



# LiveContent S1000D Delivery Server Installation and Upgrade Manual

---

***LiveContent S1000D 5.12***

**August 2022**

---

## Legal notice

Copyright and trademark information relating to this product release.

Copyright © 2009–2021 SDL as part of the RWS Holdings Plc group of companies ("RWS Group").

SDL means SDL Limited and its subsidiaries and affiliates. All intellectual property rights contained herein are the sole and exclusive rights of SDL. All references to SDL shall mean SDL Limited and its subsidiaries and affiliates details of which can be obtained upon written request.

All rights reserved. Unless explicitly stated otherwise, all intellectual property rights including those in copyright in the content of this website and documentation are owned by or controlled for these purposes by SDL. Except as otherwise expressly permitted hereunder or in accordance with copyright legislation, the content of this site, and/or the documentation may not be copied, reproduced, republished, downloaded, posted, broadcast or transmitted in any way without the express written permission of SDL.

LiveContent S1000D is a registered trademark of SDL. All other trademarks are the property of their respective owners. The names of other companies and products mentioned herein may be the trademarks of their respective owners. Unless stated to the contrary, no association with any other company or product is intended or should be inferred.

This product may include open source or similar third-party software, details of which can be found by clicking the following link: "Acknowledgments" on page 15 .

Although RWS Group takes all reasonable measures to provide accurate and comprehensive information about the product, this information is provided as-is and all warranties, conditions or other terms concerning the documentation whether express or implied by statute, common law or otherwise (including those relating to satisfactory quality and fitness for purposes) are excluded to the extent permitted by law.

To the maximum extent permitted by law, RWS Group shall not be liable in contract, tort (including negligence or breach of statutory duty) or otherwise for any loss, injury, claim liability or damage of any kind or arising out of, or in connection with, the use or performance of the Software Documentation even if such losses and/or damages were foreseen, foreseeable or known, for: (a) loss of, damage to or corruption of data, (b) economic loss, (c) loss of actual or anticipated profits, (d) loss of business revenue, (e) loss of anticipated savings, (f) loss of business, (g) loss of opportunity, (h) loss of goodwill, or (i) any indirect, special, incidental or consequential loss or damage howsoever caused.

All Third Party Software is licensed "as is." Licensor makes no warranties, express, implied, statutory or otherwise with respect to the Third Party Software, and expressly disclaims all implied warranties of non-infringement, merchantability and fitness for a particular purpose. **In no event will Licensor be liable for any damages, including loss of data, lost profits, cost of cover or other special, incidental, consequential, direct, actual, general or indirect damages arising from the use of the Third Party Software or accompanying materials, however caused and on any theory of liability. This limitation will apply even if Licensor has been advised of the possibility of such damage. The parties acknowledge that this is a reasonable allocation of risk.**

---

Information in this documentation, including any URL and other Internet website references, is subject to change without notice. Without limiting the rights under copyright, no part of this may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of RWS Group.



# Contents

---

- 1 Installing and configuring the LiveContent S1000D Delivery Server . . . . . 1**
  - Installing the LiveContent S1000D Delivery Server on Windows . . . . . 2
  - Configuring LiveContent S1000D for use with HTTPS . . . . . 3
  - Configuring Apache Tomcat for an IETM . . . . . 5
  - Configuring the LiveContent S1000D launch.ini file . . . . . 6
  - Launching the Delivery Server . . . . . 7
- 2 Verifying your LiveContent S1000D Delivery Server installation . . . . . 9**
  - Sample data for product verification . . . . . 10
  - Sample delivery and support . . . . . 10
  - Connecting to the Delivery Server from a client . . . . . 10
- 3 Uninstalling the LiveContent S1000D Delivery Server on Windows . . . . . 13**
- A Acknowledgments . . . . . 15**



# 1

## **Installing and configuring the LiveContent S1000D Delivery Server**

These topics explain how to install and configure the Delivery Server for LiveContent S1000D.

