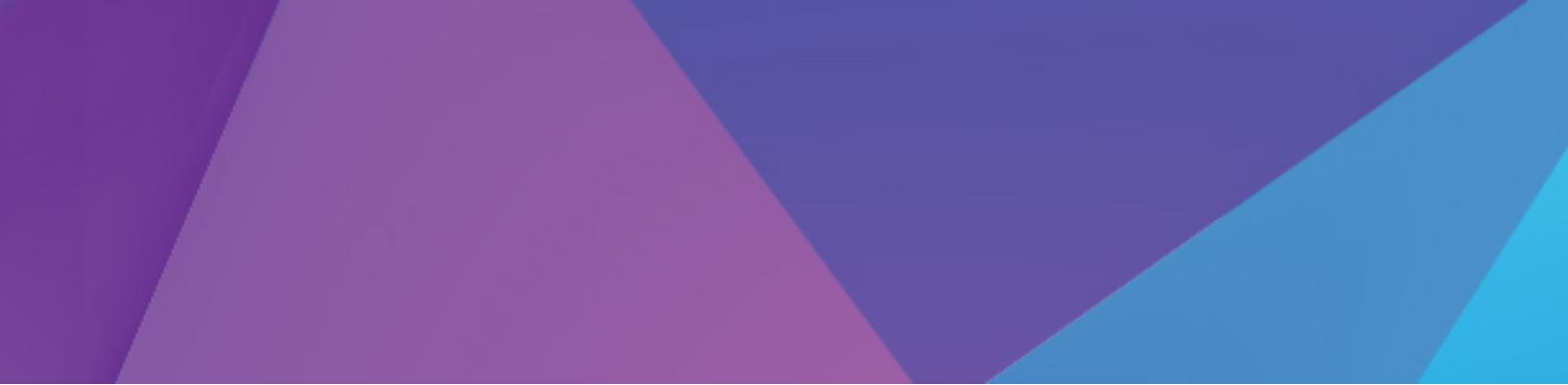




# User Guide

## Version 2019.8.0



This edition of the *User Guide* refers to version 2019.8.0 of Black Duck.

This document created or updated on Thursday, August 15, 2019.

**Please send your comments and suggestions to:**

Synopsys  
800 District Avenue, Suite 201  
Burlington, MA 01803-5061 USA

Copyright © 2019 by Synopsys.

All rights reserved. All use of this documentation is subject to the license agreement between Black Duck Software, Inc. and the licensee. No part of the contents of this document may be reproduced or transmitted in any form or by any means without the prior written permission of Black Duck Software, Inc.

Black Duck, Know Your Code, and the Black Duck logo are registered trademarks of Black Duck Software, Inc. in the United States and other jurisdictions. Black Duck Code Center, Black Duck Code Sight, Black Duck Hub, Black Duck Protex, and Black Duck Suite are trademarks of Black Duck Software, Inc. All other trademarks or registered trademarks are the sole property of their respective owners.

# Contents

<b>Chapter 1: Getting started with Black Duck .....</b>	<b>1</b>
Logging in to Black Duck .....	1
Scanning your code and mapping scans to projects .....	2
Mapping scans to projects .....	2
Administrative tasks .....	2
Importing a Protex BOM .....	3
<b>Chapter 2: About Black Duck - Binary Analysis .....</b>	<b>4</b>
<b>Chapter 3: Understanding the scanning process .....</b>	<b>5</b>
Understanding Component Scanning .....	5
Supported languages .....	7
ISO files .....	7
Supported package managers .....	7
Using Synopsys Detect (Desktop) .....	7
Downloading and installing Synopsys Detect (Desktop) .....	8
Configuring Synopsys Detect (Desktop) .....	8
Certificates .....	14
Scanning options .....	14
Creating a scan file .....	18
Managing scans .....	19
Uploading scan files to Black Duck .....	20
Viewing uploaded scans .....	21
Using the Signature Scanner .....	23
Signature Scanner client requirements .....	23
Downloading and installing the Signature Scanner CLI .....	23
Downloading the Signature Scanner CLI .....	23
Installing the Signature Scanner CLI .....	23
Defining your version of JRE for the Signature Scanner .....	24
Running a component scan using the Signature Scanner command line .....	25
Specifying the password .....	30
About package management files .....	31
Examples .....	31
Reducing the number of parameters entered on the command line for the Signature Scanner .....	32
Accessing the Black Duck server via a proxy .....	33

Running an offline component scan using the Signature Scanner .....	35
Using certificate-based authentication with the Signature Scanner .....	36
Examples .....	36
Defining the scan name .....	37
Specifying names for BOM or JSON files .....	38
Resolving memory issues .....	38
About snippet matching .....	39
Snippet scanning process .....	39
Uploading source files for snippet matching .....	41
Reviewing snippet matches .....	41
Snippet Matches and Vulnerabilities .....	41
Resolving proxy errors .....	42
About custom scan signatures - BETA .....	42
Understanding the custom scan signature process .....	42
Creating custom scan signatures .....	42
Disabling custom scan signatures .....	43
Associating custom components to custom scan signatures .....	44
<b>Chapter 4: Managing scans in the Black Duck UI .....</b>	<b>45</b>
Uploading a scan file using the Black Duck UI .....	45
Browsing scans .....	45
Viewing scan results .....	47
Mapping a scan to a project .....	48
Removing a scan from a project .....	50
Deleting a scan .....	50
Exporting a scan file .....	51
Viewing an audit log for a BOM file .....	51
<b>Chapter 5: Understanding projects in Black Duck .....</b>	<b>54</b>
Creating a project .....	56
Updating project information .....	57
Deleting a project .....	58
Managing tags .....	58
Managing project team membership .....	60
About project versions .....	67
Creating a new version of a project .....	68
Updating project version information .....	70
Cloning project versions .....	71
Enabling cloning .....	72
Deleting a project version .....	72
About project version phases .....	73
About archived project versions .....	73
<b>Chapter 6: Viewing a project version's BOM .....</b>	<b>74</b>

Understanding the information in a project version's BOM .....	74
Risk graphs .....	75
Data Table .....	75
About the hierarchical BOM .....	80
Reviewing the contents of a BOM .....	82
Managing comments .....	83
Adding a comment .....	84
Viewing a comment .....	84
Editing a comment .....	84
Deleting a comment .....	84
Managing files associated with BOM components .....	85
Accessing the Source tab .....	85
About the Source tab .....	86
Modifying matches .....	87
Identifying unmatched files .....	87
Validating matched files .....	87
Resetting files .....	88
Deleting files from a BOM .....	88
Comparing BOMs .....	88
Printing a BOM .....	91
Viewing issues in a project .....	92
Viewing component versions with encryption .....	93
About Linux distributions in Black Duck .....	95
Viewing Linux distributions in Black Duck .....	95
<b>Chapter 7: Editing a BOM .....</b>	<b>96</b>
Applying edits to all versions of a project .....	96
Persistent edit examples .....	97
Enabling or disabling persistent edits for a project .....	99
Manually adding a component to a BOM .....	99
Excluding a component from a BOM .....	101
Deleting a component from a BOM .....	102
Removing components from a BOM .....	103
Ignoring a component in a BOM .....	104
Adjusting the component and/or component version in a BOM .....	105
Editing an origin or origin ID .....	107
Selecting a different license for a component in a BOM .....	107
Selecting the license term fulfillment status .....	108
Editing license text in the BOM .....	110
Managing subprojects .....	112
Reviewing snippet matches .....	115
Snippets in the BOM .....	115

Viewing snippet matches in the Source tab .....	116
<b>Chapter 8: Managing components .....</b>	<b>121</b>
About custom components .....	122
Managing custom components .....	123
Creating custom components .....	124
Viewing custom component information .....	124
Editing custom components .....	125
Deleting custom components .....	125
Managing custom component versions .....	126
Creating additional versions for a custom component .....	126
Viewing the projects where a version is used .....	127
Editing custom component versions .....	128
Deleting a custom component version .....	129
About Black Duck KnowledgeBase components .....	130
Understanding the component information available from the Black Duck KB .....	130
Understanding the component version information available from the Black Duck KB .....	132
Modifying KB components .....	134
Resetting a Black Duck KB component's values .....	136
About the KnowledgeBase Feedback Service .....	137
Disabling the Black Duck KnowledgeBase feedback service .....	138
Setting or modify a component's status .....	138
Changing the status of components and/or versions .....	139
<b>Chapter 9: Viewing risk in Black Duck .....</b>	<b>141</b>
Dashboard pages .....	141
Project version pages .....	144
Viewing the health of your projects .....	144
Viewing overall risk for all projects .....	147
Understanding the types of project risk .....	148
Understanding the projects table .....	149
Filtering the projects list by type and severity of risk .....	149
Viewing overall risk at the project version level .....	149
Viewing the risk associated with components used your projects .....	151
Understanding the types of component risk .....	152
Understanding the component table .....	152
Filtering the dashboard .....	153
<b>Chapter 10: About security risk .....</b>	<b>154</b>
Security risk levels .....	154
Defining the default security risk calculation .....	155
Viewing all security vulnerabilities .....	157
Viewing security vulnerabilities associated with your components .....	158
Viewing the security vulnerabilities of your projects and project versions .....	161

Viewing project version vulnerabilities .....	164
Viewing vulnerability details .....	166
Black Duck Security Advisories .....	166
Overview tab .....	168
Affected Projects tab .....	169
Technical tab .....	170
CVE References tab .....	171
CVE record .....	172
Overview tab .....	172
Affected Projects tab .....	173
References tab .....	174
Remediating security vulnerabilities .....	174
Remediating a vulnerability .....	175
Getting remediation guidance for components with security vulnerabilities - BETA .....	177
<b>Chapter 11: Managing policies .....</b>	<b>179</b>
About the policy process .....	179
Viewing policy rules .....	179
Viewing policy rule violations .....	180
Overriding violations .....	181
Removing policy overrides .....	181
Default policy rules .....	181
Creating a policy rule .....	182
Policy conditions .....	182
Creating a policy .....	186
Creating whitelist and blacklist policy rules .....	187
Whitelist examples .....	187
Blacklist example .....	188
Editing a policy rule .....	189
Copying a policy rule .....	189
Deleting a policy rule .....	190
Disabling or enabling a policy rule .....	190
Overriding policy violations .....	190
Removing policy overrides .....	192
<b>Chapter 12: Managing open source licenses .....</b>	<b>193</b>
Suggested work flow .....	194
About license families .....	195
Managing license families .....	197
About custom license families .....	198
Creating custom license families .....	199
Editing custom license families .....	200
Deleting custom license families .....	202

Viewing licenses .....	202
Viewing license text .....	205
Viewing license use .....	206
Determining license risk .....	208
Default license risk .....	209
License risk - by usage .....	209
License risk by license family .....	212
About custom licenses .....	214
Creating custom licenses .....	214
Editing a custom license .....	216
Deleting custom licenses .....	218
About license terms .....	219
Suggested work flow .....	220
License terms process .....	220
Viewing license terms .....	222
About license term fulfillment .....	224
Defining fulfillment when viewing terms for a license .....	225
Creating license terms .....	227
Managing license term categories .....	231
Associating a license term to a license .....	234
Editing a custom license term .....	239
Deleting a license term .....	240
Deprecating or removing the deprecation status of a custom license term .....	241
Removing a license term .....	243
Deactivating a KnowledgeBase term .....	248
Restoring a KnowledgeBase license term .....	254
Editing a KnowledgeBase license .....	260
Restoring the original text and family of a KnowledgeBase license .....	262
Managing attribution statements .....	263
<b>Chapter 13: Running a report .....</b>	<b>266</b>
Notices File report .....	266
Excluding a component or subproject from the Notices File report .....	269
Project Version report .....	270
Vulnerability Remediation report .....	272
Vulnerability Status report .....	272
Vulnerability Update report .....	273
Deleting reports .....	274
<b>Chapter 14: Managing Black Duck user accounts .....</b>	<b>275</b>
Creating a user account .....	275
Disabling a user .....	277
Viewing a user's groups .....	280

Viewing a user's projects .....	283
Changing your Black Duck password .....	284
Changing user account information .....	285
Changing a user's password .....	288
Understanding roles .....	289
Global roles .....	289
Project roles .....	291
Managing user roles .....	292
Viewing your roles .....	295
Black Duck user role matrix .....	295
Global Roles .....	295
Project roles .....	299
Authenticating users with LDAP .....	303
Configuring secure LDAP .....	307
Obtaining your LDAP information .....	307
Importing the server certificate .....	308
About locked out user accounts .....	310
<b>Chapter 15: Managing groups in Black Duck .....</b>	<b>311</b>
Viewing your groups .....	311
Creating groups .....	312
Managing group information .....	314
Managing group projects .....	317
Managing group roles .....	321
Adding a member to a group .....	324
Removing a member from a group .....	330
Deleting groups .....	336
<b>Chapter 16: About Custom Fields .....</b>	<b>338</b>
Viewing custom field information in the Black Duck UI .....	339
Creating a custom field .....	342
Activating or deactivating a custom field .....	345
Determining the order of custom fields shown in the UI .....	346
Editing a custom field .....	347
<b>Chapter 17: Other administrative tasks .....</b>	<b>349</b>
Viewing project and project version audit information .....	349
Viewing product registration information .....	351
Updating your product registration .....	352
Managing your code size limits .....	352
Managing user access tokens .....	354
Enabling license term fulfillment .....	356
Customizing the logo .....	358
Accessing log files .....	359

Viewing jobs .....	360
<b>Appendix A: Understanding how to search in Black Duck .....</b>	<b>366</b>
Searching for projects and components .....	367
How project/component searching works .....	367
Specifying your search terms .....	368
Filtering your search results .....	368
Project and components filters .....	368
Vulnerabilities filters .....	369
Searching for security vulnerabilities .....	369
Filtering the data shown in tables .....	369
Risk Graphs .....	370
Advanced Filters .....	370
<b>Appendix B: Working with notifications .....</b>	<b>372</b>
Viewing all notifications .....	372
Viewing more information .....	372
Deleting notifications .....	373
<b>Appendix C: About the Tools page .....</b>	<b>374</b>
Downloads .....	374
Documentation .....	374
Developer's Tools .....	375
<b>Appendix D: Integrating Protex with Black Duck .....</b>	<b>376</b>
Understanding the Protex BOM integration process .....	377
Requirements .....	378
Downloading the Protex BOM tool .....	378
Exporting a Protex BOM .....	378
Exit Statuses .....	381
Viewing multiple versions of a Protex BOM in Black Duck .....	381
Examples .....	381
Importing the Protex BOM file .....	383
Mapping or unmapping a Protex BOM .....	384

## Black Duck documentation

The documentation for Black Duck consists of online help and these documents:

Title	File	Description
Release Notes	release_notes.pdf	Contains information about the new and improved features, resolved issues, and known issues in the current and previous releases.
Installing Black Duck using Docker Compose	install_compose.pdf	Contains information about installing and upgrading Black Duck using Docker Compose.
Installing Black Duck using Docker Swarm	install_swarm.pdf	Contains information about installing and upgrading Black Duck using Docker Swarm.
Installing Black Duck using Kubernetes	install_kubernetes.pdf	Contains information about installing and upgrading Black Duck using Kubernetes.
Installing Black Duck using OpenShift	install.openshift.pdf	Contains information about installing and upgrading Black Duck using OpenShift.
Getting Started	getting_started.pdf	Provides first-time users with information on using Black Duck.
Scanning Best Practices	scanning_best_practices.pdf	Provides best practices for scanning.
Getting Started with the SDK	getting_started_sdk.pdf	Contains overview information and a sample use case.

Title	File	Description
Report Database	report_db.pdf	Contains information on using the report database.
User Guide	user_guide.pdf	Contains information on using Black Duck's UI.

Black Duck integration documentation can be found on [Confluence](#).

## Customer support

If you have any problems with the software or the documentation, please contact Synopsys Customer Support.

You can contact Synopsys Support in several ways:

- Online: <https://www.synopsys.com/software-integrity/support.html>
- Email: [software-integrity-support@synopsys.com](mailto:software-integrity-support@synopsys.com)
- Phone: See the Contact Us section at the bottom of our [support page](#) to find your local phone number.

Another convenient resource available at all times is the [online customer portal](#).

## Synopsys Software Integrity Community

The Synopsys Software Integrity Community is our primary online resource for customer support, solutions, and information. The Community allows users to quickly and easily open support cases and monitor progress, learn important product information, search a knowledgebase, and gain insights from other Software Integrity Group (SIG) customers. The many features included in the Community center around the following collaborative actions:

- Connect – Open support cases and monitor their progress, as well as, monitor issues that require Engineering or Product Management assistance
- Learn – Insights and best practices from other SIG product users to allow you to learn valuable lessons from a diverse group of industry leading companies. In addition, the Customer Hub puts all the latest product news and updates from Synopsys at your fingertips, helping you to better utilize our products and services to maximize the value of open source within your organization.
- Solve – Quickly and easily get the answers you're seeking with the access to rich content and product knowledge from SIG experts and our Knowledgebase.
- Share – Collaborate and connect with Software Integrity Group staff and other customers to crowdsource solutions and share your thoughts on product direction.

[Access the Customer Success Community](#). If you do not have an account or have trouble accessing the system, click [here](#) to get started, or send an email to [community.manager@synopsys.com](mailto:community.manager@synopsys.com).

## Training

Synopsys Software Integrity, Customer Education (SIG Edu) is a one-stop resource for all your Black Duck education needs. It provides you with 24x7 access to online training courses and how-to videos.

New videos and courses are added monthly.

At Synopsys Software Integrity, Customer Education (SIG Edu), you can:

- Learn at your own pace.
- Review courses as often as you wish.
- Take assessments to test your skills.
- Print certificates of completion to showcase your accomplishments.

Learn more at <https://community.synopsys.com/s/education>.

# Chapter 1: Getting started with Black Duck

Black Duck is a risk management tool designed to help you manage the logistics of using open source software in your organization.

Using Black Duck, you can:

- Scan your code and identify open source software that exists in your code base.
- View the generated Bill of Materials (BOM) for your software projects.
- View vulnerabilities that have been identified in open source components.
- Assess your security, license, and operational risk.

Protex users can use Black Duck to view and manage security vulnerabilities in their existing BOMs.

## Logging in to Black Duck

**Note:** You must have a username and password to access Black Duck. Contact your system administrator if you do not have a username. If Black Duck is configured to use LDAP, you may be able to log in to Black Duck using those credentials.

### To log in to Black Duck

1. Using a browser, navigate to the Black Duck URL supplied by your system administrator. Typically, the URL is in the format <https://<server hostname>>.
2. Enter the username and password provided by your Black Duck administrator.

**Note:** Your password is case sensitive.

3. Click **Login**.

When you log in, Black Duck displays your dashboard page.

- For new installations of Black Duck, when you first log in after installing Black Duck, an empty Project Dashboard page appears.

For information to appear in Black Duck, you need to:

- Scan your code and map it to a project.  
and/or

- Import and map a Protex BOM.

Once these tasks are complete, you can [view the discovered components in the BOM](#) and manage your [security vulnerabilities](#).

- For existing installations of Black Duck, if this is not the first time you are logging in to Black Duck, the dashboard page that appears depends on the last main dashboard (**Projects**, **Components**, **Security**, or **Summary**) you viewed prior to previously logging out. This tab will also appear when you click the Black Duck logo in the title bar.

**Note:** You will be locked out of your account for 10 minutes if you fail to enter the correct password after 10 attempts. After the 10th failed attempt, a message appears on the login page notifying you that your account is locked. Note that there is no defined time period in which the 10 failed attempts must occur – any failed attempt will be included in the count of failed password attempts. The count resets to 0 after you successfully log in to Black Duck.

The permissions assigned to your Black Duck user account by your system administrator determine which:

- navigation elements are visible to you on each page
- projects and project data you can view on each page
- actions you can perform in Black Duck

## Scanning your code and mapping scans to projects

Use these methods to scan your code:

- [Synopsys Detect Desktop](#) which you can download from Black Duck's Tools page
- [Plugins](#).
- Synopsys Detect. Use Synopsys Detect for package management level analysis combined with signature scanning

After running a scan:

1. [Browse the available component scan results](#) in Black Duck to view the results of a component scan and the status of a scan that is in progress.
2. [View your component scan results](#) in Black Duck to view the raw component scan results.

## Mapping scans to projects

After scanning your code, [use Black Duck's UI to map your component scan](#) if you did not map the scan while scanning. Mapping connects your scan results to a Black Duck project.

A *project* is the base unit in Black Duck. A project can be both a stand-alone development project and part of another project. For example, Apache Tomcat is a project in its own right but it may also be part of other, larger projects. Projects can have [multiple versions](#).

## Administrative tasks

Other tasks for administrators include:

- [Managing users](#). Administrators need to create and manage users in Black Duck and assign [roles](#).
- [Managing groups](#). In addition to managing role assignments and project team membership at the individual user account level, administrators can manage these for multiple Black Duck users at the same time by creating a user group.

## Importing a Protex BOM

Use the Protex BOM tool to import a Protex BOM. Click [here](#) for an overview of the process and [here](#) for more information on using the Protex BOM tool.

## Chapter 2: About Black Duck - Binary Analysis

Black Duck - Binary Analysis (BDBA) identifies the open source security, compliance, and quality risks in the software libraries, executables, and vendor-supplied binaries in use within your codebase. BDBA supports expanded file type support including various firmware formats, filesystems/disk images, installation formats, and various compression and archive formats. With Black Duck - Binary Analysis, you can:

- Analyze virtually any compiled software, firmware, mobile applications, or multiple installer formats, without needing to access the source.
- Identify embedded open source usage and risks within binary executables and libraries.
- Manage code decay and improve software quality within binary dependencies.
- Monitor new vulnerabilities in previously scanned binaries.

After installing Black Duck - Binary Analysis:

1. Use Synopsys Detect to scan your software or firmware.
2. View the results of your scan in a comprehensive [project version BOM](#).

For you to easily identify these files, the BOM displays the match type as Binary.

3. Use the BOM to identify known vulnerabilities and licensing obligations within software components.

Refer to the installation guides for more information on installing Black Duck with Black Duck - Binary Analysis.

# Chapter 3: Understanding the scanning process

The process for scanning components is:

1. [Download and install Synopsys Detect Desktop](#) or the [command line](#) (CLI) version of the Signature Scanner.
2. [Upload the file using the Black Duck UI](#) if you did not use the command line or Synopsys Detect (Desktop) to upload the file.
3. [Browse the available component scan results](#) in Black Duck.
4. [View your component scan results](#) in Black Duck.
5. [View an audit log for BOM files](#).
6. [Map your component scan](#) to a project version.
7. [View the discovered components in the BOM](#) for the project version.

## Understanding Component Scanning

Black Duck Component Scanning is scanning functionality that provides an automated way to determine the set of open source software components that make up a software project. Component Scanning helps organizations manage their use of open source by identifying and cataloging components in order to provide additional metadata such as license, vulnerability, and project health for those components. Component Scanning lets users use the scanner to scan software artifacts on their local computers, which automatically generates a BOM that can be linked to a specific project in Black Duck.

Black Duck Component Scanning can extract the following archive types:

- AR
- ARJ
- CPIO
- DUMP
- TAR
- RPM
- ZIP
- 7z

Archives may optionally be compressed using any of the following compression algorithms:

- Bzip2
- Gzip
- Pack200
- XZ
- LZMA
- Snappy
- Z (compress)
- DEFLATE

During the component scan, Component Scanning examines similarities and differences between large clusters of files and can find:

- Exact matches to unmodified archives and directories of open source.
- Fuzzy matches to modified archives and directories of open source.

It scans an arbitrary file system directory or archive and matches to known components in the Black Duck KnowledgeBase (KB).

The core concept behind component scanning and discovery is the ability to compare the signatures of artifacts in the repository with the signatures of all OSS components in the Black Duck KB and quickly recognize a match. The recognition can be fuzzy—it does not need to be an exact match to be recognized. When there are multiple possible matches, Component Scanning determines the preferred match.

Component Scanning can discover and identify code that is:

- **Unmodified:** A collection of files that have not changed since they were released by the open source project.
- **Renamed:** A collection of files that have been renamed without other modification.
- **Compressed and/or recompiled:** Jars that have been compressed and/or recompiled after they were released by the open source project.
- **Modified or rebundled:** For example, with a jar:
  - Class files from more than one component jar combined into a single jar
  - Class files added to or deleted from a component jar
  - Nested component jars with jar files added or deleted

Component Scanning classifies each match based on how it was made:

- **Exact:** Component Scanning identified the set of files as an exact match to a component in the Black Duck KB.
- **File Dependency.** Component Scanning identified a match via a file dependency.
- **Files Modified:** Component Scanning identified a fuzzy match to a component in the Black Duck KB, where some of the files were modified. Sometimes this is a match to a previous or subsequent version of the component, which may have been missing from the Black Duck KB at the time that the match was made.
- **Files Added/Deleted & Modified:** The component scan identified a fuzzy match to a component in the Black Duck KB This can happen when:

- An OSS component is matched, but some of the files associated with the component have been added, deleted, or modified. This can be a match to a previous or a subsequent version of the component, which may have been missing from the Black Duck KB at the time of the match.
- A component is only matched against a common directory structure (structure-only), but because a significant number of components share this structure, the Black Duck KB may propose a match that has very little similarity to the scanned component.
- A component is only matched against a common directory structure, but because proprietary or third-party code can share a common directory structure with components, the Black Duck KB may propose a match that has very little similarity to the scanned code.

The Black Duck KB contains a 'blacklist' of very common, non-unique, directory tree structures. For example, many components include a directory that contains three subdirectories: 'css', 'img', and 'js'. This structure has been blacklisted, so that the Black Duck KB will not propose irrelevant matches.

## Supported languages

Component Scanning currently supports the ability to identify components containing Java binaries, JavaScript, or C (C, C++, C#) binaries and source code and Ruby, Python, Scala, Objective C, Swift; PHP, R, Go, Erlang, and Perl.

