AutoVue Integration SDK

Release 20.2.1
Overview

CONTENTS

<u>Preface</u>

The *AutoVue Integration SDK Overview* provides a general overview of AutoVue Integration SDK.

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 AutoVue Integration Software Development Toolkit (SDK) is intended for third-party developers who want to integrate Oracle AutoVue with their Document Management System (DMS) or Product Lifecycle Management System (PLM).

Related Documents

For more information, see the following documents in the Integration SDK documentation library:

- *Installation and Configuration Guide*
- *Design Guide*
- *User Guide*
- *Technical Guide*
- *Release Notes*
- *Java Docs*
- *Security Guide*

Conventions

The following text conventions are used in this document:

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

Introduction

The AutoVue Integration Software Development Toolkit (ISDK) is intended for third-party developers who want to integrate Oracle AutoVue with their Data Management System (DMS).

Oracle AutoVue is a thin client viewing and collaboration solution for enterprise-wide data access.

The ISDK installation package includes four sample projects that can be modified to suit your integration needs. The following sections describe these projects in more detail.

Getting Started

After you run the ISDK installer on your machine, it creates several folders. The **Quick Start.html** file found in the root folder is a top-level readme file that acts as an entry point to the rest of the documentation included in this ISDK. To view the content of this file, open it in your browser.

The following is a brief description of what is contained in each folder found in this ISDK:

- The **/docs** folder contains ISDK documentation (javadocs, installation guide, user guide, and so on).
- The **/FileSys** folder contains four subfolders:
  - The **/Repository** folder contains filesysRepository.zip which contains sample files used by the Filesys project - the Sample Integration for Filesys.
  - The **/OEVF** folder contains two GUI files used for the OEVF demo.
  - The **/WebApplication** folder contains a filesys.war file and a /filesys folder. The content in the /filesys folder is the unzipped version of the filesys.war file. The filesys.war can be imported into JDeveloper or Eclipse workspace to demo the Sample Integration for Filesys and to demo RTC & OEVF functionalities. The project contains source code for sample integration, AutoVue client and third party libraries required by the integration.
  - The **/ESAPI_Resources folder** contains the OWASP Enterprise Security API properties files: ESAPI.properties and validation.properties.
- The **/ISDKSkeleton** folder contains two subfolders:
  - The **/WebApplication** folder contains an isdk_skeleton.war file and a /isdk_skeleton folder. The content in the /isdk_skeleton folder is the unzipped version of the isdk_skeleton.war file. The isdk_skeleton.war can be imported into JDeveloper or Eclipse workspace to create the Integration SDK Skeleton project. Your integration with Java-based backend systems will be developed based on this skeleton project and fulfill the TODO comments in this project.
  - The **/ESAPI_Resources folder** contains the OWASP Enterprise Security API properties files: ESAPI.properties and validation.properties.
- The **/WebServicesIntegration** folder contains three subfolders:
  - The **WebServiceClient** folder contains the /ESAPI_resources and /WebApplication folders. The / ESAPI_Resources folder contains the OWASP Enterprise Security API properties files: ESAPI.properties and validation.properties. The /WebApplication folder contains the wsclient.war and a /wslcient folder which is the unzipped version of the WAR file.
  - The **/WSDL** folder contains the BluePrint WSDL file and the XSD file that accompanies it.
  - The **/WebServicesSampleServer** folder contains a /C# folder. The /C# folder contains the Service1.asmx.cs file and the zipped project template wsserver_VisualStudio2008_ProjectTemplate.zip. The Service1.asmx.cs file is used when creating an ISDK Web Services project manually.
- The **/etc** folder contains release notes, list of files and folders structure contained in this ISDK, and folders containing licenses of third-party software used by the ISDK.
- The **/_jvm** and **/_uninst** folders for uninstalling the ISDK.

Depending on the sample components you selected during installation, some additional configuration may be required.

For example:

If you installed the Sample Integration for Filesys project, then you must import the project into JDeveloper or Eclipse IDE, and then configure, deploy and run the sample integration. To do so, you must follow the steps outlined in the *Installation Guide*. You need to make sure that you have all the prerequisite software installed before you start deployment. For the complete list specific to your platform, refer to the "System Requirements" section found in the *Installation Guide*.

Once you have successfully deployed the Eclipse project, the next step is for you to get familiar with the sample integration. To learn more about the features and functionality provided by the sample integration, refer to the *Installation Guide* and *User Guide*.

Once you are familiar with the sample integration, the next step is for you to build your own integration. To do so, refer to the *Installation and Configuration Guide, Technical Guide* and *Javadocs.* The *Technical Guide* explains the technical details and provides step by step guidance for developing your own integration.

