



# SDL Collaborative Review installation guide

---

***Collaborative Review 7.7.0***

**July 2019**

---

## Legal notice

Copyright and trademark information relating to this product release.

Copyright © 2003–2019 SDL Group.

SDL Group means SDL PLC. and its subsidiaries and affiliates. All intellectual property rights contained herein are the sole and exclusive rights of SDL Group. All references to SDL or SDL Group shall mean SDL PLC. 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 Group. 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.

SDL Tridion Docs is a registered trademark of SDL Group. 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 2.

Although SDL 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, SDL 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 SDL Group.

---

# Contents

---

<b>1</b>	<b>Welcome to SDL Collaborative Review Installation Guide</b>	<b>1</b>
	Acknowledgments	2
	Customer support	13
<b>2</b>	<b>Collaborative Review requirements and install/upgrade preparation</b>	<b>15</b>
	Obtaining a License File before installing or upgrading	16
	Collaborative Review hardware requirements	17
	Collaborative Review software requirements	17
	Collaborative Review pre-installation and upgrade settings	19
	Enabling Secure Socket Layer (SSL) protocol for the Apache Tomcat web application server	20
	Enabling restricted http protocol for the Apache Tomcat web application server	22
<b>3</b>	<b>Install Collaborative Review</b>	<b>25</b>
	Install Collaborative Review on Windows	26
	Running the Collaborative Review installer for Windows	26
	Running the database setup script for Windows	27
	Install Collaborative Review on Linux	28
	Running the Collaborative Review installer for Linux	28
	Running the database setup script for Linux	29
	Post installation actions	30
	Manually installing a license file	30
	Configuring Collaborative Review to Support Single Sign-On (SSO) User Authentication	31
	Establishing a trust for Collaborative Review	31
	Establishing a trust: The ISHSTS example	34
	Establishing a trust: The ADFS example	35
	Setting configuration files for SSO	37
	Configuring SSO in LiveContentSSO.xml	37
	Configuring SSO in LiveContentSecurity.xml	40
	Configuring SSO in lc.properties	40
	Configuring SSO in LiveContentGroups.xml	41
	Restarting Apache Tomcat to Enable SSO	45
	Configuring the Collaborative Review Web application	46
	Configuring context.xml files	47

---

Configuring Apache Tomcat . . . . .	48
Setting the path to Linux installation logs . . . . .	48
Enabling the use of non-ASCII characters on Collaborative Review . . . . .	49
<b>4 Validating and testing Collaborative Review installation or upgrade . . . . .</b>	<b>51</b>
Validating the Installation . . . . .	52
Testing with Sample Data . . . . .	53

# 1

## **Welcome to SDL Collaborative Review Installation Guide**

This document presents the complete Collaborative Review installation procedure and information.

## Acknowledgments

SDL products include open source or similar third-party software.

### 7zip

Is a file archiver with a high compression ratio. 7-zip is delivered under the GNU LGPL License.

### **7zip SFX Modified Module**

The SFX Modified Module is a plugin for creating self-extracting archives. It is compatible with three compression methods (LZMA, Deflate, PPMd) and provides an extended list of options. Reference website <http://7zsfx.info/>.

### Akka

Akka is a toolkit and runtime for building highly concurrent, distributed, and fault tolerant event-driven applications on the JVM.

### Amazon Ion Java

Amazon Ion Java is a Java streaming parser/serializer for Ion. It is the reference implementation of the Ion data notation for the Java Platform Standard Edition 8 and above.

### Amazon SQS Java Messaging Library

This Amazon SQS Java Messaging Library holds the Java Message Service compatible classes, that are used for communicating with Amazon Simple Queue Service.

### ANTLR

ANTLR is a powerful parser generator that you can use to read, process, execute, or translate structured text or binary files.

### Apache ActiveMQ

Apache ActiveMQ is the most popular and powerful open source messaging and Integration Patterns server.

### Apache Ant

Apache Ant is a Java library and command-line tool whose mission is to drive processes described in build files as targets and extension points dependent upon each other. The main known usage of Ant is the build of Java applications. Ant supplies a number of built-in tasks allowing to compile, assemble, test and run Java applications. Ant can also be used effectively to build non Java applications, for instance C or C++ applications. More generally, Ant can be used to pilot any type of process which can be described in terms of targets and tasks.

### Apache Commons BeanUtils

The Java language provides *Reflection* and *Introspection* APIs (see the `java.lang.reflect` and `java.beans` packages in the JDK Javadocs). However, these APIs can be quite complex to understand and utilize. The *BeanUtils* component provides easy-to-use wrappers around these capabilities.



### [Apache Commons Codec](#)

Apache Commons Codec (TM) software provides implementations of common encoders and decoders such as Base64, Hex, Phonetic and URLs.

### [Apache Commons Configuration](#)

The Commons Configuration software library provides a generic configuration interface which enables a Java application to read configuration data from a variety of sources. Commons Configuration provides typed access to single, and multi-valued configuration parameters.

### [Apache Commons DBCP \(Database Connection Pools\)](#)

Many Apache projects support interaction with a relational database. Creating a new connection for each user can be time consuming (often requiring multiple seconds of clock time), in order to perform a database transaction that might take milliseconds. Opening a connection per user can be unfeasible in a publicly-hosted Internet application where the number of simultaneous users can be very large. Accordingly, developers often wish to share a "pool" of open connections between all of the application's current users. The number of users actually performing a request at any given time is usually a very small percentage of the total number of active users, and during request processing is the only time that a database connection is required. The application itself logs into the DBMS, and handles any user account issues internally. There are several Database Connection Pools already available, both within Apache products and elsewhere. This Commons package provides an opportunity to coordinate the efforts required to create and maintain an efficient, feature-rich package under the ASF license.

### [Apache Commons FileUpload](#)

The Commons **FileUpload** package makes it easy to add robust, high-performance, file upload capability to your servlets and web applications.

### [Apache Commons HttpClient](#)

HttpClient was started in 2001 as a subproject of the Jakarta Commons, based on code developed by the Jakarta Slide project.

### [Apache Commons Lang](#)

The standard Java libraries fail to provide enough methods for manipulation of its core classes. Apache Commons Lang provides these extra methods.

Lang provides a host of helper utilities for the java.lang API, notably String manipulation methods, basic numerical methods, object reflection, concurrency, creation and serialization and System properties. Additionally it contains basic enhancements to java.util.Date and a series of utilities dedicated to help with building methods, such as hashCode, toString and equals.

### [Apache Commons Logging](#)

The Logging package is an ultra-thin bridge between different logging implementations. A library that uses the commons-logging API can be used with any logging implementation at runtime. Commons-logging comes with support for a number of popular logging implementations, and writing adapters for others is a reasonably simple task.

### [Apache Commons Pool](#)

Pool provides an Object-pooling API, with three major aspects:

1. A generic object pool interface that clients and implementers can use to provide easily interchangeable pooling implementations.
2. A toolkit for creating modular object pools.

3. Several general purpose pool implementations.

### **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 HttpClient**

Although the `java.net` package provides basic functionality for accessing resources via HTTP, it doesn't provide the full flexibility or functionality needed by many applications. HttpClient seeks to fill this void by providing an efficient, up-to-date, and feature-rich package implementing the client side of the most recent HTTP standards and recommendations.

Designed for extension while providing robust support for the base HTTP protocol, HttpClient may be of interest to anyone building HTTP-aware client applications such as web browsers, web service clients, or systems that leverage or extend the HTTP protocol for distributed communication.

### **Apache HttpComponents**

The Apache HttpComponents™ project is responsible for creating and maintaining a toolset of low level Java components focused on HTTP and associated protocols.

Within the HttpComponents project, [HttpCore](#) is a set of low level HTTP transport components that can be used to build custom client and server side HTTP services with a minimal footprint. HttpCore supports two I/O models: blocking I/O model based on the classic Java I/O and non-blocking, event driven I/O model based on Java NIO

### **Apache Log4j**

Apache Log4j 2 is an upgrade to Log4j that provides significant improvements over its predecessor, Log4j 1.x, and provides many of the improvements available in Logback while fixing some inherent problems in Logback's architecture.

### **Apache Lucene, SOLR**

The Apache Lucene™ project develops open-source search software.

### **Apache Tomcat, Tomcat Embed**

Apache Tomcat is an open source software implementation of the Java Servlet and JavaServer Pages technologies.

### **Apache XBean :: Spring**

XBean :: Spring provides a schema-driven proprietary namespace handler for Spring contexts.

### **Apache Xerces**

The Apache Xerces Project is responsible for software licensed to the Apache Software Foundation intended for the creation and maintenance of:

- XML parsers
- related software components

### [Apache XML](#)

The Apache XML Project used to be the home for many XML-related subprojects, many of which have moved to top-level project status recently or are currently in migration. The Apache XML Project slowly transforms into an place where you can find pointers to XML-related projects here in The Apache Foundation.

### [AspectJ](#)

AspectJ is a seamless aspect-oriented extension to the Java programming language. It is Java platform compatible easy to learn and use.

### [AWS SDK for Amazon SQS](#)

The AWS Java SDK for Amazon SQS module holds the client classes that are used for communicating with Amazon Simple Queue Service.

### [AWS SDK for Java Core](#)

The AWS SDK for Java - Core module holds the classes that are used by the individual service clients to interact with Amazon Web Services. Users need to depend on aws-java-sdk artifact for accessing individual client classes.

### [Byte Buddy](#)

Byte Buddy is a code generation and manipulation library for creating and modifying Java classes during the runtime of a Java application and without the help of a compiler.

### [CDI APIs](#)

APIs for CDI (Contexts and Dependency Injection for Java).

### [cglib](#)

cglib is a powerful, high performance and quality Code Generation Library, It is used to extend JAVA classes and implements interfaces at runtime.

### [DITA-OT](#)

The DITA Open Toolkit is a Java-based implementation of the OASIS DITA Technical Committee's specification for DITA DTDs and schemas. It contains ANT, SAXON,...

### [DockPanel Suite](#)

.Net Docking Library for Windows Forms

### [dom4j](#)

*dom4j* is an easy to use, open source library for working with XML, XPath and XSLT on the Java platform using the Java Collections Framework and with full support for DOM, SAX and JAXP.

### [dsinfo](#)

The dsinfo library enables you to easily use Scala-side information in implementations of embedded (internal) domain-specific languages. dsinfo is implemented using Scala macros which are an experimental feature of Scala 2.10 and 2.11.

### [dsprofile](#)

The dsprofile library provides general facilities to implement domain-specific profiling in Scala and Java programs.

### [edtFTPj/Free](#)

Free Java FTP library gives Java developers extensive FTP functionality.