# Installing the LiveContent S1000D Delivery Server on Windows

When you install LiveContent S1000D Delivery Server, a limited version of the LiveContent S1000D product is installed when you install from the CD image containing the IETP.

### Before you begin

Before performing this procedure, you must configure LiveContent S1000D behavior when launched from a CD image. For more information, see “Launching CD images” on page 0 *Launching CD images* in the LiveContent S1000D documentation.

### Procedure

1. Log on to the LiveContent S1000D server as a user with administrator privileges.
2. Load the IETM from its delivery medium (CD or ISO file, for example).
3. Navigate to the root directory and right-click **autoplay.exe**, then choose **Run as Administrator**.
4. In the initial LiveContent S1000D panel, select **Local Install**.
5. In the **Introduction** panel, select **Next**.
6. In the **Choose Install Folder** panel, specify the location where you want to install LiveContent S1000D program executables and libraries, and then select **Next**.
7. In the **Choose Application Data Folder** panel, specify one of the following:
  - **Choose Directory:** Specify a directory on the computer, the workspace area, where you can specify access permissions.
  - **All Users:** Application data is available to all users that log on to this computer. If you select this option, the workspace area is in the public users folder (for example, on Windows 10, this area would be `c:\ProgramData\XyEnterprise\`).
  - **Current User:** Application data is available only to the user currently logged on to this computer. If you select this option, the workspace area is in the user folder (for example, `c:\Users\username\`, where *username* is the user name of the current user).
8. Select **Next**.
9. If you specified **Choose Directory**, you are prompted to specify the folder location and to configure user access to that folder.
  - a. In the **Choose Application Data Folder** panel, select **Choose**.
  - b. Specify the directory where you want to store application data, and then select **Next**.
  - c. In the **Shortcuts Access** panel, specify whether all users or just you should have access to this folder location and then select **Next**.



10. Review the installation settings you have specified and then select **Install** to start installing the product files.
11. When the installation completes, select **Done**.

## Results

The Delivery Server is now installed. Collections distributed from the Publishing Server to this Delivery Server initially have the same user account information as the Publishing Server at the time of distribution. From that point on, accounts may be added, deleted or modified on the Delivery Server. An administrator-level user can change the passwords.

**Tip:** If you need to troubleshoot the installation, you can examine the log file located at the following location.

```
LiveContent_Home\Uninstall_LiveContent\Logs
```

For example:

```
C:\Program Files\XyEnterprise\LiveContent\ Uninstall_LiveContent\Logs  
\LiveContent_Install_11_12_2015_04_51_21.log
```

# Configuring LiveContent S1000D for use with HTTPS

If you use the HTTPS protocol, you must configure LiveContent S1000D to use an SSL connection with the Jetty server.

## Procedure

1. Set up the SSL certificate for Jetty. Refer to <https://www.eclipse.org/jetty/documentation/jetty-9/index.php> to learn how.
2. Note down the path name of the new SSL connector keystore.
3. Create one or two obfuscated secure passwords. Refer to <https://www.eclipse.org/jetty/documentation/jetty-9/index.php> to learn how. These passwords will be used for the SSL connector keystore and the manager user.

You have several generated secure versions of one or two passwords, for example:

- OBF:20771x1b206z
- MD5:639bae9ac6b3e1a84cebb7b403297b79
- CRYPT:me/ks90E221EY

4. Note down the OBF version of the passwords for the keystore and for the manager user.
5. Navigate to the collection directory of the collection for which you want to configure an SSL connection.