Overview of ISDK Components

The following sections describes the various components included in the AutoVue Integration SDK.:

Documentation

The documentation included in the AutoVue Integration SDK is under the **/docs** folder and contains following:

- **Installation and Configuration Guide**
  - This guide contains information related to the installation, configuration and deployment of projects included in this ISDK in JDeveloper and Eclipse IDE.
- **Security Guide**
  - This guide contains information related to the security and authentication mechanisms provided in this release of Integration SDK.
- **Design Guide**
  - This guide provides a high-level overview of the Integration SDK.
- **User Guide**
  - This guide contains information related to functionality available to the end user of the sample integration included in this ISDK.
- **Technical Guide**
  - This guide contains in-depth technical information about the integration framework and describes how to implement your own integration based on the sample integration included in this ISDK.
- **JavaDocs**
  - This contains the JavaDocs of the underlying framework contained in the AutoVue Integration SDK.

ISDK Skeleton Project

The ISDK Java skeleton package has the structure for building a new VueLink. The skeleton comes with a set of `TODO` comments in places where the integrators need to add their code. The ISDK Java skeleton implementation means adding code to the skeleton codebase so that it can communicate with the repository's Java API as shown in the following figure.
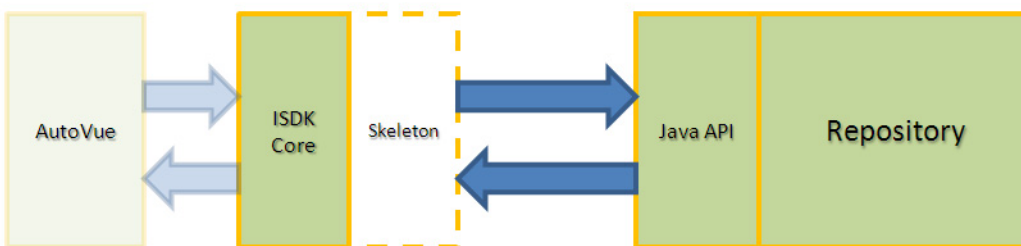
This project is located under the ISDKSkeleton/WebApplication folder and is available as single WAR (isdk_skeleton.war) as well as separate files. It can be imported into JDeveloper or Eclipse workspace.
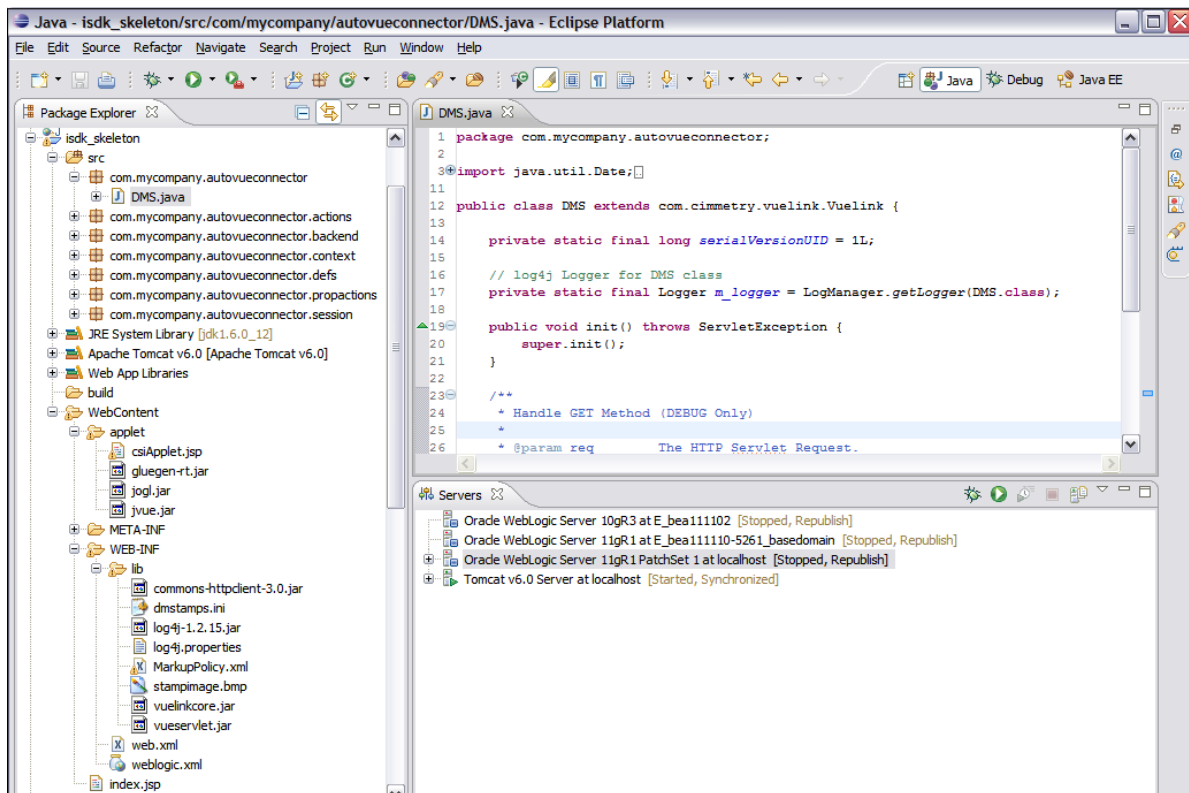
It contains four main subcomponents:

- Framework (VueLink Core): vuelinkcore.jar
- Integration Skeleton
- AutoVue Components:
  - AutoVue Applet Client (jvue.jar, jogl.jar, gluegen-rt.jar)
  - VueServlet tunneling servlet (VueServlet.jar)
- Third-Party Libraries:
  - log4j-1.2-15.jar
  - esapi-2.0.1.jar

Refer to section "Sample Integration" on page 18 for description about the two subcomponents: Framework and AutoVue Components.

It includes skeleton classes containing TODO tasks to realize the following functionalities:

- Document viewing
- Retrieve document attributes
- Create, save and review markups
- Compare document versions
- Convert documents to other formats
- Returning External References (XREFS)
- Browse DMS repository
- Search DMS repository
- Support for AutoVue in offline mode
- Support for Stamps markup entity
- Support for Set Property action
- Support for AutoVue authorization mechanism

- Support for integration between Online meeting managements and AutoVue Real-Time Collaboration (RTC)
- Support for markup save alert before applet close
- Enhanced framework to support Oracle Enterprise Visual Framework (OEVF)

The Integration Skeleton contains the following packages:

- com.mycompany.autovueconnector
- com.mycompany.autovueconnector.actions
- com.mycompany.autovueconnector.backend
- com.mycompany.autovueconnector.context
- com.mycompany.autovueconnector.defs
- com.mycompany.autovueconnector.propactions
- com.mycompany.autovueconnector.session

It includes a sample Front-UI file - applet/csiApplet.jsp - which contains HTML code for launching AutoVue applet (<applet> tag). When you develop your DMS extension on the DMS Server, you can customize this sample file.

Refer to the *Technical Guide* about steps to design your integration based the ISDK Skeleton.

ISDK Web Services Client

The Web service package includes a Web Services Description Language (WSDL) file that describes an interface for a Web service to be implemented by the repository. The package includes a client-side implementation of this WSDL. This client package itself is built using the ISDK skeleton. With the ISDK Web service package, the implementation means building a proper Web service provider based on the defined WSDL on the repository as shown in the following figure. This means more flexibility since the Web service provider can be implemented on any platform and with any programming language.
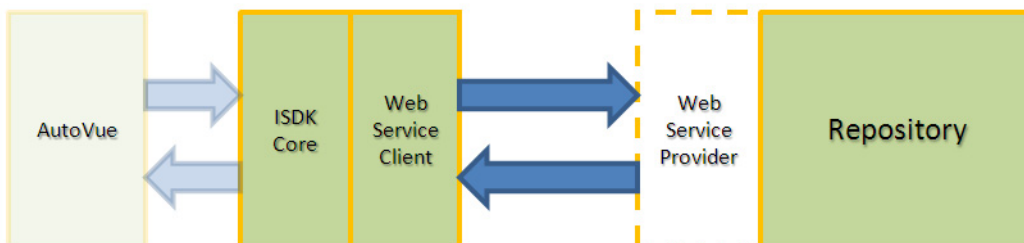
The ISDK Web Services Client project is located under the WebServiceClient/WebApplication folder and is available as a single WAR (wsclient.war) and separate files. It can be imported into JDeveloper or Eclipse workspace. The BluePrint.wsdl is located in WebServiceClient/WSDL folder.
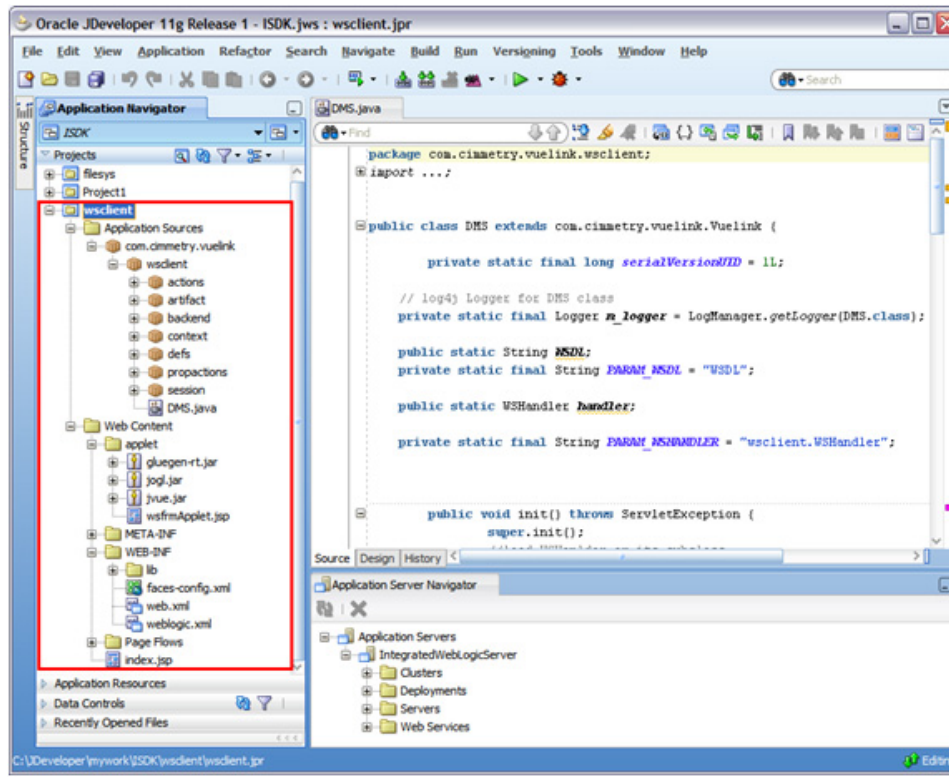
It contains also four main subcomponents:

- Framework (VueLink Core): vuelinkcore.jar
- Web Services Client
- AutoVue Components:
  - AutoVue Applet Client (jvue.jar, jogl.jar, gluegen-rt.jar)
  - VueServlet tunneling servlet (VueServlet.jar)
- Third-Party Libraries:
  - log4j-1.2-15.jar
  - esapi-2.0.1.jar

Refer to the section "Sample Integration" on page 18 for description about the three subcomponents: Framework, AutoVue Components and Third-Party Libraries.

It provides APIs to provide the same functionalities as the ISDK Skeleton does.

The Web Services Client project contains the following packages:

- com.cimmetry.vuelink.wsclient
- com.cimmetry.vuelink.wsclient.actions
- com.cimmetry.vuelink.wsclient.artifact
- com.cimmetry.vuelink.wsclient.backend
- com.cimmetry.vuelink.wsclient.context
- com.cimmetry.vuelink.wsclient.defs
- com.cimmetry.vuelink.wsclient.propactions
- com.cimmetry.vuelink.wsclient.session

It includes a sample Front-UI file - applet/wsfrmApplet.jsp - which contains HTML code for launching AutoVue applet (<applet> tag). When you develop your DMS extension on the DMS Server, you can customize this sample file.

Sample Projects

The installation of the ISDK includes two sample projects:

- Sample Integration for Filesys DMS
- Web Services Sample Server

These projects provide a good introduction to the many uses of the ISDK. The following sections provide an introduction to each project. For full configuration information, refer to the *Oracle AutoVue Integration SDK Installation and Configuration Guide*.

Sample Integration for Filesys Project

The AutoVue Integration SDK bundles a sample integration into a JDeveloper-based or Eclipse-based project. This project is located under the FileSys\WebApplication folder and is available as single web archive (WAR) filesys.war file and separate files. This provides you with an option of either importing the project from single WAR file into your workspace or manually creating a project and adding individual pieces to it.