### **Elasticsearch RESTful client**

A RESTful client for the Elasticsearch search engine.

### **Fast Serialization**

Fast Serialization reimplements Java Serialization with focus on speed (up to 10 times faster), size and compatibility. This allows the use of FST with minimal code change.

### **Fonto Editor**

Fonto is an online XML editor designed for people with no knowledge of XML or any other technology that comes with structured content authoring.

### **GeckoFX**

Gecko is a free and open source layout engine used in many applications developed by the Mozilla Foundation and the Mozilla Corporation (notably the Firefox web browser).

### **globalize**

JavaScript globalization and localization. Formats and parses strings, dates and numbers in over 350 cultures.

### **GNU Aspell**

GNU Aspell is a Free and Open Source spell checker designed to eventually replace Ispell. It can either be used as a library or as an independent spell checker. Its main feature is that it does a superior job of suggesting possible replacements for a misspelled word than just about any other spell checker out there for the English language. Unlike Ispell, Aspell can also easily check documents in UTF-8 without having to use a special dictionary. Aspell will also do its best to respect the current locale setting. Other advantages over Ispell include support for using multiple dictionaries at once and intelligently handling personal dictionaries when more than one Aspell process is open at once.

Specifically we are using GNUASpell dictionaries for de-CH, de-DE, en-CA, en-GB, en-US, es-ES, fr-FR, fr-CH, nl-NL.

### **google-code-prettify**

google-code-prettify is a Javascript module and CSS file that allows syntax highlighting in an html page.

### **google-gson**

google-gson is a Java library to convert JSON to Java objects and vice-versa.

### **Google Guava**

The Guava project contains several of Google's core libraries that we rely on in our Java-based projects: collections, caching, primitives support, concurrency libraries, common annotations, string processing, I/O, and so forth.

### **GraphQL-Java**

The Java implementation of GraphQL.

### **Hibernate**

Hibernate is a high-performance Object/Relational persistence and query service. The most flexible and powerful Object/Relational solution on the market, Hibernate takes care of the mapping from Java classes to database tables and from Java data types to SQL data types. It provides data query and retrieval facilities that significantly reduce development time. Hibernate's design goal is to relieve the developer from 95% of common data persistence-related programming tasks by eliminating the need for manual, hand-crafted data processing using SQL and JDBC.

## **HK2 Framework**

HK2 is a light-weight and dynamic dependency injection framework.

## **HSQldb (HyperSQL DataBase)**

HSQldb (HyperSQL DataBase) is the leading SQL relational database engine written in Java. It offers a small, fast multithreaded and transactional database engine with in-memory and disk-based tables and supports embedded and server modes. It includes a powerful command line SQL tool and simple GUI query tools.

## **Hunspell**

Hunspell is the spell checker of LibreOffice, OpenOffice.org, Mozilla Firefox 3 & Thunderbird, Google Chrome, and it is also used by proprietary software packages, like Mac OS X, InDesign, MemoQ, Opera and SDL Trados Studio.

## **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.

## **Jackson tooling**

Inspired by the quality and variety of XML tooling available for the Java platform (StAX, JAXB, etc.), the Jackson is a multi-purpose Java library for processing JSON data format. Jackson aims to be the best possible combination of fast, correct, lightweight, and ergonomic components for developers.

## **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).

## **JavaBeans Validation**

Bean Validation (JSR-303) API.

## **Javassist (*Java Programming Assistant*)**

Javassist (*Java Programming Assistant*) makes Java bytecode manipulation simple. It is a class library for editing bytecodes in Java; it enables Java programs to define a new class at runtime and to modify a class file when the JVM loads it. Unlike other similar bytecode editors, Javassist provides two levels of API: source level and bytecode level. If the users use the source-level API, they can edit a class file without knowledge of the specifications of the Java bytecode. The whole API is designed with only the vocabulary of the Java language. You can even specify inserted bytecode in the form of source text; Javassist compiles it on the fly. On the other hand, the bytecode-level API allows the users to directly edit a class file as other editors.

## **javax.annotation**

JSR 250 Common Annotations For The Java Platform.

## **javax.cache**

Caching Java API

### Java Expression Language

Expression Language Java API

### javax.inject

Dependency Injection Java API

### 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.

### JBoss Java Annotation Indexer (Jandex)

A Java Annotation Indexer for JBoss

### JBoss Logging Framework

The JBoss Logging Framework.

### jedis

A blazingly small and sane Redis Java client.

### Jersey RESTful WS

Developing RESTful Web services that seamlessly support exposing your data in a variety of representation media types and abstract away the low-level details of the client-server communication is not an easy task without a good toolkit. In order to simplify development of RESTful Web services and their clients in Java, a standard and portable JAX-RS API has been designed. Jersey RESTful Web Services framework is open source, production quality, framework for developing RESTful Web Services in Java that provides support for JAX-RS APIs and serves as a JAX-RS (JSR 311 & JSR 339) Reference Implementation.

### 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.

### JLine

JLine is a Java library for handling console input. It is similar in functionality to BSD editline and GNU readline. People familiar with the readline/editline capabilities for modern shells (such as bash and tcsh) will find most of the command editing features of JLine to be familiar.

### JMESPath Java

JMESPath is a query language for JSON. You can extract and transform elements from a JSON document. This is a Java implementation

### Joda-Convert

Joda-Convert provides a small set of classes to provide round-trip conversion between Objects and Strings. It does not tackle the wider problem of Object to Object transformation.

### Joda-Time

Joda-Time provides a quality replacement for the Java *date* and *time* classes. The design allows for multiple *calendar* systems, while still providing a simple API. The 'default' calendar is the [http://www.joda.org/joda-time/cal\\_iso.html](http://www.joda.org/joda-time/cal_iso.html) standard which is used by XML. The Gregorian, Julian, Buddhist, Coptic, Ethiopic and Islamic systems are also included, and we welcome further additions. Supporting classes include time zone, duration, format and parsing.

### 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-cookie

jQuery plugin for reading, writing and deleting cookies.

### jquery.datatables

DataTables is a plug-in for the jQuery Javascript library. It is a highly flexible tool, based upon the foundations of progressive enhancement, which will add advanced interaction controls to any HTML table.

### jquery.dataTables.columnFilter

Adds advanced filter capabilities to the DataTables. JS file.

### jQueryFileUpload

File Upload widget with multiple file selection, drag&drop support, progress bar, validation and preview images, audio and video for jQuery.

### jquery.TypeScript.DefinitelyTyped

TypeScript Definitions (d.ts) for jquery.

### jQuery Highlight

Highlights the search keywords/terms in a preview.

### jQuery UI

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

### JSON-js

JSON is a light-weight, language independent, data interchange format. See <http://www.JSON.org> / The files in this collection implement JSON encoders/decoders in JavaScript. JSON became a built-in feature of JavaScript when the ECMAScript Programming Language Standard - Fifth Edition was adopted by the ECMA General Assembly in December 2009. Most of the files in this collection are for applications that are expected to run in obsolete web browsers. For most purposes, json2.js is the best choice.



### Json.NET

Json.NET is a popular high-performance JSON framework for .NET.

### JTA (Java Transaction API)

The `javax.transaction` package. It is appropriate for inclusion in a classpath, and may be added to a Java 2 installation.

### Kiama

The Kiama test library contains a collection of examples, tests that use those examples, and useful test support code.

### Knockout JavaScript library

Knockout is a JavaScript library that helps you to create rich, responsive display and editor user interfaces with a clean underlying data model. Any time you have sections of UI that update dynamically (e.g., changing depending on the user's actions or when an external data source changes), KO can help you implement it more simply and maintainably.

### kXML 2

kXML is a small XML pull parser, specially designed for constrained environments such as Applets, Personal Java or MIDP devices. In contrast to kXML 1, kXML 2 is based on the XML pull API.

### Logback

Logback is intended as a successor to the popular log4j project, picking up where log4j leaves off.

### MVC Web Projects

Auxiliary MVC Web Project libraries.

### MXP1

MXP1 is a stable XmlPull parsing engine that is based on ideas from XPP and in particular XPP2 but completely revised and rewritten to take the best advantage of latest JIT JVMs such as Hotspot in JDK 1.4+.

### Objenesis

Objenesis is a small Java library that serves one purpose: to instantiate a new object of a particular class.

### NHunspell

NHunspell brings the spell checking, hyphenation and thesaurus to the Microsoft® .NET Framework. NHunspell is C# library and wraps native libraries for Hunspell, Hyphen and MyThes. One design goal of this library and wrapper is to keep the source code of the included libraries as unmodified as possible. New versions of the base libraries can therefore easily adopted to NHunspell.

The integrated libraries are used in OpenOffice and they work with the dictionaries published on OpenOffice.org.

### NLog

NLog is a free logging platform for .NET, Silverlight and Windows Phone with rich log routing and management capabilities. NLog makes it easy to produce and manage high-quality logs for your application regardless of its size or complexity.



**[okhttp](#)**

An HTTP+HTTP/2 client for Android and Java applications.

**[okio](#)**

A modern I/O API for Java.

**[PATRICIA Trie in Java](#)**

An implementation of the Practical Algorithm to Retrieve Information Coded in Alphanumeric (PATRICIA).

**[Postal.Mvc5](#)**

Generate emails using ASP.NET MVC views

**[PS Cmdlet Help Editor](#)**

PowerShell Cmdlet Help Editor is the tool that helps you to create and edit XML-based help files for your PowerShell modules and PSSnap-Ins.

**[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.

**[Rx .NET](#)**

Reactive Extensions for .NET library used to validate entered values

**[Scallop](#)**

Scallop is a command line parser.

**[Scala](#)**

The Scala programming language fuses object-oriented and functional programming in a statically typed programming language. It is aimed at the construction of components and component systems.

**[SitemapGen4j](#)**

SitemapGen4j is a library to generate XML sitemaps in Java.

**[SLF4j](#)**

The Simple Logging Facade for Java (SLF4j) serves as a simple facade or abstraction for various logging frameworks (e.g. java.util.logging, logback, log4j) allowing the end user to plug in the desired logging framework at deployment time.

**[SnakeYAML](#)**

YAML is a data serialization format designed for human readability and interaction with scripting languages. SnakeYAML is a YAML parser and emitter for the Java programming language.

**[SNMP4j](#)**

SNMP4j is an enterprise class free open source and state-of-the-art SNMP implementation for Java™ 2SE 1.4 or later. SNMP4j supports command generation (managers) as well as command responding (agents). Its clean object oriented design is inspired by SNMP++, which is a well-known SNMPv1/v2c/v3 API for C++.