6. Open `wietmsd_prg.xml` for editing.
7. Within the `<application>` section, add the following:

```
<!-- Set <value> to 1 to use an SSL connection with the Jetty server -->
<configitem name="app.secure_connection">
  <value>1</value>
</configitem>
<!-- SSL configuration. Use only if app.secure_connection is set to 1.
-->
<configitem name="app.keystore_location">
  <value>KEYSTOREPATH</value>
</configitem>
<configitem name="app.keystore_password">
  <value>KEYSTOREPASSWORD</value>
</configitem>
<configitem name="app.manager_password">
  <value>MANAGERPASSWORD</value>
</configitem>
```

where:

- `KEYSTOREPATH` is the path name of the new SSL connector keystore.
  - `KEYSTOREPASSWORD` is the obfuscated version of the keystore password (including `OBF:` at the beginning).
  - `MANAGERPASSWORD` is the obfuscated version of the private key stored in the keystore (including `OBF:` at the beginning).
8. If you want to change the default port used by LiveContent S1000D, then within the `<application>` section, find the following section and change the value of `<value>`:

```
<configitem name="app.port">
  <comments>Server port</comments>
  <value>XXX</value>
</configitem>
```

In this fragment, set `XXX` to the default port used (for HTTPS, this would by default be 443).

9. Save and close `wietmsd_prg.xml`.
10. In the same location, open `launch.ini` for editing.
11. Find the property called `AccessURL1` in this file and make the following changes to its value:
  - a. Change the protocol at the beginning of the URL from `http://` to `https://`.
  - b. Change the host name to the fully qualified host name as defined in your SSL certificate.
  - c. If you do not use the default HTTPS port, 443, add `:PORTNUMBER` after the host name, where `PORTNUMBER` is the non-default number of your HTTPS port.

Your property now resembles one of the following:

- `AccessURL1=https://ietm.example.com/servlets3/wietmsd` (if running on the default HTTPS port, 443)
- `AccessURL1=https://ietm.example.com:886/servlets3/wietmsd` (if running on a

non-default HTTPS port, 886 in this example)

12. Find the property called `AccessURL2` in the file and make the same change to its value.
13. Save and close `launch.ini`.

## Configuring Apache Tomcat for an IETM

LiveContent S1000D provides its own custom HTTP server that can serve content across the Web, but you may want to use an application server such as Apache Tomcat for larger implementations with multiple concurrent viewers.

### About this task

In the steps and code samples below, read the following italicized strings as follows:

- For *LiveContent\*, read your LiveContent directory, which defaults to either `%ProgramFiles%\XyEnterprise\LiveContent\` (on Windows) or to `/opt/XyEnterprise/LiveContent/` (on Linux).
- For *LiveContentData\*, read your LiveContent data directory, which defaults to either `%HomePath%\LiveContentData\` (on Windows) or to `/opt/XyEnterprise/LiveContentData/` (on Linux).

### Procedure

1. Open *LiveContent\WEB-INF\web.xml* for editing and specify the locations of your collection configuration file and your log4j configuration file, as indicated:

```
<servlet>
  <servlet-name>wietmsd</servlet-name>
  <servlet-class>server.skxmlsrv</servlet-class>
  <init-param>
    <!-- Location of the configuration file -->
    <param-name>properties</param-name>
    <param-value>
      LiveContentData\collections\default\wietmsd_prg.xml
    </param-value>
    <!-- The <param-value/> value includes the collection name
         of the sample data (default). You should substitute
         the name of your collection. -->
  </init-param>
  <init-param>
    <!-- Location of the log4j configuration file -->
    <param-name>log4j-conf</param-name>
    <param-value>
      LiveContent\servlet-log4j.conf
    </param-value>
  </init-param>
</servlet>
```

2. Copy the JAR files.

Copy all files from here:

*LiveContent\lib*

to here:

`LiveContent\WEB-INF\lib`

3. Map the publication data to a location that Tomcat can access.

Examine the `wietmsd.xml` file to ensure that the following code block exists:

```
<collection name="default">
  <configitem name="http_name">
    <value>IETMBooks3</value>
  </configitem>
</collection>
<!-- The <collection name="default"> element includes the
      collection name of the sample data (default).
      You should substitute the name of your collection. -->
```

