings that represent the list of attendees who have been invited to the RTC by AutoVue.

Note: This value is not supported by the current AutoVue server.

- FILENAME is a file among the list of documents intended to be reviewed during the RTC meeting.

```
var session = 'CSI_ClbDMS=' + DMS + ';' +  
  'CSI_ClbSessionData=' + MEETINGID + ';' +  
  'CSI_ClbSessionSubject = DemoRealTimeCollaboration;' +  
  'CSI_ClbSessionType = public;' +  
  'CSI_ClbUsers = ' + CLBUSERS + ';;';
```

When instantiating an AutoVue JavaScript Object (call it **myAvApp**) to initiate RTC for the first time, the client parameters *FILENAME* and *COLLABORATION* should be provided using one of the two methods described in the previous section and then invoke the start API method to launch AutoVue client as described in [Installing AutoVue Client Components](#). For example, to set them at the initialization stage, you need to include them in the argument *INIT_PARAMS* provided to AutoVue JavaScript object constructor:

```
INIT_PARAMS={..., 'FILENAME':FILE_NAME, 'COLLABORATION': 'INIT:' + session,...}
```

When reusing an AutoVue JavaScript object for RTC, the following needs to be set using AutoVue scripting APIs. *FILE_NAME* is the new file to collaborate on.

```
myAvApp.setFile(FILE_NAME);  
myAvApp.collaborationInit(session);
```

Guests Join Real-Time Collaboration

The following information is needed for guests to launch AutoVue to join a RTC. Note that only *CSI_ClbDMS* and *CSI_ClbSessionData* are needed.

```
var session = 'CSI_ClbDMS=' + DMS + ';' +  
  'CSI_ClbSessionData=' + MEETINGID + ';;';
```

When instantiating an AutoVue JavaScript Object for joining a RTC for the first time, the COLLABORATION parameter should be provided. There is no need for FILENAME parameter.

```
INIT_PARAMS={..., 'COLLABORATION': 'INIT:' + session,...}
```


H.1.2 Scripting AutoVue Client

This section provides the basic definition needed to start an AutoVue client, and the advanced scripting functionality, using the JavaScript API provided within AutoVue JavaScript Object (autovue.js).

H.1.2.1 Basic Client

The basic JavaScript code needed to start an AutoVue client is:

```
<script language="JavaScript">
    /* The JVUESERVER parameter specifies a semi-colon separated list of connection
    methods to use to communicate with the AutoVue server. Below: the client
    tunnels through the Servlet installed under
    http://www.webserver.com/Servlet/VueServlet */
    var JVUESERVER = 'http://www.webserver.com/Servlet/VueServlet';

    /* This specifies the servlet connection to a JNLP file generator. Review section
    "Deploying JNLP Components" for more information about this component. Below used
    a VueJNLPServlet deployed on http://www.webserver.com */
    var JNLP_HOST = 'http://www.webserver.com/Servlet/VueJNLPServlet';

    /* This specifies the location of jvue.jar, jogl.jar, gluegen-rt.jar and
    jsonrpc4j.jar. The WEB Browser will download them from this location */
    var CODEBASE = 'http://www.webserver.com/AutoVue';

    /* This specifies the list of ports and port intervals to try for communication
    between the browser and AutoVue Client. The first one available within the list
    will be used. Below, the port 2345 will be tried, then 7500, 7501, ..., 7510, 8500,
    8501, ..., 8510, and finally 8888*/
    var CLIENT_PORTS = [2345, [7500, 7510], [8500, 8510], 8888];

    /* The VERBOSE parameter is optional. It indicates the level of diagnostic output
    messages to appear on the Browser's Java Console */
    Var VERBOSE='info';

    // Initialization Parameters
    var INIT_PARAMS = {'JVUESERVER':JVUESERVER,'VERBOSE':VERBOSE};

    // Instantiate an AutoVue JavaScript Object
    var myAvApp = new AutoVue(JNLP_HOST,CODEBASE,CLIENT_PORTS,INIT_PARAMS);
    // Start AutoVue Client
    myAvApp.start();
</script>
```

H.1.2.2 Advanced Scripting Functionality

When integrating the AutoVue client in dynamic Web pages, a public scripting API is provided within the interface of the AutoVue JavaScript Object. It allows the browser to interact with AutoVue client to set files, switch pages, open markups, etc. It mirrors the scripting API in the main Java client class JVueAPP.