The following high-level architectural diagram shows how various components included in the Sample Integration for Filesys are related to each other as well as to others. For detailed description about this architectural diagram, please refer to *Technical Guide*.
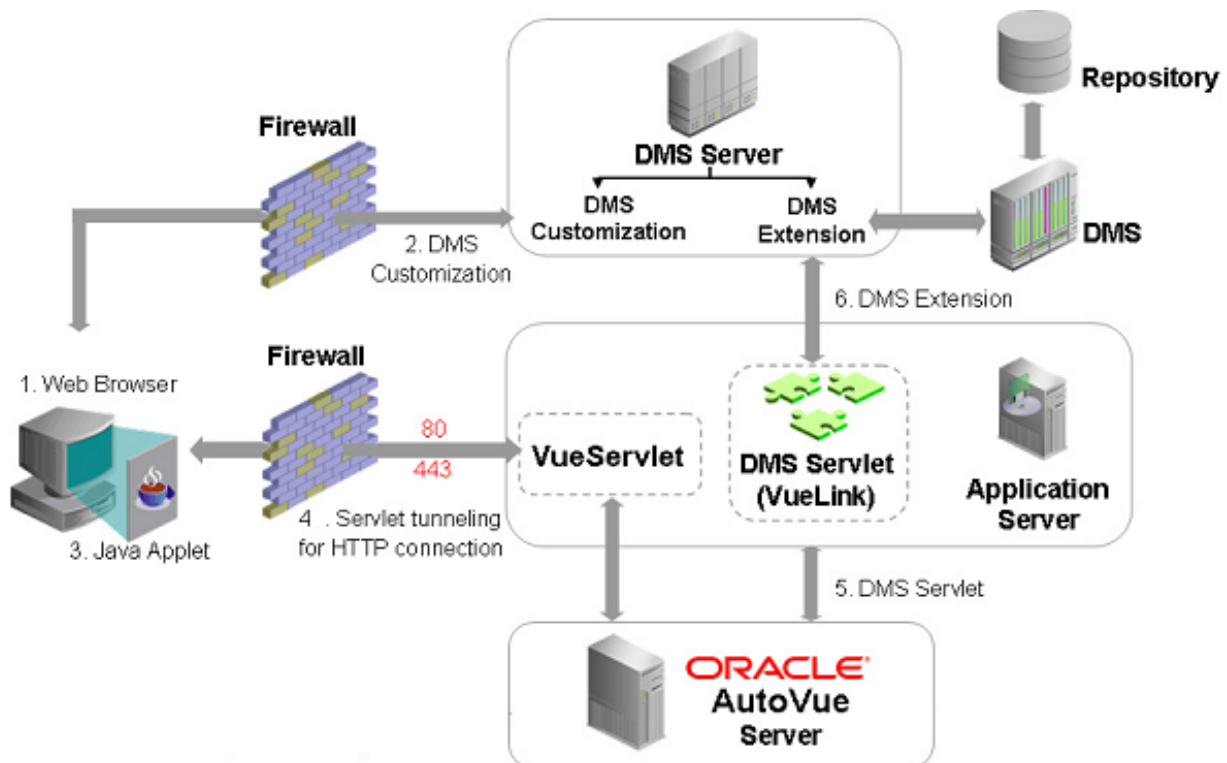
The following is a sample integration project on Eclipse.
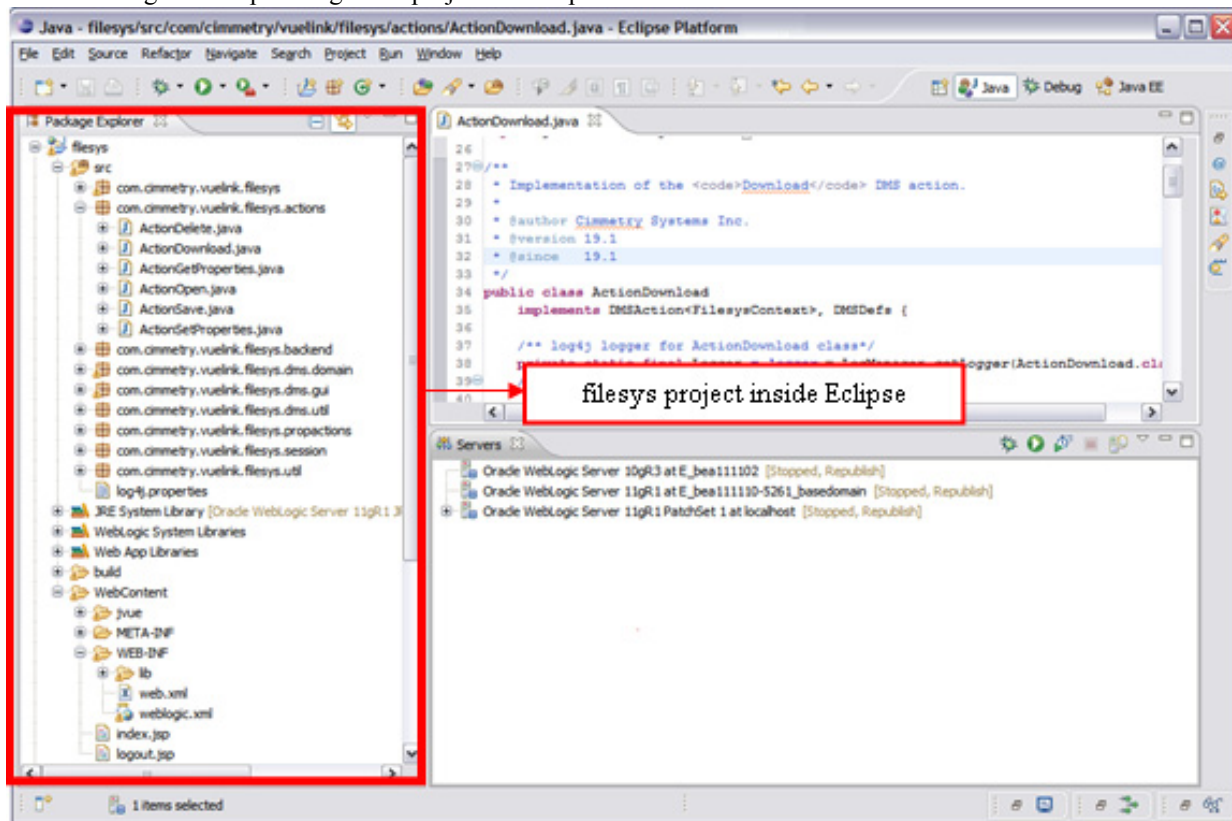
The following is sample integration project on JDeveloper.