4. Navigate to the `conf\` subfolder of your Apache Tomcat root folder, for example, `Apache Software Foundation\Tomcat 8.0\conf\`
5. Open `server.xml` in this location for editing, and modify the `docBase` parameters:

```
<Host>
  <Context docBase="LiveContent\" path="/servlets3"/>
  <Context
    docBase="LiveContent\templates\FullSupport\httpdocs\"
    path="/wietmsd3"/>
  <Context docBase="LiveContentData\publications\"
    path="/IETMBooks3"/>
</Host>
```

6. Stop the LiveContent S1000D server.
7. Restart the LiveContent S1000D server by accessing the Windows **Start** menu and navigating to **Programs > XyEnterprise > LiveContent > COLLECTION**, where **COLLECTION** is the name of your collection.
8. Stop and then restart the Apache Tomcat service.

## Configuring the LiveContent S1000D launch.ini file

You can configure LiveContent S1000D Delivery Server behavior by modifying the `launch.ini` file in the application data folder.

### Procedure

1. Open the `launch.ini` file in a text editor.

This file is located in the application data folder you specified during the LiveContent S1000D installation.

For example, if you specified that the data folder was called `LiveContentData` and that the CD image was called `s1000D_sample_40`, the location of the `launch.ini` file might be:

```
%ProgramFiles%\XyEnterprise\LiveContentData\collections\S1000D_Sample_40\
launch.ini
```

2. To configure Delivery Server behavior, you can modify the following settings in `launch.ini`:

#### ShowServer

Defaults to 0. Set to 1 to display the LiveContent S1000D console window.

#### Startup

Defaults to 0. Set to 1 to force users to connect directly to the server through a browser using the server URL. This URL defaults to `http://HOST.DOMAIN:2245/servlets3/wietmsd` where *HOST* is your host name and *DOMAIN* is your domain name.

#### AutoShutDown

Set to 1 to terminate the LiveContent S1000D server process after the last attached client process ends. Set to 0 to keep the server process running even when you close a viewing browser session.

#### AutoStartup

Defaults to 1. Set to 0 to force users to connect directly to the server through a browser using the server URL. This URL defaults to `http://HOST.DOMAIN:2245/servlets3/wietmsd` where *HOST* is your host name and *DOMAIN* is your domain name.

The `Startup` and `AutoStartup` settings interact as follows:

- Both settings cannot be set to 0.
- Both settings cannot be set to 1.
- If `Startup` is set to 1 and `AutoStartup` is set to 0, launching of the IETP collection is suppressed.
- If `Startup` is set to 0 and `AutoStartup` is set to 1, launching of the IETP collection is enabled.

3. Save and close `launch.ini`.

## Launching the Delivery Server

As the final step of your Delivery Server installation, launch the server.

### Procedure

1. Access the machine on which you installed Delivery Server.
2. On Windows, navigate to **All Programs > XyEnterprise > LiveContent Bike IETM** to run LiveContent against the default collection.  
LiveContent S1000D is now running, as indicated by the blue globe in the system tray.
3. If the `AutoStartup` setting in the `launch.ini` file is set to 1, you are prompted to log in. If this happens, log in as **guest**, password **guest**, and click the **Bike** publication.  
The IETP is launched.



# 2

## **Verifying your LiveContent S1000D Delivery Server installation**

To verify that your Delivery Server was installed correctly, connect to the server from a client machine and view some of the sample content.

# Sample data for product verification

RWS provides sample data (known as bike data) you can use to help verify your installation or upgrade. To do so, publish the sample data to LiveContent S1000D and compare your results to the sample IETP provided. Note that samples, by their nature, are not covered by maintenance support.

# Sample delivery and support

Samples are not covered by maintenance support and are not necessarily consistent across S1000D specification versions.

## Support for samples

Because the styles that RWS delivers are samples, they are not covered by maintenance support. They are delivered "as is". If you have requirements that these samples do not meet, you can enlist the help of RWS.

Typically, RWS Professional Services can meet your requirements. This may include custom style or transform modifications that necessitate changes to the publish process itself. To contact RWS Professional Services for a quote, send an e-mail to <mailto:ProServRequest@sdl.com>.