Note: For more information about AutoVue JavaScript API, refer to "JavaScript API" section of the API Guide.

Example H-1 Script Language

```
<script language="JavaScript">
    // Open the specified document
    myAvApp.setFile('http://www.machine.com/AutoVue/samples/acad12.dwg');
```

```
// Switch to the second page
myAvApp.setPage(2);

// Load all markups
myAvApp.openMarkup('*');

/* Print the current page in force-to-black mode */
myAvApp.printFile({'forceToBlack':true, 'pages':{'choice':1}});
</script>
```

H.2 Customizing the GUI

This section discusses the GUI file to choose and modify.

H.2.1 Choosing the GUI File

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

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

H.2.2 Modifying the GUI

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

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

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

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

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

H.2.2.1 Role-Based GUI

Using the GUI customization capability, you can create a role-based UI when integrating AutoVue with a third-party application. Depending on the role assigned to the user, a subset of the functionality AutoVue offers may only be required.

The recommended approach is to create multiple GUI files where each one presents the user interface needed for a specific role. The integration to the third-party system determines the role of the user (using a connection to an LDAP server, for instance) and then selects the GUI file to show the user the role-based UI.

Refer to the following sections for information on how the integration defines and uses the GUI files.

H.2.2.2 Structure and Syntax of GUI Files

The GUI definition file describes which controls (corresponding to available actions in AutoVue, like Rotate, Open, and so on) are to be added to which context (like MenuBar, ToolBar, and so on), thus allowing users to have complete control over the functionality of the AutoVue interface.

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

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

The [Table H-2, "GUI Keyword"](#) lists each GUI keyword for each mode:

Table H-2 GUI Keyword

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

H.2.2.3 GUI Configuration Syntax

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

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

Table H-3 GUI Configuration Blocks

GUI Configuration Blocks

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

*UI_mode_configuration =
{menu_bar_configuration | {toolbar_configuration}* |status_bar_
configuration|RMB_popup_menu_configuration}
```

Table H–3 (Cont.) GUI Configuration Blocks**GUI Configuration Blocks**

```

menu_bar_configuration =
MENUBAR BEGIN {popup_menu_configuration}* END

toolbar_configuration =
TOOLBAR NORTH|WEST|EAST BEGIN {button_control}* END

status_bar_configuration =
STATUSBAR SOUTH BEGIN {button_control|pane_control} * END

RMB_popup_menu_configuration =
RMB BEGIN {popup_menu_configuration|menu_item_control}* END

popup_menu_configuration =
POPUP IDS_{FILE |EDIT |VIEW |OPTIONS |HELP |MANIPULATE |ANALYSIS |MARKUP
|COLLABORATION} BEGIN {popup_menu_configuration|menu_item_control|
SEPARATOR }* END

button_control =
BUTTONAction_control

menu_item_control =
MENUITEMAction_control

pane_control =
PANEAction_control

action_control =
control_name,control_key_list,permissions

control_name: For list of available control names refer to Control Names.

control_key_list: For the control key list for different controls refer to Control Names.

permissions: All action names need "PERM_READ ".

```

These are the exceptions to this rule:

```

VueActionFilePrint needs: PERM_READ|PERM_HEADERS|PERM_WATERMARK
EcadActionSelect needs: PERM_HIDE
SMActionSelect needs: PERM_HIDE

```

Example:

To define a very basic user interface that only allows users, through menu items, to open or print a file and get the file information without changing watermark/headers/footers:

```

BEGIN UI VIEW
  MENUBAR BEGIN
    POPUP IDS_FILE BEGIN
      MENUITEM VueActionFileOpen, , PERM_READ
      MENUITEM VueActionFileProperties, , PERM_READ
      MENUITEM VueActionFilePrint, , PERM_READ
    END
  END
END

```

H.2.2.4 Control Names

The following table lists available Control Names and their functionality.

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

V - View

C - Compare

M - Markup

Table H-4 Control Names

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

Table H-4 (Cont.) Control Names

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

Table H-4 (Cont.) Control Names

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

The columns indicate:

- **Control Name:** Column shows the list of available control names.
- **UI modes:** Column specifies in which modes (View, Markup, and Compare) the control names can be used.
Example: **VueActionFileOpen** can be added to View and Compare Modes, but not Markup mode.
- **Functionality:** Column specifies which functionalities are provided when this control is added to a context.
Example: Adding **VueActionFileMarkup** to any context enables you to switch to Markup mode.
- **Control key list:** Column provides the optional functionalities that can be added to a context.
 - If there is no entry for a control name in this list, it means that there is only one action to invoke. A list specifies sub-actions. For example, for **VueActionFileOverlays**, there is no entry in the control key list, so adding it to a popup menu will provide both select and modify functionalities for overlays. The entry will look like this:

```
MENUITEM VueActionFileOverlays, , PERM_READ
```

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

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

Whereas

```
BUTTON VueActionViewZoom, , PERM_READ
```

is interpreted as

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

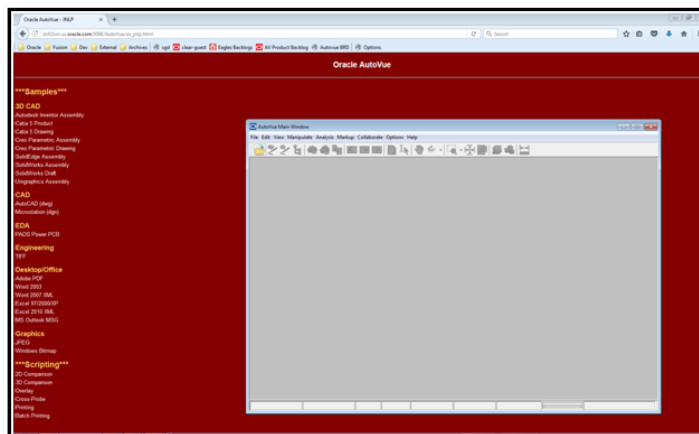
- **Contexts:** Column provides the contexts to which you can add the control to.
Example: You can add an entry in a popup menu of the menu bar, but not in an RMB configuration

H.3 Customizing the Example AutoVue Client Pages

The AutoVue client can be customized by setting parameters in the *av_jnlp.html* file located in the <AutoVue Install Root>\html directory. The HTML code in *av_jnlp.html* holds the "INIT_PARAMS" variable with the customizable parameters and provides a JavaScript method called *setFile()* to allow *av_jnlp.html* to dynamically change the file displayed in AutoVue. For more information, see [Scripting AutoVue Client](#). For a list of parameters you can set, refer to [AutoVue Client Parameters](#).

Note: During installation, if you select the Example Client Application, you can view a test HTML page *av_jnlp.html*. The test HTML page is strictly a sample Web page and is not a required component for your deployment. This sample Web page does, however, provide a good example of how to configure AutoVue Client. The file *av_jnlp.html* appears only if you choose to install the sample files during AutoVue installation.

Figure H-1 AutoVue Client



AutoVue Server Configuration Options

You can configure the AutoVue server by modifying `javueserver.properties` located in the `<AutoVue 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.

I.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 I-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

I.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`.

I.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 I-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 <code>processPoolSize</code> 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 <code>processPoolSize</code> 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

I.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 I-3 Process Pool Size Option

Parameter	Description	Default
<code>javueserver.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 rule of thumb, 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 `javueserver.properties`.

I.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 `javueserver.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.