For example, scanning can identify open source components based on directories that contain C, C++, or C# source files (i.e., files with the following extensions: .c, .c++, .cc, .cpp, .cs, .cxx, .h, .hh, .hpp, .hxx, .h++, .idl, .rc). That is to say, collections of these files (or partial collections) that are re-used from their original OSS packages with similar directory structures will be matched. As noted above, some files in these collections may be modified, added or removed, and matching will still occur. Individual source files appearing "out of context", however, will probably not be recognized. In the case of C, C++ and C# binary files, the scanner can do identification based on a single .dll, .exe, .so or .obj file, in addition to recognizing a directory containing multiple binary files.

## ISO files

The Signature Scanner cannot scan an ISO file: you must first mount the file to your local file system and then scan the file system.

## Supported package managers

Component Scanning currently supports scanning npm and RubyGems package managers.

## Using Synopsys Detect (Desktop)

Synopsys Detect (Desktop) provides a new interface to make it easier to scan code.

With Synopsys Detect (Desktop), you can:

- [Scan](#) source directories, binaries and executables, and docker images and distributions.
- [Create a scan file](#) to be uploaded at a later time.
- [Manage scan files](#).
- [Upload scan files](#) directly to Black Duck.
- [View uploaded scans](#).

To use Synopsys Detect (Desktop):

1. Download and install Synopsys Detect (Desktop).
2. Configure Synopsys Detect (Desktop) with your Black Duck server settings and complete the installation process.
3. Use Synopsys Detect (Desktop) to scan and/or upload your files.

**Note:** An error message appears if you exceed the scan size limit, which is 5 GB (6 GB for Black Duck - Binary Analysis). Contact Customer Support if you receive this message.

## Downloading and installing Synopsys Detect (Desktop)

1. Log in to Black Duck.
2. Navigate to the drop-down menu under your username and select **Tools**.
3. Select the operating system you wish to use in the **Downloads Synopsys Detect (Desktop)** section to download the executable from Google Cloud Storage.
4. Run the executable to install Synopsys Detect (Desktop).

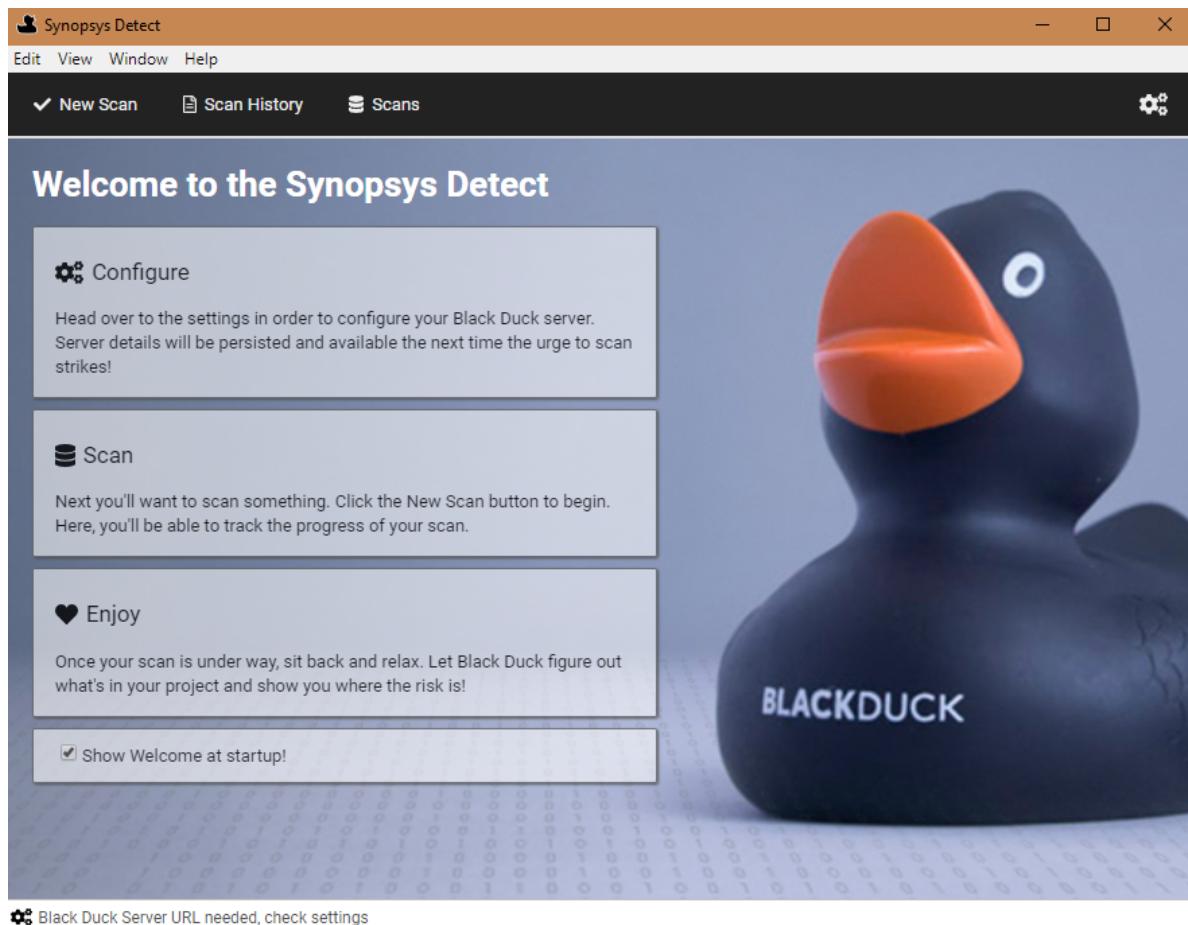
If you are upgrading from a previous version of Synopsys Detect (Desktop), an option appears to migrate data from the previous version.

**Note:** As the application installs into a directory related to its name, Synopsys Detect (Desktop) will not uninstall previous versions of Black Duck Detect Desktop. It also will not uninstall versions of Synopsys Detect (Desktop) that were installed in a non-default directory. You must manually uninstall all previous versions of Black Duck Detect Desktop, versions of Synopsys Detect (Desktop) installed in the non-default directory, and fix or delete any shortcuts.

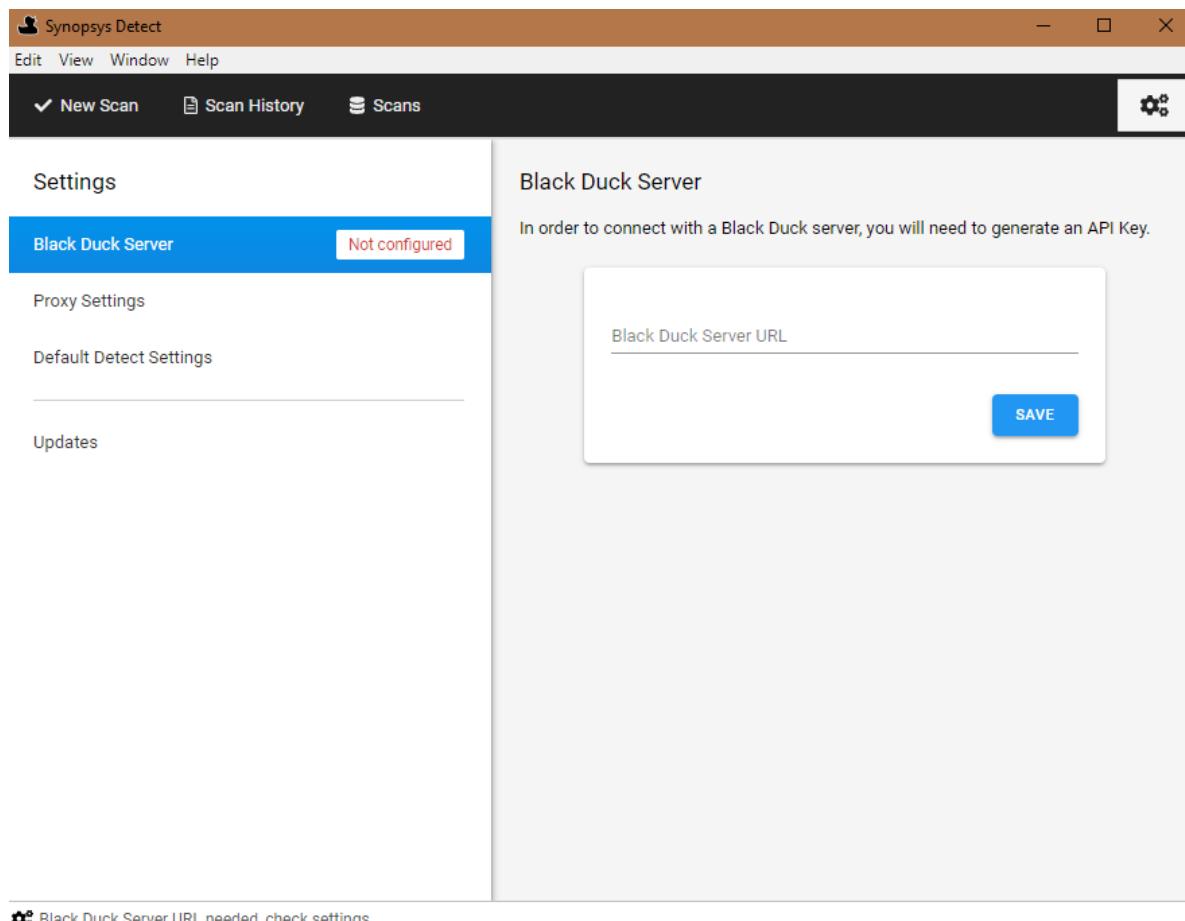
## Configuring Synopsys Detect (Desktop)

After installing Synopsys Detect (Desktop), continue the installation process by configuring your Black Duck settings.

1. After installing or upgrading to Synopsys Detect (Desktop), the Welcome page appears.



2. Select **Configure** to display the Settings page.



⚙️ Black Duck Server URL needed, check settings



You can also click , located in the upper right corner, to display this page.

3. As described below, select one of the following tabs and complete the installation and configuration process:
  - Black Duck Server
  - Proxy Settings
  - Default Detect Settings
  - Updates

### Black Duck server settings

1. Specify the Black Duck Server URL. Enter the URL to the Black Duck server as you would type it in the browser, for example <https://servername:8443/>  
If required, enter context information, for example, if the X-Forwarded-Prefix header is being specified in a proxy server/load balancer configuration.
2. Click **Save**. Synopsys Detect (Desktop) connects to the Black Duck server and displays the version of Black Duck you are connected to.

3. Generate or enter an API key (user access token). This information appears after you enter the Black Duck Server URL.

- To generate a new API key:
  - a. Select **Generate New API Key**.
  - b. Enter a key name, your username, and password.
  - c. Click **Generate**.
- To enter an API key:
  - a. Select **Enter API Key**.
  - b. Enter the API key in the field.
  - c. Click **Save**.

## Proxy settings

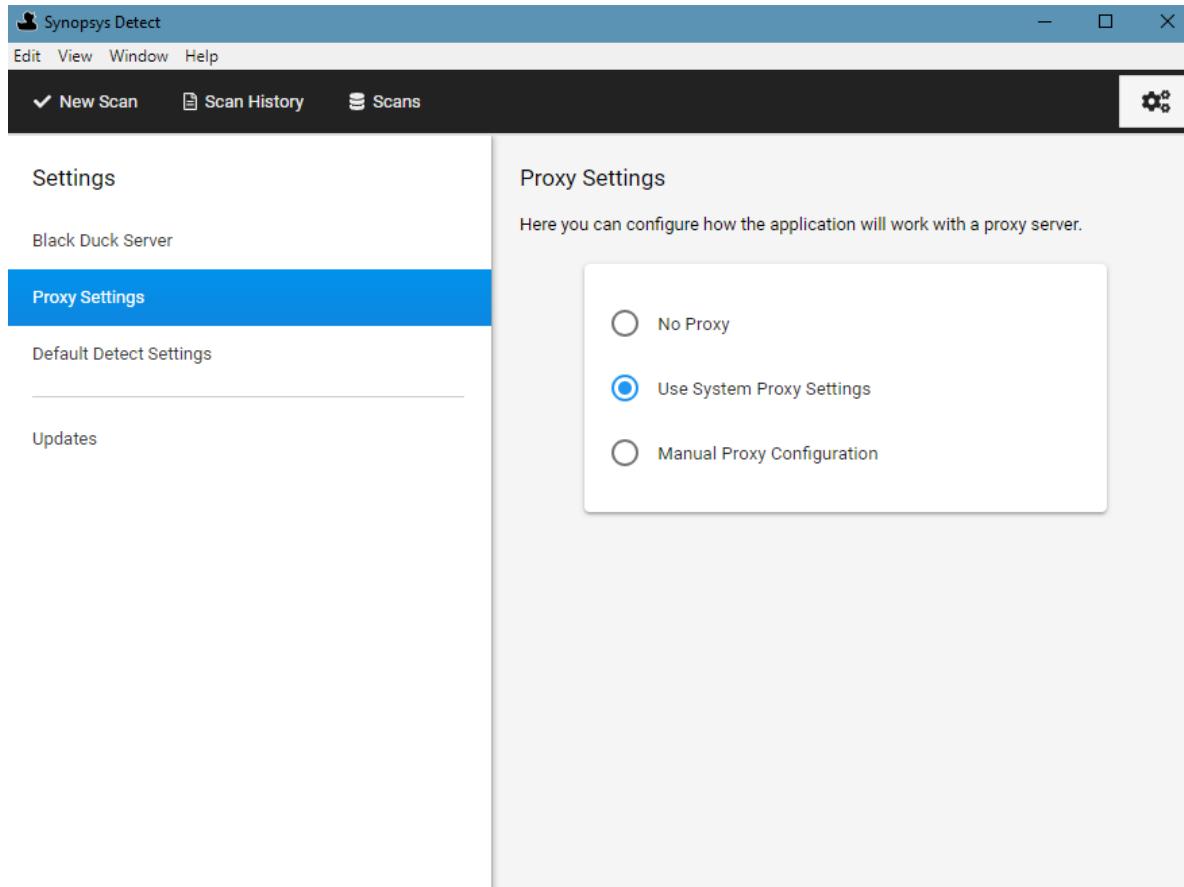
Accessing Synopsys Detect (Desktop) through a proxy is supported. Synopsys Detect (Desktop) automatically uses your local system proxy setup.

If you are required to manually enter your proxy settings or you do not require a proxy, you can modify these default settings.

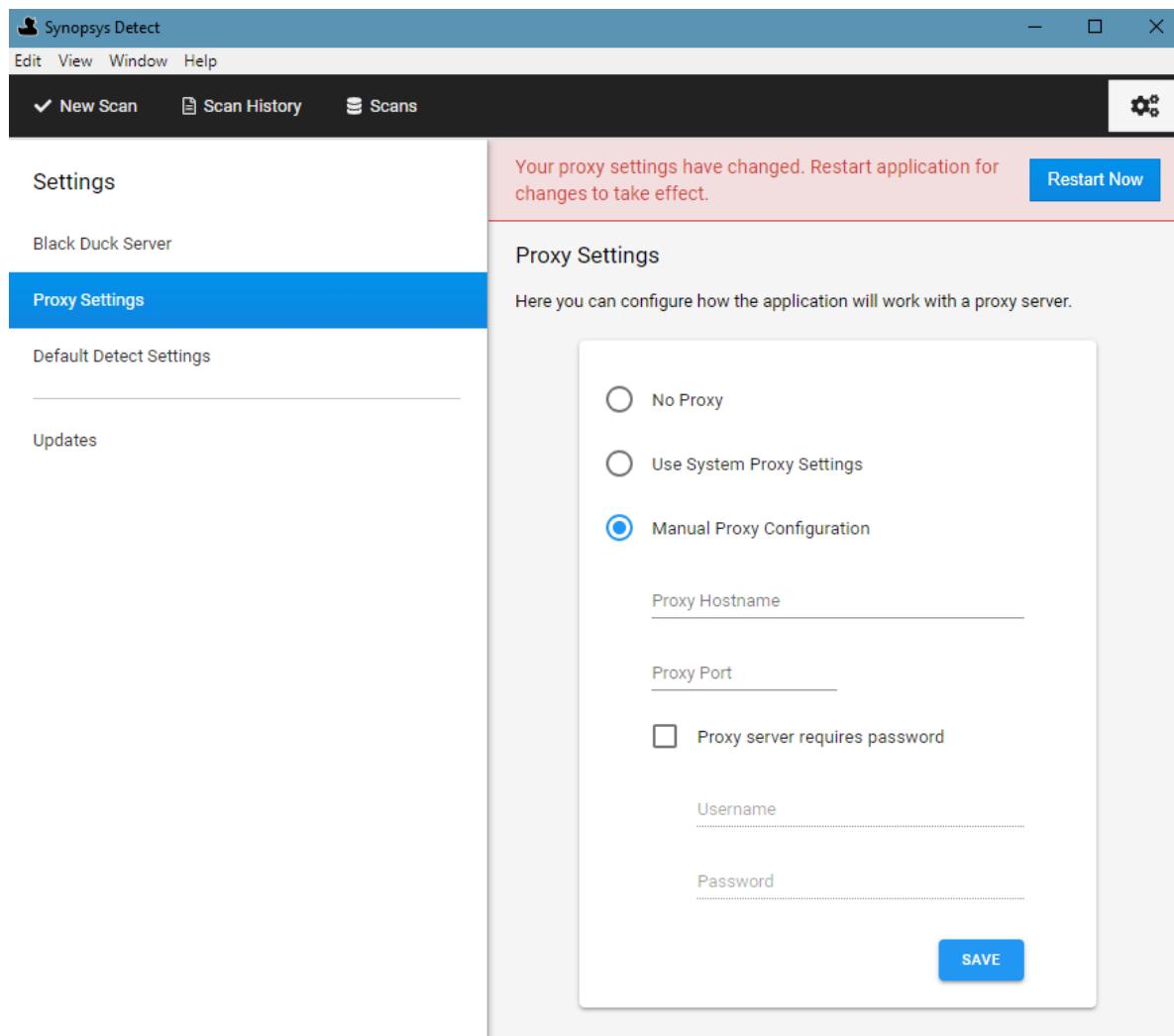
⚙ To modify the default proxy settings



1. Click **[gear icon]** to display the Settings page and select the **Proxy Settings** tab.



2. Select either **No Proxy** or **Manual Proxy Configuration**.
3. If you select a manual proxy configuration:



- a. Enter the following information:
  - Your proxy host name.
  - Port number.
  - Whether authentication is required.
  - Your username and password.

If a proxy is enabled and authentication is required, you may have to re-enter your username and password.

- b. Click **Save**.

4. Restart the application.

## Configuring Synopsys Detect settings

Optionally, select **Default Detect Settings** and if necessary, define any Synopsys Detect settings, clear any build tools you do not want to use, or manually configure the path to the build tools.

## Checking for updates

You can check to see if there are updates to the Synopsys Detect (Desktop) by selecting the **Updates** tab. The page lists the last time you checked for updates. Click **Check for updates** to view if there are newer versions available. This option is only available for Windows and MacOS systems.

## Certificates

When connecting to Black Duck: if you connect to a Black Duck instance with an insecure SSL certificate, you are prompted to view and trust the certificate. Select the **Always trust <Black Duck instance sever name> to trust** option.

**Note:** On the Mac OS, even though you have accepted the certificate, your key store may display more options than were originally presented. For the SSL certificate, you must select the *Always trust* option. This prevents future prompts asking you about trusting certificates.

## Scanning options

The Synopsys Detect (Desktop) makes it easier to scan:

- Source directories
- Binaries or executables
- Docker images or distributions

By default, all scans are uploaded to the Black Duck server and mapped to a project version. However, you can create a scan file as described [here](#), to output the scan to a file which you can later upload to Black Duck.

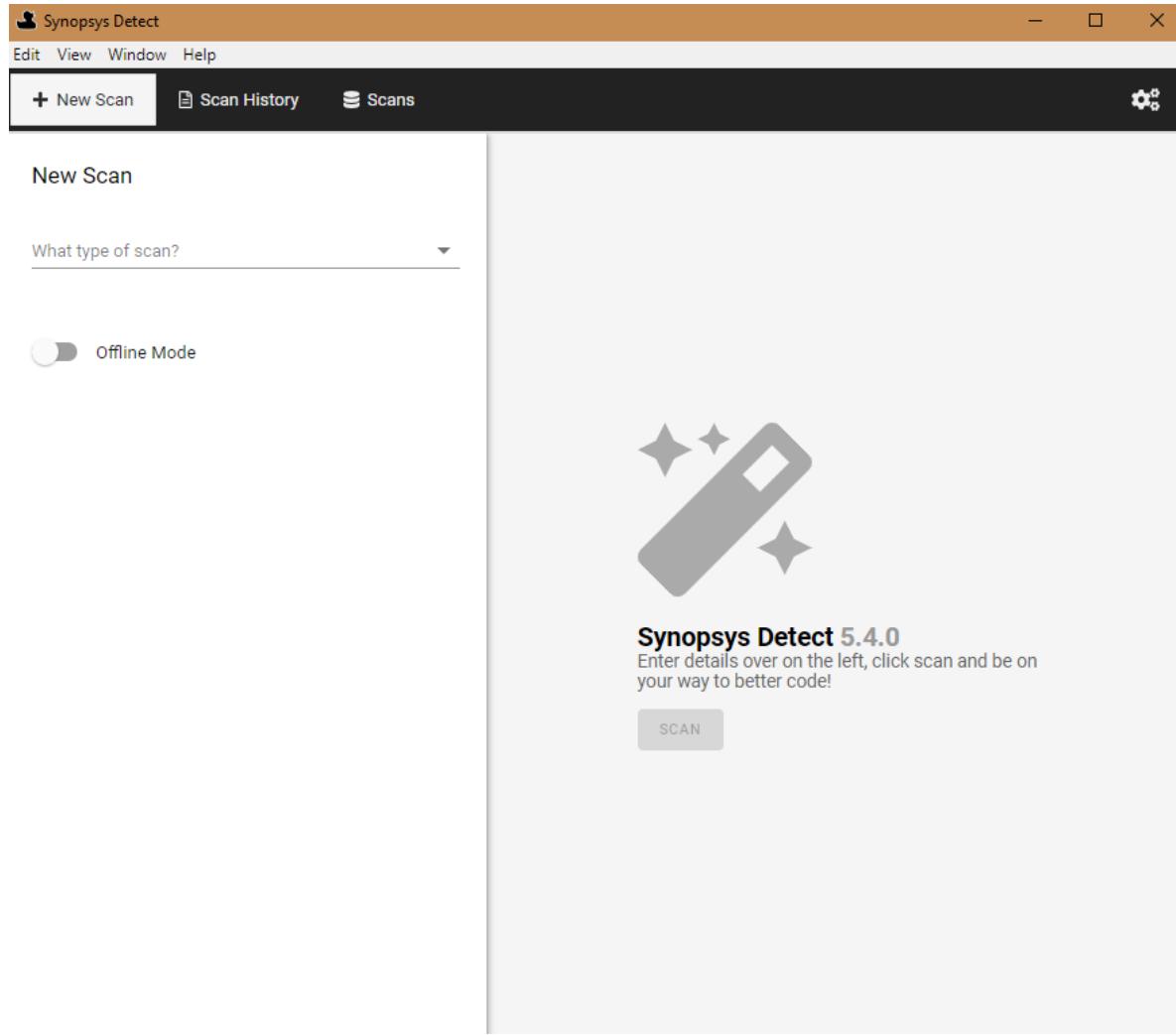
To specify project and/or version names:

1. Click **ADD** located next to **Project Settings**.
2. Select **Project Name** and/or **Version Name**. The fields appear in the UI.
3. Specify the values for the field(s).

## Scanning Source Directory

### ⚙️ To scan a source directory

1. Click **New Scan**.



2. From the **What type of scan?** list, select **Source Directory**,
3. Click to select the directory you would like to scan.
4. Optionally, modify or configure any project or scan settings by clicking **ADD** and selecting the setting.

If you have purchased a snippet scanning license and want to enable snippet scanning, select **Snippet Scanning** from the **Settings** options and enable it.

5. Click **Scan**.

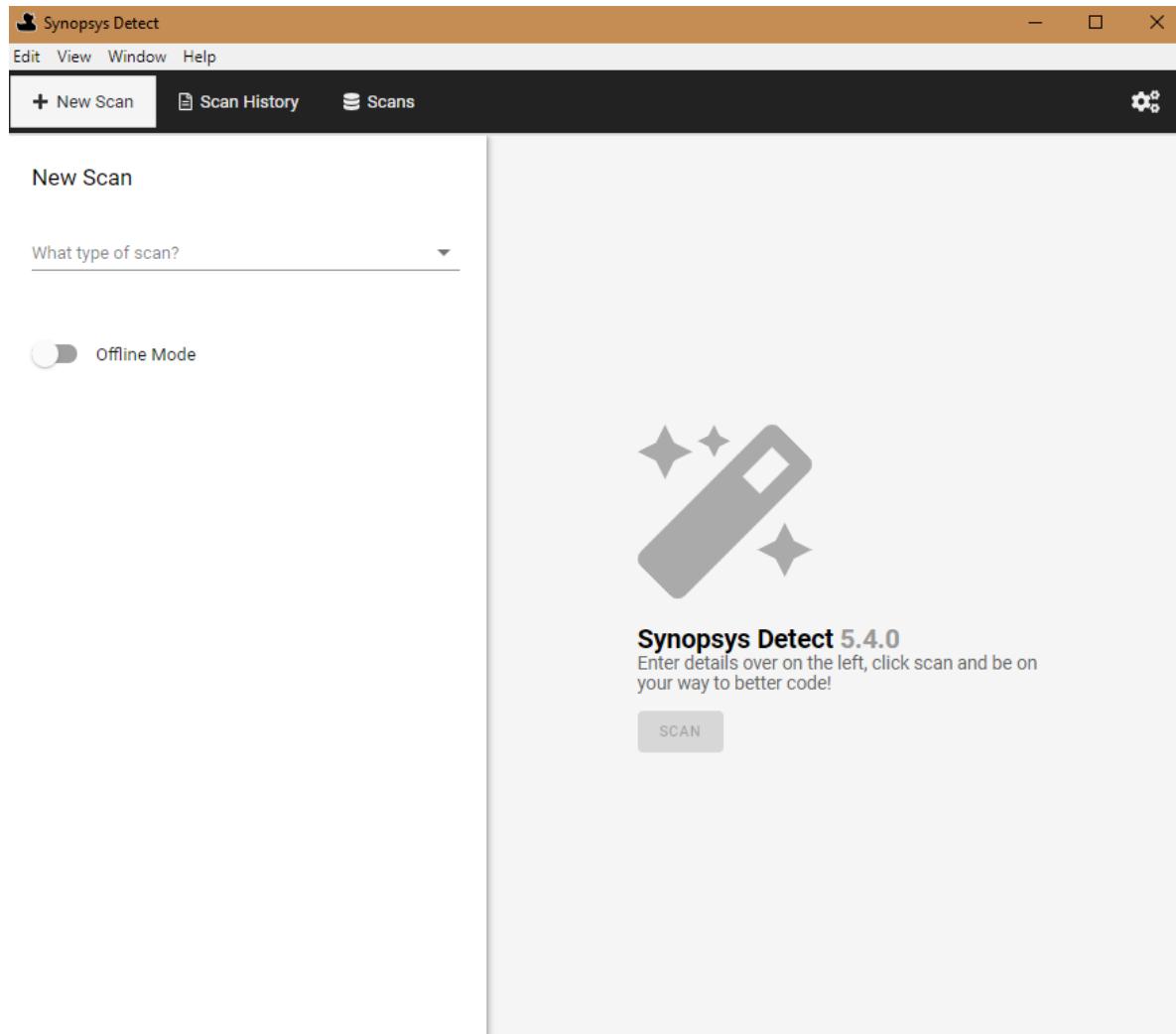
The status of the scan appears along with an option to cancel the scan.