In some cases, your request may necessitate more than a customization: a change to the product itself, either as a product enhancement or because an issue in the product needs to be resolved. If your request does not constitute a customization (or if you are unsure), create a support ticket for RWS Customer Support in the [RWS Support Gateway](#).

Customer Support ensures that the appropriate parties evaluate your request, and informs you of the status of your request. Requests that go beyond customization are placed on the product backlog of the product development team and prioritized against other issues, requests for enhancements and planned product features.

## Delivery and updates

Samples for LiveContent S1000D are delivered with LiveContent S1000D releases. Each release includes any additional sample development completed during that release schedule. The samples provided are not necessarily consistent across S1000D specification versions.

# Connecting to the Delivery Server from a client



View sample data from a client that connects to the Delivery Server to confirm that the Delivery Server works correctly.

### Procedure

1. On a client machine that can connect to the Delivery Server machine, launch a supported browser (such as Safari, Internet Explorer, or Mozilla Firefox).
2. Depending on your setup, do one of the following:
  - If your Delivery Server runs on Jetty (the Web server it ships with), access `http://DELIVERYSERVER:2245/servlets3/wietmsd` or, if you use HTTPS, `https://DELIVERYSERVER:2245/servlets3/wietmsd`

---

**Note:** In these samples, 2245 is the port number, which may be different at your site.

---

- Alternatively, if your Delivery Server runs on Apache Tomcat, access `http://DELIVERYSERVER/servlets3/wietmsd` or, if you use HTTPS, `https://DELIVERYSERVER/servlets3/wietmsd`

where *DELIVERYSERVER* is one of the following:

- The name of the Delivery Server machine if your client machine is on the same network as the Delivery Server machine and your protocol is HTTP
- The fully qualified hostname if your client machine is connecting remotely or through HTTPS (or both)

If you can view content, your Delivery Server is installed correctly.



# 3

## **Uninstalling the LiveContent S1000D Delivery Server on Windows**

To uninstall the Delivery Server, you must uninstall the limited version of the LiveContent S1000D product that was installed when you installed from the CD image of the IETP. If your machine doubles as a Publishing Server, this task also uninstalls the Publishing Server.

**Procedure**

1. Log on to the LiveContent S1000D server as a user with administrator privileges.
2. In Windows Explorer, navigate to the **Uninstall LiveContent** directory. This directory is the `Uninstall_LiveContent\` subdirectory of your LiveContent S1000D directory, which defaults to `%ProgramFiles%\XyEnterprise\LiveContent\`.
3. Double-click `Uninstall LiveContent S1000D.exe`.
4. In the **Uninstall LiveContent** panel, click **Uninstall**.
5. When the uninstallation process completes, click **Done**.



# Acknowledgments

LiveContent S1000D includes open source or similar third-party software.

### **ActiveState ActivePerl**

ActivePerl is the industry-standard, commercial-grade Perl distribution used by millions of developers around the world for easy Perl installation and quality-assured code.

### **AdoptOpenJDK**

AdoptOpenJDK provides prebuilt OpenJDK binaries from a fully open source set of build scripts and infrastructure.

### **Annogen**

Annogen is a framework which helps you work with JSR175 Annotations. In a nutshell, Annogen generates a proxy layer in front of your Annotations.

### **Apache Axis2**

Apache Axis2 is a Web Services / SOAP / WSDL engine, the successor to the widely used Apache Axis SOAP stack. There are two implementations of the Apache Axis2 Web services engine - Apache Axis2/Java and Apache Axis2/C.

### **Apache Batik SVG Toolkit**

Batik is a Java-based toolkit for applications or applets that want to use images in the Scalable Vector Graphics (SVG) format for various purposes, such as display, generation or manipulation.