### [SpringFox](#)

Automated JSON API documentation for API's built with Spring.

### [Spring Framework](#)

The Spring Framework provides a comprehensive programming and configuration model for modern Java-based enterprise applications - on any kind of deployment platform. A key element of Spring is infrastructural support at the application level: Spring focuses on the "plumbing" of enterprise applications so that teams can focus on application-level business logic, without unnecessary ties to specific deployment environments.

### [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.

### [Swagger](#)

Swagger is a simple yet powerful representation of your RESTful API. With the largest ecosystem of API tooling on the planet, thousands of developers are supporting Swagger in almost every modern programming language and deployment environment. With a Swagger-enabled API, you get interactive documentation, client SDK generation and discoverability.

### [Swashbuckle.Core](#)

Seamlessly adds a Swagger to WebApi projects.

### [Thinktecture IdentityServer](#)

Front-end Secure Token Service to serve SAML tokens.

### [TwelveMonkeys Common](#)

TwelveMonkeys Common library contains common utility classes relating to languages, I/O and images.

### [TwelveMonkeys ImageIO](#)

TwelveMonkeys ImageIO is a collection of plugins and extensions for Java's ImageIO. These plugins extends the number of image file formats supported in Java, using the `javax.imageio.*` package. The main purpose of this project is to provide support for formats not covered by the JRE itself.

### [ua-parser](#)

A multi-language port of Browserscope's user agent parser.

### [Xalan-Java](#)

Xalan-Java is an XSLT processor for transforming XML documents into HTML, text, or other XML document types. It implements XSL Transformations (XSLT) Version 1.0 and XML Path Language (XPath) Version 1.0 and can be used from the command line, in an applet or a servlet, or as a module in other program.

### [Thinktecture IdentityServer](#)

Front-end Secure Token Service to serve SAML tokens.

### [WiX](#)

The WiX toolset builds Windows installation packages from XML source code. The tool-set integrates seamlessly into build processes.

#### **Woodstox**

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

#### **XML Pull Parsing**

An XML Pull Parsing API.

#### **XStream**

XStream is a simple library to serialize objects to XML and back again.

#### **XULRunner**

XULRunner is a runtime environment developed by the Mozilla Foundation to provide a common back-end for previewing.

## Customer support

To contact Technical Support, connect to the Customer Support Web Portal at <https://gateway.sdl.com> and log a case for your SDL product. You need an account to log a case. If you do not have an account, contact your company's SDL Support Account Administrator.



# 2

## **Collaborative Review requirements and install/ upgrade preparation**

In order to ensure that your target system meets the hardware and software requirements for installing or upgrading and running Collaborative Review , run the checks and perform the tasks in this section.

# Obtaining a License File before installing or upgrading

We strongly recommend that you obtain your license file before installing Collaborative Review , which requires the Reprise license manager.

### About this task

To contact Technical Support, connect to the Customer Support Web Portal (<http://www.sdl.com/support/>), and then click LOG A TICKET link in the "Current support system" column for your SDL product.

You will need an account to log a ticket. If you do not have an account, contact the designated representative at your site, as identified in your service agreement.

Collaborative Review requires a license file.

---

**Note:** You may choose to install a server without a license file, but it will run with the following limited functionality until you obtain and install a valid license file:

- Only 20 concurrent users can view publications.
  - Documents and graphics cannot be uploaded.
  - Publications cannot be prepared.
  - Packages cannot be created.
- 

### Procedure

1. Provide your site code.

Your site code is listed in the implementation package under contact information for your company. The site code also appears as your Customer Number on SDL invoices and in existing license files.

2. Provide the host name of the designated server. To find the host name, do one of the following:
  - a. On Windows, right click the **My Computer** icon on the desktop, then click **Properties**. On the Computer Name tab, the host name is displayed as the *Full computer name*.
  - b. On Linux, at a command prompt, type `hostname`.  
Indicate only the host name (for example, `neptune`) and not the fully qualified host name (for example, `neptune.xyenterprise.com`).
3. Provide the host ID (on Windows, MAC Address) of the designated server.
  - a. On Windows, at a DOS prompt, type `ipconfig /all`.  
The host ID is the value listed for Physical Address under Ethernet adapter Local Area Connection in the following format: 00-00-00-00-00-00. If you have multiple Ethernet connections, use the physical address for the first Ethernet connection in the list.

- b. On Linux, at a command prompt, type `/sbin/ifconfig eth0`.  
The host ID is the HWaddr (without the colons).
4. Your license file will be sent via email from SDL as an attachment. Save the file attachment to `c:\temp` (Windows) or `/tmp` on the Collaborative Review computer.  
This is only a temporary location for the file; the installation program installs the license file into the correct directory.

## Collaborative Review hardware requirements

Before you install or upgrade Collaborative Review, you must check the hardware requirements prior to completing the pre-installation tasks.

Know that the hardware required for a specific implementation depends on the specific requirements and settings of the project (for example, the number of concurrent users). The exact definition of the hardware requirements is typically done at the beginning of the project.

### Web and Application layer

The minimum server configuration: A recent quad core system(s) containing 8 GB of RAM or more. Virtualized environments are supported if they are guaranteed to behave like a Windows OS installed on a physical machine. If performance is, or becomes an issue, you are advised to use a physical server.

The recommended server configuration: A quad core Xeon® X5550 2.66 GHz processor system with at least 12 MB Level 3 cache and 8 GB RAM, dual port Gigabit Ethernet, and a smart array RAID controller with 256 MB memory.

### Network requirements

A 10 Mbit network connection provides a more than acceptable throughput.

## Collaborative Review software requirements

Information about third-party or client software that is packaged, configured and tested for this software version release.

The following overview includes information about:

- Third-party software that is configured or integrated in this server-side component release.
- Third-party software, such as the operating systems, databases, and runtimes that are quality-assurance tested.
- Client hardware and software compatibility.

### Third Party Software supported versions

---

**Note:** Names, trademarks, designs, logos, service marks, intellectual property, and so on, of the products shown are exclusive property of their respective owners.

---

#### Application server

Depending on the installation of the hotfix delivered with Tridion Docs 13 SP2:  
`2018.10.26_hotfix_CollaborativeReview_7.7.0-SRQ-9266.`

- Microsoft Windows Server 2016 (64-bit)
- Microsoft Windows Server 2012R2 (64-bit)
- Red Hat Enterprise Linux version 7.3 (64-bit)
- Red Hat Enterprise Linux version 7.5 (64-bit)
- Before hotfix:
  - Java Runtime 10.0.1 (64-bit)
  - Java Development Kit 10.0.1 (64-bit)
  - Apache Tomcat 8.5.31
  - Apache Tomcat 9.0.8
- After hotfix
  - Eclipse Temurin OpenJDK 11 (formerly known as AdoptOpenJDK OpenJDK 11) with Hotspot 11+28 (64-bit)
  - Apache Tomcat 8.5.34
  - Apache Tomcat 9.0.12

---

**Note:** All examples in this documentation use Apache Tomcat as third-party server. A GUI application may then be used for fine-tuning Tomcat.

---

#### Client

- Microsoft Windows 8/8.1 (64-bit)
- Microsoft Windows 10 (64-bit)
- iOS 9 on iPad (not supported on iPhone)
- Internet Explorer 11
- Google Chrome (latest version)
- Mozilla Firefox (latest version)

---

**Note:** The browser configuration needs to allow cookies, JavaScript execution, and pop-up windows. Pop-up windows are triggered when the Content Manager Web Client makes calls to Draft Space.

---

Browser plugins (additional plugins may be required to support rich media display):

- Adobe Flash Player version 11 (32-bit and 64-bit version) depending on the web browser in use. Required to upload Flash content to Collaborative Review using the interactive interface.



- CGM: Any viewer, such as IsoView (requires an ActiveX control in the Collaborative Review data).

**Note:** Designed for a minimum resolution of 1024 x 768 pixels. Optimal resolution: 1280 x 1024 pixels or higher.

## Collaborative Review pre-installation and upgrade settings

Making sure a list of configuration or variables are set correctly before the installation will make the operation considerably smoother.

Any user who installs or upgrades Collaborative Review must have administrator rights.

The specific Java environment depends on whether or not you have installed the Collaborative Review hotfix released with Tridion Docs 13 SP2: 2018.10.26\_hotfix\_CollaborativeReview\_7.7.0-SRQ-9266. This has an impact on the environment variables value.

Before the hotfix:

- Make sure that the `JAVA_HOME` environment variable is set to the installation path for the Java SE Development Kit. Ex: `C:\Program Files\Java\jdk<version_number>`.
- Make sure that the `JRE_HOME` environment variable is set to the installation path for the Java SE Runtime Environment. Ex: `C:\Program Files\Java\jre<version_number>`.
- Make sure that Java's `java.library.path` property has a value that includes `c:\windows`.
- Make sure that the `%JAVA_HOME%` path and the Apache Tomcat JVM path are synchronized. For example:

Apache Tomcat JVM	JAVA_HOME env. var.
<code>C:\Program Files\Java\jdk1.8.0_144\jre\bin\server\jvm.dll</code>	<code>C:\Program Files\Java\jdk1.8.0_144</code>

After the hotfix:

- Make sure that the `JAVA_HOME` environment variable is set to the installation path for OpenJDK . Ex: `C:\Program Files\Java\jdk-11+28-jre`.
- Make sure that Java's `java.library.path` property has a value that includes `c:\windows`.
- Make sure that the `%JAVA_HOME%` path and the Apache Tomcat JVM path are synchronized. For example:

Apache Tomcat JVM	JAVA_HOME env. var.
<code>C:\Program Files\Java\jdk-11+28-jre\bin\server\jvm.dll</code>	<code>C:\Program Files\Java\jdk-11+28-jre</code>

# Enabling Secure Socket Layer (SSL) protocol for the Apache Tomcat web application server

Enabling the Apache Tomcat web application server to support the Secure Socket Layer (SSL) protocol and make it capable to manage secure connections (HTTPS), is a prerequisite. In most cases, the secure connection is already enabled in Apache Tomcat for other applications. If not, you must obtain and load the appropriate SSL certificates.

## Before you begin

Secure Socket Layer (SSL) protocol requires the use of a certificate.

This documentation is written based on the assumption that you acquire this certificate commercially, which is the recommended option.

---

**Note:** Alternatively, it is possible to create a certificate using open SSL or the Windows Domain Certificate Services. In this case the certificate chain will not be fully trusted by Java and that trust must be manually configured using tools such as `keytool`.