6. When the scan is complete, select the **Scan History** tab to view information on the completed scan. From this tab, you can [manage your scan](#). You can also view the uploaded scan using the **Scans** tab.

## Scanning binary/executable

### To scan a single binary or executable

1. Click **New Scan**.



2. From the **What type of scan?** list, select **Binary/Executable**,
3. Click  to select the binary or executable you would like to scan.
4. Optionally, modify or configure any project settings by clicking **ADD** and selecting the setting.
5. Click **Scan**.

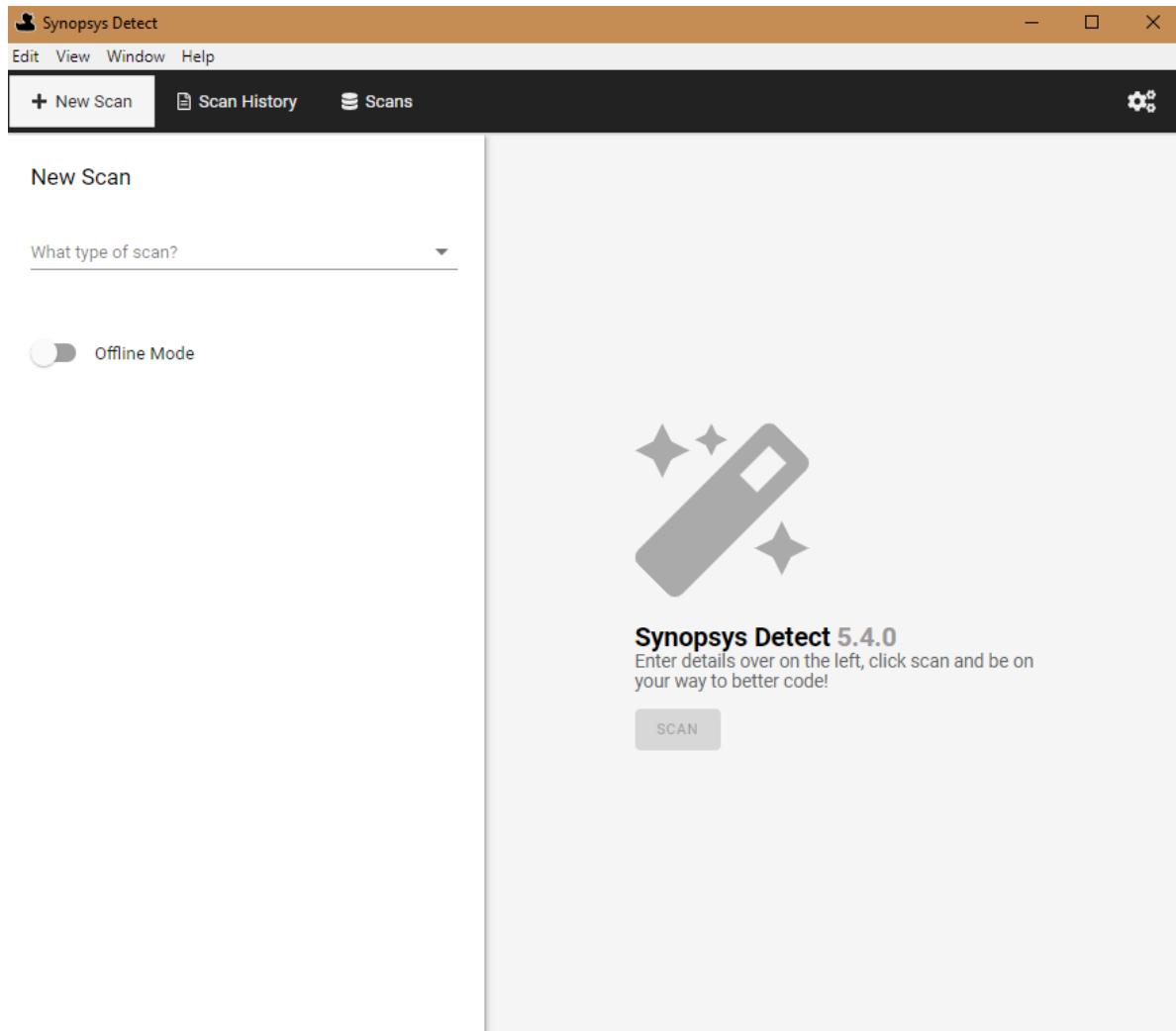
The status of the scan appears along with an option to cancel the scan.

6. When the scan is complete, select the **Scan History** tab to view information on the completed scan. From this tab, you can [manage your scan](#). You can also view the uploaded scan using the **Scans** tab.

## Scanning a Docker image or distribution

### To scan a Docker image or distribution (.tar file)

1. Click **New Scan**.



2. From the **What type of scan?** list, select **Docker**,
3. Do one of the following:
  - Enter the Docker image name.
  - Select **Choose Docker File (.tar)** and click  to select the directory you would like to scan.
4. Optionally, modify or configure any project settings by clicking **ADD** and selecting the setting.

5. Click **Scan**.

The status of the scan appears along with an option to cancel the scan.

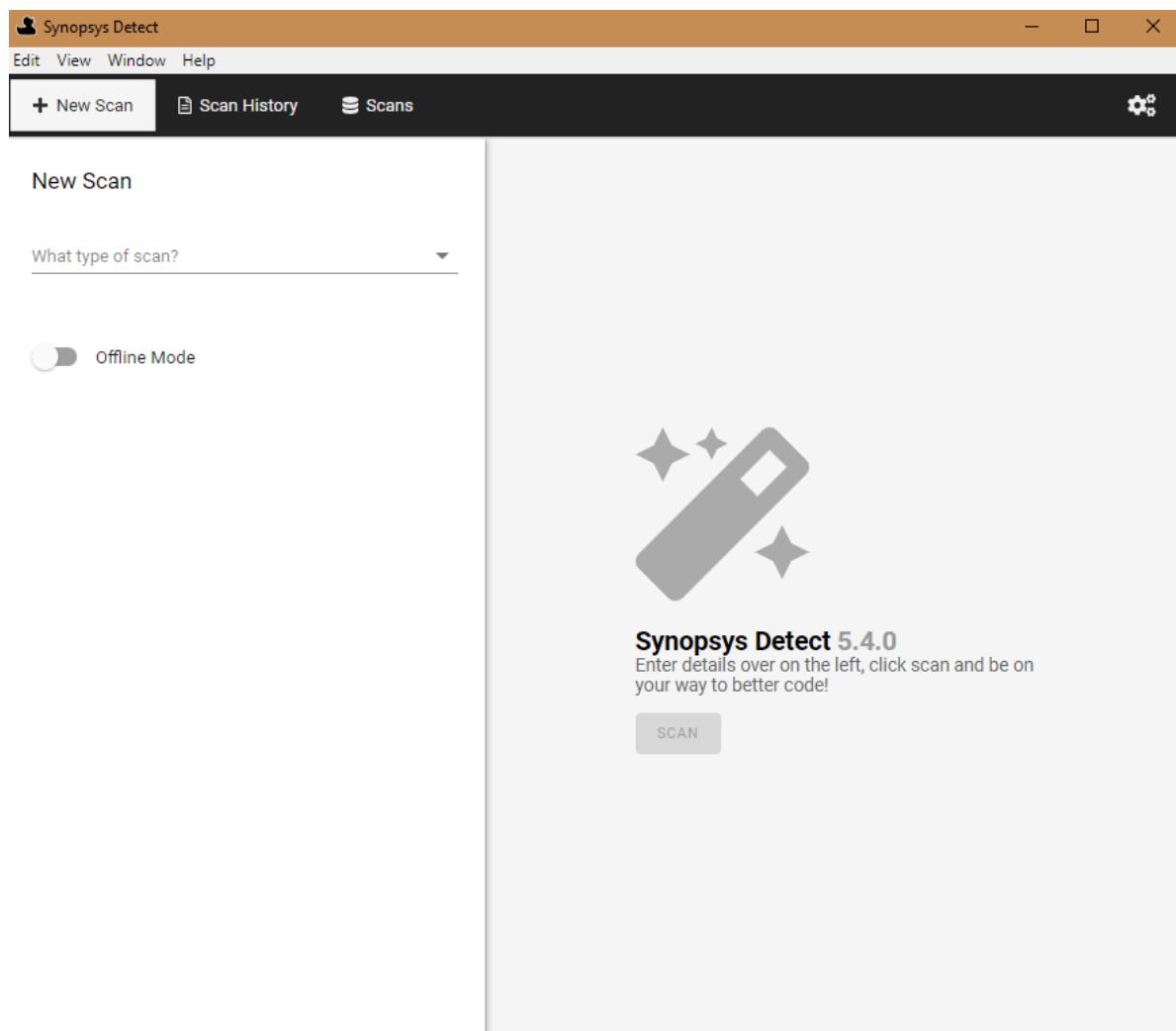
6. When the scan is complete, select the **Scan History** tab to view information on the completed scan. From this tab, you can [manage your scan](#). You can also view the uploaded scan using the **Scans** tab.

## Creating a scan file

**Note:** Snippet scanning cannot be completed offline as it requires communication with the Black Duck server.

 To create a scan file:

1. Click **New Scan**.



2. Select the type of scan (**Source Directory**, **Binary/Executable**, or **Docker**).

3. Optionally, modify or configure any project or, for source directory scanning, scan settings by clicking **ADD**

and selecting the setting.

4. Select **Offline Mode**.

5. Click **Scan**.

The status of the scan appears along with an option to cancel the scan.

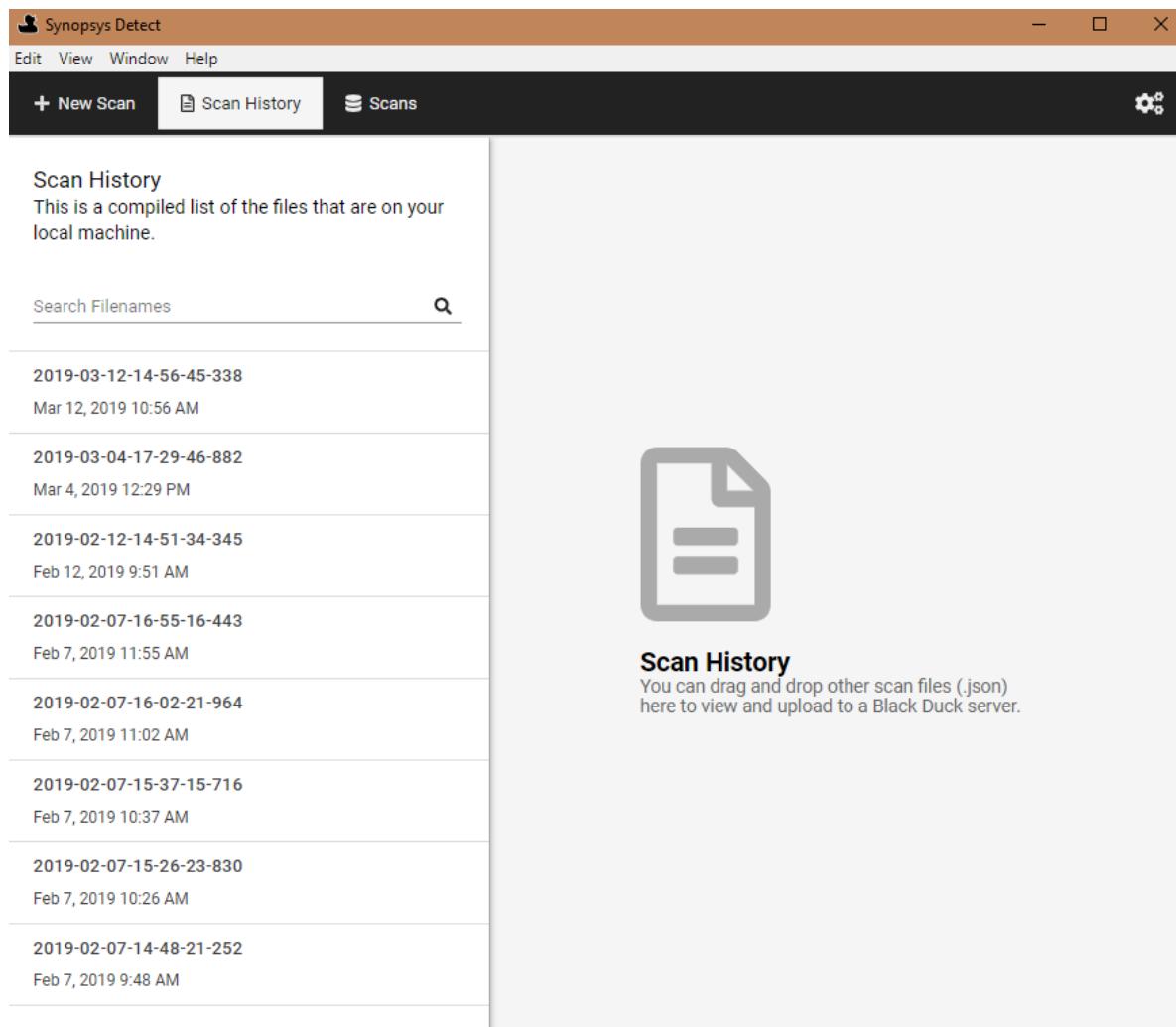
6. When the scan is complete, select the **Scan History** tab to view information on the completed scan.

## Managing scans

Use the **Scan History** tab to manage your scans.

1. Click **Scan History**.

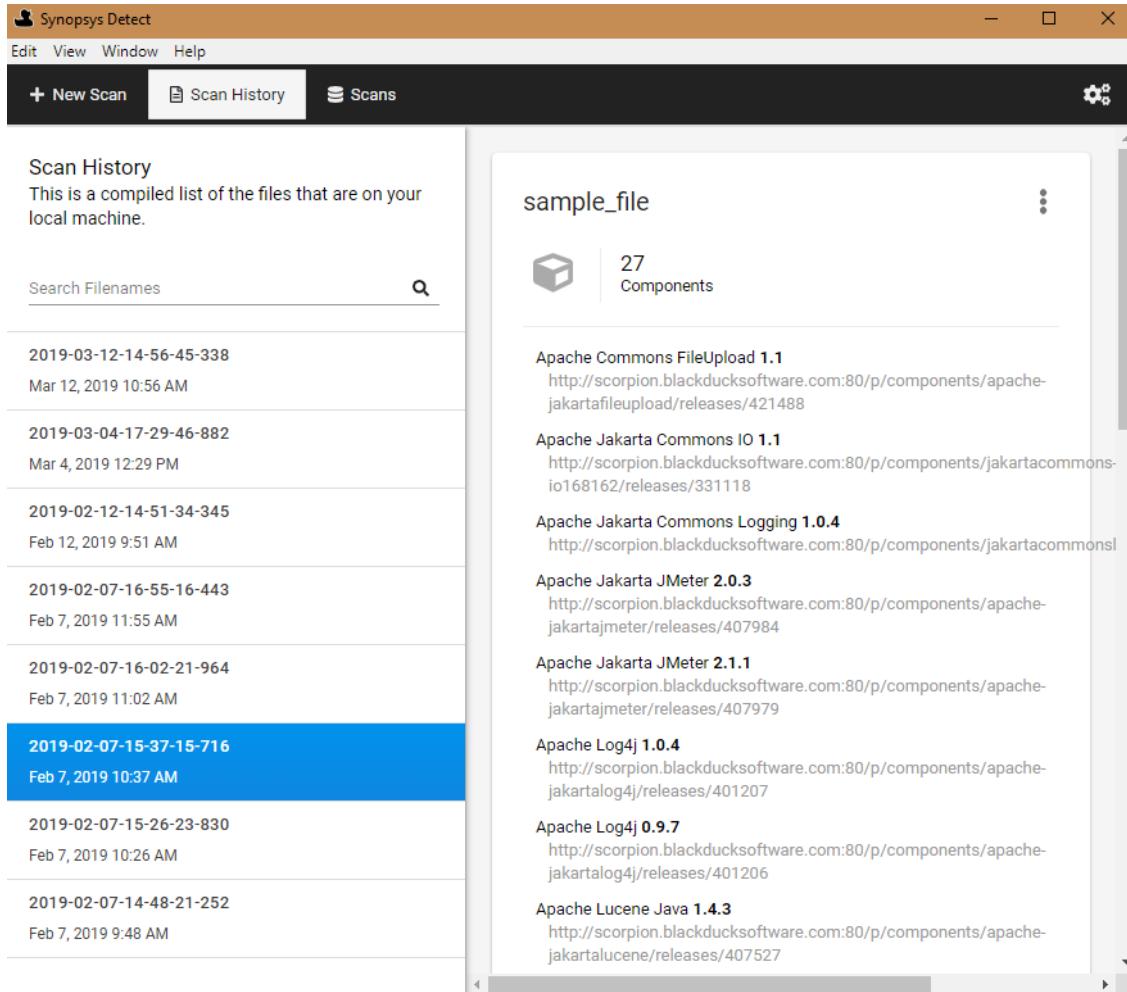
A list of scans on your local system appears in the left column of the tab.



Drag and drop scans from your local machine to this tab to manage them.

From this tab, select a scan and:

- View information on the contents of the scan:

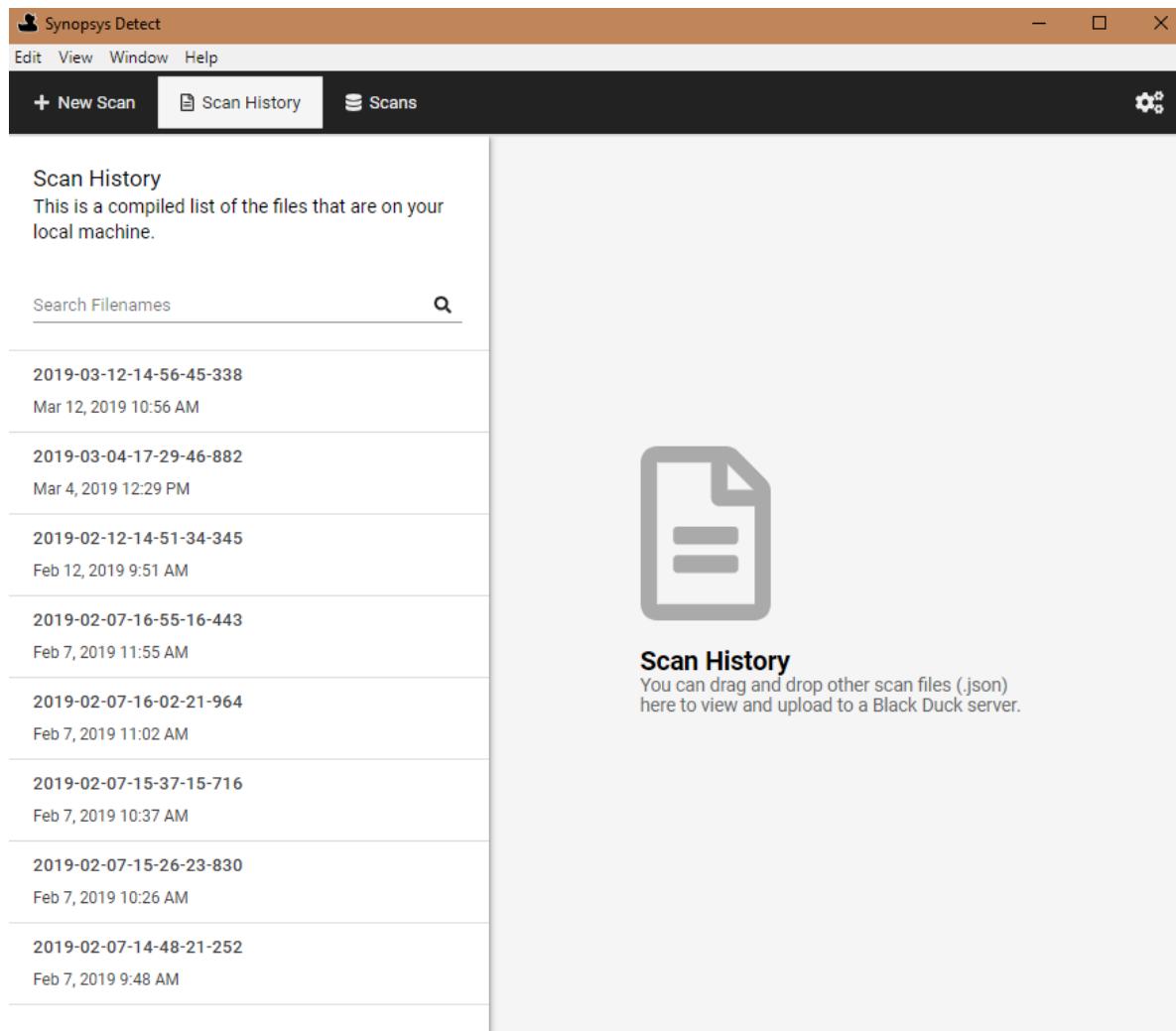


- View the location of the file on your system by clicking and selecting **Show File**.
- Upload the file, as described in the next section.
- Delete the scan by hovering over the scan name in the left column and clicking **Delete**. Click **Yes** to confirm.

## Uploading scan files to Black Duck

You can use Synopsys Detect (Desktop) to upload scan files to Black Duck.

## 1. Click **Scan History**.



2. If the file is on your local system, you can drag and drop the scan file from your local machine to the **Scan History** tab.
3. Select the file to upload and click  in the upper right corner to display the file options.
4. Click **Upload Scan File to Black Duck**. The Upload Progress window appears showing you the status of the upload. Close the window when the process is complete.

You can confirm that the scan has been uploaded by clicking **Scans** and viewing the uploaded file.

## Viewing uploaded scans

You can view the scans that have been uploaded to Black Duck's UI by clicking **Scans**:

The screenshot shows the Synopsys Detect (Desktop) interface. At the top, there's a menu bar with options like Edit, View, Window, Help, New Scan, Scan History, and Scans. A search bar labeled "Search Scans" is also present. On the left, a section titled "Scans" describes the list of scans on the Black Duck Server. Below it is a "By Status" section with a large green circular progress bar containing the number "27" and the word "Scans". To the right, a list of completed scans is shown in a grid format:

Scan Details	Date
blackduck-alert/Black Ducky Alert/6.6.6 scan Black Ducky Alert 6.6.6	May 3, 2019
blackduck-alert/blackduck-alert/4.0.0 npm/bom Black Ducky Alert 6.6.6	May 3, 2019
blackduck-alert/alert-common/blackduck-alert/alert-common/4.2.0-SNAPSHOT gradle/bom Black Ducky Alert 6.6.6	May 3, 2019
blackduck-alert/alert-database/blackduck-alert/alert-database/4.2.0-SNAPSHOT gradle/bom Black Ducky Alert 6.6.6	May 3, 2019
blackduck-alert/com.blackducksoftware.integration/blackduck-alert/4.2.0-SNAPSHOT gradle/bom Black Ducky Alert 6.6.6	May 3, 2019
hub-alert/blackduck-alert/4.2.0-SNAPSHOT scan blackduck-alert 4.2.0-SNAPSHOT	May 3, 2019
hub-alert/blackduck-alert/4.0.0 npm/bom blackduck-alert 4.2.0-SNAPSHOT	May 3, 2019

This tab displays the following information:

- The left side of the tab shows uploaded scans by status (in progress, completed, or error).  
Use the search field to find a scan or limit the scans shown.
- The right side of the page lists the scans and shows the following information for each scan:
  - Name
  - Project and project version scan is mapped to or indicates that the scan is not mapped to a project.
  - Date the scan was uploaded to Black Duck.

Select a scan to open the *Scan Name* page in Black Duck for the selected scan.

**Note:** The number of scanned bytes displayed in Synopsys Detect (Desktop) may differ from the number of scanned bytes shown in Black Duck. This is because of how Black Duck calculates and counts the number of bytes used. This is normal and is expected to occur in some scans.

## Using the Signature Scanner

Synopsys recommends using Synopsys Detect or Synopsys Detect (Desktop) for scanning. However, you may want to use the Signature Scanner CLI to scan your code.

### Signature Scanner client requirements

A Windows 7 or later, Mac OS X 10.9 or later, or Linux 64-bit system is required to run the Signature Scanner. Client systems must have a minimum of 6 GB of RAM.