```
javueserver.http.proxyhost=my.proxyserver.com:80
javueserver.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.

I.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 I-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 I-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

I.7 DMS Options

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

Table I-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
dms.save.length= [TRUE FALSE]	Set to TRUE so that the multipart save requests contain content length. Note: We recommend that you do not modify this parameter.	FALSE

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

I.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 I-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 <code>jvueserver.collaboration.id.min + jvueserver.collaboration.id.range</code> 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

I.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 I-7 Log4j and Diagnostics Options

Parameter	Description	Default
jvueserver.log4j.configurea ndwatch = [TRUE FALSE]	Set this to TRUE 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.

I.10 Modified With JavaScript Option

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

Table I-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

I.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 I-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

I.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 I-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

I.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 I-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

I.14 DLL Version Option

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

Table I-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.	

I.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 I-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>

I.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 I-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 installation directory>\bin\Profiles

Table I-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 installation directory>\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 Installation Directory>\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

I.17 Markup Options

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

Table I-15 Markup Options

Option	Description	Default
juvueserver.markup.nativetype	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 I-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

I.18 Online Help Options

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

Table I-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."

I.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 I-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 I-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 I-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

I.20 Linux-Specific Options

The following section lists Linux-specific parameters that can be configured in jvueserver.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.

I.20.1 Preload Java Class Option

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

Table I-19 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>

I.20.2 Xvfb Options

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

Table I-20 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

I.20.3 WINE Options

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

Table I-21 WINE Options

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

I.20.4 OEM Copyright Notice

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

Table I-22 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.	

I.21 VueServlet Configuration Options

The following table describes VueServlet initialization parameters that can be set in webdefault.xml when deploying on Jetty. 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 I-23 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
JVueServer=[server hostnames]	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. 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. 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. Note: The port listed in this option should match the port listed in the jvueserver.socket.port option in the jvueserver.properties file.	local hostname:5099

Table I–23 (Cont.) VueServlet Initialization Parameters

Parameter	Description	Default
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

Deploying the VueServlet on Application Servers

The VueServlet allows the AutoVue client to communicate with the AutoVue server using HTTP tunneling. This has two advantages:

- The client and the AutoVue server can generally communicate across firewalls since the standard HTTP ports (for example, 80) are used.
- The client can be configured to use the HTTPS protocol to communicate with the VueServlet. This ensures that all communications are secure.

The AutoVue client encodes requests from the HTTP/HTTPS protocol and attempts to invoke the VueServlet on the specified server. The VueServlet decodes the parameters included in the request and forwards the request to the AutoVue server using a socket connection. The VueServlet also replies to the client machine using the same HTTP/HTTPS protocol. You can deploy the VueServlet with any application server you choose. For a list of application servers that are certified by Oracle, refer to [System Requirements](#).

The exact steps to set up the VueServlet on your application server depend on the software you are using. This section describes the steps to setup the VueServlet for several popular Application Servers/Servlet Engines. Generally, you can follow similar steps to deploy with any application server. Refer to your application server documentation for specific instructions.

For information on configuring the VueServlet, refer to [VueServlet Configuration Options](#).

J.1 Generic Steps to Deploy the WAR File

1. Launch the administrative console of your application server.
2. Select **Install a new Web application**.
3. Browse and select **VueServlet.war**.
4. Specify **VueServlet** for the context name.
5. Deploy **VueServlet.war**.

We provide you with instructions for deploying **VueServlet.war** with some application servers in the following section.