---

Depending on the certificate's provided format, it might be necessary to convert it. The following steps assume that you have one of the following:

- A Java key store file `server_name.jks`
- A PFX certificate file `server_name.pfx`

## Procedure

1. Copy `server_name.jks` OR `server_name.pfx` to the `conf` subdirectory underneath the Apache Tomcat installation.  
Typically, this directory is located:  
On Windows:  
`C:\Program Files\Apache Software Foundation\Tomcat 8.0\conf\`  
On Linux:  
`/usr/Apache Software Foundation/Tomcat 8.0/conf/`
2. Configure the Apache Tomcat web application server to use the keystore file.
  - a. Open the `server.xml` file, found underneath the `conf` subdirectory, in a text editor.
  - b. Add a connector using the matching certificate type.

On Windows (non-PFX certificate)

```
<Connector port="8443" protocol="org.apache.coyote.http11.  
Http11NioProtocol"  
    SSLEnabled="true" sslProtocol="TLS" clientAuth="false"  
    maxThreads="150" scheme="https" secure="true"  
    useBodyEncodingForURI="false"  
    URIEncoding="UTF-8"  
    connectionTimeout="7200000"  
    compression="on"  
    compressionMinSize="10"
```

```
compressableMimeType="text/html,text/xml,application/xml,text/javascript,text/css,application/x-javascript"
maxHttpHeaderSize="65536"
keystoreFile="C:\Program Files\Apache Software Foundation\Tomcat 8.0\conf\server_name.jks" keystorePass="*****"
/>
```

#### On Linux (non-PFX certificate)

```
<Connector port="8443" protocol="org.apache.coyote.http11.
Http11NioProtocol"
    SSLEnabled="true" sslProtocol="TLS" clientAuth="false"
    maxThreads="150" scheme="https" secure="true"
    useBodyEncodingForURI="false"
    URIEncoding="UTF-8"
    connectionTimeout="7200000"
    compression="on"
    compressionMinSize="10"
    compressableMimeType="text/html,text/xml,application/xml,
    text/javascript,text/css,application/x-javascript"
    maxHttpHeaderSize="65536"
    keystoreFile="./server_name.pfx" keystorePass="*****"
/>
```

#### On Windows (PFX certificate)

```
<Connector port="8443" protocol="org.apache.coyote.http11.
Http11NioProtocol"
    SSLEnabled="true" sslProtocol="TLS" clientAuth="false"
    maxThreads="150" scheme="https" secure="true"
    useBodyEncodingForURI="false"
    URIEncoding="UTF-8"
    connectionTimeout="7200000"
    compression="on"
    compressionMinSize="10"
    compressableMimeType="text/html,text/xml,application/xml,
    text/javascript,text/css,application/x-javascript"
    maxHttpHeaderSize="65536"
    keystoreFile="C:\Program Files\Apache Software Foundation
    \Tomcat 8.0\conf\server_name.pfx" keystoreType="PKCS12"
    keystorePass="*****"
/>
```

#### On Linux (PFX certificate)

```
<Connector port="8443" protocol="org.apache.coyote.http11.
Http11NioProtocol"
    SSLEnabled="true" sslProtocol="TLS" clientAuth="false"
    maxThreads="150" scheme="https" secure="true"
    useBodyEncodingForURI="false"
    URIEncoding="UTF-8"
    connectionTimeout="7200000"
    compression="on"
    compressionMinSize="10"
    compressableMimeType="text/html,text/xml,application/xml,
    text/javascript,text/css,application/x-javascript"
    maxHttpHeaderSize="65536"
    keystoreFile="/usr/Apache Software Foundation/Tomcat
    8.0/conf/server_name.pfx" keystoreType="PKCS12"
    keystorePass="*****"
/>
```

where \*\*\*\*\* is the keystore password.

- If necessary replace the port number 8443 to a more suitable one.
3. Restart the Apache Tomcat service.
  4. Check if the Tomcat home page displays correctly with a secure connection by using the link `https://example.com:8443/`

# Enabling restricted http protocol for the Apache Tomcat web application server

Enabling the Apache Tomcat web application server to support a limited access http protocol, is a prerequisite for the database administration operations to be carried out.

## Before you begin

It is recommended that all connections to Collaborative Review are encrypted using `HTTPS` schema. Therefore, when configuring Tomcat to allow inbound `HTTP` connections, you should make it an exception. This is done by binding the Tomcat connector to a specific IP.

A standard exception is for the requests to originate locally. *127.0.0.1 is the loopback Internet protocol (IP) address also referred to as the localhost. The address is used to establish an IP connection to the same machine or computer being used by the end-user.*

If you want to execute database administration from a different remote system, then instead of 127.0.0.1 use the IP of that remote system. If you want to allow `HTTP` access to all remote systems then don't specify the binding address in the `Connector`.

## Procedure

1. Configure the Apache Tomcat web application server.
  - a. Open the `server.xml` file, found underneath the `conf` subdirectory, in a text editor.
  - b. Add a connector with the following configuration:

```
<Connector port="8080" protocol="HTTP/1.1"
  redirectPort="8443"
  address="localhost"
  useBodyEncodingForURI="false"
  URIEncoding="UTF-8"
  connectionTimeout="72000000"
  compression="on"
  compressionMinSize="10"
  compressableMimeType="text/html,text/xml,application/xml,text/
  javascript,text/css,application/x-javascript"
  maxHttpHeaderSize="65536"
/>
```

where `port` is the desired port for `HTTP` and `redirectPort` is the chosen port for the `HTTPS/SSL` enabled `Connector`.

To entirely remove this limitation, remove the `address` attribute. Tomcat will bind `port` to all available IPs.

---

**Note:** If you need to bind to multiple IPs then you need to configure multiple `Connectors` nodes where each `Connector` binds to each IP using the `address` attribute.

---

2. Restart the Apache Tomcat service.
3. Check if the Tomcat home page displays correctly with a non-secure connection by using the link `http://localhost/`.



3

## **Install** Collaborative Review

To install Collaborative Review, select an installer and a setup script based on the operating system you use, and run them.

## Install Collaborative Review on Windows

Run the Windows installer and the Windows setup script to install or upgrade Collaborative Review on your Windows target machine.

### Running the Collaborative Review installer for Windows

Run the installer for Windows if you are installing or upgrading on a Microsoft Windows system.

#### Before you begin

#### Procedure

1. Log on to your target machine as an administrator-level user.
2. On your target machine, stop your Web application server.
3. From your target machine, access your Collaborative Review download location for SDL Tridion Docs 14.
4. Download and run the executable called `20180516.SDLTridionDocs.13SP1.CollaborativeReview.7.7.0.0.Windows.exe` as an administrator (right-click the executable and from the context menu that opens, select **Run as administrator**).
5. In the Collaborative Review (Install) step, select **Next**.
6. In the **License Agreement** step, read the license terms, confirm that you agree with them by selecting **I accept the terms of the License Agreement**, and select **Next**.
7. In the **Choose Web App Context** step, specify the application context name, that is, the last part of the URL. For example, given the URL `http://delivery.lc.example.com/CollaborativeReview` (where `lc` refers to an example related to output, and `delivery` specifies it further as a delivery server), the context is `CollaborativeReview`. In a content distribution model, the distribution server and all of its delivery server must have the same context. Then select **Next**.
8. In the **Choose Backup Schedule** step, specify how often you would like SDL Tridion Docs to back up the Collaborative Review database and select **Next**.
9. In the **Install License** step, do one of the following:
  - If you have a license file, select **Yes** to proceed to the **Choose a License File** step. In this step, locate your license file and select **Next**.
  - If you do not yet have a license file, select **No** to continue with the installation. You can install a license file later, but until you do, you will not be able to work with Collaborative Review.
10. In the **Install Summary** step, select **Install**. The installer installs the software.
11. In the **Installation Successful - SDL Tridion Docs Collaborative Review** step, note down the provided command prompt syntax and select **Next**.



12. In the **Install Complete** step, select **Done** to close the installer and save an installation log to the `logs\` subfolder of your SDL Tridion Docs home folder.

## Running the database setup script for Windows

The database setup script installs or upgrades your Collaborative Review database.

### Procedure

1. Log on to your target machine as an administrator-level user.
2. Make sure that the `JAVA_HOME` environment variable is set to a supported version of Java.
3. Ensure that Java's `java.library.path` property has a value that includes `c:\windows`.
4. If your Web application server is Apache Tomcat, do the following to enable management of the Collaborative Review Web application from your Apache Tomcat Web Application Manager:
  - a. In Windows Explorer, navigate to the SDL Tridion Docs home folder, and then to its `WEB-INF\lib\` subfolder.
  - b. From this location, copy `rlmnum.jar` (where *num* is a series of digits, for example 923).
  - c. Navigate to the Apache Tomcat home folder, and then to its `lib\` subfolder, and paste the file.
5. Open the Windows Start menu and navigate to **All Programs > Accessories**.
6. Right-click **Command Prompt** and from the context menu that opens, select **Run As Administrator**. A command prompt with administrator-level access opens.
7. Navigate to the SDL Tridion Docs home folder and then into its `WEB-INF\` subfolder.
8. Enter the following command:

```
setup.bat install
```

Add the following switches if necessary:

**-Dlc.protocol=HTTPS**

Add this to the end of your command if your Collaborative Review Web application runs securely over an HTTPS protocol.

**-Dlc.port=PORTNUMBER**

Add this to the end of your command if your Collaborative Review Web application runs on a different port than 8080, indicated by *PORTNUMBER*.

**-Dlc.context=CONTEXT**

Add this to the end of your command if your Collaborative Review Web application runs in a specific application context other than the default context `/ContentDelivery`, indicated by *CONTEXT* (must start with a `/`).

**-Dlc.host=HOST**

Add this to the end of your command if your Collaborative Review Web application on a host other than the default, `localhost`, indicated by `HOST`.

**-jh "C:\Program Files\Java\jdk-11+28-jre"**

Add this to the command if you haven't already set your `JAVA_HOME` environment variable appropriately. It specifies the directory where Java is installed (use this value if you have installed OpenJDK).

---

**Note:** Unlike other options, `-jh` is used before the command name not after. E.g.  
`setup.bat jh "C:\Program Files\Java\jdk-11+28-jre" install.`

---

The setup script initializes the database and uploads program files and skins to it.

## Install Collaborative Review on Linux

Run the Linux installer and the Windows setup script to install or upgrade Collaborative Review on your Linux target machine.

## Running the Collaborative Review installer for Linux

Run the installer for Linux if you are installing or upgrading on a Red Hat Linux Windows system.