## Downloading and installing the Signature Scanner CLI

### Downloading the Signature Scanner CLI

The Signature Scanner CLI is packaged as a .zip file. Download it from the Black Duck application.

Before downloading the Signature Scanner CLI, be sure that:

- Your Black Duck license is enabled for Component Scanning.
- Your Black Duck account has the Global or Project Code Scanner role.

**Note:** Java Runtime Environment (JRE) is included with the download of the the Signature Scanner. However, there may be situations that require you to use your version of JRE, for example you have self-signed certificates stored in a preferred version of Java or your company policy only allows you to run a specific version of JAVA or JRE. In these instances, you need to set the BDS\_JAVA\_HOME environment variable prior to running the Signature Scanner. See the Black Duck online help for more information.

#### ⚙️ To download the Signature Scanner CLI from the Black Duck user interface

1. Log in to Black Duck.
2. Navigate to the drop-down menu under your username, and select **Tools**.
3. On the Tools page under **Legacy Downloads**, select **Toggle All** to view and select the download link for the Linux, Mac OS X, or Windows CLI of the Signature Scanner,

### Installing the Signature Scanner CLI

Install the scanner on the computer that contains the archives to be scanned. You cannot scan archives on a remote server.

#### ⚙️ To install the Signature Scanner CLI

1. Unzip the Signature Scanner CLI.

The following is the directory structure for Windows:

Name	Type
bin	File folder
jre	File folder
lib	File folder

**Note:** For Mac OS X or Linux users, refer to the [partnerships documentation website](#) for the Google Cloud Platform for information on the Google Cloud script (`scan.gcloud.sh`).

## Defining your version of JRE for the Signature Scanner

The Java Runtime Environment (JRE) is included with the download of the Signature Scanner. As a result, you do not need to configure the JRE or the `JAVA_HOME` environment variable.

However, there may be situations that require you to use your version of JRE, for example you have self-signed certificates stored in a preferred version of Java or your company policy prohibits using the version of the JRE included with Signature Scanner. In these instances, you can set the `BDS_JAVA_HOME` environment variable to define the installed version of JRE Signature Scanner should use. The Signature Scanner will then use this version when scanning components.

**Note:** If you do not configure `BDS_JAVA_HOME`, the Signature Scanner uses the version of JRE packaged with the download of the Scanner.

### To configure the `BDS_JAVA_HOME` environment variable on Windows

1. Access the System Properties dialog box. For example, in Windows 7, click **Start > Control Panel > System > Advanced System Settings**.
2. Select the **Advanced** tab and click **Environment Variables**.
3. In the Environment Variables dialog box, under **System Variables**, click **New**.
4. Enter the following information:  
**Variable name:** `BDS_JAVA_HOME`  
**Variable value:** <path to JRE>
5. Click **OK**.

### To configure the `BDS_JAVA_HOME` environment variable on Linux or Mac OS X

1. Start a terminal session.
2. At the command line, type  

```
export BDS_JAVA_HOME=<path to JRE>
```
3. Close the terminal session.

## Running a component scan using the Signature Scanner command line

You run a component scan to identify the components contained in an archive or a directory of files.

**Note:** An error message appears if you exceed the scan size limit, which is 5 GB. Contact Customer Support if you receive this message.

The usage is:

```
scan.cli.bat [parameter1]...[parameterN]...<scan_path>
```

Parameter	Description
<b>-?, --help</b>	Shows help for this tool.
<scan_path>	Path to the file directory location or archive that you want to scan.
<b>--cloneFrom &lt;version&gt;</b>	<p>Specifies the name of an existing project version to use as a <a href="#">clone</a>.</p> <p>To clone a project version, use the:</p> <ul style="list-style-type: none"><li>• <b>--project</b> parameter to specify the project you wish to clone from.</li><li>• <b>--release</b> parameter to specify the new project version.</li><li>• <b>--cloneFrom</b> parameter to specify the project version to use as a clone.</li></ul> <p>For example, to clone version 1.0 of project SampleProject to a new version called 2.0, you would include these parameters:</p> <pre>--project SampleProject --release 2.0 --cloneFrom 1.0</pre>
<b>--context &lt;context&gt;</b>	Additional URL context. Use this parameter, for example, if the X-Forwarded-Prefix header is being specified in a proxy server/load balancer configuration.
<b>--dryRunReadFile &lt;data directory&gt;</b>	Specifies the directory, including the file name, from a dryRun scan and posts the scan to the Black Duck server.
<b>--dryRunWriteDir &lt;data directory&gt;</b>	Specifies the directory to which the scanner outputs a JSON file with the original file metadata used for scanning. The scanner does not connect to or post the scan to the Black Duck server. Note that the <code>data</code> directory is created inside the specified directory.
<b>--exclude &lt;pattern&gt;</b>	Excludes a directory or several directories from scanning.
<b>--exclude-from &lt;filename&gt;</b>	The scanner automatically excludes these directories and the contents of these directories: <ul style="list-style-type: none"><li>■ CVS</li><li>■ .svn</li><li>■ .git</li><li>■ .hg</li><li>■ .bzr</li></ul>

Parameter	Description
	<ul style="list-style-type: none"> <li>■ <code>__MACOSX</code></li> </ul> <p>The scanner automatically excludes files named:</p> <ul style="list-style-type: none"> <li>■ <code>.cvsignore</code></li> <li>■ <code>.git</code></li> <li>■ <code>.gitignore</code></li> <li>■ <code>.gitattributes</code></li> <li>■ <code>.gitmodules</code></li> <li>■ <code>.hgignore</code></li> <li>■ <code>.hgsub</code></li> <li>■ <code>.hgsubstate</code></li> <li>■ <code>.hgtags</code></li> <li>■ <code>.bzrignore</code></li> <li>■ <code>vssver.scc</code></li> <li>■ <code>.DS_Store</code></li> </ul> <p>To exclude other directories, use <b>--exclude</b> to exclude a single directory; <b>--exclude-from</b> to specify a file that lists directories that should be excluded.</p> <p>Exclusion guidelines:</p> <ul style="list-style-type: none"> <li>■ Leading and trailing forward slashes are required.</li> </ul> <p>For example, if you enter <code>exclude /directory</code>, a warning message will appear and the directory will not be excluded. If you enter <code>/directory</code> in the file, the directory will not be excluded.</p> <ul style="list-style-type: none"> <li>■ Directory names cannot contain double asterisks (**).</li> <li>■ Specify one directory per line in the file. Include the complete direct path. The path must be a relative path rather than an absolute path.</li> <li>■ You cannot exclude archives or contents within archives.</li> </ul> <p>There are two additional methods you can use to exclude directories from scanning:</p> <ul style="list-style-type: none"> <li>■ Create an <code>ignore</code> file located in the <code>\$HOME/config/blackduck</code> directory.</li> </ul> <p>Use this file to list excluded directories, relative to root. This option provides you with the ability to use one location to list all directories that need to be excluded.</p> <ul style="list-style-type: none"> <li>■ Create individual <code>.bdignore</code> files which can be located in any directory.</li> </ul> <p>Use this file to list the excluded subdirectories in the directory where the <code>.bdignore</code> file is located. You must create a <code>.bdignore</code> file in each</p>

Parameter	Description
	<p>directory that has subdirectories you want to exclude.</p> <p>You do not need to use the <b>--exclude-from</b> parameter with these files: the scanner reads these files while scanning and excludes the listed directories.</p> <p>You must also follow the exclusion guidelines as described above when using either of these methods.</p> <p><b>Tip:</b> Use the <b>debug</b> parameter when excluding directories to ensure that the scanner visited and excluded the directory.</p>
<b>--host &lt;host&gt;</b>	Server hosting the Black Duck installation.
<b>-insecure</b>	Ignores TLS validation errors, allowing the scanner to connect to the Black Duck server.
<b>--logDir &lt;log directory&gt;</b>	<p>Location of the <code>log</code> directory which contains all scanner log files.</p> <p>By default, the <code>log</code> directory is located at:</p> <p><b>Linux/MAC OS X:</b></p> <pre>/opt/blackduck/hub/scan.cli-2019.8.0/scan.cli-2019.8.0</pre> <p><b>Windows:</b></p> <pre>C:\scan.cli-2019.8.0\scan.cli-2019.8.0</pre>
<b>--name &lt;scan name&gt;</b>	<p>Unique name identifying this scan. This name is displayed on the Scans page. Click <a href="#">here</a> for more information.</p> <p><b>Note:</b> The <b>--name</b> parameter is not supported when specifying multiple scan paths in a single command line.</p>
<b>--no-prompt</b>	<p>Non-interactive mode.</p> <p>Instead of the <b>--no-prompt</b> parameter, you can set the <code>BD_HUB_NO_PROMPT</code> environment variable to enable non-interactive mode.</p>
<b>--project &lt;project&gt;</b>	<p>Name of the <a href="#">project</a> to which you want to map the scan results.</p> <p>If you specify a project, you must specify a version.</p> <ul style="list-style-type: none"> <li>■ If the project and project version exist, the scanner maps or remaps the scan results.</li> <li>■ If the project exists, but the version does not, the scanner creates the version and maps the scan results.</li> </ul>

Parameter	Description
<b>--password &lt;password&gt;</b>	<p>Forces the scanner to prompt you for the password for the user account with the code scanner role:</p> <ul style="list-style-type: none"> <li>Specifying the <b>--password</b> parameter without the <i>password</i> value results in the scanner prompting you for the password.</li> <li>Specifying the <i>password</i> value displays a warning message notifying you that specifying the password on the command line will not be supported in future versions of Black Duck; the scan then runs.</li> </ul> <p>Set the BD_HUB_PASSWORD environment variable with the Black Duck server password instead of passing an argument to the <b>--password</b> parameter:</p> <ul style="list-style-type: none"> <li>If you set this environment variable <i>and</i> specify the <b>--password</b> parameter, the scanner prompts you for the password; it does not check the password value against the value specified in the environment variable.</li> <li>If you set this environment variable <i>and do not</i> specify the <b>--password</b> parameter, the scanner does <i>not</i> prompt you for the password.</li> </ul> <div style="background-color: #FFFACD; padding: 10px;"> <p><b>Important:</b> Set the BD_HUB_PASSWORD environment variable with the Black Duck server password. If you supply the <b>password</b> parameter, an error message appears and the scan will not complete.</p> </div>
	<p>If this environment variable is <i>not</i> set, the scanner prompts you for the password whether you include or omit the <b>--password</b> parameter.</p> <div style="background-color: #F0F0F0; padding: 10px;"> <p><b>Note:</b> If the <b>password</b> parameter is the parameter immediately before <i>&lt;scan_path&gt;</i> use <b>--</b> to indicate you are finished passing parameters, for example <b>--password -- &lt;scan_path&gt;</b>. Otherwise, the scanner will try to use the <i>&lt;scan_path&gt;</i> value as the password.</p> </div>
	<p>Instead of specifying a username and password, use the <b>BD_HUB_TOKEN</b> environment variable to specify a Black Duck API token.</p>
<b>--port &lt;port&gt;</b>	<p>Port on which the Black Duck server instance is listening.</p>
<b>--release&lt;release&gt;</b>	<p>Name of the project version to which you want to map the scan results.</p> <p>If you specify a version, you must specify a project.</p> <ul style="list-style-type: none"> <li>If the project and project version exist, the scanner maps or remaps the scan results.</li> <li>If the project exists, but the version does not, the scanner creates the version and maps the scan results.</li> </ul>
<b>--scheme &lt;scheme&gt;</b>	<p>Protocol to use to connect to the server hosting the Black Duck installation. Possible values are http or https; https is the default value. You must include <b>-scheme https</b> to specify the https protocol.</p>

Parameter	Description
<b>--statusWriteDir &lt;directory&gt;</b>	Specifies the directory to which the scanner outputs a JSON file which contains the complete scan status information.
<b>--selfTest</b>	Performs a self-test; will not connect to or post the scan to the Black Duck server.
<b>--snippet-matching</b> <b>--snippet-matching-only</b> <b>--full-snippet-scan</b>	<p>Select one of the following for snippet matching:</p> <ul style="list-style-type: none"> <li>• <b>--snippet-matching.</b> Selecting this parameter enables a two-phase approach to scanning. First, a component scan is completed whereby only files that have changed since the previous scan are scanned. Once that component scan is completed, a snippet scan runs on those newly scanned files only: if a previously scanned file has not changed, it will not be rescanned for snippets.</li> </ul> <p>Black Duck Software recommends using this parameter for snippet scanning.</p> <ul style="list-style-type: none"> <li>• <b>--snippet-matching-only.</b> Selecting this parameter runs a snippet scan only on files that have changed; a component scan is not performed.</li> </ul> <p>You must have successfully completed a full file scan prior to selecting this parameter.</p> <ul style="list-style-type: none"> <li>• <b>--full-snippet-scan.</b> Selecting this parameter performs a snippet scan on all files.</li> </ul> <p>This parameter must be used with the <b>--snippet-matching</b> or <b>--snippet-matching-only</b> parameter:</p> <ul style="list-style-type: none"> <li>• With the <b>--snippet-matching</b> parameter: First, a component scan is completed whereby only files that have changed since the previous scan are scanned. Once that scan is completed, a snippet scan is performed on <i>all</i> files.</li> <li>• With the <b>--snippet-matching-only</b> parameter: A snippet scan is performed on all files; a component scan is <i>not</i> completed.</li> </ul> <p>To upload source files, you must use the <b>--upload-source</b> parameter, as described below.</p>
<b>--tlscertpass</b>	<p>Forces the scanner to prompt you for the password for the client certificate.</p> <p>You can specify the <b>--tlscertpass</b> parameter and/or set the BD_HUB_CLIENTCERT_PASS environment variable which specifies the private key password for the client certificate, for example, when <b>--tlscert</b> points to an encrypted PKCS #12 key store.</p> <p>The result of specifying the <b>--tlscertpass</b> parameter depends on whether the key is encrypted.</p> <ul style="list-style-type: none"> <li>■ If the key <i>is</i> encrypted, the scan will fail if you do not set the BD_HUB_CLIENTCERT_PASS environment variable or specify the <b>--tlscertpass</b> parameter.</li> </ul>

Parameter	Description
	<ul style="list-style-type: none"> <li>If you set the environment variable <i>and</i> specify the <b>--tlscertpass</b> parameter, the scanner prompts you for the password; it does not check the password value against the value specified in the environment variable.</li> <li>If the key <i>is not</i> encrypted, regardless of whether the BD_HUB_CLIENTCERT_PASS environment variable is set: <ul style="list-style-type: none"> <li>Specifying the <b>--tlscertpass</b> parameter forces the scanner to prompt you for the password for the client certificate. The scan will fail unless the password is empty.</li> <li>If you do not specify the <b>--tlscertpass</b> parameter, the scan will succeed.</li> </ul> </li> </ul>
<b>-tlskey &lt;keyFile&gt;</b>	Black Duck client certificate private key file. Automatically sets <b>--scheme</b> to https.  <b>Note:</b> This parameter is optional as the key and certificate can be included in the key store file specified with <b>--tlscert</b> .
<b>--tlscert &lt;certFile&gt;</b>	Black Duck client certificate chain file or key store file. Automatically sets <b>--scheme</b> to https. Click <a href="#">here</a> for more information on using certificate-based authentication.
<b>--upload-source</b>	Uploads the source file.  This parameter must be used with the <b>--snippet-matching</b> or <b>--snippet-matching-only</b> parameters.
<b>--username &lt;username&gt;</b>	Black Duck user account with the code scanner <a href="#">role</a> .  Instead of specifying a username and password, use the <b>BD_HUB_TOKEN</b> environment variable to specify a Black Duck API token.
<b>-V, --version</b>	Shows the version information of this tool.
<b>-v, --verbose</b>	Sets the logging level to verbose.
<b>--debug</b>	Shows debug output.
Other environment variable:  • <b>BD_HUB_TOKEN</b>	Used to specify the Black Duck API token which is the preferred authentication method over username and password.  Use the <a href="#">Profile page</a> in the Black Duck UI or the api-token-rest-server API to manage API tokens.

## Specifying the password

Set the BD\_HUB\_PASSWORD environment variable with the Black Duck server password. If you supply the **password** parameter, the scan will not complete.

## About package management files

By default, the scanner does not include components declared in supported package management files. Use Synopsys Detect to discover declared dependencies.

## Examples

The following are examples of using the command line to run the Signature Scanner CLI.

- Scanning and sending scan data to Black Duck
- Scanning and mapping the scan data

Note that:

- In all examples, the user has a code scanner role. Contact your Black Duck administrator for more information.
- The examples show only required parameters.

### To scan and send the scan data to Black Duck

1. Open a command prompt.
2. Go to the directory where the Signature Scanner is installed.

For example:

#### **Linux/MAC OS X:**

```
/opt/blackduck/hub/scan.cli-2019.8.0/scan.cli-2019.8.0/bin
```

#### **Windows:**

```
C:\scan.cli-2019.8.0\scan.cli-2019.8.0\bin
```

4. Run the following command to configure and initiate the scan.

For example:

#### **Linux/Mac OS X:**

```
./scan.cli.sh --username <username> --host <host> --port <port> <scan_path>
```

#### **Windows:**

```
scan.cli.bat --username <username> --host <host> --port <port> <scan_path>
```

The Signature Scanner sends the scan data to Black Duck's server. Log in to Black Duck to view the status of the component scan and its results and to map the component scan to a project, which adds the identified components to the project BOM.

- ⚙️ To scan over HTTPS, sending the scan data to Black Duck, and automatically mapping scan to a project

1. Open a command prompt.
2. Go to the directory to which the scanner is installed.

**Linux/MAC OS X:**

```
/opt/blackduck/hub/scan.cli-2019.8.0/scan.cli-2019.8.0/bin
```

**Windows:**

```
C:\scan.cli-2019.8.0\scan.cli-2019.8.0\bin
```

4. Run the following command to configure and initiate the scan.

**Linux/Mac OS X:**

```
./scan.cli.sh --username <username> --host <host> --port <port> --  
scheme HTTPS --project <project> --release <release> <scan_path>
```

**Windows:**

```
scan.cli.bat --username <username> --host <host> --port <port> --  
scheme HTTPS --project <project> --release <release> <scan_path>
```

The Signature Scanner sends the scan data to the Black Duck server and automatically maps the scan to the version of the project you specified.

## Reducing the number of parameters entered on the command line for the Signature Scanner

You may need to scan numerous times using the same values for some or all of the parameters. To make this procedure easier, use the alias command in Linux and Mac OS X or the DOSKEY utility in Windows to reduce the number of parameters you must enter on the command line.

- ⚙️ To reduce the number of parameters in Linux and Mac OS X

Create an alias that runs the Signature Scanner and specifies those parameters that will not change.

1. Open a terminal window and optionally, go to the directory where the Signature Scanner is installed.
2. Create an alias. The alias command has the following syntax:

```
alias <AliasName>="<PathToCommand> --<Parameter1> <Value1> --<Parameter2>  
<Value2>...--<ParameterN> <ValueN>"
```

The following example contains all required parameter excluding the <scan path> value and password:

```
alias HubScan=". ./scan.cli.sh --host hostName --port 80 --username sysadmin  
--project projectName --release releaseNumber"
```

### 3. Run the alias command.

```
AliasName --<RemainingParameter1> <Value 1>... --<RemainingParameterN>
<ValueN>
```

The following example runs the alias command with the password and path to the file directory specified:

```
HubScan /path/to/file/to/scan --password passwordValue
```

#### To reduce the number of parameters in Windows

Use the DOSKEY utility to create a macro that executes the Signature Scanner.

1. Open a command prompt and optionally go to the directory where the Signature Scanner is installed.
2. Create the macro. DOSKEY has the following syntax:

```
DOSKEY <Macro_Name>=<path to command> -<Parameter1> <Value1> -<Parameter2>
<Value2>...-<ParameterN> <ValueN>$*
```

The following example contains all required parameter excluding the **<scan path>** value and password:

```
DOSKEY HubScan=scan.cli.bat -host hostName -port 80 -username sysadmin -
project projectName -release releaseNumber $*
```

**Note:** DOSKEY must have \$\* at the end in order to specify additional parameters when the macro is called.

### 3. Run the DOSKEY command.

```
DOSKEYName -<RemainingParameter1> <Value1>... -<RemainingParameterN>
<ValueN>
```

The following example runs the DOSKEY command with the password and path to the file directory specified:

```
HubScan /path/to/file/to/scan -password passwordValue
```

## Accessing the Black Duck server via a proxy

If the client running the component scans communicates with Black Duck via a proxy server, for example, the Black Duck instance is located outside of your company and your company policy requires a proxy server, you must set a SCAN\_CLI\_OPTS environment variable prior to running the client. If this environment variable is not configured, scans will fail.

The Black Duck scan client supports Digest, Basic, and NTLM authentication.

For an HTTP proxy server:

```
SCAN_CLI_OPTS=-Dhttp.proxyHost=<ProxyHostName> -Dhttp.proxyPort=<ProxyPort> -
Dhttp.nonProxyHosts=<NonProxyHostName> -Dhttp.proxyUser=<Username> -
Dhttp.proxyPassword=<Password>
```

For an HTTPS proxy server:

```
SCAN_CLI_OPTS=-Dhttps.proxyHost=<ProxyHostName> -Dhttps.proxyPort=<ProxyPort>
-Dhttp.nonProxyHosts=<NonProxyHostName> -Dhttp.proxyUser=<Username> -
Dhttp.proxyPassword=<Password>
```

For NTLM authentication:

```
SCAN_CLI_OPTS=-Dhttp.proxyHost=<ProxyHostName> -Dhttp.proxyPort=<ProxyPort> -
Dhttp.proxyUser=<Username> -Dhttp.proxyPassword=<Password> -
Dhttp.auth.ntlm.domain=<ntlmDomain> -
Dblackduck.http.auth.ntlm.workstation=<ntlmWorkstation>
```

where

- (required) **<ProxyHostName>** The name of the proxy server host.
- (required) **<ProxyPort>** The port on which the proxy server host is listening.
- (optional) **<NonProxyHostName>** The name of any non-proxy hosts. These are servers that are trusted and do not need to go through the proxy server.
- (optional) **<Username>** Username to access the proxy server.
- (optional) **<Password>** Password to access the proxy server.
- (if required by proxy server for NTLM authentication) **<ntlmDomain>** The domain to authenticate within.
- (if required by proxy server for NTLM authentication) **<ntlmWorkstation>** The workstation the authentication request is originating from. Essentially, the computer name for this machine.

 To configure the SCAN\_CLI\_OPTS environment variable in Linux or Mac OS X

1. Start a terminal session.
2. At the command line, type

```
export SCAN_CLI_OPTS=<variable values>"
```

3. Close the terminal session.

 To configure the SCAN\_CLI\_OPTS environment variable in Windows

1. Access the System Properties dialog box. For example, in Windows 7, click **Start > Control Panel > System > Advanced System Settings**.
2. Select the **Advanced** tab and click **Environment Variables**.
3. In the Environment Variables dialog box, under **System Variables**, click **New**.
4. Enter the following information:

**Variable Name:** SCAN\_CLI\_OPTS

**Variable value:** <Variable Values>

5. Click **OK**.

For information on resolving proxy errors in Black Duck refer to [Resolving Proxy Errors](#).

## Running an offline component scan using the Signature Scanner

If a client does not have access to Black Duck, you can [use the command line for the Signature Scanner](#) to run an offline scan to identify the open source software (OSS) components contained in an archive or a directory of files. Running an offline scan lets you:

- Use the Signature Scanner to run a scan and save the results to a data file.
- Upload the data file from a client that does have access to Black Duck to create a BOM.

**Note:** An error message appears if you exceed the scan size limit, which is 5 GB. Contact Customer Support if you receive this message.

### To run an offline component scan

1. Be sure that you have a code scanner [role](#).
2. Using a client that has access to Black Duck, [download the Signature Scanner CLI](#) for the platform where the offline scan will occur.
3. Move the zip file to the client that does not have access to Black Duck and extract the files.
4. From the client that does not have access to Black Duck, go to the directory where the Signature Scanner is installed and enter the command to run the scan.

For example:

#### **Linux/Mac OS X:**

```
./scan.cli.sh --dryRunWriteDir <data_directory> <scan_path>
```

#### **Windows:**

```
scan.cli.bat --dryRunWriteDir <data_directory> <scan_path>
```

5. Move the `data` directory that contains the JSON file to a client that has access to Black Duck.
6. From the client that has access to Black Duck, send the scan data to Black Duck using the user interface or the Signature Scanner.

### To send the data using the user interface

1. Log in to Black Duck.
2. Select the expanding menu () icon and select **Scans**.
3. In the Scans page, click **+Add** and select **Scan File**.
4. Use the Upload Scan File dialog box to locate the JSON file, and click **Close**.

### ⚙️ To send the data using the Signature Scanner CLI

1. Open a command prompt.
2. Go to the directory to which the Signature Scanner is installed and run the following command:

For example:

#### Linux/Mac OS X:

```
./scan.cli.sh --dryRunReadFile <data directory> --username <username> --  
host <host> --port <port>
```

#### Windows:

```
scan.cli.bat --dryRunReadFile <data directory> --username <username> --  
host <host> --port <port>
```

## Using certificate-based authentication with the Signature Scanner

You can use a client certificate, also known as a signed key pair, to authenticate to a TLS-enabled server.

From the command line, enter the **--tlscert <certFile>** and optionally the **--tlskey <keyFile>** parameters. These two parameters represent both the signed public key and the private key, respectively, used to authenticate to the TLS-enabled server.

Optionally you can specify the **--tlscertpass** parameter to force a password prompt for the client certificate or use the **BD\_HUB\_CLIENTCERT\_PASS** environment variable to specify the password for the private key. Click [here](#) for more information.

## Examples

The following are examples of using certificate-based authentication with a certificate that does and does not include a separate private key file.

Note that:

- The examples show only required parameters.
- The key is encrypted and the **BD\_HUB\_CLIENTCERT\_PASS** environment variable has been set. Therefore, the **--tlscertpass** parameter is not included.