### **Apache FOP**

Apache FOP (Formatting Objects Processor) is a print formatter driven by XSL formatting objects (XSL-FO) and an output independent formatter. It is a Java application that reads a formatting object (FO) tree and renders the resulting pages to a specified output. Output formats currently supported include PDF, PS, PCL, AFP, XML (area tree representation), Print, AWT and PNG, and to a lesser extent, RTF and TXT. The primary output target is PDF.

### **Apache Geronimo**

Apache Geronimo is an open source server runtime that integrates the best open source projects to create Java/OSGi server runtimes that meet the needs of enterprise developers and system administrators.

### **Apache Neethi**

Apache Neethi provides general framework for the programmers to use WS Policy. It is compliant with latest WS Policy specification which was published in March 2006. This framework is specifically written to enable the Apache Web services stack to use WS Policy as a way of expressing its requirements and capabilities.

### **Apache XML Graphics Commons**

Apache™ XML Graphics Commons is a library that consists of several reusable components used by Apache Batik and Apache FOP. Many of these components can easily be used separately outside the domains of SVG and XSL-FO.

### **Apache XMLSchema**

XMLSchema is a lightweight Java object model that can be used to manipulate and generate XML schema representations. You can use it to read XML Schema (xsd) files into memory and analyze or modify them, or to create entirely new schemata from scratch.

### **backport-util-concurrent**

This package is the backport of java.util.concurrent API, introduced in Java 5.0 and further refined in Java 6.0, to older Java platforms. The backport is based on public-domain sources from the JSR 166 CVS repository, the dl.util.concurrent package, and the Doug Lea's collections package.

### **Hammer JS**

Javascript library for recognizing touch events and gestures.

### **InstallAnywhere**

InstallAnywhere is the leading multi-platform development solution for application producers who need to deliver a professional and consistent cross installation experience for physical, virtual and cloud environments. From a single project file and build environment, InstallAnywhere creates reliable installations for on-premises platforms - Windows, Linux, Apple OS X, Solaris, AIX, HP-UX, and IBM iSeries - and enables you to take existing and new software products to a virtual and cloud infrastructure.

### **iTextSharp**

iText is a PDF library that allows you to CREATE, ADAPT, INSPECT and MAINTAIN documents in the Portable Document Format (PDF):

- Generate documents and reports based on data from an XML file or a database
- Create maps and books, exploiting numerous interactive features available in PDF
- Add bookmarks, page numbers, watermarks, and other features to existing PDF documents
- Split or concatenate pages from existing PDF files
- Fill out interactive forms
- Serve dynamically generated or manipulated PDF documents to a web browser

iText is used by Java, .NET, Android and GAE developers to enhance their applications with PDF functionality. iTextSharp is the .NET port.

### **ICU (International Components for Unicode)**

ICU is a mature, widely used set of C/C++ and Java libraries providing Unicode and Globalization support for software applications. ICU is widely portable and gives applications the same results on all platforms and between C/C++ and Java software.

### **Infonyte-DB**

Infonyte-DB provides a comprehensive, lightweight solution for querying and storing large, distributed XML documents. It is based on two major components, the PDOM engine which is a persistent implementation of the W3C DOM (Document Object Model) API, and the XQL engine which is a web-aware query engine supporting the XQL query language.

### **iOS**

iOS is a mobile operating system created and developed by Apple Inc. exclusively for its hardware.

### **Jalopy**

Jalopy is a source code formatter/beautifier/pretty printer for the Java programming language. It is aimed to provide a full-featured, yet free alternative to the well-known

Jindent. Plug-ins for Ant, Eclipse, IDEA, JBuilder, JDeveloper, jEdit, NetBeans.

### **Java Runtime Environment (JRE)**

This is part of Java Development Kit (JDK), a set of programming tools for developing Java applications.

### **JAXB**

The goal of the JAXB project is to develop and evolve the code base for the Reference Implementation (RI) of JAXB, the Java Architecture for XML Binding. The JAXB specification is developed through the Java Community Process following the process described at [jcp.org](http://jcp.org). This process involves an Expert Group with a lead that is responsible for delivering the specification, a reference implementation (RI) and a Technology Compatibility Kit (TCK). The primary goal of an RI is to support the development of the specification and to validate it. Specific RIs can have additional goals; the JAXB RI is a production-quality implementation that is used directly in a number of products by Oracle and other vendors.