### Procedure

1. Log on to your target machine as an administrator-level user.
2. On your target machine, stop your Web application server.
3. From your target machine, access your Collaborative Review download location for SDL Tridion Docs 14.
4. Run the executable called `20180516.SDLTridionDocs.13SP1.CollaborativeReview.7.7.0.0.Linux.bin`.
5. In the **Collaborative Review (Install)** step, select **Next**.
6. In the **License Agreement** step, read the license terms, confirm that you agree with them by selecting **I accept the terms of the License Agreement**, and select **Next**.
7. In the **Choose Web Application Deployment Folder** step, navigate to the directory in which your Web application server deploys its Web applications (for Apache Tomcat, this directory is the `webapps` subdirectory of the Apache Tomcat home directory), and select **Next**.
8. In the **Choose Web App Context** step, specify the application context name, that is, the last part of the URL. For example, given the URL `http://delivery.lc.example.com/CollaborativeReview` (where `lc` refers to an example related to output, and `delivery` specifies it further as a delivery server), the context is `CollaborativeReview`. In a content distribution model, the distribution server and all of its delivery server must have the same context. Then select **Next**.
9. In the **Choose Backup Schedule** step, specify how often you would like SDL Tridion

Docs to back up the Collaborative Review database and select **Next**.

10. In the **Install License** step, do one of the following:
  - If you have a license file, select **Yes** to proceed to the **Choose a License File** step. In this step, locate your license file and select **Next**.
  - If you do not yet have a license file, select **No** to continue with the installation. You can install a license file later, but until you do, you will not be able to work with Collaborative Review.
11. In the **Install Summary** step, select **Install**. The installer installs the software.
12. In the **Installation Successful - Collaborative Review** step, note down the URL for Collaborative Review and select **Next**.
13. In the **Install Complete** step, select **Done** to close the installer and save an installation log to the `logs\` subfolder of your SDL Tridion Docs home folder.

## Running the database setup script for Linux

The database setup script installs or upgrades your Collaborative Review database.

### Procedure

1. Log on to your target machine as an administrator-level user.
2. Make sure that the `JAVA_HOME` environment variable is set to a supported version of Java.
3. If your Web application server is Apache Tomcat, do the following to enable management of the Collaborative Review Web application from your Apache Tomcat Web Application Manager:
  - a. Navigate to the SDL Tridion Docs home folder, and then to its `WEB-INF/lib/` subfolder.
  - b. From this location, copy `r1mnum.jar` (where *num* is a series of digits, for example 923).
  - c. Navigate to the Apache Tomcat home folder, and then to its `lib/` subfolder, and paste the file.
4. Start your Web application server and wait for the Collaborative Review Web application to be fully initialized. You can verify that the Web application is loaded by checking if you can access its URL.
5. In a command shell, navigate to the SDL Tridion Docs home folder and then into its `WEB-INF/` subfolder.
6. Enter the following command:

```
setup.sh install
```

Add the following switches if necessary:

**-Dlc.protocol=HTTPS**

Add this to the end of your command if your Collaborative Review Web application runs securely over an HTTPS protocol.

**-Dlc.port=PORTNUMBER**

Add this to the end of your command if your Collaborative Review Web application runs on a different port than 8080, indicated by *PORTNUMBER*.

**-Dlc.context=CONTEXT**

Add this to the end of your command if your Collaborative Review Web application runs in a specific application context other than the default context */ContentDelivery*, indicated by *CONTEXT* (must start with a */*).

**-Dlc.host=HOST**

Add this to the end of your command if your Collaborative Review Web application on a host other than the default, *localhost*, indicated by *HOST*.

**-jh "C:\Program Files\Java\jdk-11+28-jre"**

Add this to the command if you haven't already set your *JAVA\_HOME* environment variable appropriately. It specifies the directory where Java is installed (use this value if you have installed OpenJDK).

---

**Note:** Unlike other options, *-jh* is used before the command name not after. E.g. `setup.bat jh "C:\Program Files\Java\jdk-11+28-jre" install`.

---

The setup script initializes the database and uploads program files and skins to it.

## Post installation actions

After you have installed Collaborative Review, follow these tasks to establish the Single Sign On and configure the capability.

## Manually installing a license file

If you did not have a license file when you installed or upgraded Collaborative Review, or if you have a new license file to install, follow this procedure.

### About this task

---

**Note:** In case of issue, check out the troubleshooting chapters for a solution.

---

### Procedure

1. Stop your application or service. For example, if you are using Apache Tomcat, stop the Apache Tomcat service.
2. Stop the Collaborative Review license service by doing one of the following:
  - On Windows, stop the **SDL\_License** service.
  - On Linux, log in as the root user, and type the following at a command prompt:  
`/etc/init.d/SDL_License stop`
3. Copy the new license file to the license directory.

By default, this is `C:\Program Files\XyEnterprise\SDL_License` on Windows and `/opt/XyEnterprise/SDL_License` on Linux. The license file can be named anything, but must have a `.lic` extension. Make sure there is only one file in this directory with the `.lic` extension; the license manager loads all files with a `.lic` extension.

4. Start the Collaborative Review license service by doing one of the following:
  - On Windows, start the **SDL\_License** service.
  - On Linux, log in as the root user, and type the following at a command prompt:  
`/etc/init.d/SDL_License start`
5. Start your application or service. For example, if you are using Apache Tomcat, start the Apache Tomcat service.

## Configuring Collaborative Review to Support Single Sign-On (SSO) User Authentication

To enable Single Sign On (SSO) for Collaborative Review, you must load the secure socket layer (SSL) and security token service (STS) certificates, then configure the STS service to recognize the web application, and make some minor configuration file modifications.

### Establishing a trust for Collaborative Review

Establish a trust for Collaborative Review with a Security Token Service.

For Collaborative Review to integrate with a Security Token Service we need to first establish a trust on the Security Token Service.

To achieve this we first need the identifier of Collaborative Review. This is the URL of Collaborative Review. e.g. `https://example.com/LC/`.

The Security Token Service generates tokens that are processed by Collaborative Review. For the token to be valid and useful the following conditions have to be met:

- The token format must be SAML1.1 (urn:oasis:names:tc:SAML:1.0:assertion). For example in the generated token there should be an element

```
<t:TokenType>urn:oasis:names:tc:SAML:1.0:assertion</t:TokenType>
```

- You have a set of claims to drive the Collaborative Review authorization.

The generated token's attribute composition must be as follows:

Name	Claim type	Required
Name identifier		Yes
Given name	<code>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/givenname</code>	Yes

Name	Claim type	Required
Surname	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/surname	Yes
Email address	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress	Yes
Role	http://schemas.microsoft.com/ws/2008/06/identity/claims/role	No
Group	http://schemas.xmlsoap.org/claims/group	No
Display Name	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/displayname	No
Department	http://schemas.xmlsoap.org/ws/2005/05/identity/claims/department	No

**Note:** The name identifier has no claim type and is defined as part of the `subject`.

**Note:** At least one of the `Role` and `Department` must be present in the token.

Here is an example token where private information is deducted:

```
<t:RequestSecurityTokenResponse xmlns:t="http://schemas.xmlsoap.org/ws/2005/02/trust">
  <t:Lifetime>
    <wsu:Created xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">2015-11-05T09:55:42.162Z</wsu:Created>
    <wsu:Expires xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">2015-11-05T10:55:42.162Z</wsu:Expires>
  </t:Lifetime>
  <wsp:AppliesTo xmlns:wsp="http://schemas.xmlsoap.org/ws/2004/09/policy">
    <wsa:EndpointReference xmlns:wsa="http://www.w3.org/2005/08/addressing">
      <wsa:Address>https://example.com/LiveContent/</wsa:Address>
    </wsa:EndpointReference>
  </wsp:AppliesTo>
  <t:RequestedSecurityToken>
    <saml:Assertion MajorVersion="1" MinorVersion="1" AssertionID="_2609f1cf-a664-49eb-bffd-68ab598724e9" Issuer="deducted" IssueInstant="2015-11-05T09:55:42.169Z" xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
      <saml:Conditions NotBefore="2015-11-05T09:55:42.162Z" NotOnOrAfter="2015-11-05T10:55:42.162Z">
        <saml:AudienceRestrictionCondition>
          <saml:Audience>https://example.com/LiveContent/</saml:Audience>
        </saml:AudienceRestrictionCondition>
      </saml:Conditions>
      <saml:AttributeStatement>
        <saml:Subject>
```

```

<saml:NameIdentifier>user@example.com</saml:NameIdentifier>
<saml:SubjectConfirmation>
<saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer
</saml:ConfirmationMethod>
</saml:SubjectConfirmation>
</saml:Subject>
<saml:Attribute AttributeName="emailaddress" AttributeNamespac
e="http://schemas.xmlsoap.org/ws/2005/05/identity/claims">
<saml:AttributeValue>user@example.com</saml:AttributeValue>
</saml:Attribute>
<saml:Attribute AttributeName="givenname" AttributeNamespace="h
ttp://schemas.xmlsoap.org/ws/2005/05/identity/claims">
<saml:AttributeValue>deducted</saml:AttributeValue>
</saml:Attribute>
<saml:Attribute AttributeName="surname" AttributeNamespace="htt
p://schemas.xmlsoap.org/ws/2005/05/identity/claims">
<saml:AttributeValue>deducted</saml:AttributeValue>
</saml:Attribute>
<saml:Attribute AttributeName="role" AttributeNamespace="http:/
/schemas.microsoft.com/ws/2008/06/identity/claims">
<saml:AttributeValue>deducted</saml:AttributeValue>
</saml:Attribute>
<saml:Attribute AttributeName="group" AttributeNamespace="http:
//schemas.xmlsoap.org/claims">
<saml:AttributeValue>deducted</saml:AttributeValue>
</saml:Attribute>
<saml:Attribute AttributeName="displayname" AttributeNamespace=
"http://schemas.xmlsoap.org/ws/2005/05/identity/claims/display
name">
<saml:AttributeValue>deducted</saml:AttributeValue>
</saml:Attribute>
<saml:Attribute AttributeName="department" AttributeNamespace="
http://schemas.xmlsoap.org/ws/2005/05/identity/claims">
<saml:AttributeValue>deducted</saml:AttributeValue>
</saml:Attribute>
</saml:AttributeStatement>
<saml:AuthenticationStatement AuthenticationMethod="urn:federatio
n:authentication:windows" AuthenticationInstant="2015-11-05T09:55:
42.106Z">
<saml:Subject>
<saml:NameIdentifier>user@example.com</saml:NameIdentifier>
<saml:SubjectConfirmation>
<saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer
</saml:ConfirmationMethod>
</saml:SubjectConfirmation>
</saml:Subject>
</saml:AuthenticationStatement>
<ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
<ds:SignedInfo>
<ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10
/xml-exc-c14n#" />
<ds:SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsi
g-more#rsa-sha256" />
<ds:Reference URI="#_2609f1cf-a664-49eb-bffd-68ab598724e9">
<ds:Transforms>
<ds:Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enve
loped-signature" />
<ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n
#" />
</ds:Transforms>
<ds:DigestMethod Algorithm="http://www.w3.org/2001/04/xmlenc#sh
a256" />
<ds:DigestValue>deducted</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>deducted</ds:SignatureValue>
<KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
<X509Data>

```

```

        <X509Certificate>deducted</X509Certificate>
      </X509Data>
    </KeyInfo>
  </ds:Signature>
</saml:Assertion>
</t:RequestedSecurityToken>
<t:TokenType>urn:oasis:names:tc:SAML:1.0:assertion</t:TokenType>
<t:RequestType>http://schemas.xmlsoap.org/ws/2005/02/trust/Issue</t:RequestType>
<t:KeyType>http://schemas.xmlsoap.org/ws/2005/05/identity/NoProofKey</t:KeyType>
</t:RequestSecurityTokenResponse>

```

## Establishing a trust: The ISHSTS example

The configuration steps when using Content Manager STS (ISHSTS) as the Secure Token Service.