### ⚙️ To use a certificate that does not includes the private key (that is, a key store)

1. Open a command prompt.
2. Go to the directory where the Signature Scanner is installed.

#### Linux/MAC OS X:

```
/opt/blackduck/hub/scan.cli-2019.8.0/scan.cli-2019.8.0/bin
```

#### Windows:

C:\scan.cli-2019.8.0\scan.cli-2019.8.0\bin

4. Run the following command to configure and initiate the scan.

**Linux/Mac OS X:**

```
./scan.cli.sh --host <host> --port <port> --tlskey <keyFile> --tlscert <certFile> <scan_path>
```

**Windows:**

```
scan.cli.bat --host <host> --port <port> --tlskey <keyFile> --tlscert <certFile> <scan_path>
```

 To use a certificate that includes the private key (that is, a key store)

1. Open a command prompt.
2. Go to the directory where the Signature Scanner is installed.

**Linux/MAC OS X:**

/opt/blackduck/hub/scan.cli-2019.8.0/scan.cli-2019.8.0/bin

**Windows:**

C:\scan.cli-2019.8.0\scan.cli-2019.8.0\bin

4. Run the following command to configure and initiate the scan.

**Linux/Mac OS X:**

```
./scan.cli.sh --host <host> --port <port> --tlscert <certFile> <scan_path>
```

**Windows:**

```
scan.cli.bat --host <host> --port <port> --tlscert <certFile> <scan_path>
```

The Signature Scanner sends the scan data to the Black Duck server. Log in to Black Duck to view the status of the component scan and its results and to map the component scan to a project, which adds the identified components to the project BOM.

## Defining the scan name

By default, the name of a scan, as shown on the Scans page, is a combination of the host name of the server that ran the scan and the path to the code. This name is created when you run the scan. You may want to specify a different name.

Some examples of why you may want to specify a scan name are:

- You are using a continuous integration build system and have multiple slave/client servers running a scan. Each slave/client server has a different host name. Depending on which slave/client server completes the scan, there can be duplicate scan files for the same scan. Your BOM may also be inaccurate as old scans are included although the code has been rescanned.

By entering a unique scan name, duplicate scan files are eliminated. Your BOM no longer contains old scans as multiple slaves/clients can now run the same scan: the newest scan replaces the existing scan as the most current scan for given code.

- You have many different build system work spaces that you scan and you want to reuse the same workspace for multiple projects. By using a different name for the scans, you can use the same workspace and have the code point to different projects.

To specify a name, use the **--name** parameter when using the [command line](#) and provide a unique name for a scan. This name appears on the Scans page.

Note the following:

- Scan names are case insensitive. Scan1, scan1, and SCAN1 are considered the same name.
- Scans with the same host and path but different names are considered different scan files.
- The host name of the server that ran the scan and the path to the code are shown in the **Scan Details** table in the *Scan Name* page.

## Specifying names for BOM or JSON files

You can change the default scan name specified in BOM files (such as from Maven, Gradle, or from the Protex BOM tool) and in JSON files (such as the file that is output when using the **--dryrunWrite** parameter).

To change the existing name, open the file using an application such as Notepad and enter a new value for the **spdx:name** parameter:

```
spdx:name : "Scan Name"
```

## Resolving memory issues

You may receive the following error when trying to run Signature Scanner:

```
ERROR: Insufficient memory <Value>
```

To resolve this error, increase the memory that is available for use by the Signature Scanner. You can accomplish this by using the **SCAN\_CLI\_OPTS** environment variable to increase the values for the initial and maximum heap size.

**Note:** The value you specify for the maximum heap size must be larger than that value shown in the error message.

The instructions shown below describe how to use the command line to configure the environment variable. These instructions can be adapted so you can create an alias definition in Linux or Mac OS X or use the Control Panel in Windows.

## ⚙️ To configure the SCAN\_CLI\_OPTS environment variable in Linux or Mac OS X

1. Start a terminal session.

2. At the command line, type:

```
export SCAN_CLI_OPTS="-Xms<Initial heap size> -Xmx<Maximum heap size>"
```

For example, to set the minimum size to 1 GB and the maximum to 6 GB:

```
export SCAN_CLI_OPTS="-Xms1g -Xmx6g"
```

3. Close the terminal session.

## ⚙️ To configure the SCAN\_CLI\_OPTS environment variable in Windows

1. At the command line, type:

```
set SCAN_CLI_OPTS=-Xms<Initial heap size> -Xmx<Maximum heap size>
```

For example, to set the minimum size to 1 GB and the maximum to 6 GB:

```
set SCAN_CLI_OPTS=-Xms1g -Xmx6g
```

**Note:** There are limits in the maximum scan size when scanning with a 32-bit system as the increase in addressable memory is restricted by the limitations of the 32-bit system.

## About snippet matching

Snippets are small reusable pieces of computer code. A snippet of open source software can easily find its way into your proprietary files. For example, a developer may find a useful function from an open source program and cut and paste that code into their program.

Snippet matching is beneficial managing legal risk and detecting possible license infringement. A snippet match occurs when a portion of code in your file matches code in one or more KnowledgeBase files.

As the use of open source software is managed through licenses that allow you to use, modify, and/or share the software under defined terms and conditions, it is important to identify the open source software used in your proprietary code so that you can manage the legal risk and detect possible license infringement. Although your proprietary code may include only a portion of open source software code, you still must comply with the license associated with that open source software.

Snippet matching finds these fragments of open source code used in your proprietary files or files moved into proprietary directories and matches that code with open source code found in one or more Black Duck KB files.

## Snippet scanning process

All scanning methods have an option to enable snippet scanning. Enabling the snippet scanning option scans files not identified as open source (proprietary files). The methods to scan your code for snippets are by using:

- Signature Scanner command line
- Synopsys Detect (Desktop)

- Synopsys Detect

The process for snippet scanning is:

1. The component scan is completed first. This identifies the OSS components using directory/file-level signatures.
2. Optionally, a second-pass scan is completed. This scan analyzes files that were not matched in the initial component scan.
3. Snippet fingerprints are generated and sent to Black Duck and then sent to the KB Snippet Matching Service.
4. The user reviewing matches details.

Each snippet scanning option is discussed as follows.

### Using the Signature Scanner command line

The command line has three parameters you can select for snippet matching:

- **--snippet-matching**. Selecting this parameter enables a two-phase approach to scanning. First, a component scan is completed whereby only files that have changed since the previous scan are scanned. Once that component scan is completed, a snippet scan runs on those newly scanned files only: if a previously scanned file has not changed, it will not be rescanned for snippets.

Black Duck recommends using this parameter for snippet scanning.
- **--snippet-matching-only**. Selecting this parameter runs a snippet scan only on files that have changed; a component scan is not performed.

You must have successfully completed a full file scan prior to selecting this parameter.
- **--full-snippet-scan**. Selecting this parameter performs a snippet scan on *all* files.

This parameter must be used with the **--snippet-matching** or **--snippet-matching-only** parameter:

- With the **--snippet-matching** parameter: First, a component scan is completed whereby only files that have changed since the previous scan are scanned. Once that scan is completed, a snippet scan is performed on *all* files.
- With the **--snippet-matching-only** parameter: A snippet scan is performed on *all* files; a component scan is *not* completed.

Use this option to take advantage of new signatures in the Black Duck KB.

Click [here](#) for more information on using the command line.

**Note:** Snippet scanning cannot be completed offline as it requires communication with the Black Duck server.

### Using Synopsys Detect (Desktop)

To enable scanning for snippets, select the select **Snippet Scanning** from the **Settings** options and enable it. Selecting this option runs the scanner using the command line **--snippet-matching** parameter, as described

above.

## Using Synopsys Detect

Use the **--detect.blackduck.signature.scanner.snippet.mode** property to enable snippet scanning in Synopsys Detect. With this property enabled, Synopsys Detect uses the command line **--snippet-matching** parameter, as described above.

## Uploading source files for snippet matching

Black Duck provides the ability for you to upload your source files so that BOM reviewers can see the file contents for reviewing snippet matches from within the Black Duck UI. When source files are uploaded, Black Duck provides a side-by-side comparison of the source file to the match which can help BOM reviewers in the evaluation and review of the snippet match.

After your administrator has enabled source uploads, as described in the installation guides, use the Signature Scanner and include the **--upload-source** parameter when using the **--snippet-matching** or **--snippet-matching-only** parameter.

## Reviewing snippet matches

It can be difficult to determine where a snippet of code originated; in other words, which open source supplied the snippet of code. The matching process attempts to select the best match for a snippet of code by selecting a component and version in the following order:

1. Highest KB ranked component/version.
2. Highest license risk component/version.
3. Earliest version of component by release date.
4. Component with the most versions for which a match appears.

As snippet matching is an imprecise technique, snippet matches must be reviewed prior to including these matches in your BOM. Use the **Source** tab, as described [here](#), to determine if the snippet match is relevant; in other words, does this snippet belong in your BOM? If so, determine if the snippet match is correct.

After reviewing the snippet match, add it to your BOM. The component is shown with:

- Match type = Snippet
- Usage = Source Code

Any policies you have created execute.

## Snippet Matches and Vulnerabilities

Black Duck does not include any vulnerabilities related to components/versions that are identified through snippet matching *only*: vulnerabilities are not counted when showing the total number of vulnerabilities for a project/project version and are also excluded from vulnerability reports. Black Duck will add vulnerabilities/security risk identified by a snippet match if another type match type (for example, exact) identified the same component/version.

## Resolving proxy errors

Black Duck version 4.5.0 introduced a larger HTTP header size. The larger header size may cause problems with the load balancer. If this occurs, the larger header size may cause authentication errors in Black Duck environments running a proxy server. To prevent possible authentication errors and to support HTTP responses from Black Duck, Black Duck Software recommends increasing the allowed maximum HTTP header size in Black Duck versions 4.5.0 and higher to 8192.

## About custom scan signatures - BETA

Your software projects may contain a mix of open source, third-party, and proprietary software components. While the Black Duck KnowledgeBase can identify your open source components, it cannot identify third-party or proprietary software components. As such, your BOM may not include all the software components used in your code.

To ensure that your BOM tracks all your code, you can enable custom scan signatures which you can use to identify third-party and proprietary software used in your code. Once identified, and displayed in your BOM, you can track the use of proprietary code within your organization and ensure that you meet the license obligations required by your third-party software,

If you are interested in using this beta feature, please contact your Synopsys Customer Success Representative or Account Executive for more information about enabling this feature and potential impacts on scan performance.

**Important:** This is a beta version of the custom code signatures feature. As such, this feature may not perform as expected. Also, there may be *significant* performance issues seen when using this feature.

## Understanding the custom scan signature process

Custom code signatures is an optional feature. Once enabled, the match service uses these signatures to identify custom code when scanning other projects.

Unlike the Black Duck KnowledgeBase, custom scan signatures reside on your local Black Duck instance (whether the server is on premises or hosted by Synopsys).

### Identifying custom code signatures in your code

As the scan client scans the code, it generates “signatures” of the files and directories it is scanning. After the scan completes, these signatures are initially sent to the Black Duck KnowledgeBase (KB) web service where the match service uses the signatures to identify the open source components/versions that are contained in the code being scanned. After identifying the open source components, these signatures are then sent to your local Black Duck instance where the match service compares the signatures to the custom scan signatures. After identifying the custom code signatures that are in the scanned code, the BOM is then created.

**Note:** To improve performance, custom scan signatures have been limited to the top four levels in the directory structure.

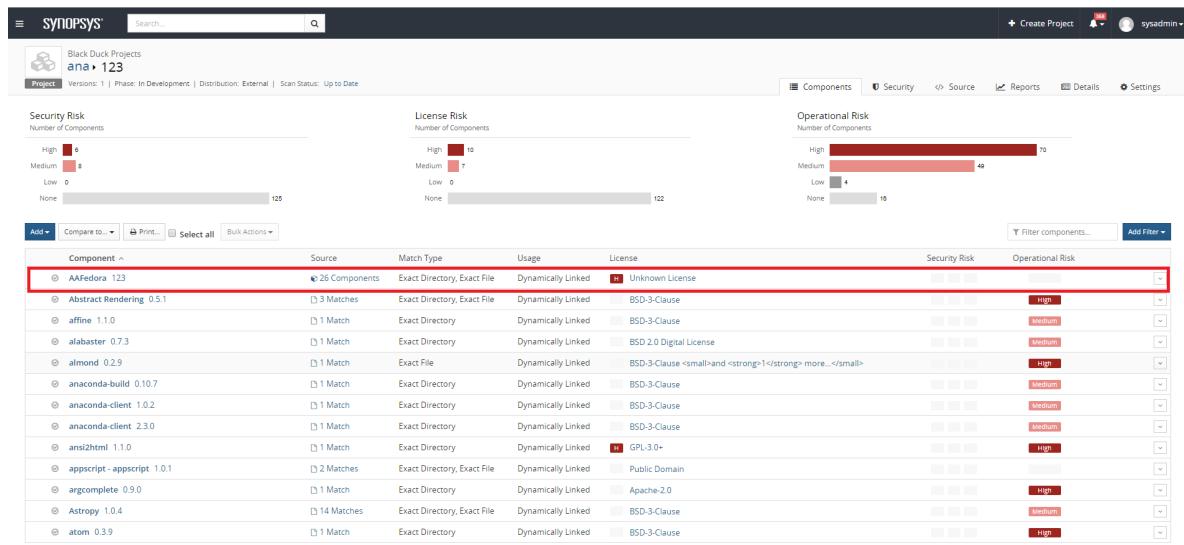
### Creating custom scan signatures

Custom code signatures are managed as projects and after identifying the code the custom code signatures

are pulled into the BOM as a [subproject](#).

### To create custom scan signatures

1. Scan the third-party or proprietary code you wish to identify as a custom scan signature.
2. Map the scan to a project version.
3. Identify this project as a custom scan signature by selecting the option to let this project be automatically matched during scans and clicking **Save** in the project's **Settings** tab:
4. Scan your code. The custom scan signature appears in your BOM as a subproject:



The **Source** column displays the number of components in the subproject.

Note the following:

- If a project contains several versions of a custom scan signature project, the BOM will display only one match to one version of the custom code signature project.
- If the custom scan signature project contains open source components, values for security and operational risk may also appear in the BOM.
- Although you may have selected only one custom code signature project, if you have scanned several projects, you will experience performance issues.
- Policy violations within the subproject will not appear in the BOM. However, a policy violation will appear in the BOM for the subproject if a policy rule is violated at the project level.
- Users who do not have permission to the subproject will not be able to drill down to view additional data about that project version.
- A Custom Scan Signature filter has been added to the Project dashboard and the BOM page to help you find custom scan signature projects.

### Disabling custom scan signatures

If you experience significant performance degradation in scanning, you can disable this feature.

### To disable custom scan signatures

1. Clear the custom scan signature option for *all* projects.
2. Rescan your code.

## Associating custom components to custom scan signatures

1. [Create the custom component.](#)

Users with the Component Manager role can create custom components.

2. Create a custom scan signature, as described above:
  - a. Scan the code for the custom component and map the scan(s) to a project version.
  - b. In the project's **Settings** tab, select the option to allow automatic matching.
3. Select to view the project version created in step 2.
4. From the BOM page, select the **Source** tab and select the top node.
5. Modify the match for the custom component to associate the custom scan signature to the custom component:
  - a. Click **Edit**, located in the lower-left corner of the page, to open the Edit Component dialog box.
  - b. Select the custom component created previously and click **Update**.

Click [here](#) for more information on using the **Source** tab.

# Chapter 4: Managing scans in the Black Duck UI

Use Black Duck's UI to manage scans:

- [Uploading a scan file using the Black Duck UI.](#)
- [Browsing component scans.](#)
- [Viewing component scan results.](#)
- [Mapping a scan to a project.](#)
- [Removing a scan from a project.](#)
- [Deleting a scan.](#)
- [Viewing an audit log for a BOM file.](#)

## Uploading a scan file using the Black Duck UI

If you output the scan to a file, you can import the file into Black Duck using the UI.

### To upload a file

1. Log in to Black Duck and click the expanding menu () icon.
2. Select **Scans**.



Status	Name	Scan Size	Last Updated	Mapped to
<input type="checkbox"/>	cowboy-mac#/Users/cowboy/node_modules/get-stdin	3.58 KB	Aug 13, 2018	testScan2 2
<input type="checkbox"/>	cowboy-mac#/Users/cowboy/node_modules/iodash	823.26 KB	Aug 13, 2018	testScan1 2
<input type="checkbox"/>	FitNesseScanCodeLocation_2	184.28 KB	Aug 13, 2018	
<input type="checkbox"/>	hubui_10518	148.89 MB	Aug 13, 2018	

3. In the Scans page, click **Upload Scans**.
4. Use the Upload Scans dialog box to locate the file and upload it.
5. Click **Close** after uploading the file.

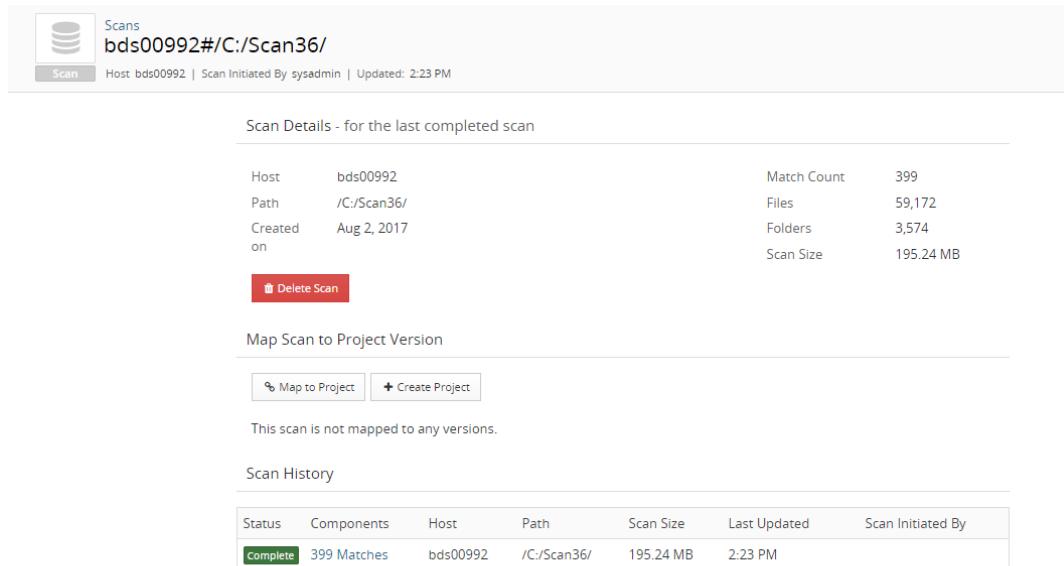
After uploading the file(s), use Black Duck to [map the file to a project](#).

## Browsing scans

You can view the results of a scan and the status of a scan that is in progress on the Scan Name page.

## ⚙ To browse component scans

1. Log in to Black Duck and click the expanding menu ( ) icon.
2. Select **Scans**.
3. Select the path of the scan that you want to view the results to open the *Scan Name* page.



The screenshot shows the 'Scan Details' page for a completed scan named 'bds00992#/C:/Scan36/'. The top navigation bar includes a 'Scans' icon, the scan name, host information ('Host bds00992 | Scan Initiated By sysadmin | Updated: 2:23 PM'), and a 'Scan' button. The main content area is divided into three sections:

- Scan Details - for the last completed scan:** Displays summary statistics:

Host	bds00992	Match Count	399
Path	/C:/Scan36/	Files	59,172
Created on	Aug 2, 2017	Folders	3,574
		Scan Size	195.24 MB

A red 'Delete Scan' button is located below these details.
- Map Scan to Project Version:** Contains buttons for 'Map to Project' and 'Create Project'. A note states: 'This scan is not mapped to any versions.'
- Scan History:** A table showing the history of this scan:

Status	Components	Host	Path	Scan Size	Last Updated	Scan Initiated By
Complete	399 Matches	bds00992	/C:/Scan36/	195.24 MB	2:23 PM	

The top of the page lists the:

- Host name of the machine where the latest scan was performed
- Last time this scan was uploaded and who initiated the scan

The **Scan Details** section provides the following information:

- Host name of the machine where the latest scan was performed
- Path to the code
- Match count, number of files, folders, and the size of the scan.

The **Map Scan to Project Version** section displays the project and project versions to which the scan is currently mapped. If this scan is unmapped, you can use the section to map this scan to a project or create a project and/or version.

The **Scan History** section displays the following information about each of the scans:

- Status of a scan. Possible values are:
  - **Not Started.** The scan has not started.
  - **Scanning.** The scanner is scanning. If the scanner could not complete the scan, a status of **Scan**

**Error** appears.

- **Paused.** A user paused scanning before it was completed.
  - **Pending.** The scan is pending.
  - **In Progress.** The scan or the building of the BOM is in progress.
  - **Error.** A schema error has occurred.
  - **Unknown.** The status of the scan is unknown.
  - **Saving Scan Data.** The scanner has completed scanning and has posted the scan results, which are a set of SHA1 hashes of the files and directories it has scanned, to the Black Duck server in a JSON file. The scan results are then persisted into the Black Duck database.
  - **Save Scan Data Complete.** The scan results have been saved in the Black Duck database. If the results are not saved correctly to the database, a status of **Saving Scan Data Error** appears.
  - **Request Match Job.** After the scan results have been saved on the Black Duck server, the Black Duck application initiates a job request to perform a match of the results to OSS components in the Black Duck Knowledge Base.
  - **Matching.** The Black Duck application is comparing the SHA1 hashes from the scan to the Black Duck KB to identify OSS components. If there are errors, a status of **Matching Error** appears.
  - **Building BOM.** Black Duck is building the BOM. If an error results, a status of **Building BOM Error** appears.
  - **BOM Version Check.** Black Duck is checking the version of the BOM.
  - **Complete.** The scan and matching process is complete and a BOM is available for review.
  - **Cancelled.** A user cancelled the scan before it was completed.
- Link to [view the components for this scan](#).
  - Host name of the machine where the latest scan was performed.
  - Path to the code.
  - Scan size.
  - Time the scan was created.
  - User who initiated the component scan.

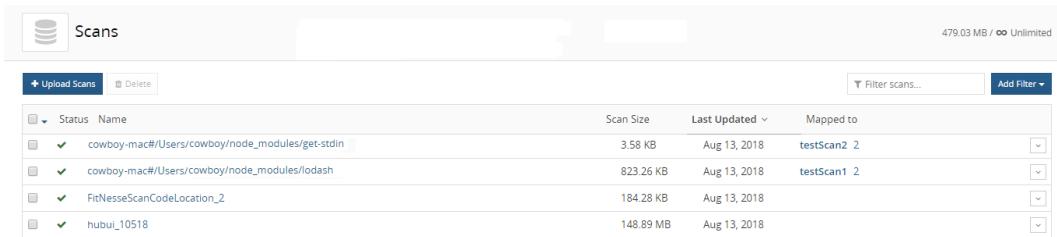
## Viewing scan results

After you scan an archive on your local computer, you can view the raw scan results. This is the raw data created during the scan process. Once the scan is [mapped to a project version](#), this data is used to automatically populate the BOM.

### To view scan results



1. Log in to Black Duck and click the expanding menu (  ) icon.
2. Click **Scans**.

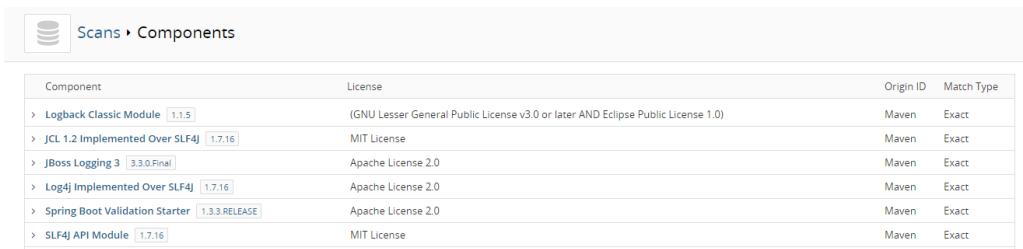


The screenshot shows the 'Scans' page with a table of scanned archive files. The columns are 'Status', 'Name', 'Scan Size', 'Last Updated', and 'Mapped to'. There are four entries:

Status	Name	Scan Size	Last Updated	Mapped to
✓	cowboy-mac#/Users/cowboy/node_modules/get-stdin	3.58 KB	Aug 13, 2018	testScan2_2
✓	cowboy-mac#/Users/cowboy/node_modules/lodash	823.26 KB	Aug 13, 2018	testScan1_2
✓	FitNesseScanCodeLocation_2	184.28 KB	Aug 13, 2018	
✓	hubui_10518	148.89 MB	Aug 13, 2018	

3. Select the name of the scan that you want to view to open the **Scan Name** page.
4. Select **XX Matches** in the table shown in the **Scan History** section.

The scan results are shown in the Scans > Components page that opens in a new browser tab or window.



The screenshot shows the 'Components' page with a table of identified components. The columns are 'Component', 'License', 'Origin ID', and 'Match Type'. There are six entries:

Component	License	Origin ID	Match Type
> Logback Classic Module [1.1.5]	(GNU Lesser General Public License v3.0 or later AND Eclipse Public License 1.0)	Maven	Exact
> JCL 1.2 Implemented Over SLF4J [1.7.16]	MIT License	Maven	Exact
> JBoss Logging 3 [3.3.0.Final]	Apache License 2.0	Maven	Exact
> Log4j Implemented Over SLF4J [1.7.16]	Apache License 2.0	Maven	Exact
> Spring Boot Validation Starter [1.3.3.RELEASE]	Apache License 2.0	Maven	Exact
> SLF4J API Module [1.7.16]	MIT License	Maven	Exact

The Scans > Components page contains the following information for each component identified in the scanned archive files:

- **Component:** The name and version of the component discovered in the scanned archive files.
- **License:** The license associated with the discovered component.
- One of the following:
  - **Forge name:** Where the component can be downloaded
  - **Origin ID:** Name of Linux distribution
- **Match Type:** The type of match made between the scanned archive and the component in the Black Duck KB.

Click > to view additional information.