J.2 Deploying the WAR File with WebLogic

1. Logon to the Administrative Console for WebLogic.

2. Select **Deployments** from the tree.
3. Click on **Install**.
4. Browse to the folder containing VueServlet.war and select **VueServlet.war**.
5. Enter **VueServlet** for the **Application Name**.
6. Select the Server to which you wish to deploy VueServlet.
Example: myserver
7. Click **Activate Changes**.
8. Select **Deployments** again and select the VueServlet application.
9. Click **Start** and select **Servicing all requests**.

The application starts.

Once the deployment is successful, verify the deployment. To do so, connect to:

`http://<host name>:<port>/VueServlet/servlet/VueServlet`

where *<host name>* is the name of your Application Server host machine and *<port>* is the port your application server is running on.

J.3 Deploying the VueServlet with Tomcat

1. Copy vueservlet.war to your Tomcat *webapps* directory.
2. Restart Tomcat.

The VueServlet is deployed automatically.

J.4 Deploying the VueServlet on non-J2EE Application Servers

This section comprises of the following:

- [Setting up the VueServlet](#)
- [Deploying on Jetty](#)

J.4.1 Setting up the VueServlet

Below are generic instructions for deploying the VueServlet with a non-J2EE application server.

For Tomcat users, refer to [Deploying the VueServlet with Tomcat](#).

1. Copy the file vueservlet.jar to your Servlet Engine's servlet directory.
2. Add vueservlet.jar to your Servlet Engine's CLASSPATH.
3. Create an alias for VueServlet to com.cimmetry.servlet.VueServlet.
4. If your AutoVue server is running on a different machine, specify the init parameter *JVueServer* to be *my.jvueserver.com:5099* where *my.jvueserver.com* specifies the machine on which AutoVue server is running. 5099 specifies the socket port that the AutoVue server uses. If the server is using a different socket port, specify the correct socket port in parameter *JVueServer*.
5. For the changes to take effect, restart the servlet engine.

Note: The default socket port is 5099.

J.4.2 Deploying on Jetty

1. Add `VueServlet.jar` to Jetty's class path.
2. Edit `startjetty.bat` and add the full path to `VueServlet.jar` to the *CLASSPATH* variable.
3. Edit `webdefault.xml` and add the following:

```
<servlet id="VueServlet">
  <servlet-name>VueServlet
  </servlet-name>
  <servlet-class>com.cimmetry.servlet.VueServlet
  </servlet-class>
  <init-param>
    <param-name>JVueServer
    </param-name>
    <param-value>www.jvueserver.com:5099
    </param-value>
  </init-param>
  <init-param>
    <param-name>Verbose
    </param-name>
    <param-value>>false
    </param-value>
  </init-param>
  <init-param>
    <param-name>DebugLevel
    </param-name>
    <param-value>0
    </param-value>
  </init-param>
  <load-on-startup>0
  </load-on-startup>
</servlet>
```

4. Replace `www.jvueserver.com` with the name of the machine on which the AutoVue server is running. *5099* specifies the socket port that the AutoVue server uses. If the server is using a different socket port, specify the correct socket port.
5. Start Jetty and the AutoVue server.
6. Test that the VueServlet is installed properly; Open a Web browser and enter the URL to the VueServlet:

`http://<machine name>:5098/servlet/VueServlet`

Non-Interactive Installations

This chapter discusses the installation of AutoVue in a non-interactive mode.

K.1 Installation

To install AutoVue in non-interactive mode, you need to specify a configuration file that contains the required installation parameters. To do so, you must generate the configuration file manually. In the file, you must specify one of two sets for AutoVue server authentication protocol (Kerberos) and SSL.

Note: If you do not specify a set for Kerberos and/or SSL, the installation will abort.