### Procedure

1. In a web browser, go to the STS web application.

For example:

`https://example.com/ISHSTS/`

---

**Note:** You must use the secure `https://` protocol.

---

2. Sign in using a Content Manager administrative user account.  
The **administration** option is visible.
3. Click **administration > Relying Parties & Resources**  
You see the list of **Enabled Relying Parties**.
4. Under the list of **Enabled Relying Parties** click **New**.
5. Check the **Enabled** checkbox.
6. Fill in the **Display Name**. The value must start with `LC` and can be followed by anything.  
Example

LC: example.com

7. Fill in the **Realm/Scope Name** with the URL of the Collaborative Review web application.  
Example

`https://example.com/KnowledgeCenterAppName/`

8. Click **Create** to insert the new relying party.  
A new entry appears with display name in the **Enabled Relying Parties** list.



## Results

Under **administration > Relying Parties & Resources** the newly created relying party is visible in the tree and in the **Enabled Relying Parties** list.

## Establishing a trust: The ADFS example

The configuration steps when using ADFS as the secure token service.

### Procedure

1. Log on to the ADFS server using administrative credentials
2. From the Server Manager, click **Tools > AD FS Management** to start the ADFS Management Console.
3. In the tree follow **AD FS > Service > Claim Descriptions**  
You see a list of predefined claim descriptions. A claim description is a combination of name and claim type. Claim descriptions simplify the configuration of claims transformation rules.
4. Check if a claim description for **Display Name** with a claim type having the description `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/displayname` exists. If it doesn't, then:
  - a. Click **Add Claim Description...**
  - b. Fill in **Display Name** for the **Display Name**.
  - c. Fill in `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/displayname` for the **Claim Identifier**.
  - d. Check the checkbox **Publish the claim description in the federation metadata as a claim type that this Federation Service can send**.  
A claim description for **Display Name** with claim type having the description `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/displayname` exists.
5. Check if a claim description for **Department** with claim type having the description `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/department` exists. If it doesn't then:
  - a. Click **Add Claim Description...**
  - b. Fill in **Department** for the **Display Name**.
  - c. Fill in `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/department` for the **Claim Identifier**.
  - d. Check the checkbox **Publish the claim description in the federation metadata as a claim type that this Federation Service can send**.  
A claim description for **Department** with claim type having the description `http://schemas.xmlsoap.org/ws/2005/05/identity/claims/department` exists.
6. In the tree, follow **Trust Relationships > Relying Party Trusts**.
7. Click **Add Relying Party Trust**.  
The Add Relying Party Trust Wizard starts.
8. Click **Start**.
9. On the Select Data Source page, select **Enter data about the relying party manually**. Click **Next**.

10. On the Specify Display Name page specify a name under **Display name** (for example, Collaborative Review), and then specify any additional notes to describe the application (for example, it might be helpful to specify the server name and IP address). Click **Next**.
11. On the Choose Profile page, select **AD FS profile**. Click **Next**.
12. On the Configure Certificate page just click **Next**.
13. On the Configure URL page:
  - a. Select **Enable support for the WS-Federation Passive protocol**.
  - b. Under **Relying party WS-Federation Passive protocol URL**, specify the URL to the web application for the web application for Collaborative Review.

For example:

`https://example.com/KnowledgeCenterAppName/`

---

**Note:** The trailing slash (/) is important.

---

- c. Click **Next**.
14. On the Configure Identifiers page, just click **Next**.

The URL filled in the previous step is already populated in the list.
15. On the Configure Multi-Factor authentication type now? page, just click **Next**.

The expected default value is `I do not want to configure multi-factor authentication settings for this relying party trust at this time`.
16. On the Choose Issuance Authorization Rules page, just click **Next**.

The expected default value is **Permit all users to access this relying party**.
17. On the Ready to Add Trust page, confirm your selections by exploring the tabs, and then click **Next** to add the relying party trust to the ADFS configuration database.
18. On the Finish page, select **Open the Edit Claim Rules dialog for this relying party trust when the wizard closes** and then click **Close**.

A dialog box for editing the claim rules for the Collaborative Review product appears.
19. On the Issuance Transform Rules tab, click **Add Rule**.

The Add Transform Claim Rule Wizard opens.
20. On the Select Rule Template page, under **Claim rule template**, select **Send LDAP Attributes as Claims** from the list.

---

**Note:** Unless specified otherwise, clicking **Next** is an assumed step to proceed for all subsequent steps.

---

21. On the Configure Claim Rule page, specify these settings:

**Claim rule name**

Specify a descriptive name (for example, `Mappings for Output`).

**Attribute store**

Select **Active Directory** from the list.

**Mapping of LDAP attributes to outgoing claim types**

Specify these mappings:

LDAP attribute	Outgoing claim type
Department	Department
E-Mail-Addresses	E-Mail Address
Given-Name	Given Name
Surname	Surname
Title	Role
User-Principal-Name	Name ID
Display-Name	Display Name

22. Click **Finish** and **Apply** to apply the claim rule changes, and then click **OK** to exit the wizard.

**Results**

A relying party trust was created on ADFS for Collaborative Review. With this trust, a user can authenticate on ADFS for Collaborative Review and then ADFS will issue a token populated with a claim set generated by the claims transformation rules.

## Setting configuration files for SSO

In order to achieve federation for Collaborative Review, you need to properly adapt settings in 3 configuration files. These procedures apply to an installation that uses an Apache Tomcat web application server.

### Configuring SSO in LiveContentSSO.xml

Configure the integration with a Security Token Service.

**Before you begin**

To configure this file you need to gather and prepare the following values

- The WS Federation endpoint. The value is provided by the owner of the target STS. Some examples:
  - For ADFS the typical endpoint is `https://ads.example.com/ads/ls/`.
  - For ISHSTS the typical endpoint is `https://example.com/ISHSTS/issue/wsFed`.
- A Java key store file with the STS's token signing certificate. The public key of the token signing certificate is provided by the owner of the target STS.
- The subject name of the token signing certificate. The value is provided by the owner of the target STS.

For example the owner of the STS has provided a `token.signing.cer` certificate.

#### Procedure

1. Prepare a Java key store file containing the `token.signing.cer`.
  - a. Choose a location and a name for the key store file. For example in `INSTALL_PATH\token.signing.jks`.
  - b. Using java's `keytool`, import the `token.signing.cer` into a new Java key store file.  
For Windows the command is

```
keytool -import -keystore INSTALL_PATH\token.signing.jks -alias
mystscert -trustcacerts -file token.signing.cer -noprompt -storepass
password
```

For Linux the command is

```
keytool -import -keystore INSTALL_PATH/token.signing.jks -alias
mystscert -trustcacerts -file token.signing.cer -noprompt -storepass
password
```

where `INSTALL_PATH` is the Collaborative Review installation path and `password` is the password for the key store.

2. From the Collaborative Review installation folder, open `\webapp\WEB-INF\LiveContentSSO.xml`.
3. Make sure the name in `contextConfig` node matches the **Web App Context** from the installation.  
The default is `ContentDelivery`.
4. Set the `audienceItem` to Collaborative Review URL.  
You can add additional entries if necessary.
5. Add a `issuer` element in the `trustedIssuers` to accept the token signing certificate. Use a regular expression based on the subject name of the certificate.

**Note:** Every `issuer` element defines a `subject` attribute that is a regular expression to match the signing certificate's subject name for an incoming token. You can add multiple issuers to be trusted by Collaborative Review. For every STS you need to import its token signing certificate into the `token.signing.cer` key store.

```
<issuer subject="CN=.*TokenSigningSubjectName.*" certificateValidation="
ChainTrust" name="NameOfSTS" />
```

where `TokenSigningSubjectName` is the subject name of the token signing certificate and `NameOfSTS` the name of the STS. The `NameOfSTS` has no functional purpose but helps annotate the entry.

6. Add a `issuer` element in the `protocol` to provide the WS Federation endpoint from the Security Token Service.  
For ISHSTS it looks like this

```
<issuer>https://example.com/ISHSTS/issue/wsfed</issuer>
```

For ADFS it looks like this

```
<issuer>https://adfs.example.com/adfs/ls/</issuer>
```

7. Add a `keyStore` element in the `trustManager` to enable the `ChainTrust` validation. The `keyStore` element is a reference to a Java key store. Use the one containing the token signing certificate `INSTALL_PATH\conf\token.signing.jks`.

```
<!--Windows-->
<keyStore file="INSTALL_PATH\token.signing.jks" password="password"
type="JKS" />
<!--Linux-->
<keyStore file="INSTALL_PATH/token.signing.jks" password="password"
type="JKS" />
```

where `INSTALL_PATH` is the Collaborative Review installation path and `password` is the password for the key store.

### Example of LiveContentSSO.xml content

This example applies to ADFS. For a ISHSTS application, follow the recommendations provided at the start of this task.

```
<FedizConfig>
  <contextConfig name="/ContentDelivery">
    <audienceUri>
      <audienceItem>https://example.com/ContentDelivery/</audienceItem>
    </audienceUri>
    <certificateStores>
      <trustManager>
        <keyStore file="C:\Program Files\Apache Software Foundation
          \Tomcat 8.0\webapps\LiveContentSTS\token.signing.jks" password=
            "password" type="JKS" />
      </trustManager>
    </certificateStores>
    <trustedIssuers>
      <issuer subject="CN=TokenSigningSubjectName.*" certificateValida-
        tion="ChainTrust" name="STS" />
    </trustedIssuers>
    <maximumClockSkew>1000</maximumClockSkew>
    <protocol xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:type="federationProtocolType" version="1.0.0">
      <issuer>https://example.com/ISHSTS/issue/wsfed</issuer>
      <roleDelimiter>,</roleDelimiter>
      <roleURI>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/
        role</roleURI>
      <freshness>10000</freshness>
      <claimTypesRequested>
        <claimType type="a particular claim type" optional="true" />
      </claimTypesRequested>
    </protocol>
  </contextConfig>
</FedizConfig>
```

## Configuring SSO in LiveContentSecurity.xml

Adapting LiveContentSecurity.xml forces Collaborative Review to redirect the user to the STS whenever the user is not properly authenticated. Therefore a non-authenticated user will not go past the login page.