5. Select a component to display the **Details** tab of the Black Duck KB [Component Name Version](#) page.

## Mapping a scan to a project

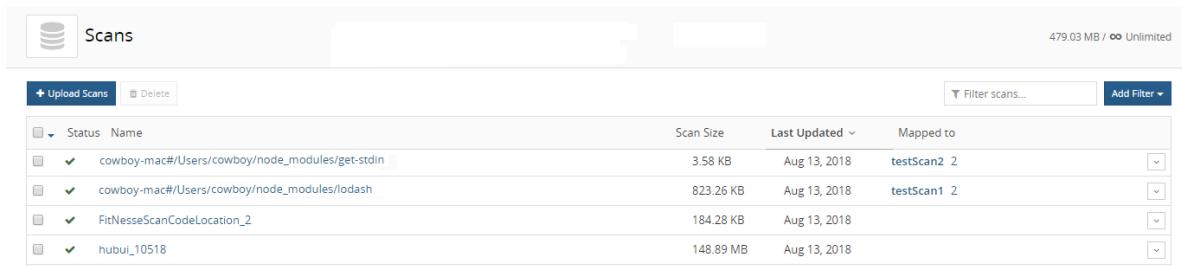
Mapping a scan adds the scan data to the BOM of a project version.

**Note:** You can scan a Docker image or file directory location or archive more than once, but you only have to map it to a project version once. As long as the host and path used to uniquely identify the scanned location or image does not change, Black Duck automatically updates the BOM of the project with any new information discovered during subsequent scans.

## To map a scan to a project

1. Log in to Black Duck and click the expanding menu () icon.

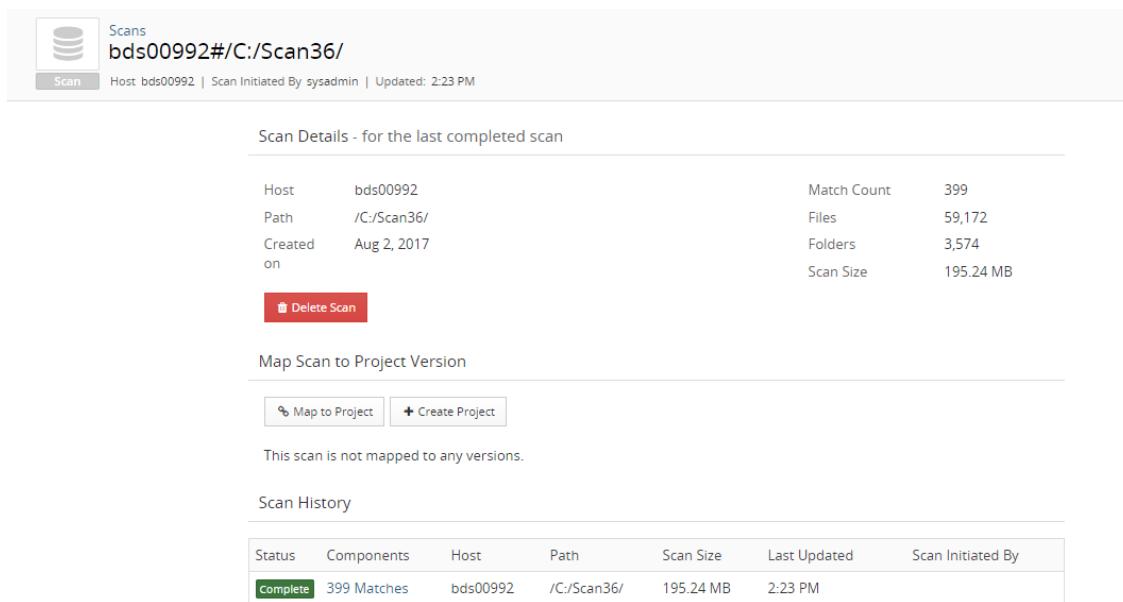
2. Select **Scans**.



Status	Name	Scan Size	Last Updated	Mapped to
✓	cowboy-mac#/Users/cowboy/node_modules/get-stdin	3.58 KB	Aug 13, 2018	testScan2 2
✓	cowboy-mac#/Users/cowboy/node_modules/lodash	823.26 KB	Aug 13, 2018	testScan1 2
✓	FitNesseScanCodeLocation_2	184.28 KB	Aug 13, 2018	
✓	hubui_10518	148.89 MB	Aug 13, 2018	

3. Do one of the following:

- Click  and select **Map to Project** in the row of the scan that you want to map.
- Select the path of the scan you want to map to open the *Scan Name* page.



Scan Details - for the last completed scan

Host	bds00992	Match Count	399
Path	/C:/Scan36/	Files	59,172
Created on	Aug 2, 2017	Folders	3,574
		Scan Size	195.24 MB

**Delete Scan**

Map Scan to Project Version

This scan is not mapped to any versions.

Scan History

Status	Components	Host	Path	Scan Size	Last Updated	Scan Initiated By
Complete	399 Matches	bds00992	/C:/Scan36/	195.24 MB	2:23 PM	

### Select **Map to Project**.

4. Start typing the name of a project to progressively display matches in the **Project** field.  
If necessary, select **Create Project** to create a new project and version.
5. Select the project version to which you want to map the component scan.

If necessary, select **Create Version** to create a new version for a project.

6. Click **Save**.

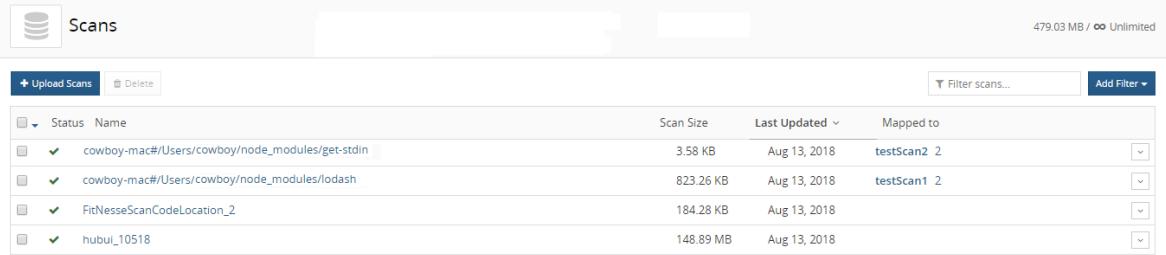
**Note:** Black Duck displays an aggregate project version BOM. If a component version appears more than once in an archive, it is only displayed in the BOM once.

## Removing a scan from a project

Removing the mapping of a scan removes the scan data from the BOM.

 To remove a mapping

1. Log in to Black Duck and click the expanding menu () icon.
2. Select **Scans**.



Status	Name	Scan Size	Last Updated	Mapped to
✓	cowboy-mac#/Users/cowboy/node_modules/get-stdin	3.58 KB	Aug 13, 2018	testScan2 2
✓	cowboy-mac#/Users/cowboy/node_modules/lodash	823.26 KB	Aug 13, 2018	testScan1 2
✓	FitNesseScanCodeLocation_2	184.28 KB	Aug 13, 2018	
✓	hubui_10518	148.89 MB	Aug 13, 2018	

3. Click  and select **Unmap from Project** in the row of the scan that you want to remove the mapping.
4. Click **Remove** to confirm.

## Deleting a scan

If you have scanned an incorrect path or Docker image, no longer require the scan, or want to free up space, you can delete the scan(s).

- Users with the global code scanner role can delete any scan.

 To delete a scan

1. Log in to Black Duck and select the expanding menu () icon.
2. Select **Scans**.
3. Select the scan(s) you want to delete by using the checkbox(es) and click **Delete**.  
You can also click  and select **Delete** in the row of the scan that you want to delete.
4. In the Delete Scan dialog box, confirm that you have selected the correct scan(s), and click **Delete**.

Black Duck removes the scan.

## Exporting a scan file

You may need a scan file, which is a file of a scan that has been imported to Black Duck, similar to a dry run file. For example, you may need to provide Customer Support with the scan file if you are experiencing scanning issues, as this file may help them investigate the issue.

**Note:** This feature is not available if you initially scanned using Black Duck version 5.x or earlier. If the option does not appear, delete the code location and re-scan.

### To export a file

1. Log in to Black Duck and click the expanding menu (≡) icon.
2. Select **Scans**.

The screenshot shows the 'Scans' page in Black Duck. At the top, there are buttons for 'Upload Scans' and 'Delete'. Below that is a search bar labeled 'Filter scans...' and a 'Add Filter' button. The main area displays a table of scans with columns: Status, Name, Scan Size, Last Updated, and Mapped to. There are five entries listed:

Status	Name	Scan Size	Last Updated	Mapped to
✓	cowboy-mac#/Users/cowboy/node_modules/get-stdin	3.58 KB	Aug 13, 2018	testScan2 2
✓	cowboy-mac#/Users/cowboy/node_modules/lodash	823.26 KB	Aug 13, 2018	testScan1 2
✓	FitNesseScanCodeLocation_2	184.28 KB	Aug 13, 2018	
✓	hubui_10518	148.89 MB	Aug 13, 2018	

3. Click and select **Download Scan** in the row of the scan that you want to obtain a scan file.

The file is downloaded with a .bdio extension.

## Viewing an audit log for a BOM file

Use the BOM Import Log to view your results of importing a BOM file. This log lists the components and licenses that were mapped to Black Duck. It also provides details for any items that were unable to be mapped.

### To view an audit log

1. Log in to Black Duck.
2. Select the expanding menu icon (≡) and select **Scans**.

The Scans page appears.

The screenshot shows the 'Scans' page with a list of completed scans. The table has columns for Status, Name, Scan Size, Last Updated, and Mapped to. There are five rows of data:

Status	Name	Scan Size	Last Updated	Mapped to
✓	cowboy-mac#/Users/cowboy/node_modules/get-stdin	3.58 KB	Aug 13, 2018	testScan2 2
✓	cowboy-mac#/Users/cowboy/node_modules/lodash	823.26 KB	Aug 13, 2018	testScan1 2
✓	FitNesseScanCodeLocation_2	184.28 KB	Aug 13, 2018	
✓	hubul_10518	148.89 MB	Aug 13, 2018	

At the top right, it says '479.03 MB / ∞ Unlimited'. There are buttons for '+ Upload Scans' and 'Delete'. A search bar says 'Filter scans...' and a dropdown says 'Add Filter'.

3. Select the scan you wish to view.

The *Scan Name* page appears.

The screenshot shows the 'Scan Name' page for 'bds00992#C:/Scan36'. It includes sections for 'Scan Details', 'Map Scan to Project Version', and 'Scan History'.

**Scan Details - for the last completed scan**

Host	bds00992	Match Count	399
Path	/C:/Scan36/	Files	59,172
Created on	Aug 2, 2017	Folders	3,574
		Scan Size	195.24 MB

**Delete Scan**

**Map Scan to Project Version**

This scan is not mapped to any versions.

**Scan History**

Status	Components	Host	Path	Scan Size	Last Updated	Scan Initiated By
Complete	399 Matches	bds00992	/C:/Scan36/	195.24 MB	2:23 PM	

4. In the **Scan History** section, click in the row of the scan you wish to view the log.

The BOM Import Log appears.

BOM Import Log		
Host	https://[REDACTED]	
Path	s-eval-201607	
Scan Details	complete Jul 11, 2016 / Queen Test	
The following is a list of components and licenses that were mapped to the Hub. Items that failed to map will contain a description of the failure.		
<b>23</b> Components Mapped	<b>3</b> Components Not Found	<b>17</b> Licenses Mapped
<b>0</b> License Not Found		
		<b>Add Filter ▾</b>
Import Name	Hub Name	Status
> cfitsio		Component Not Found
> PLYFormatConversion master-20100911		Component Not Found
> postgresql-8.4.4 master-20101210		Component Not Found
↳ Apache License 2.0	Apache License 2.0	License Mapped
↳ CFITSIO License	CFITSIO License	License Mapped
↳ Independent JPEG Group License	Independent JPEG Group License	License Mapped
↳ cad2octree - dime 0.9.1	cad2octree 0.9.1	Component Mapped
↳ PostgreSQL License	PostgreSQL License	License Mapped
↳ [template] Basic Proprietary Commercial License	[template] Basic Proprietary Commercial License	License Mapped
↳ Boost Software License 1.0	Boost Software License 1.0	License Mapped
↳ BSD 3-clause "New" or "Revised" License	BSD 3-clause "New" or "Revised" License	License Mapped
↳ Creative Commons Attribution 2.5	Creative Commons Attribution 2.5	License Mapped
↳ GNU General Public License v2.0 or later	GNU General Public License v2.0 or later	License Mapped

The host, path, and scan details (status of the scan, date, or time (if the date is today) the scan completed, and the username of the user that ran the scan) appear at the top of the page. The number of components mapped, components not found, licenses mapped, and licenses not found appear above the table. The table lists all components with unmatched items listed at the top.

- Click > in the row of unmatched items to view more information.
- Click **Add Filter** to view the table by a selected status.

# Chapter 5: Understanding projects in Black Duck

Black Duck helps project teams manage project information and the OSS components that are being used in each of the versions of a project.

At the project level, team members can:

- [Update the project](#) and [project version](#) information.  
This information is searchable in Black Duck.
- [Manage tags associated with the project](#).  
This information is searchable in Black Duck.
- [Create a new version of the project](#).
- [Manage project team membership](#).
- [Delete a project](#) or [project version](#).

The [Projects tab on the Dashboard page](#) lists all projects where you are a member or where you have project-group privileges. Select the name of the project to go to the *Project Name* page which displays the **Overview** tab by default.

The screenshot shows the 'Sample Project' overview page. At the top, there's a navigation bar with 'Black Duck Projects' and a 'Project' icon. Below it is a 'Create Version' button. The main area has tabs for 'Overview' (which is selected) and 'Settings'. On the left, there's a sidebar with a 'Create Version' button. The main content area has sections for 'Description' (No description), 'Created' (Aug 13, 2018 by sysadmin), 'Updated' (Aug 14, 2018 by sysadmin), and 'Tags' (No Tags). Below these sections is a table showing two project versions:

Version	Phase	Last Updated	License	Security Risk	License Risk	Operational Risk
1.0	In Planning	Aug 23, 2018	Unknown License	■ (red)	■ (red)	■ (red)
2.0	In Planning	Aug 27, 2018	Unknown License	■ (red)	■ (grey)	■ (grey)

At the bottom of the table, it says 'Displaying 1-2 of 2'.

This tab provides the following information for each version in this project:

Column	Description
N/A	Icons shown to the left of the version name: <ul style="list-style-type: none"><li>■  <a href="#">Policy violation</a>.</li><li>■  <a href="#">Policy violation has been overridden</a>.</li></ul>
Version	Name of the project version.
Phase	The development phase of this version. The possible values are:

Column	Description
	<ul style="list-style-type: none"> <li>• In Planning</li> <li>• In Development</li> <li>• Pre-release</li> <li>• Released</li> <li>• Deprecated</li> <li>• Archived</li> </ul> <p>The value in this field is used to calculate risk for the project. Archived versions are not included in project risk calculations. Click <a href="#">here</a> for more information about project version phases.</p>
Last Updated	<p>When this project version was last updated. Hover over the value to see:</p> <ul style="list-style-type: none"> <li>• When the scan mapped to this version of the project was last scanned. If there are multiple scans mapped to this version of the project, this is when any of those scans was most recently scanned.</li> <li>• When the BOM was last updated. There are several ways that the BOM could have been updated, including manual adjustments, new scans of existing code or Docker images, and newly-mapped scans.</li> </ul>
License	Name of the license for this project version.
Security Risk	<p>Bars show the high (100% red), medium (50% red), and low (100% gray) security risk levels for the OSS components in this version of the project.</p> <p>Select a bar to view the number of affected components.</p>
License Risk	<p>Bars show the high (100% red), medium (50% red), and low (100% gray) license risk levels for the OSS components in this version of the project.</p> <p>Select a bar to view the number of affected components.</p>
Operational Risk	<p>Bars show the high (100% red), medium (50% red), and low (100% gray) operational risk levels for the OSS components in this version of the project.</p> <p>Select a bar to view the number of affected components.</p>

To the right of the table, the following information is shown:

- **Description.** Description of this project. Select the **Settings** tab to create or revise the description.
  - **Created.** The user who created this project and the date it was created.
  - **Updated.** The user who last updated this project (by modifying any project information or by adding a member) and the date it was last updated.
- Updates do not include adding or modifying a project version.
- **Tags.** Any [tags](#) for this project.

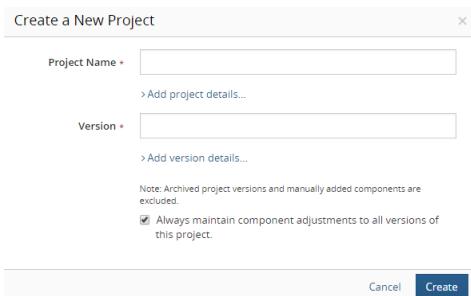
## Creating a project

A project is the base unit in Black Duck. A project can be both a stand-alone development project and part of another project. For example, Apache Tomcat is a project in its own right but it may also be part of other, larger projects. You must create the projects that you want to make available for search by other developers in your organization.

Note that a project or application is limited to 10GB of Managed Code base.

### To create a project

1. Log in to Black Duck.
2. Click **+ Create Project** at the top of any page.



3. In the Create a New Project dialog box, enter a project name. This name must be unique among projects in Black Duck, although it can have the same name as a project in the Black Duck KB.

**Tip:** As a best practice, you should think about how other users will search for your projects when creating project names. For example, if your project is related to 3D graphics, naming it "3DGraphics" means that the user must type the entire project name in order to find your project. If you use a space or an underscore in the name, for example, "3D Graphics" or "3D\_Graphics", the additional separator characters will allow users to locate the project using the search term "3D".

4. Optionally, select **Add project details** to enter additional information such as:

- Description.

**Tip:** As a best practice, you should think about how other users will search for your projects when creating project descriptions. The description should be specific about what the project does and how it is unique, so that it is easily distinguishable from other similar projects.

- Name of the project owner in the **Owner** field.

**Note:** If the user you add is not already a project member, Black Duck adds the user to the project team.

By default, the user creating the project is the project owner. The owner has the ability to assign their projects to users and groups.

- Select a tier.<sup>1</sup>
- 5. Type the version for this project in the **Version** field.
- 6. Click **Create**.

Black Duck displays the *Project Name* page.

Version	Phase	Last Updated	License	Security Risk	License Risk	Operational Risk
2.0	In Planning	Oct 22, 2018	Unknown License	<span style="width: 10%;">Low</span>	<span style="width: 20%;">Medium</span>	<span style="width: 80%;">High</span>

Displaying 1-1 of 1

**Description**  
No description.

**Created**  
Sep 4, 2018 by sysadmin

**Updated**  
Sep 4, 2018 by sysadmin

**Tags**  
No Tags

## Updating project information

Project team members can update project settings, such as the:

- Project name.
- Project description.
- Project owner.
- Tier.<sup>2</sup>
- [Ability to apply edits to all versions of a project.](#)
- [Cloning settings.](#)
- [Custom fields.](#)
- Application ID.<sup>3</sup>

### To configure project settings for a project

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the **Settings** tab.

---

<sup>1</sup>A tier lets you categorize projects in terms of importance to your company. Tier 1 projects are defined as those that are most critical to the company, where Tier 5 projects are defined as least critical.

<sup>2</sup>A tier lets you categorize projects in terms of importance to your company. Tier 1 projects are defined as those that are most critical to the company, where Tier 5 projects are defined as least critical.

<sup>3</sup>A field that can be used to store an external mapping ID for the project to an external system, such as an asset management system or application catalog.

The screenshot shows the 'Settings' tab for the 'Sample Project'. The 'Project Details' sidebar lists 'Members', 'Groups', and 'Activity'. The main area has sections for 'Project Name' (set to 'Sample Project'), 'Description' (empty), 'Owner' (a dropdown menu with the placeholder 'start typing to select owner...'), and 'Tier' (a dropdown menu with the placeholder 'select tier...'). Below these are notes: 'Note: Archived project versions and manually added components are excluded.' followed by a checked checkbox for 'Always maintain component adjustments to all versions of this project.' There's also a section for cloning attributes with checkboxes for 'Component Edits', 'Remediation Details', and 'License Fulfillment Status', all of which are checked. At the bottom, there's a note about automatic matching with a checkbox for 'Allow automatic matching [Beta]' and a 'Save' button.

5. Select **Project Details** and update the information, as needed.

**Note:** If you remove a project owner, the user remains a member of the project. If you add a project owner who is not already a project member, Black Duck adds the user as a member.

6. Click **Save**.

**Note:** You can also use the page to delete the project.

## Deleting a project

**Caution:** Once you delete a project, you cannot restore it. You can create another project with the same name, but the new project will not have any of the version or BOM information associated with the deleted project.

### To delete a project

1. Log in to Black Duck.
2. On the **Projects** tab on the Dashboard page, locate the project that you want to delete.
3. Click in the row of the project you want to delete.
4. In the Delete Project dialog box, confirm that you have selected the correct project, and click **Delete**.

The project is deleted.

## Managing tags

You can add tags to projects and custom components to describe them and provide additional metadata, such

as the programming language, frameworks, operating systems, purpose, and any other information that you think might help other users find it. Tags act as keywords when searching and filtering.

- Tags for components in the Black Duck KB have been created by the users at [The Open Hub](#).
- Tags for projects are created by project team members.
- Tags for custom components are created by users with the Component Manager [role](#).

Best practices for tagging projects:

- Use a few, specific tags rather than many tags. Tags are limited to 20 for each project or custom component.
- Tags must be at least one character long (nulls not allowed) and are limited to 50 characters in length. You can use letters and numbers to create tags.
- The only special characters supported in tags are the underscore (\_) the plus sign (+), and parentheses (). You cannot use spaces in tags.
- Do not use punctuation unless it is necessary for the tag, for example, C vs. C# vs. C++.
- Use singular nouns, for example, "server" instead of "servers."

#### To add tags to a project

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Hover over the **Tags** area of the page and click  to display the tags field.
5. Type the tag and press **Enter**.

The tag is added to the project.

#### To add tags to a custom component

1. Log in to Black Duck with the Component Manager role.
2. Click the expanding menu icon () and select **Component Management**.  
The Component Management page appears.
3. Select the name of the custom component to go to the *Custom Component Name* page.
4. Hover over the **Tags** area of the page and click  to display the tags field.
5. Type the tag and press **Enter**.

The tag is added to the custom component.

### ⚙️ To edit a tag

1. Hover over the **Tags** area of the page and click to display the tags field.
2. Select **X** next to the tag you wish to edit.
3. Type the revised text in the field and press **Enter**.

### ⚙️ To remove a tag

1. Hover over the **Tags** area of the page and click to display the tags field.
2. Select **X** next to the tag.

## Managing project team membership

Once you have been added to a project team, you can add and remove other users as team members in one of two ways:

- As users:
  - [Add users to the project team](#)
  - [Remove users from the project team](#)
- As groups, which contain several users:
  - [Add groups to the project team](#)
  - [Remove groups from the project team](#)

### ⚙️ To add users to the project team

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the **Settings** tab and then select **Members** to view the list of members for this project.

Username	First Name	Last Name	Email	Status
PolicyMgr	First	Last		Active
sysadmin			noreply@blackducksoftware.com	Active

5. Click **+ Add Member**.
6. In the Add Member dialog box, type the name of the user that you want to add. The list is type-ahead

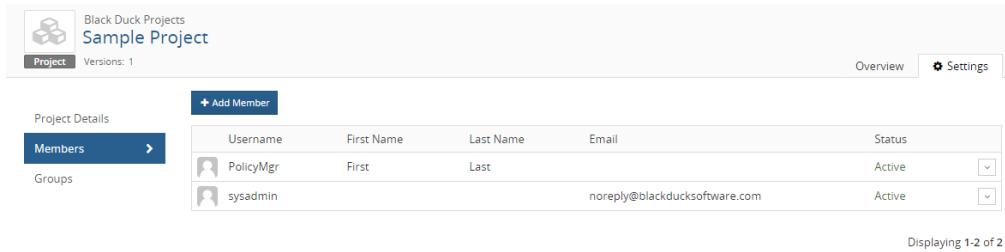
enabled, so you can see a list of available usernames that contain the text you have typed and whether those users are active.

7. Select the username to add this user to the project team.
8. Optionally, to add multiple users, type and select the name of additional users.
9. Select the [roles](#) for this user for this project.
10. Click **Add**.

The user(s) are added to the project team.

#### To remove a member from the project team

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the **Settings** tab and then select **Members** to view the list of members for this project.



Username	First Name	Last Name	Email	Status
PolicyMgr	First	Last	noreply@blackducksoftware.com	Active
sysadmin				Active

5. Click  in the row of the user you want to remove from the project team and select **Remove**.
6. In the Remove Member dialog box, click **Remove**.

The user is removed from the project team.

#### To add a group to the project team

You can manage project membership from the *Project Name* page or from the *Group Name* page.

From the *Project Name* page:

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the **Settings** tab and select **Groups** to view the list of groups for this project.

The screenshot shows the 'Groups' section of a project named 'Sample Project 1'. The table lists one group, 'Htest', with the status 'Active'. A button '+ Add Group' is located at the top left of the table area.

5. Click **+ Add Group**.
6. Type the name of the group that you want to add. The list is type-ahead enabled, so you can see a list of available groups that contain the text you have typed and whether the group is active.
7. Select the [roles](#) for this group for this project.
8. Click **Add**.

The group is added to the project team.