The project contains four main subcomponents:

- "Framework (VueLink Core)" on page 17
- "Sample Integration" on page 18
- "AutoVue Components" on page 19
- "Third-Party Libraries" on page 20

Framework (VueLink Core)

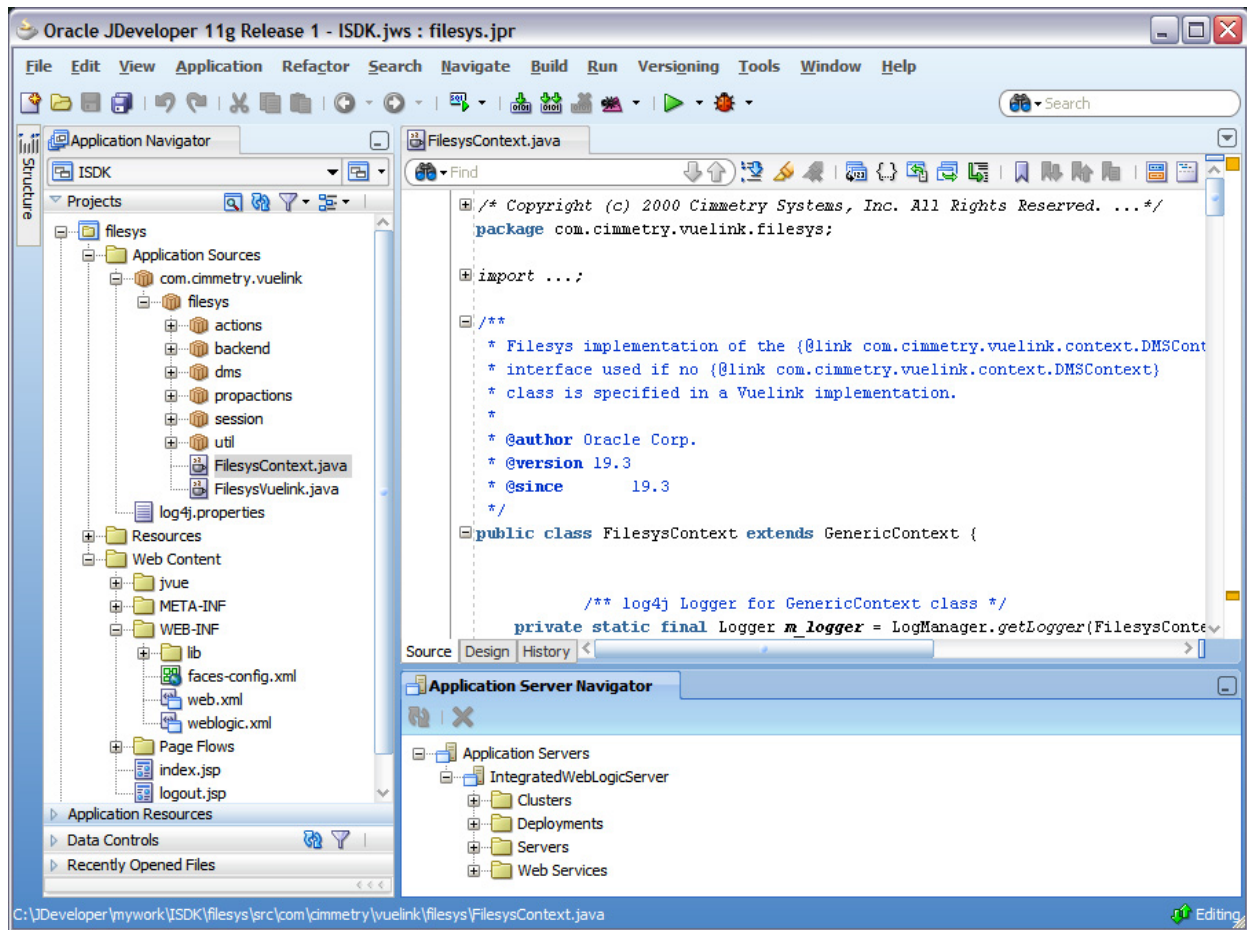The framework is implemented based on Java Servlet API provided by SUN as part of J2EE. The main class is called `com.cimmetry.Vuelink` which extends `javax.servlet.http.HttpServlet`. Servlets do not run on their own, they require a servlet engine such as Oracle WebLogic Server.