### Procedure

1. From the Collaborative Review installation folder, open \webapp\WEB-INF\LiveContentSecurity.xml and make these changes:
2. Enable all <security-constraint> in the file.

### Example LiveContentSecurity.xml

```
<security-role>
  <role-name>*</role-name>
</security-role>
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Protected Area</web-resource-name>
    <url-pattern>/*</url-pattern>
  </web-resource-collection>
  <auth-constraint>
    <role-name>*</role-name>
  </auth-constraint>
</security-constraint>
<security-constraint>
  <web-resource-collection>
    <web-resource-name>Public</web-resource-name>
    <description>XmlRpc calls</description>
    <url-pattern>/xmlrpc/*</url-pattern>
  </web-resource-collection>
</security-constraint>
```

## Configuring SSO in lc.properties

lc.properties needs to be adapted so that Collaborative Review's internal pipeline is compatible with SSO.

### Procedure

1. From the Collaborative Review installation folder, open \webapp\WEB-INF\lc.properties and make these changes:
2. Change the setting for identity.provider from internal to sso.

### Example of lc.properties content

```
identity.provider=sso
sso.anonymous_users=disable
sso.saml_attribute_list=role,givenname,department,surname,emailaddress,
displayname
sso.permission.group=role,department
```

## Configuring SSO in LiveContentGroups.xml

Configure authorization for Collaborative Review.

Collaborative Review authorization is controlled by the configuration in the file `\webapp\WEB-INF\LiveContentGroups.xml`. The purpose of the file is to help the authorization pipeline to map attributes coming from a security token into the Collaborative Review groups.

Collaborative Review acknowledges the following groups:

- Consumers
- Tech Docs
- Contributors
- Contribution Managers
- Publication Managers
- Developers

In the configuration file the mapping can be configured through two distinct segments:

- User mapping to Collaborative Review group
- Attribute mapping to Collaborative Review group

The target Collaborative Review group is referenced with `app-group`.

User mapping has priority over attribute mapping configuration and attribute is even skipped when a user mapping is matched.

**Note:** When there is not match for either user or attribute mapping, then the user is not assigned any Collaborative Review group and the sign in will fail with a 401 `Permission Denied` error message.

### User mapping to Collaborative Review group

A `user` node in the `users` maps a user's `username` with a group. For example the user with `username user@example.com` is mapped to the Collaborative Review `Consumers` group.

```
<user>
  <username>user@example.com</username>
  <app-group>Consumers</app-group>
</user>
```

**Note:** The username matches the subject name identifier from the incoming token.

```
<saml:Subject>
  <saml:NameIdentifier>user@example.com</saml:NameIdentifier>
  <saml:SubjectConfirmation>
    <saml:ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</saml:
      ConfirmationMethod>
    </saml:SubjectConfirmation>
  </saml:Subject>
```

### Attribute mapping to Collaborative Review group

A `group` node in the `groups` maps a user's specific attribute with a Collaborative Review group. The attribute is specified with the XML attributes:

- `attribute`
- `value`

For example the user having an attribute `role` with value `RoleA` is mapped to the Collaborative Review `Consumers` group.

```
<group attribute="role" value="RoleA">
  <app-group>Consumers</app-group>
</group>
```

**Note:** Multiple Collaborative Review groups can be assigned to one user.

**Note:** in the `\webapp\WEB-INF\lc.properties` the value of `sso.permission.group` plays an important role. By default the value is

```
sso.permission.group=role,department
```

When the security token has at least one role attribute then only attribute mappings for role will be considered.

```
<group attribute="role" value="RoleA">
  <app-group>Consumers</app-group>
</group>
```

Only when no role attributes are found will attribute mappings for department be considered.

### Using a default group

If you are using your own certificate provider, ADFS for example, any user that is not included in a specific group will be allowed access to Collaborative Review through a default group.

That group is named `Visitors` by default. You can change this value in the file `LiveContentGroups.xml`, under the `<defaultgroup>` element.

### User mapping for Collaborative Review publishing and synchronization.

Content Manager pushes content to Collaborative Review as part of a function known as publishing or synchronizing to Collaborative Review. Both actions execute under a pre-configured user. This user's security token must be mapped to the `dba` role on Collaborative Review.

The executing user depends on Content Manager input parameter `issuerwstrustbindingtype`.

- When `UserNameMixed` then value is derived from input parameter `serviceusername`. For example `ServiceUser`.



- When `WindowsMixed` then value is derived from input parameter `osuser`. For example `osuser@example.com`.

The following example reflects both cases but you only need to use only one.

```
<user>
  <username>osuser@example.com</username>
  <app-group>dba</app-group>
</user>
<user>
  <username>ServiceUser</username>
  <app-group>dba</app-group>
</user>
```

### Simple example with role mapping

To setup a configuration with conditions:

- User `admin@example.com` has always Collaborative Review dba rights.
- All users except (`admin@example.com`) with role `Manager` have Collaborative Review Contribution Managers and Publication Managers rights.
- All users except (`admin@example.com`) with role `Administrator` have Collaborative Review dba rights.

use the following configuration

```
<mappings>
  <groups>
    <group attribute="role" value="Administrator">
      <app-group>dba</app-group>
    </group>
    <group attribute="role" value="Manager">
      <app-group>Contribution Managers</app-group>
    </group>
    <group attribute="role" value="Manager">
      <app-group>Publication Managers</app-group>
    </group>
  </groups>
  <users>
    <user>
      <username>admin@example.com</username>
      <app-group>dba</app-group>
    </user>
  </users>
</mappings>
```

### Simple example with department mapping

To setup a configuration with conditions:

- User `admin@example.com` has always Collaborative Review dba rights.
- No user has a `role` attribute.
- All users except (`admin@example.com`) with department `Managers` have Collaborative Review Contribution Managers and Publication Managers rights.
- All users except (`admin@example.com`) with department `Administrators` have Collaborative Review dba rights.

use the following configuration

```
<mappings>
  <groups>
    <group attribute="role" value="Administrators">
      <app-group>dba</app-group>
    </group>
    <group attribute="role" value="Managers">
      <app-group>Contribution Managers</app-group>
    </group>
    <group attribute="role" value="Managers">
      <app-group>Publication Managers</app-group>
    </group>
  </groups>
  <users>
    <user>
      <username>admin@example.com</username>
      <app-group>dba</app-group>
    </user>
  </users>
</mappings>
```

#### Example for Content Manager ISHSTS

Content Manager provides a user repository where every user is assigned to one or more roles. ISHSTS will populate the role attribute with the values of the Content Manager field with name `FISHUSERROLES` and value type `element`. For example:

- `VUSERROLEADMINISTRATOR`
- `VUSERROLEAUTHOR`
- `VUSERROLEREVIEWER`
- `VUSERROLETRANSLATOR`

Collaborative Review should map these roles as:

Content Manager role	Collaborative Review group
VUSERROLEADMINISTRATOR	dba
VUSERROLEAUTHOR	Publication Managers
VUSERROLEREVIEWER	Contribution Managers
VUSERROLETRANSLATOR	VUSERROLETRANSLATOR

For this example we will assume that ISHSTS is deployed in the default mode of username/password authentication mode. This is the case when input parameter `issuerwstrustbindingtype` is `UserNameMixed`. Therefore we need to guarantee a `dba` group for Collaborative Review for the user defined in the input parameter `serviceusername`.

A suitable configuration is:

```
<mappings>
  <groups>
    <group attribute="role" value="VUSERROLEADMINISTRATOR">
      <app-group>dba</app-group>
    </group>
  </groups>
```

```

    <group attribute="role" value="VUSERROLEAUTHOR">
    <app-group>Publication Managers</app-group>
    </group>
    <group attribute="role" value="VUSERROLEREVIEWER">
    <app-group>Contribution Managers</app-group>
    </group>
    <group attribute="role" value="VUSERROLETRANSLATOR">
    <app-group>Developers</app-group>
    </group>
  </groups>
  <users>
    <user>
      <username>ServiceUser</username>
      <app-group>dba</app-group>
    </user>
  </users>
</mappings>

```

As an alternative, the ISHSTS can be deployed in windows authentication mode. This is the case when input parameter `issuerwstrustbindingtype` is `WindowsMixed`. Therefore we need to guarantee a `dba` group for Collaborative Review for the user defined in the input parameter `osuser`.. For example `osuser@example.com`.

The example becomes this

```

<mappings>
  <groups>
    <group attribute="role" value="VUSERROLEADMINISTRATOR">
    <app-group>dba</app-group>
    </group>
    <group attribute="role" value="VUSERROLEAUTHOR">
    <app-group>Publication Managers</app-group>
    </group>
    <group attribute="role" value="VUSERROLEREVIEWER">
    <app-group>Contribution Managers</app-group>
    </group>
    <group attribute="role" value="VUSERROLETRANSLATOR">
    <app-group>Developers</app-group>
    </group>
  </groups>
  <users>
    <user>
      <username>osuser@example.com</username>
      <app-group>dba</app-group>
    </user>
  </users>
</mappings>

```

## Restarting Apache Tomcat to Enable SSO

How to restart the Apache Tomcat service on the Collaborative Review server.

### Procedure

1. Once all the SSL, SSO, and STS configuration steps have been completed, restart the Apache Tomcat service on the Collaborative Review server to enable the appropriate user authentication behavior.

### Results

After you restart Apache Tomcat, any attempt to access the Collaborative Review URI (Collaborative Review) summons the authentication page specified for the STS implementation at your site.

When you are authenticated, you are directed to the Collaborative Review landing page.

## Configuring the Collaborative Review Web application

Optimize Collaborative Review Web application performance by setting Java's initial and maximum heap size and by overriding session timeouts.