Table K-1 Installation Parameters

Authentication	Set 1	Set 2
Kerberos (JAAS)	JVUESERVER_AUTH_CONFIG	JVUESERVER_AUTH_CONFIGURE_LATER=1
	JVUESERVER_AUTH_KRB5_REALM	
	JVUESERVER_AUTH_KRB5_KDC	
SSL	SSL_CA_CERTIFICATE	SSL_CONFIGURE_LATER=1
	SSL_IDENTITY_KEYSTORE	
	SSL_IDENTITY_KEYSTORE_PASSWORD	

For example, specifying the variables for Set 1 enables Kerberos authentication protocol. Where as, setting JVUESERVER_AUTH_CONFIGURE_LATER=1 disables Kerberos protocol. For more information on these parameters. Refer to [Silent Installation Parameters](#).

K.1.1 Sample Silent Installation for Windows OSes

```
#Specify Installation Directory
#-----
USER_INSTALL_DIR=C:\\Oracle\\AutoVue
#Select Shortcut Folder
#-----
USER_SHORTCUTS=C:\\Documents and Settings\\Administrator\\Start
Menu\\Programs\\Oracle AutoVue
#Select Features (Available: ProgFiles,UserDocs,Website,SampleFiles,APIEx)
#-----
CHOSEN_INSTALL_FEATURE_LIST=ProgFiles,UserDocs,Website,SampleFiles,APIEx
```

```
#Specify host name for AutoVue Server
#-----
JVUESERVER_HOST=avserver1
#Specify Kerberos set. By setting these parameters, Kerberos protocol is enabled.
#-----
JVUESERVER_AUTH_CONFIG=<Full path to JAAS login configuration file>
JVUESERVER_AUTH_KRB5_REALM=<realm>
JVUESERVER_AUTH_KRB5_KDC=<kdc>
#Specify SSL set. By setting these parameters, SSL is enabled.
#-----
SSL_CA_CERTIFICATE=<Full path to CA-issued certificate>
SSL_IDENTITY_KEYSTORE=<Full file path to the Identity JKS keystore>
SSL_IDENTITY_KEYSTORE_PASSWORD=<Specify the password for the Identity JKS
keystore>
```

K.1.2 Sample Silent Installation for Linux OSes

```
#Specify Installation Directory
#-----
USER_INSTALL_DIR=/home/apps/AutoVue
#Select Features (Available: ProgFiles,UserDocs,Website,SampleFiles,APIEx)
#-----
CHOSEN_INSTALL_FEATURE_LIST=ProgFiles,UserDocs,Website,SampleFiles,APIEx
#Specify host name for AutoVue Server
#-----
JVUESERVER_HOST=avserver
#Specify Kerberos set. By setting these parameters, Kerberos protocol is enabled.
#-----
JVUESERVER_AUTH_CONFIG=<Full path to JAAS login configuration file>
JVUESERVER_AUTH_KRB5_REALM=<realm>
JVUESERVER_AUTH_KRB5_KDC=<kdc>
#Specify SSL set. By setting these parameters, SSL is enabled.
#-----
SSL_CA_CERTIFICATE=<Full path to CA-issued certificate>
SSL_IDENTITY_KEYSTORE=<Full file path to the Identity JKS keystore>
SSL_IDENTITY_KEYSTORE_PASSWORD=<Specify the password for the Identity JKS
keystore>
```

K.1.3 Silent Installation Parameters

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

Table K-2 Silent Installation Parameters

Parameter	Description	Default Value
USER_INSTALL_DIR=<file path>	Specify the path where you want to install the AutoVue server.	
USER_SHORTCUTS=<file path>	Specify the shortcut path. Note: This parameter is only for Windows OS installations.	