From the *Group Name* page:

1. Log in to Black Duck.

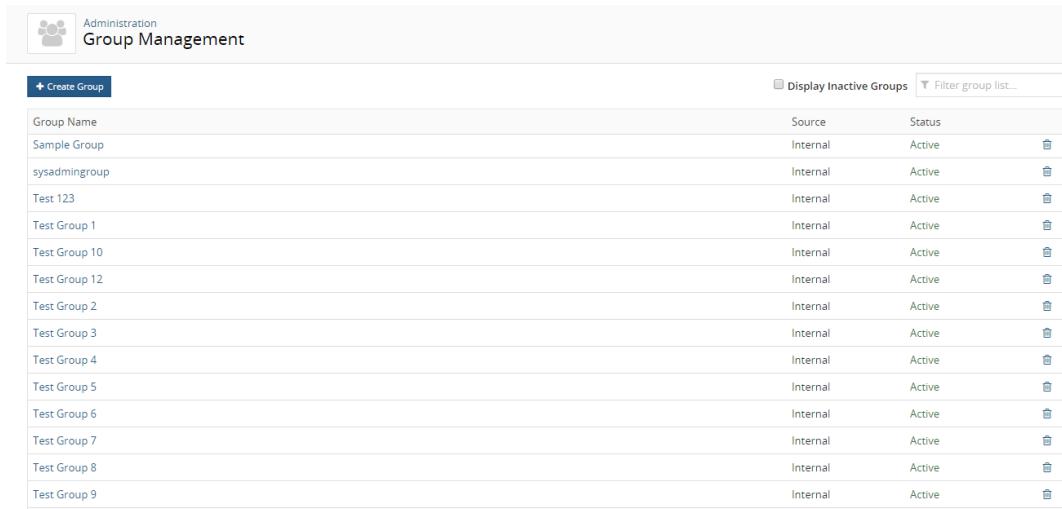
2. Click the expanding menu icon ( ) and select **Administration**.

The Administration page appears.

The Administration page is a grid of nine cards:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. Shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. Shows Using 2.55 MB out of unlimited Codebase storage, Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. Shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.

3. Select **Group Management** to display the Group Management page.



Group Name	Source	Status
Sample Group	Internal	Active
sysadmingroup	Internal	Active
Test 123	Internal	Active
Test Group 1	Internal	Active
Test Group 10	Internal	Active
Test Group 12	Internal	Active
Test Group 2	Internal	Active
Test Group 3	Internal	Active
Test Group 4	Internal	Active
Test Group 5	Internal	Active
Test Group 6	Internal	Active
Test Group 7	Internal	Active
Test Group 8	Internal	Active
Test Group 9	Internal	Active

4. Select the name of the group you want to add.

The screenshot shows the 'Administration / Group Management' section for 'Sample Group 1'.  
**Group Details:** Shows 'Group Name \*' as 'Sample Group 1' and 'Active Group' checked. Buttons for 'Delete Group' (red) and 'Save' (blue).  
**Overall Roles:** A list of roles with descriptions:

- Component Manager**: This role can create, update and delete custom components.
- Global Code Scanner**: This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project.
- Global Project Viewer**: This role has read only access to all projects.
- License Manager**: Ability to create/modify/delete licenses.
- Policy Manager**: Ability to create/modify/delete policies.
- Project Creator**: This role can create projects/versions and edit their settings.
- Super User**: This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**: This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application.

  
**Group Members:** A section with a '+ Add Member' button and a message 'No Results Found'.  
**Group Projects:** A section with a '+ Add Project' button and a message 'No Results Found'.

5. Click **Add Project** in the **Group Projects** section to display the Add Project dialog box.

6. Enter the name of the project.

7. Select the project [roles](#) for this group and click **Add**.

**To remove a group from the project team**

You can manage project membership from the *Project Name* page or from the *Group Name* page.

From the *Project Name* page:

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.

- Select the **Settings** tab and then select **Groups** to view the list of groups for this project.

The screenshot shows the Black Duck Project interface for 'Sample Project 1'. The 'Groups' tab is selected. A table lists one group: 'Htest' with 'Status' set to 'Active'. At the bottom right, it says 'Displaying 1-1 of 1'.

- Click in the row of the group that you want to remove from the project team and select **Remove**.
- In the Remove Group dialog box, click **Remove** to confirm.

The group is removed from the project team.

From the *Group Name* page:

- Log in to Black Duck.

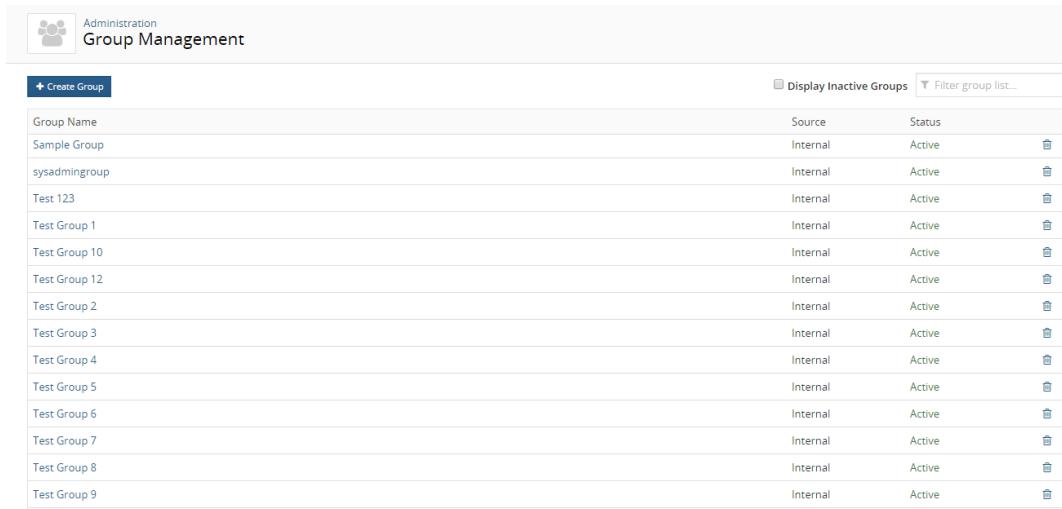
- Click the expanding menu icon ( ) and select **Administration**.

The Administration page appears.

The screenshot shows the Black Duck Administration page with various management options:

- Jobs**: View current and past jobs.
- User Management**: Manage users and roles.
- Product Registration**: Set product registration key and view licensed features.
- Group Management**: Organize users and add groups to projects.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Provide additional behaviors and functionality.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View current system configuration.

- Select **Group Management** to display the Group Management page.



Group Name	Source	Status
Sample Group	Internal	Active
sysadmingroup	Internal	Active
Test 123	Internal	Active
Test Group 1	Internal	Active
Test Group 10	Internal	Active
Test Group 12	Internal	Active
Test Group 2	Internal	Active
Test Group 3	Internal	Active
Test Group 4	Internal	Active
Test Group 5	Internal	Active
Test Group 6	Internal	Active
Test Group 7	Internal	Active
Test Group 8	Internal	Active
Test Group 9	Internal	Active

4. Select the name of the group you want to remove.

The screenshot shows the Black Duck Group Management interface. At the top left is a user icon. To its right, the path 'Administration / Group Management' and the group name 'Sample Group 1' are displayed. Below this is a 'Group Details' section with a 'Group Name \*' field containing 'Sample Group 1'. A checked checkbox labeled 'Active Group' is present. Below these are two buttons: a red 'Delete Group' button and a blue 'Save' button. The next section, 'Overall Roles', lists several role options with their descriptions:

- Component Manager**: This role can create, update and delete custom components.
- Global Code Scanner**: This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project.
- Global Project Viewer**: This role has read only access to all projects.
- License Manager**: Ability to create/modify/delete licenses.
- Policy Manager**: Ability to create/modify/delete policies.
- Project Creator**: This role can create projects/versions and edit their settings.
- Super User**: This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**: This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application.

The 'Group Members' section contains a blue '+ Add Member' button and a message 'No Results Found'. The 'Group Projects' section also contains a blue '+ Add Project' button and a message 'No Results Found'.

5. In the **Group Projects** section, click  in the row of the group you want to remove and select **Remove**.
6. Click **Remove** to confirm.

## About project versions

Use the **Details** tab to obtain information about a project version.

This tab provides the following information:

- The **Where Used** table lists the project name, project version, tier, release date, distribution, and phase for all projects where this project version is a subproject.
- To the right of the table, the following information is shown:
  - **Description**. Description of this project. Select the **Settings** tab for the project to create or revise the description.

- **Created.** The user who created this project version and the date it was created.
- **Updated.** The user who last updated this project version settings and the date it was last updated.
- **Last Scan.** Date and time the latest scan(s) mapped to this project version completed.
- **Last KnowledgeBase Update.** Date and time of the last KnowledgeBase update.
- **Tags.** Any [tags](#) for this project version.

⚙ To view the project version Details tab

1. Locate the project using the **Projects** tab on the Dashboard by selecting the name of the project to go to the *Project Name* page.
2. Select the version name which opens the **Components** tab.
3. Select the **Details** tab.

## Creating a new version of a project

When you create a project, it has one version. You can create more project versions as needed.

⚙ To add a new project version

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.

**Note:** If you wish to clone an existing version, click  in the row of the version of the project you want to clone and select **Clone**. The Create a New Version dialog box appears with the information in the **Version to Clone** field completed.

4. Click **+ Create Version**.

The Create a New Version dialog box appears.

The screenshot shows a 'Create a New Version' dialog box with the following fields:

- Version: An input field containing a single character.
- License: A dropdown menu.
- Notes: A text area.
- Nickname: A text area.
- Release Date: An input field with a calendar icon.
- Phase: A dropdown menu showing 'In Planning'.
- Distribution: A dropdown menu showing 'External'.
- Version to Clone: A dropdown menu.

A note at the bottom states: "By selecting a version to clone, any configured project cloning attributes will be applied to the new version."

At the bottom are 'Cancel' and 'Save' buttons.

5. Required. Type a name for this version of the project. This name can be a numerical release number, a text description of the version, or any combination of both.
  6. From the drop-down list in the **License** field, select the license for this project version. This value is used, for example, for the license of this project version when it is a subproject.
  7. In the **Notes** field, type any information about this version of the project that distinguishes it from other project versions, or that will be useful to other developers working on the version or searching for it.
  8. If appropriate, in the **Nickname** field type a nickname for the project version. This might be a development code name or a shortened name by which this version of the project is commonly called.
  9. If known, in the **Release Date** field, click to select the anticipated release date for the project version or the actual date on which the project version was released.
  10. From the drop-down list in the **Phase** field, select the development phase that this version of the project is currently in. The available options are:
    - In Planning (Default)
    - In Development
    - Pre-release
    - Released
    - Deprecated
    - Archived
- Note:** The value in this field is used to calculate risk for the project. Archived versions are not included in project risk calculations. Click [here](#) for more information about project version phases.
11. From the drop-down list in the **Distribution** field, select the method by which this version of the project is

being released. The available options are:

- External (Default)
- SaaS (Software as a Service)
- Internal
- Open Source

**Note:** The value in this field is used to calculate risk for the project. Project versions that are internally distributed are not included in the risk calculations for the project.

12. Specify the name of an existing project version to use as a [clone](#) for this version.

**Note:** By default, component edits, remediation details, and the license fulfillment status are cloned. Use the project's **Settings** tab to [modify these values](#).

13. Click **Save**.

Black Duck saves the project version.

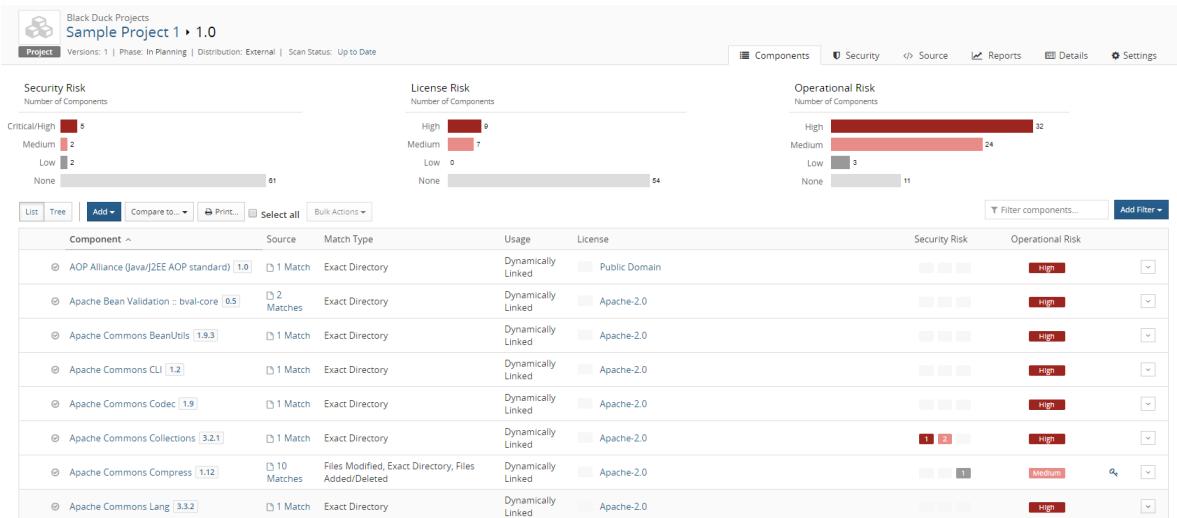
## Updating project version information

You can rename a project version and update its information.

### To update project version information

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the version name of the project that you want to manage.

The **Components** tab for the version opens.



5. Select the **Settings** tab and select **Version Details** to update the version information.

**Note:** The ability to delete a version is also available in the **Version Details** section, if there is more than one version of a project. You cannot delete a project version if that version is a [subproject in a BOM](#): you must remove the project version from all BOMs before you can delete it. Select the **Details** tab to view where this project version is used as a subproject.

- ## 6. Click **Save**.

Black Duck saves the project version information.

## Cloning project versions

When creating a new project version, Black Duck now lets you select an existing project version and clone its component edits, remediation details, and/or license term fulfillment status to the new project version. Use cloning to help reduce your workload by using the analysis and resolutions you defined in an existing project version as a baseline for a new version.

Unlike persistent edits which synchronize edits made in one version to all other versions of that project, edits made to the baseline version do not propagate to the cloned version. This gives you the ability to experiment with the cloned version while keeping the original version intact.

To successfully use cloning:

1. Enable cloning, as described below.
  2. Run a scan to the new version.

Cloned information appears in the cloned project version for components that are the *same* as in the original project version: if a component is not included in the newly scanned files, then that component *will not* be included in the new cloned project version. Cloned information *will* appear in the cloned project version for components that were manually added in the original project version.

By default, the component edits, remediation details, and the license fulfillment status are cloned:

- Component Edits:
  - Component and/or version information
  - Review flag
  - License
  - Usage
  - Ignored components
  - Comments
  - Manually added components
  - Confirmed snippet adjustments
  - Policy violation overrides and comments
- Remediation Details:
  - Remediation status
  - Target date
  - Actual date
  - Remediation comments
- License Fulfillment Status. For license terms requiring fulfillment:
  - Fulfillment status (fulfilled or unfulfilled)
  - For fulfilled license terms, the user who fulfilled the term and the date it was fulfilled

You can modify these settings by using the **Cloning Details** field in the **Project Details** section of the **Settings** tab of a project, as described [here](#).

**Note:** You cannot clone individual component or remediation values.

## Enabling cloning

You enable cloning when creating a project version when using:

- Black Duck's UI as described [here](#).
- the **--cloneFrom** parameter when using [the command line](#) to scan and create a project version.

## Deleting a project version

You can delete a version from a project.

**Note:** You cannot delete a project version if that version is a **subproject in a BOM**: you must remove the project version from all BOMs before you can delete it. Select the **Details** tab to view where this project version is used as a subproject.

### To delete a project version

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.

3. Select the name of the project to go to the *Project Name* page.
4. Click  in the row of the version of the project you want to delete and select **Delete**.
5. Click **Delete** to confirm.

Black Duck removes the project version.

## About project version phases

Projects versions include a phase which you can use to manage your development projects in Black Duck. Possible phase values are:

- In Planning
- In Development
- Pre-release
- Released
- Deprecated
- Archived

You can select the phase when [creating](#) or [editing](#) a project version. By default, a project version is in the "In Planning" phase.

**Note:** You can "lock" a project version BOM against any component and license changes from the Black Duck KnowledgeBase by select the archived phase, as described below.

## About archived project versions

You can modify archived project versions, as you would a project version in any other phase, for example, manually adding components or modifying licenses.

However, archived project versions are treated differently than all other project version phases.

- Archived project versions are excluded from project risk calculations.  
Project versions with any other phase are included in project risk calculations.
- If you enabled [persistent edits](#):
  - Your edits made to a project version are *not* propagated to archived project versions.
  - Your edits made to an archived project version are propagated to all other non-archived project versions.

Those edits are *not* applied to any other archived project version.
- Updates from the Black Duck KnowledgeBase regarding security vulnerabilities are applied to archived project versions.  
Other updates from the Black Duck KB, such as updates to license information, are *not* applied to archived project versions.

# Chapter 6: Viewing a project version's BOM

Once you have mapped a component scan or a Protex BOM to a project version, the results automatically create the [project version's BOM](#).

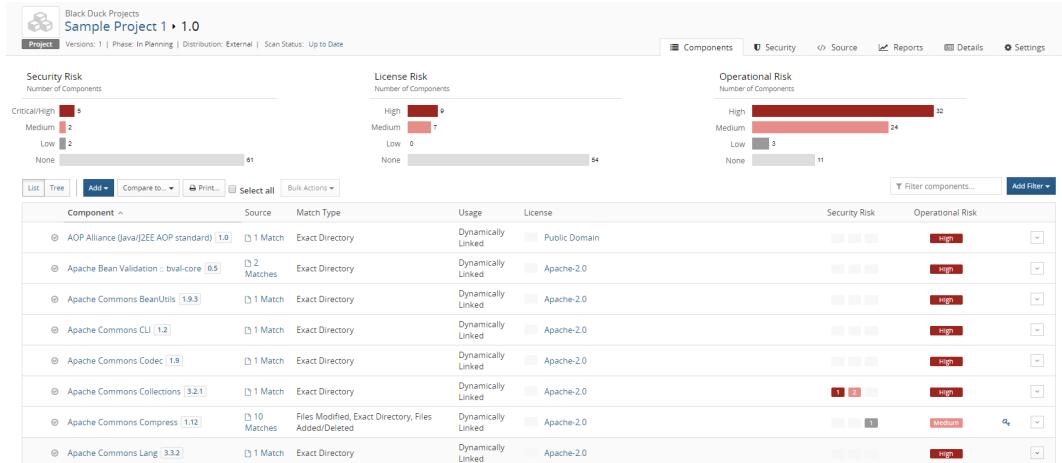
## ⚙️ To view a project version's BOM

1. Select the project using the **Projects** tab on the Dashboard.

The *Project Name* page appears.

2. Select the version that you want to view.

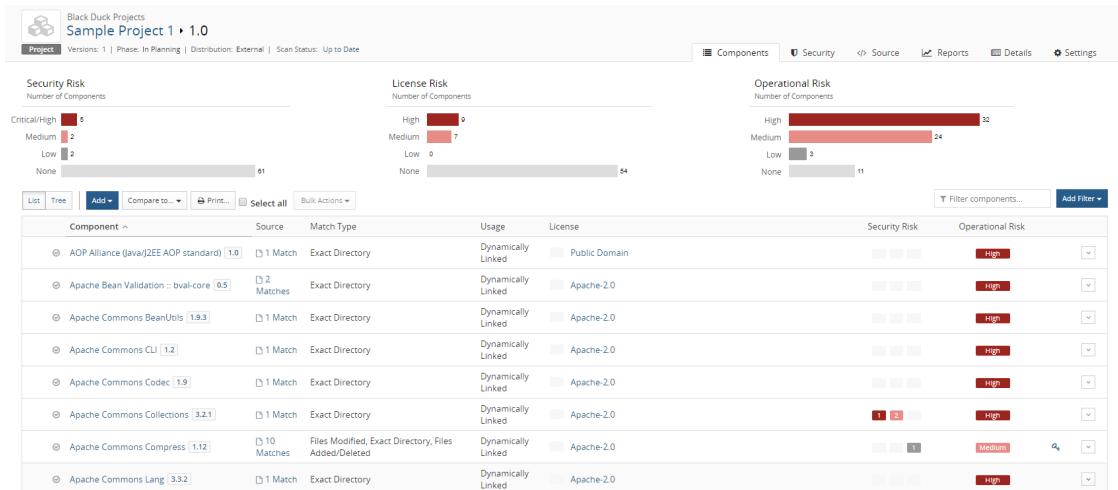
The **Components** tab shows you the BOM. The example below is what appears for a user with the BOM Manager [role](#) using the List view:



**Tip:** Refer to Black Duck online help system for information on how users with the BOM Manager, Super User, and Project Manager role can modify the project version's BOM to reflect how you are actually using the OSS components in the project.

## Understanding the information in a project version's BOM

On a project version page (from the Dashboard, select **Project** tab > **Project Name** > **Project Version**), the **Components** tab displays the BOM. The page displays risk graphs and a data table.



The above example is the list (or "flat") view of the BOM. You can also view a [hierarchical version](#) of the BOM.

## Risk graphs

At the top of the page are security, license, and operational risk graphs:

- The number displayed before the risk severity bars in the risk graphs indicates the number of components (listed in the table and in subprojects) in this BOM that have that type of risk.
  - The color of the bars in the risk graphs and in the table corresponds to the severity of risk that they represent:



- **High or Critical/High** risk: 100% red
  - **Medium** risk: 50% red
  - **Low** risk: 100% gray
  - **None**: 50% gray

To filter the table by risk category and severity:

- Select a severity label/graph to filter the table to show only those components and subprojects that have a specific type and severity of risk.
  - Use the [advanced filters feature](#) to select risk categories and severity levels.

## Data Table

The table contains the information about the components and subprojects in this version of the project.

In the component list view of the BOM, click

 located in the far-right column to [modify](#), [ignore](#), and (for manually added components), [delete](#) components or subprojects from the BOM.

When you edit a component (using the BOM or [Source tab](#)), a BOM adjustment indicator () appears in the table row to indicate that a manual adjustment was made to this component:

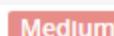
Component List							Add	Compare to...	Print...	Select all	Bulk Actions	Filter components...	Add Filter
Component	Source	Match Type	Usage	License	Security Risk	Operational Risk							
Apache Commons Logging 1.2.0	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	High								

Click  to open the Component Details dialog box which displays the edits made to this component.

Column	Description
N/A	<p>Icons shown to the left of the component or subproject name:</p> <ul style="list-style-type: none"> <li> <a href="#">Policy violation</a>.</li> <li> Policy violation in a <a href="#">child component</a>. This icon appears next to the parent component when the child components are not displayed.</li> <li> Policy violation has been <a href="#">overridden</a>.</li> <li> Policy violation has been <a href="#">overridden</a> at the <a href="#">child component level</a>.</li> <li> Component or subproject has not been <a href="#">reviewed</a>.</li> <li> Component or subproject has been reviewed.</li> </ul>
Component	<p>For <a href="#">subprojects</a>: name and version of the project.</p> <p>Select a subproject version to open the <b>Details</b> tab for this project version. This page lists the projects where this project version is included as a subproject.</p> <p>For components: name, version, and if applicable, distribution of the component in use in this version of your project.</p> <p>Components shown are top-level (parent) and subcomponents (children).</p> <ul style="list-style-type: none"> <li>Select the version number to open <a href="#">the Black Duck KB component version page</a> which displays a list of the projects and project versions in which this version of the component is used.</li> <li>Select ?, which indicates an unknown version, to open the <a href="#">Black Duck KB component page</a> which provides general information about the component.</li> <li>Mouse over the version to view the origin and origin ID.</li> </ul> <p><b>Note:</b> If a component has more than one origin for a version, the table displays the highest risk values.</p>
Source	<p>For components: Number of archives or files that match. For example:  4 Matches</p> <p>For automatic matches, the number of files that were identified in the component scan and</p>

Column	Description
	<p>matched to this version of the component appears. Select the text to open the <a href="#">Source tab</a>.</p> <p>For <a href="#">parent components</a>, this value does not include child component values.</p> <p>For subprojects: Number of components in the subproject. For example:  <a href="#">83 Components</a></p> <p>Select the value to open the BOM for this project version. The BOM only appears if you have permission to view the project.</p>
<b>Match Type</b>	<p>Indicates how the match between the component in use in this version of your project and a specific version of a project in the Black Duck KB was made.</p> <p>Possible values are:</p> <ul style="list-style-type: none"> <li>• <b>Binary</b>. Scanning identified the binaries in use within your codebase. This match type only appears if <a href="#">Black Duck - Binary Analysis</a> is enabled.</li> <li>• <b>Exact Directory</b>. Scanning identified the archive as an exact match to a directory or archive as an exact match to a directory or archive in the Black Duck KB.</li> <li>• <b>Exact File</b>. Scanning identified an exact match to a single file as an exact match to a single file in the Black Duck KB.</li> <li>• <b>File Dependency</b>. Scanning identified a match via a file dependency. Note that this match type remains for files scanned prior to version 5.0.0. For files scanned in version 5.0.0 and later, files dependencies are identified as either direct or transitive dependencies.</li> <li>• <b>Files Added/Deleted</b>. Scanning identified a fuzzy match to a component in the Black Duck KB, where some of the OSS component's files were added, deleted, or modified in the scanned archive. Sometimes this is a match to a previous or subsequent version of the component, which may have been missing from the Black Duck KB at the time that the match was made.</li> <li>• <b>Files Modified</b>. Scanning identified a fuzzy match to a component in the Black Duck KB, where some of the archive files were modified. Sometimes this is a match to a previous or subsequent version of the component, which may have been missing from the Black Duck KB at the time that the match was made.</li> <li>• <b>Direct Dependency</b>. Scanning identified a match to a component in the Black Duck KB via a direct (declared) dependency.</li> <li>• <b>Transitive Dependency</b>. Scanning identified a match to a component in the Black Duck KB via a transitive dependency.</li> <li>• <b>Manually Added</b>. The component was manually added to the BOM.</li> <li>• <b>Manually Identified</b>. An unmatched file was manually matched to a component. The table displays an icon in the table row to indicate that the component was manually adjusted.</li> </ul> <p>The following are automatic matches from an imported Protex BOM:</p> <ul style="list-style-type: none"> <li>• <b>Exact</b></li> <li>• <b>Partial</b></li> <li>• <b>File Dependency</b></li> </ul> <p>Click <a href="#">here</a> for more information.</p> <p>The match type for subprojects is <b>Manually Added</b>.</p>
<b>Usage</b>	For components: Indicates how this component is intended to be included in the project when