### Procedure

1. Based on the amount of RAM available on your target machine, check and set the initial and maximum memory pool (heap size) used by Java in your Web application server. For example:
  - If your Web application server is Apache Tomcat running on a Windows machine, run the `tomcatVERSION.exe` tool, where *VERSION* is the Tomcat version (say 8), select the **Java** tab and set **Initial memory pool** and **Maximum memory pool** to your preferred size (**Thread stack size** can be `null`).
  - If your Web application server is Apache Tomcat running on a Linux machine, access the `bin/` subdirectory of your Apache Tomcat home directory and edit the file `setenv.sh`. In this file, include a value for `JAVA_OPTS` by including a line like the following:

```
export JAVA_OPTS="-XmsINITPOOL -XmxMAXPOOL"
```

where *INITPOOL* is your initial pool size, and *MAXPOOL* is your maximum pool size.

If your Tomcat server has at least 8 GB of RAM, your initial pool size can be set to 2048, and your maximum pool size to 6144.

2. Specify the Tomcat session timeout override for Collaborative Review sessions created by Content Manager publish tasks.

The Tomcat web application server has a timeout setting for each Collaborative Review session. When user inactivity is detected in Collaborative Review, a publish process may be terminated based on Tomcat's session timeout. Tomcat's session timeout can be overridden for publishing by adding the `pub.session.timeout` configuration item, so a long publish will not be interrupted.

---

**Note:** By default, `pub.session.timeout` is set to 600 (minutes). When working with publications including several thousands of topics or more, it is suggested to at least double this number.

---

- a. From the landing page, click **Manage Application > Global Config**.
  - b. In the right pane, under **Customized Config**, click **Add Item**.
  - c. In the **Name** field, enter `pub.session.timeout`.
  - d. In the **Value** field, enter a timeout value in minutes.
  - e. Click **Save**.
3. Specify the Tomcat session timeout override for all Collaborative Review sessions.

**Example:** Tomcat has a default session timeout of 30 minutes that is imposed on every web application that the Tomcat instance serves. In the case of the Collaborative Review application, that timeout means that you will have to log in again after 30 minutes of inactivity. You can override the Tomcat session timeout defined in the `$tomcatdir\conf\web.xml` file by editing the `ContentDelivery_home\WEB-INF\LiveContentSecurity.xml` file to include a `session-timeout` setting.

For example, to set the timeout to 1 hour (60 minutes):

```
<session-config>
  <session-timeout>60</session-timeout>
</session-config>
```

**Note:** To avoid many dead sessions that may linger, and possibly degrade performance of Tomcat and/or Collaborative Review, do not specify a session timeout greater than 2 hours (120 minutes).

## Configuring context.xml files

When you proceed to an installation, we recommend that you make specific changes in Tomcat `context.xml` file and in Collaborative Review `context.xml` file.

### Tomcat context.xml file

That file is typically located under `<Tomcat_home>/config`.

In case of restart, Tomcat restores the session from disk by default and that does not go well with an ongoing publication or other Collaborative Review running processes. In order to prevent this, un-comment this line in the `context.xml` file.

```
<Manager pathname="" />
```

If it is not in the file, add it.

As an alternative for preventing session restores from disk, you can also set `<Manager saveOnRestart>` to `false` in the `context.xml` file.

```
<Manager saveOnRestart='false' .../>
```

**Collaborative Review context.xml file**

That file is typically located in META-INF. A change in Apache Tomcat 8.0.30 on URI redirection made a parameter setting mandatory for using Collaborative Review 7.4 and later with Tomcat 8.0.30 and later. Without this setting the system will return the following error: Invalid URI: The format of the URI could not be determined.

Add the attribute and value `useRelativeRedirects= "false"` to the Context element of the `context.xml` file in META-INF

```
C:\Program Files\Apache Software Foundation\Tomcat 8.0\webapps\ISHCD\META-INF\context.xml
<Context xmlBlockExternal="false" useRelativeRedirects="false">
```

## Configuring Apache Tomcat

Apache Tomcat must be configured for Java.

**About this task**

The configuration actions depend on how you start Tomcat.

**Procedure**

1. If you start Tomcat via `catalina.bat (sh)` or `startup`:
  - a. Navigate to `\Tomcat\bin\catalina.bat (sh)`
  - b. Search for the comment or `CATALINA_HOME/endorsed` exists
  - c. If it exists, delete all endorsed attributes that follow (the `set ENDORSE_PROP` commands).
2. (Ignore this step if you use OpenJDK) If you start Tomcat via **Tomcat Monitor**:
  - a. Navigate to the **Java** tab.
  - b. In **Java options**, search for the `-Djava.endorsed.dirs` option. If it exists, delete it.
  - c. In **Java [version] options**, add `--add-modules=java.se.ee`

## Setting the path to Linux installation logs

By default, when you run `startup.sh`, your installation log files are saved in a `logs/` subdirectory relative to the directory from which you run `startup.sh`. For example, in `$tomcat`, running `bin/startup.sh` makes logs appear in `$tomcat/logs/`. To log to the same directory at all times, in Collaborative Review's `WEB-INF/log4j.xml`, specify the full path to your `$tomcat/logs` directory.

## Enabling the use of non-ASCII characters on Collaborative Review

Only if you use Apache Tomcat 7 or earlier, add `URIEncoding= "UTF-8"` to every HTTP connector in your `server.xml` file. (For Apache Tomcat 8 or higher, this attribute is present by default.)

### Procedure

1. On your Collaborative Review server, navigate to `%CATALINA_HOME%\ conf\ .`
2. Open the file `server.xml` for editing.
3. Find all `Connector` elements that define an HTTP connection. For example:

```
<Connector
  port="8080"
  protocol="HTTP/1.1"
  connectionTimeout="20000"
  redirectPort="8443" />
```

4. Add an attribute called `URIEncoding` to this element and set it to the value `UTF-8`. For example:

```
<Connector
  port="8080"
  protocol="HTTP/1.1"
  connectionTimeout="20000"
  URIEncoding="UTF-8"
  redirectPort="8443" />
```

5. Save and close `server.xml`.
6. Restart Apache Tomcat.





# 4

## **Validating and testing Collaborative Review installation or upgrade**

Once the Collaborative Review installation or upgrade has been completed, the database restored, the certificates installed, make sure the system runs correctly.

## Validating the Installation

After you have installed or upgraded Collaborative Review , perform these steps to ensure everything is working correctly.

### Before you begin

---

#### Important:

If you plan on running anti-virus software, you must ensure that you exclude from the scan the Collaborative Review database path. Typically, this is the `WEB-INF\data` folder (`WEB-INF/data` on Linux) in the Collaborative Review application.

For example, if you are running Apache Tomcat, you would exclude this directory from the scan:

On Windows:

`ApacheTomcat_home\webapps\CollaborativeReview\WEB-INF\data`

On Linux:

`ApacheTomcat_home/webapps/CollaborationReview/WEB-INF/data`

where `ApacheTomcat_home` is the Apache Tomcat installation directory.

---

#### Procedure

1. Browse to the Collaborative Review home page at `https://delivery.lc.example.com/CollaborativeReview` (where `lc` refers to an example related to output, and `delivery` specifies it further as a delivery server).  
If you installed on a Windows Server, use another computer to browse to the Tridion Docs home page.
2. If you are on a Collaborative Review page, select **Home** in the breadcrumb trail.  
The Collaborative Review landing page appears.
3. In the Administration Tools pane, select **Manage Application**.  
The Manage Application page appears.
4. If this is a new installation, specify a password for the administrator user.

---

**Note:** If this is an upgrade installation, you do not need to change the administrator user password. You can omit this step.

---

- a. Select **Manage Users**.
- b. Adjacent to **admin** user name, hover over the icons to find **Change password** and then select that icon.

- c. Enter a password, then reenter the password, and then select **Save**.
5. Select **System Status**.
6. Confirm that the file paths, operating system, and other status items are correct.
7. Upload the sample data set and view the publication.  
See *"Testing with Sample Data"* on page 53.

---

**Note:** The `lcapi.properties` file is used to load API topics from the `LiveContentDoc.pac` file, and will always be included as an unused resource when running validation. See *Testing with Sample Data*.

---

8. Manually select items in the table of contents to ensure that all `<topicref>` elements work properly. You may then tune the database for optimal performance (search for *Tuning performance*).

## Testing with Sample Data

When you install Collaborative Review, some sample data is included that you may upload and view to test your system. This procedure provides instructions on creating a publication and language version, uploading content, and preparing the publication.

### About this task

`ContentDelivery_home` refers to `ContentDelivery_InstallDir\WEB-INF` on Windows and `ContentDelivery_InstallDir/WEB-INF` on Linux.

### Procedure

1. On a client machine, browse to the Collaborative Review home page:  
`http://delivery.lc.example.com/CollaborativeReview` (where `lc` refers to an example related to output, and `delivery` specifies it further as a delivery server)
2. Log in as a user who belongs to a group with permission to manage publications.  
The initial user is admin. For new installations, there is no password set initially.
3. If you are on a Collaborative Review page, select **Home** in the breadcrumb trail.  
The Collaborative Review landing page appears.
4. In the Administration Tools pane, select **Manage Publications**.  
The Manage Publications page appears.
5. Select **Add Publication**.
6. Do the following:
  - a. In the **Publication Name** field, type a publication name.

**Example:** For example, *Sample Publication*.

This is the title that appears in the left pane. The title that appears in the publication list is the title contained in the map, *Tridion Docs Sample Data*.

- b. Select **Standard**, and then **DITA** from the drop-down list.
- c. In the **Default Language** field, type `en`.  
In this example, the sample data is in English.
- d. Select **Save**.
7. In the left pane, navigate to the publication and language version you just created.
8. Select **Upload Resources**.
9. Select **Select Resources**.
10. Navigate to the `CollaborativeReview_home/samples/DITA`,  
where *CollaborativeReview\_home* is the directory where Collaborative Review is installed.
11. Select all the files in the folder, then **Open**.
12. Select **Upload**.
13. Select **Configuration**, then do the following:
  - a. Under **Top-Level TOC Resource**, select **sample.ditamap**.
  - b. Under **Filter Resource**, select **ProductDefinition.xml**.
  - c. Select **Save**.
14. Select **Prepare Publication**.  
You can only prepare a publication if you have installed a valid license.
15. Check **Make publication visible after successful prepare**.
16. Check **Set publication to show draft comments when viewed**, because some of the topics in the sample data use draft comments to explain their content references or other sample features.
17. Select **Prepare**.  
This resolves all the references in the publication and makes it viewable, and indexes its content so that it is searchable.
18. Select **Close**.
19. In the Administration Tools pane, select **Manage Publications**.  
The Manage Publications page appears.
20. Select the publication title, *Tridion Docs Sample Publication*, and view the content in the right pane.  
The resulting publication contains several topics, some of which contain graphics and Shockwave Flash animations.