Table K-2 (Cont.) Silent Installation Parameters

Parameter	Description	Default Value
CHOSEN_INSTALL_FEATURES=[ProgFile,UserDocs,Website,SampleFiles,APIEx]	Specify the features to install. The comma-separated list can contain the following features: ProgFile: Installs Oracle AutoVue. UserDocs: Installs AutoVue end-user documentation. Website: Installs AutoVue client components onto a Web server. SampleFiles: Installs sample files. APIEx: Installs examples of how Oracle AutoVue features can be added to third-party applications using APIs.	ProgFile, UserDocs, Website
JVUESERVER_AUTH_CONFIG=<file path>	Specify the full file path to the JAAS login configuration file.	
JVUESERVER_AUTH_KRB5_REALM=<realm>	Specify the realm for the Kerberos authentication protocol.	
JVUESERVER_AUTH_KRB5_KDC=<kdc>	Specify the Kerberos Key Distribution Center (kdc) for the Kerberos authentication protocol.	
JVUESERVER_AUTH_CONFIGURE_LATER=1	Specify whether to configure authentication later manually (JVUESERVER_AUTH_CONFIGURE_LATER=1) or to configure the Kerberos authentication protocol on installation (JVUESERVER_AUTH_CONFIGURE_LATER=0).	
JVUESERVER_HOST=<AutoVue Server host name>	Specify the AutoVue server host name.	
SSL_CA_CERTIFICATE=<file path>	Specify the full file path to the CA-issued certificate.	
SSL_IDENTITY_KEYSTORE=<file path>	Specify the full file path to the Identity JKS keystore.	
SSL_IDENTITY_KEYSTORE_PASSWORD=<password>	Specify the password for the Identity JKS keystore. The password must be at least 6 characters in length.	
SSL_CONFIGURE_LATER=1	Specify whether to configure SSL later manually (SSL_CONFIGURE_LATER=1) or to install SSL during AutoVue installation (SSL_CONFIGURE_LATER=0).	

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

For Windows OSes:

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

For Linux OSes:

```
InstallClientServer_lin.bin -i silent -f <full path to configuration file>
```

K.2 Uninstallation

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

For Windows OSes:

```
<AutoVue Install Root>\uninstall\uninstall.exe
```

For Linux OSes:

```
<AutoVue Install Root>/uninstall/uninstall
```

Configuring AutoVue Plug-in for Enterprise Manager

An AutoVue plug-in can be added to Oracle Enterprise Manager to enable monitoring of AutoVue servers.


L.1 Prerequisites

- Oracle AutoVue 21.0.1 Client/Server Deployment
- Oracle Enterprise Manager 11g or above

Important: The plug-in is configured to work with Jetty that is included with AutoVue server. If you plan to use the plug-in to monitor AutoVue server usage, make sure to start up Jetty that is included with AutoVue. You must set the `VueServlet` parameter `EnableEM` to **TRUE** in Jetty's `webdefault.xml`.

Since the plug-in reports sensitive information about the AutoVue server, it is recommended that the `VueServlet` instance that communicates with Enterprise Manager is secure and is within the intranet. This `VueServlet` instance should only be used for Enterprise Manager reporting.

L.2 Installing the Plug-in

1. Connect to Oracle Enterprise Manager from a Web browser.
2. Enter user login information.
3. From the Oracle Enterprise Manager Grid Control home page, click **Setup**. The Overview of Setup page appears.
4. From the Overview of Setup section, click **Management Plug-ins**. The Management Plug-ins page appears.
5. Click **Import**. The Import Management Plug-ins page appears.
6. To import the plug-in, click **Browse**.
7. Select the plug-in, `oracle_autovue.jar` located in the `bin` directory and then click **OK**.
8. Click the **Deploy** icon .
9. Click **Add Agents**. The Search and Select Agents page appears.
10. Click **Go**, select an agent from the results, and then click **Select**. The Deploy Management Plug-in: Select Targets page appears.
11. Click **Next**. The Deploy Management Plug-in: Review page appears.