The framework links AutoVue system with a third-party DMS. Integration framework receives requests from AutoVue, obtains information from the DMS, and then builds a response back to AutoVue.

The framework provides plumbing for parsing XML requests received from AutoVue Server as well as constructing XML responses sent back to AutoVue Server. The framework uses XML parser libraries included in your servlet container.

The framework defines a set of interfaces and classes that facilitate the integration task. The framework is packaged into vuelinkcore.jar and contains many packages including following:

- com.cimmetry.vuelink
- com.cimmetry.vuelink.authentication

- com.cimmetry.vuelink.backend
- com.cimmetry.vuelink.context
- com.cimmetry.vuelink.defs
- com.cimmetry.vuelink.io
- com.cimmetry.vuelink.property
- com.cimmetry.vuelink.prosaction
- com.cimmetry.vuelink.query
- com.cimmetry.vuelink.session
- com.cimmetry.vuelink.util
- com.cimmetry.vuelink.xml

The framework uses log4j for logging messages into application server log file or console.

Sample Integration

The ISDK includes a sample integration of a simplified file system management (Filesys). This sample includes "VueLink for filesys DMS" and aims to act as a starting point for developing your own integration as well as familiarizing yourself with the integration framework.

The Filesys DMS comes with a database repository that is preloaded with some sample files in 2D/3D formats. To simplify things, the structure of this repository is based on local file system. Markups and renditions are stored back into this content repository.

The sample integration demonstrates how you can add basic and advanced functionalities to your own integration including:

- Document viewing of native formats
- Retrieve document attributes
- Create, save and review markups
- Browse DMS repository
- Search DMS repository
- Compare document versions
- Convert documents to other formats
- Support for AutoVue in offline mode
- Support for Stamps markup entity
- Support for Set Property action (with Pick List support)
- Enhanced framework to support Oracle Enterprise Visual Framework (OEVF)
- Support for markup save alert before applet close
- Support for browser Pop-up blocker notification
- Support for AutoVue authorization mechanism (encrypted Authorization block and password)
- Improved performance with support for distributed file servers
- Support for integration between Online meeting managements and AutoVue Real-Time Collaboration (RTC)
- Support for saving/deleting on-line master markups based on default markup policy
- Support for read-only markups
- Bundled demos for OEVA and RTC

The sample integration for Filesys contains many packages including the following:

- com.cimmetry.vuelink.filesys
- com.cimmetry.vuelink.filesys.actions

- com.cimmetry.vuelink.filesys.backend
- com.cimmetry.vuelink.filesys.dms
- com.cimmetry.vuelink.filesys.dms.gui
- com.cimmetry.vuelink.filesys.propactions
- com.cimmetry.vuelink.filesys.session
- com.cimmetry.vuelink.filesys.util

The sample integration includes a front-end UI which allows users to navigate the Filesys DMS data structure. This UI consists of:

- A default home page: index.jsp. It provides links for Filesys demo, RTC demo and OEVF demo.
- RTC demo pages: RTCDemo.jsp, RTCDemo_init.jsp, RTCDemo_join.jsp
- OEVF demo pages: OEVFDemo.html, OEVFDemoDes.html
- Filesys demo pages: jVue.html, frmApplet.jsp and a single servlet called
  com.cimmetry.vuelink.filesys.dms.gui.ListDirServlet.
  The main filesys demo page contains two frames as shown in the following figure.
  - The frame on the left displays the structure of Filesys DMS data. The content of this frame is displayed by ListDirServlet which allows you to navigate the Filesys DMS by expanding folders and selecting documents to view.
  - The frame on the right displays the AutoVue applet using frmApplet.jsp. When you click a document in the frame on the left, it displays in the frame on the right.
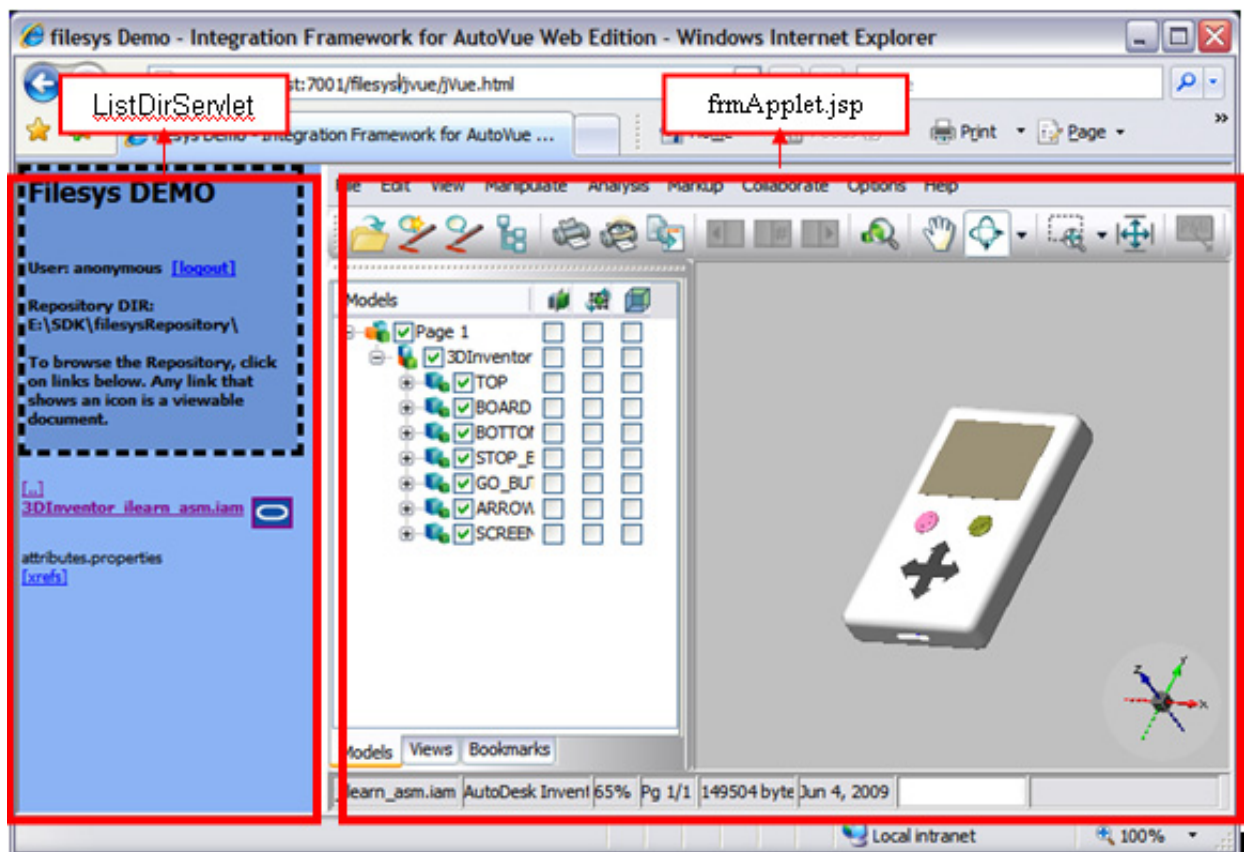
AutoVue Components

The ISDK bundles following two components of Oracle AutoVue:

- AutoVue Applet Client (jvue.jar, jogl.jar,gluegen-rt.jar)
  - Referenced by frmApplet.jsp which contains HTML code for AutoVue Applet (<applet> tag).

- VueServlet tunneling servlet (vueservlet.jar)
 - This servlet is used to allow AutoVue applet connects to AutoVue Servlet.

The AutoVue Integration SDK does not bundle AutoVue Server. You need to download and install it separately. Refer to the *Installation Guide* for more information.

Third-Party Libraries

The Sample Integration for Filesys bundles third-party open-source libraries needed by the framework. For information, refer to the *Acknowledgments* document.

ISDK Web Services Sample Server Project

The Web Services Sample Server project is a sample implementation of the Web Services provider and uses the Filesys repository as the backend DMS.
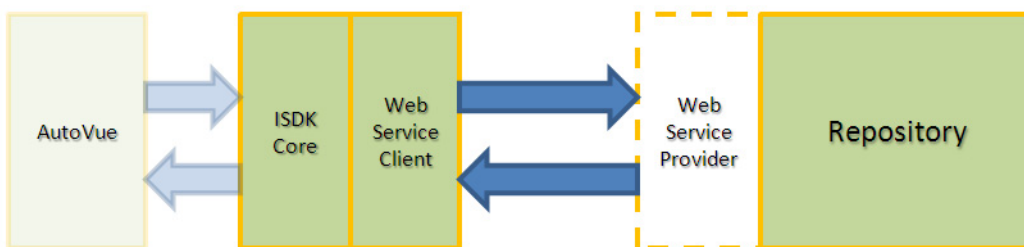
It implements the Web Services methods defined in the BluePrint WSDL file. The ISDK Web Services Sample Server project is located under the WebServiceIntegration/WebSserviceSampleFolder folder. Refer to the *ISDK Installation and Configuration Guide* for more information.

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.

General AutoVue Information

Telephone   +1.514.905.8434 or 1.800.363.5805

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

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

Oracle Customer Support

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

My Oracle Support AutoVue Community

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

Sales Inquiries

E-mail: autovuesales_ww@oracle.com