### **Jettison**

Jettison is a collection of Java APIs (like STaX and DOM) which read and write JSON. This allows nearly transparent enablement of JSON based web services in services frameworks like CXF or XML serialization frameworks like XStream.

### **Jetty**

The Jetty Web Server provides an HTTP server and Servlet container capable of serving static and dynamic content either from a standalone or embedded instantiations. Starting from Jetty version 7, the Jetty webserver and other core components are hosted by the Eclipse Foundation.

### **JiBX**

JiBX is a tool for binding XML data to Java objects. It's extremely flexible, allowing you to start from existing Java code and generate an XML schema, start from an XML schema and generate Java code, or bridge your existing code to a schema that represents the same data. It also provides very high performance, outperforming all other Java data binding tools across a wide variety of tests.

### **jQuery**

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

### **jQuery Highlight**

Highlights the search keywords/terms in a preview.

### **jQuery-JSON**

A JSON plugin for jQuery, provides simple ways to convert to JSON and back again.

### **jQuery Splitter**

A splitter plugin for jQuery.

### **jQuery UI**

jQuery UI is a set of user interface interactions, effects, widgets, and themes built on top of the jQuery JavaScript Library.



**jQuery UI Touch Punch**

Javascript plugin for adding touch support to jQuery.

**Oracle JavaBeans Activation Framework**

With the JavaBeans Activation Framework standard extension, developers who use Java technology can take advantage of standard services to determine the type of an arbitrary piece of data, encapsulate access to it, discover the operations available on it, and to instantiate the appropriate bean to perform said operation(s).

**Oracle JavaMail**

The JavaMail API provides a platform-independent and protocol-independent framework to build mail and messaging applications. The JavaMail API is available as an optional package for use with the Java SE platform and is also included in the Java EE platform.

**Red Hat Linux**

Red Hat Enterprise Linux OpenStack Platform delivers an integrated foundation to create, deploy, and scale a secure and reliable public or private OpenStack cloud. Red Hat Enterprise Linux OpenStack Platform combines the world's leading enterprise Linux and the fastest-growing cloud infrastructure platform to give you the agility to scale and quickly meet customer demands without compromising on availability, security, or performance.

**Saxon**

Saxon is an XSLT processor for transforming XML documents into HTML, text, or other XML document types. It implements XSL Transformations (XSLT) and XML Path Language (XPath) and can be used from the command line, in an applet or a servlet, or as a module in other program.

**SDI Convert**

SDI Convert provides graphics file conversion for CAD, CAE, Maps, Seismic & Well Logs.

**StAX**

StAX is a standard XML processing API that allows you to stream XML data from and to your application. This StAX implementation is the standard pull parser implementation for JSR-173 specification.

**svg-pan-zoom**

JavaScript library that enables panning and zooming of an SVG in an HTML document, with mouse events or custom JavaScript hooks.

**Syncro Soft Oxygen XML Author**

XML editor for structured authoring.

**Woden**

The Woden project is an incubation sub-project of the Apache Web Services Project to develop a Java class library for reading, manipulating, creating and writing WSDL documents, initially to support WSDL 2.0 but with the longer term aim of supporting past, present and future versions of WSDL.

**Woodstox**

Woodstox is a high-performance validating namespace-aware StAX-compliant (JSR-173) Open Source XML-processor written in Java.

### **Xerces Java Parser**

The Xerces Java Parser 1.4.4 supports the XML 1.0 recommendation and contains advanced parser functionality, such as support for the W3C's XML Schema recommendation version 1.0, DOM Level 2 version 1.0, and SAX Version 2, in addition to supporting the industry-standard DOM Level 1 and SAX version 1 APIs.