12. Click **Finish**.
13. From the Oracle Enterprise Manager Grid Control home page, click **Setup**. The Overview of Setup page appears.
14. Click **Agent**. The Management Agents page appears.
15. Click the agent link **hostname:3872**.
16. From the Add drop-list, select **Oracle AutoVue** and then click **Go**. The Add Oracle AutoVue page appears.
17. Enter the following information:
 - A descriptive name in the Name field (for example, AutoVue server).
 - The name or the IP address of the machine where the VueServlet that enables EM is installed.
 - The servlet port number (for example, :5098) in the Oracle AutoVue Server Servlet Port field.
18. Click **OK**.
19. Edit Jetty's webdefault.xml and VueServlet parameter `EnableEM` and set it to **TRUE** (lines in bold below):

```
<servlet id="VueServlet">
  <servlet-name>VueServlet</servlet-name>
  <servlet-class>com.cimmetry.servlet.VueServlet</servlet-class>
  <init-param>
    <param-name>JVueServer</param-name>
    <param-value>mtloaga13.cimmetrysystems.com:5099</param-value>
  </init-param>
  <init-param>
    <param-name>Verbose</param-name>
    <param-value>>false</param-value>
  </init-param>
  <init-param>
    <param-name>EnableEM</param-name>
    <param-value>TRUE</param-value>
  </init-param>
  <init-param>
    <param-name>DebugLevel</param-name>
    <param-value>0</param-value>
  </init-param>
  <load-on-startup>0</load-on-startup>
</servlet>
```

20. Restart Jetty for the changes to take effect.
21. From Enterprise Manager's Monitored Targets section, click the created AutoVue target (for example, AutoVue server) to view AutoVue server status information.

Note: If `EnableEM` on the `VueServlet` is set to `FALSE`, the Oracle Enterprise Manager will not be able to retrieve the AutoVue server status and will keep pinging the server through the `VueServlet`. In this situation, you will see the following message on the *VueServlet:Enterprise Manager is not enabled*.

Samples and API Examples Included with AutoVue

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

M.1 API Examples

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

- VueActionSample

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

To use this sample:

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

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

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

- BatchSample

This provides an example of how to implement batch stamping using the VueAction. For more information on VueAction, refer to the *Oracle AutoVue API Guide* and the *VueBean JavaDocs*. To use this sample:

- Update the Java code as required and compile the sample
- Bundle the class file into BatchStamping.jar.
- Copy the custom GUI batch.gui file to <AutoVue Installation Directory>\bin\Profiles.
- Make sure your DMS is configured for stamps and that you have defined fixed-size stamps.
- Launch jVueBatchStamping.bat.
- Open a DMS file from the AutoVue window and enter Markup mode.
- A Batch Stamping button is available in the Markup toolbar.
- Click on Batch Stamping.
- A dialog appears showing the list of fixed-size stamps available. Click on the stamp you want to add.
- The selected stamp is placed on all pages of the document.

M.2 Sample Files

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

- Sample 2D, 3D, EDA, Office and Graphics files are installed at <AutoVue Install Root>\html\samples
- Sample web pages to invoke the AutoVue application are installed at <AutoVue Install Root>\html. You will need to update AutoVue first (JNLP_HOST) and second (CODEBASE) constructor parameters to the URLs of the VueJNLPServlet and the location of the client JAR files, respectively. You will also need to update the URL of VueKeyPairServlet in the line including it (if you use it). These updates need to be done in the file – av_jnlp.html, which is the web page to invoke AutoVue from the client.

Note: If a web server is detected on the machine where you install AutoVue, the samples and the web pages are copied to the web server doc root.

- A Batch Printing JavaScript is also installed at <AutoVue Install Root>\html. The files is batchPrint_jnlp.html. Make sure to update also AutoVue constructor parameters in order to use this sample.
- av_jnlp.html contains the following JavaScript samples:
 - 2D Comparison
 - 3D Comparison
 - Overlay
 - Cross-Probe
 - Printing

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

N.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`

To resolve this issue, you must install these packages.



Feedback

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

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

0.1 General AutoVue Information

Web Site <http://www.oracle.com/us/products/applications/autovue/index.html>

Blog <http://blogs.oracle.com/enterprisevisualization/>

0.2 Oracle Customer Support

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

0.3 My Oracle Support AutoVue Community

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

0.4 Sales Inquiries

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