Column	Description
	<p>this version is released. For example, if scanning identified development tools in scanned code or a Docker image, you will want to indicate in the BOM that they will not actually be included in the released version of the project.</p> <p><b>Tip:</b> To remove components from the project version's risk calculations because they will not be released with the project, <a href="#">exclude them from the BOM</a>.</p> <p>The possible usage statuses are:</p> <ul style="list-style-type: none"> <li>• <b>Dynamically linked.</b> A moderately-integrated component that is dynamically linked in, such as with DLLs or .jar files. This is the default value.</li> <li>• <b>Statically linked.</b> A tightly-integrated component that is statically linked in and distributed with your project.</li> <li>• <b>Source Code.</b> Source code such as .java or .cpp files.</li> <li>• <b>Separate Work.</b> Intended for loosely-integrated components. Your work is not derived from the component. To be considered a separate work, your application has its own executables, with no linking between the component and your application. An example is including the free Acrobat PDF Viewer with your distribution media.</li> <li>• <b>Merely Aggregated.</b> Intended for components that your project does not use or depend upon in any way, although they may be on the same media. For example, a sample version of an unrelated product included with your distribution.</li> <li>• <b>Implementation of Standard.</b> Intended for cases where you implemented according to a standard. For example, a Java spec request that ships with your project.</li> <li>• <b>Prerequisite.</b> Intended for components that are required but not provided by your distribution.</li> <li>• <b>Dev. Tool / Excluded.</b> Component will not be included in the released project. For example, a component that is used internally for building, development, or testing. Examples are unit tests, IDE files, or a compiler.</li> </ul> <p>For subprojects, usage defaults to <b>Dynamically Linked</b>, as described above.</p>
<b>License</b>	<p>Declared license of the component or subproject in use in this version of your project.</p> <ul style="list-style-type: none"> <li>•  indicates that the component/subproject has a high license risk.</li> <li>•  indicates that the component/subproject has a medium license risk.</li> <li>•  indicates that the component/subproject has a low license risk.</li> <li>• (white box) indicates that there is no license risk.</li> </ul> <p>For known licenses, select the license name to view license details and license text.</p> <p>For <a href="#">parent components</a>, the license risk is for the parent component only.</p> <p>In the component list view, if the license text on the BOM page indicates that there is more than one license for this component version (for example the text states "Apache 2.0 and 3 more..."), hover over the license name to view the names of all licenses.</p> <p>Click <a href="#">here</a> for more information on how license risk for a component is determined.</p>
<b>Security</b>	Number of critical/high or high risk (100% red), medium risk (50% red), and low risk (100%

Column	Description
<b>Risk</b>	<p>gray) vulnerabilities associated with this version of the component or with the subproject:</p>  <p>Select a value to open the <a href="#">project version page Security tab</a> which displays the vulnerabilities for that component or subproject.</p> <p>For subprojects, the value shown is the total number of vulnerabilities for all components. Note that the values shown here may not match the values shown on the subproject version's BOM page as that lists the number of components with a vulnerability.</p> <p><b>Note:</b> If you do not have permission to view the project, you will not be able to access this page.</p> <p>For <a href="#">parent components</a>, this column shows the security risk of the parent and all of its children.</p>
	Indicates that this component version has <a href="#">encryption algorithms</a> .
<b>Operational Risk</b>	<p>Operational risk level for the component or subproject in use in this version of your project:</p> <ul style="list-style-type: none"> <li> High risk</li> <li> Medium risk</li> <li> Low risk</li> </ul> <p>The operational risk level in this version of your project is calculated using a combination of:</p> <ul style="list-style-type: none"> <li>Version status. Part of the component's operational risk calculation is based on the version of the component used compared to the number of newer versions that have been released and the time since the newest version was released. Using older versions of a component is considered risky when newer versions are available.</li> <li>Activity status. Part of the component's operational risk calculation is based on the commit activity trend for the component over the last 12 months. Increasing or stable commit activity over the time frame is considered less risky than decreasing commit activity over that time frame.</li> </ul> <p>The final operational risk will be the higher of these two risk calculations.</p> <p>In the component list view, for components, hover over the value to view the factors that determined the value shown:</p>

Column	Description
	<p><b>Operational Risk Factors</b></p> <p>In the component list view, for subprojects, hover over the value to see the number of components in this project version for each operational risk level:</p> <p><b>Note:</b> The values shown here may not match the values shown on the subproject version's BOM page. As a subproject, the value shown is the total number of components that have an operational risk. As listed on the BOM page, the operational risk values are for top-level components.</p> <p>For <a href="#">parent and child components</a>, use the component list view to hover over the value to obtain more information.</p>

## About the hierarchical BOM

By default, the BOM page displays a "flat" view of components – all components found during a scan – regardless of the directory where the component was found – are listed at the same level on the BOM page. This can make it difficult to determine where a component came from.

Black Duck provides a hierarchical view which is based on file system relationships. Use this view to see parent components and the children subcomponents which were brought in by the parent component.

**Note:** This feature is disabled by default. Refer to the installation guide for information on enabling this feature.

To view a hierarchical view of the BOM, select **Tree**,

Component	Source	Match Type	Usage	License	Security Risk	Operational Risk
ceph 13.2.0.39+geb7f429568	2 Matches	Exact File, Exact Directory	Dynamically Linked	<span>H</span> GNU General Public License v2.0 only and 5 more...	<span>Low</span>	<span>Low</span>
ceph 13.2.1	12 Matches	Files Modified	Dynamically Linked	<span>M</span> GNU Lesser General Public License v2.1 or later	<span>2</span> <span>1</span>	<span>Low</span>
+ ceph v14.0.0	104 Matches	Exact Directory, Files Added/Deleted, Files Modified	Dynamically Linked	<span>H</span> Creative Commons Attribution Share Alike 3.0 and 7 more...	<span>Low</span>	<span>Low</span>
Chart.js ?	1 Match	Direct Dependency	Dynamically Linked	<span>H</span> Unknown License	<span>Low</span>	<span>Low</span>
core.js ?	1 Match	Direct Dependency	Dynamically Linked	MIT License	<span>Low</span>	<span>Low</span>
Flot 0.8.3	2 Matches	Exact File	Dynamically Linked	MIT License	<span>High</span>	<span>High</span>

The hierarchical BOM displays parent components and those components with no child subcomponents. It also includes components found via a dependency scan.

Click **+** to view child subcomponents.

+ Apache Xerces2-j 2.3.0	1 Match	Files Modified	Statically Linked	Apache License 2.0	<span>2</span>	<span>High</span>
Apache XML Commons 1.0.02	1 Match	Files Added/Deleted	Statically Linked	Apache License 1.1	<span>Low</span>	<span>High</span>
Commons IO 1.1	1 Match	Exact Directory	Statically Linked	Apache License 2.0	<span>Low</span>	<span>High</span>
db-charmer 1.6.10	1 Match	Exact Directory	Statically Linked	MIT License	<span>Low</span>	<span>High</span>
- Mercurial Toolbar Initial Release	1 Match	Exact Directory	Statically Linked	<span>H</span> GNU General Public License v2.0 or later	<span>Low</span>	<span>High</span>
Mercurial Toolbar Initial Release	1 Match	Exact File	Statically Linked	<span>H</span> GNU General Public License v2.0 or later	<span>Low</span>	<span>High</span>

In the hierarchical view of the BOM:

- Security risk is rolled up to the parent component: parent components show the security risk of the parent and all of its children. Clicking **+** to display the child subcomponents shows the individual security risk of the parent and each child component's security risk.

License and operational risk are not rolled up to the parent component. The values shown for the license, license risk, and operational risk are for the parent component only. Click **>** to view the values for each child component.

- Policy violations for a child component appear at the parent level.
  - ✗ indicates that this component has a policy violation at the parent level.
  - ✗ indicates that a child has a policy violation. This icon appears when the child components are not displayed.



Click **+** to view the policy violations for the child components.

- ⓘ indicates that the parent or child component policy violation has been overridden.

This icon appears for child components when they are displayed in the table.



- ⓘ indicates that a child's policy violation has been overridden. This icon appears when the child components are not displayed.



- As this hierarchical view displays components based on file relationships, components that were manually added in the component list view do not appear in this view of the BOM.
- The number of archives or files that match, as identified in the **Source** column, applies to the parent component only. Click + to view the values for each child component.
- You must use the component list view to:
  - Indicate that a component has been reviewed.
  - Edit a component.
  - Ignore a component.
  - Manage (add, edit, delete) comments.
  - Override a policy violation or remove a policy override.
  - View more information on licenses and operational risk.
- If you modify the BOM using the List view, there is a delay of 15 minutes for those changes to propagate to the hierarchical version of the BOM.

## Reviewing the contents of a BOM

Any user that can edit a BOM can review the contents and indicate that a component version or subproject is correctly included in that BOM.

**Note:** Project members with no roles assigned to them cannot flag BOM contents as reviewed.

In the component list view of the BOM, next to each component or subproject name is an icon which indicates whether this item has been reviewed:

Component	
Apache Struts	2.0.4
Apache Tomcat	8.5.4

- ⓘ - Not reviewed
- ⓘ - Reviewed

Use this icon to flag component versions and subprojects as reviewed: the icon is a toggle – select it to change its status.

### To review multiple component versions or subprojects

Use the bulk review feature to indicate that all component versions and/or subprojects that appear on a *single* page are reviewed or unreviewed.

1. Optionally, filter the BOM so that the component versions and subprojects you wish to review/unreview appear on the page.
2. Select **Select all**.  
All components and/or subprojects on this page are selected.  
You can select individual rows so that they are not included.
3. From the **Bulk Actions** menu, select one of the following:
  - **Mark as reviewed** to indicate the component/subproject has been reviewed.
  - **Unmark as reviewed** to indicate the component/subproject has not been reviewed.
4. Click **Review** or **Unreview** in the confirmation dialog box.
5. Refresh the page to view your changes. It may take some time for the review status to appear.

**Tip:** To review or unreview multiple pages, repeat steps 2-5 for each additional page in the BOM.

Note:

- Hover over the Reviewed icon () to view the username of the user who reviewed this component version/subproject and the date and time when it was reviewed.
- If you selected to [apply edits to all versions](#) of a project, the review status will persist if you rescan the same code into a new project version.
- Use the filters on the BOM page to view the BOM page by review status.
- The `component.csv` file in the [Project Version report](#) includes the review status, the username of reviewers, and the review date.
- Changing the review status does not cause the [BOM adjustment indicator](#) () to appear.
- The review status cannot be changed in the [hierarchical view of the BOM](#).

## Managing comments

Comments apply to a specific component version or subproject in a BOM. For example, you can use comments to explain why a component version was ignored or why a policy violation was overridden.

Note:

- Comments are applied to a component version or [subproject](#):
  - If the component version or subproject is deleted in a BOM, the comment is deleted. If the component version or subproject is then added back to the BOM, the comment(s) will reappear.
  - If the version of a component or subproject is changed in a BOM, the comment no longer appears.
- Comments do not [persist to all versions of a project](#).

- Comments by users who become inactive still appear in the BOM.
- A component version or subproject can have multiple comments.
- The search feature is not available for comments.
- Comments cannot be added to the [Hierarchical view of the BOM](#).

## Adding a comment

1. [Display the project version BOM](#).
2. Click  in the row where you want to add a comment and select **Comment**.

The *Component/Subproject Name Version* Comment dialog box appears.



3. Enter the comment and click **Add Comment**.

A comment icon () appears in the component version or subproject row indicating a comment was added. The number shown in the icon indicates the number of comments for this component version or subproject.

Component ^	Source	Match Type	Usage	License	Security Risk	Operational Risk
 Apache Struts 2.0.4	Manually Added	Dynamically Linked	Apache-2.0	  	High	 

## Viewing a comment

Click  in the row where you want to view a comment.

## Editing a comment

Only the original writer can edit their comment.

1. Click  in the row where you want to edit a comment and select **Comment**.
2. Click  next to the comment you want to edit and select **Edit**.
3. Edit the comment, click **Update**, and then select **Close**.

## Deleting a comment

Only the original writer of the comment, BOM Manager, Super User, or Project Manager can delete a comment.

1. Click  in the row where you want to edit a comment and select **Comment**.
2. Click  next to the comment you want to delete and select **Delete**.

## Managing files associated with BOM components

Use the **Source** tab to manage the files associated with BOM components. Common cases include:

- Analyzing and identifying unmatched files. Unmatched files can be related to a component, a proprietary component, or a third-party component. Review these files to determine if they must be matched to a component version or if they can be excluded.
- Validating files that were matched to a component. Review these files to determine if they were matched to the correct component version or if they were incorrectly matched. Incorrectly matched files can be associated with the correct component version or excluded.
- [Reviewing snippet matches](#).

### Accessing the Source tab

You can access the **Source** tab to view all files in a project or automatically filtered to view specific matches.

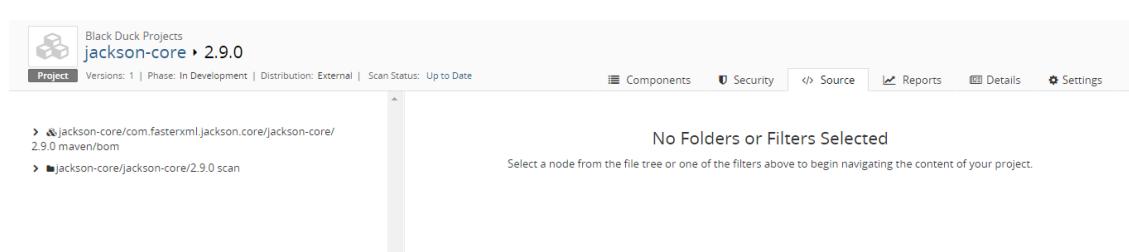
1. Select the project using the **Projects** tab on the Dashboard.

The *Project Name* page appears.

2. Select the version of the project that you want to view.

The BOM page appears.

3. Do one of the following:
  - Select the **Source** tab to view all files in this BOM.



The screenshot shows the Black Duck Project interface. At the top, there is a header bar with the project name "jackson-core 2.9.0" and various status indicators like "Versions: 1", "Phase: In Development", "Distribution: External", and "Scan Status: Up to Date". Below the header, there are several tabs: Components, Security, Source (which is highlighted in blue), Reports, Details, and Settings. The main content area is titled "No Folders or Filters Selected" and contains the message "Select a node from the file tree or one of the filters above to begin navigating the content of your project." On the left side, there is a file tree with two items: "jackson-core/com.fasterxml.jackson.core/jackson-core/2.9.0/maven/bom" and "jackson-core/jackson-core/2.9.0/scan".

Select an item in the left pane to see information in the table.

- Select a value in the **Match Count** column to view the **Source** tab filtered to that component.

Name	Component	Match Type	License	Usage
io	Apache ORO 2.0.8	Files Modified	Apache-1.1	Dynamically Linked
util	Apache ORO 2.0.8	Files Modified	Apache-1.1	Dynamically Linked

## About the Source tab

The **Source** tab consists of:

- A left pane which shows the tree structure of the files. Use this pane to navigate and select the information shown in the table.

Select an item in the left pane to display the information in the table for the selected item.

The table displays the files/directories directly under the selected item in the left pane.

Information about the selected item, such as the component name and version, path, and scan size in the area below the tree.

Click to copy the path to your clipboard.

- A table which provides the following information on the item selected in the pane.

- Name.

Hover over the name to view the path.

Select the name to filter the information shown in the table. The item you selected is also highlighted in the tree shown in the left pane.

- Component. Name and version of the OSS component in use in this version of your project.

Select the component name or version to open [the Black Duck KB component version page](#) which displays more information of the component version, such as a list of the projects and project versions in which this version of the component is used.

- Match type. Indicates how the match between the component in use in this version of your project and a specific version of a project in the Black Duck KB was made.

- License. Declared license of the component in use in this version of your project.

- Usage. Indicates how this file is intended to be included in the project when this version is released. Click [here](#) for more information on usage.

- Filters located above the table, to filter the information shown on the tab.

The tab uses the following icons:

-  Archive
-  Directory
-  or  File
-  Snippet match. Click [here](#) for more information.
-  Source file. Click [here](#) for more information.

## Modifying matches

### To modify a match

1. Open the **Source** tab as described above.
2. Do one of the following:
  - Select an item in the tree and click **Edit** located in the bottom left pane.
  - Select one or more items in the table and click **Edit** located above the table.
3. In the Edit Component (if you selected one item) or Bulk edit (if you selected multiple items) dialog box, modify the component, version, origin ID, and/or usage.  
Click [here](#) for more information about modifying snippet matches.
4. Click **Update**.

## Identifying unmatched files

1. Open the **Source** tab as described above.
2. Click  and select **Match type > Unmatched** and click **OK**.
3. Select one or more entries and click **Edit**. The Edit Component dialog box (if you selected one item) or Bulk Edit dialog box (if you selected multiple items) appears.
  - If the file is part of a component that is in use, enter the name in the **Component** field and specify a version.
  - If the file should not be included in the project, select **Dev. Tool / Excluded** from the **Usage** list.
4. Click **Update**.

 A  appears in the BOM in the row of the component you selected to indicate that a manual adjustment was made to this file. The match type changes to **Manually Identified**.

## Validating matched files

1. Open the **Source** tab as described above.
2. Click  and select **Match Type > Type of match(es)** and click **OK**.
3. Select one or more entries and click **Edit**. The Edit Component dialog box (if you selected one item) or

Bulk Edit dialog box (if you selected multiple items) appears.

- If the file was incorrectly matched to a component during the scan, enter the new name in the **Component** field and specify a version in the **Version** field.
- If the file was incorrectly matched to an origin or origin ID, specify a different value using the **Origin** and **Origin ID** fields.
- If the file should not be included in the project, select **Dev. Tool / Excluded** from the **Usage** list.

#### 4. Click **Update**.

A  appears in the BOM for this component to indicate that a manual adjustment was made to this file.

## Resetting files

You can revert manually adjusted files to their original match type.

This option is not available for unmatched files and is not enabled if the file cannot be reset.

1. Open the **Source** tab as described above.
2. Click **Add filter** and select **Adjusted**.
3. Select one or more files and click **Reset Adjustments**.  
If you select multiple files, only those files that can be reverted are reset.
4. Click **Save**.

## Deleting files from a BOM

You cannot delete files that were automatically added to a component. You can [ignore a component](#) in the BOM that contains the file so that it is not included when calculating the security, license, and operational risks for this version of your project.

To remove an automatically-added scanned component from a project version's BOM, you must remove it from your source code or Docker image and then rescan that code or Docker image. This will automatically update the project version's BOM to reflect only those component's that were automatically discovered in the mapped scans and manually added to the BOM.

To remove an automatically-added component from a Protex BOM, you must remove it in Protex and then use the Protex BOM tool to re-import the Protex BOM. This will automatically update the project version's BOM to reflect the changes in the Protex BOM.

## Comparing BOMs

Use the Project Comparison window to view the differences between two project version BOMs. You can view the differences between two versions of the same project or between two versions of different projects.

**Note:** You can only compare projects which you have permission to view.

## To view a comparison of two project version BOMs

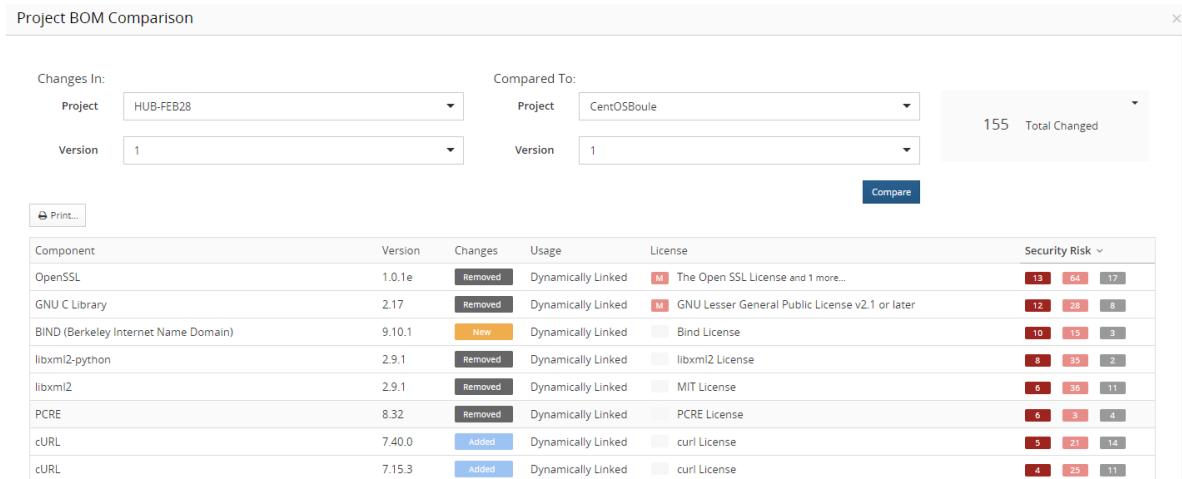
**Note:** While you can compare any two versions of a BOM for the same or different projects, this page uses the terms "current" and "compared to" to differentiate the versions.

1. Locate the project using the **Projects** tab on the Dashboard.
2. Select the name of the project to open the *Project Name* page.
3. Select the version name to open the **Components** tab and view the BOM.

This is the "current" version of the BOM.

4. Select **Compare to** and then select a different version of this BOM or select **Other project** to select a different project and version.

The Project BOM Comparison window appears.



The screenshot shows the 'Project BOM Comparison' window. At the top, there are dropdown menus for 'Changes In' (Project: HUB-FEB28, Version: 1) and 'Compared To' (Project: CentOSBoule, Version: 1). A summary box indicates '155 Total Changed'. Below this is a 'Print...' button and a 'Compare' button. The main area is a table with the following columns: Component, Version, Changes, Usage, License, and Security Risk. The table lists various components like OpenSSL, GNU C Library, BIND, libxml2-python, libxml2, PCRE, cURL, and curl, showing their status (e.g., Removed, New, Added) and security risk levels across different license categories.

Component	Version	Changes	Usage	License	Security Risk
OpenSSL	1.0.1e	Removed	Dynamically Linked	The Open SSL License and 1 more...	[13, 64, 17]
GNU C Library	2.17	Removed	Dynamically Linked	GNU Lesser General Public License v2.1 or later	[12, 28, 8]
BIND (Berkeley Internet Name Domain)	9.10.1	New	Dynamically Linked	Bind License	[10, 15, 3]
libxml2-python	2.9.1	Removed	Dynamically Linked	libxml2 License	[8, 35, 2]
libxml2	2.9.1	Removed	Dynamically Linked	MIT License	[6, 36, 11]
PCRE	8.32	Removed	Dynamically Linked	PCRE License	[6, 3, 4]
cURL	7.40.0	Added	Dynamically Linked	curl License	[5, 21, 14]
curl	7.15.3	Added	Dynamically Linked	curl License	[4, 25, 11]

At the top of the page are the projects and versions being compared. The "current" project and version of the BOM appears in the **Changes In** column.

- If you selected to compare a different version of the same project, that project name and version appears in the **Compared To** column and the table shows the comparison of the two BOMs.
- If you selected **Other project**, the table is empty; use the **Project** and **Version** fields to select the BOM to be compared and click **Compare**.

This is the "compared to" version of the BOM.

This window shows the adjustments to components or subprojects that occurred in the BOM and the associated change to the security risk. Adjustments to components consist of:

- New components/subprojects. Components or subprojects in the "current" version of the BOM that were not in the "compared to" version of the BOM.
- Updated components/subprojects. While the components or subprojects were in the "compared to" version of the BOM, one or more of the following changed:

- Component/Subproject version
  - Usage
  - License
- Removed components/subprojects. The components or subprojects that were in the "compared to" version of the BOM that are not in the "current" version of the BOM.

Note the following:

- There is only a top-level comparison of subprojects: the components in subprojects are not compared.
- If you selected to [maintain component adjustments to all versions of a project](#), the Project Comparison window may show little to no changes between versions of the same project.
- Only confirmed snippets are compared.

To view and work with the information that is important to you:

- Filter the information shown by the type of adjustment.

Select the **# New Components**, **# Removed Components**, or **# Updated Components** filters located at the top right section of the window to filter the information shown in the table.

Select **# Total Changed** to view all information. This is the default view.

- Print the information shown in the window.



1. Click . A print dialog box appears.
2. Configure the print settings and print the comparison.

Column	Description
Component	Component or subproject name.
Version	Component or subproject version.
Changes	<p>Possible values are:</p> <ul style="list-style-type: none"> <li>• <b>Added</b>. The component or subproject is in the "current" and "compared to" version of the BOM, however, it had a different version in the "compared to" version of the BOM. The version shown here is the version in the "current" version of the BOM.</li> <li>• <b>Modified</b>. The usage or license for this component/subproject version has changed.</li> <li>• <b>New</b>. The component or subproject is new – it was not in the "compared to" version of the BOM.</li> <li>• <b>Removed</b>. The component/subproject was in the "compared to" version of the BOM, however, it is not in the "current" version of the BOM.</li> <li>• <b>Replaced</b>. The component/subproject is in the "current" and "compared to" version of the BOM, however, there is a different version in the "current" version of the BOM. The version shown here is the version in the "compared to" version of the BOM.</li> </ul> <p>For modifications to ignored components:</p> <ul style="list-style-type: none"> <li>• Components ignored in both versions are not compared.</li> </ul>

Column	Description										
	<ul style="list-style-type: none"> <li>Components ignored in the "compared to" version but not ignored in the "current" version have a value of <b>Added</b>.</li> <li>Components ignored in the "current" version but not ignored in the "compared to" version have a value of <b>Removed</b>.</li> </ul> <p>Note that for a modification to the version:</p> <ul style="list-style-type: none"> <li>The component/subproject and original version are shown with <b>Replaced</b> as the value in the <b>Changes</b> column.</li> <li>The component/subproject and new version are shown with <b>Added</b> as the value in the <b>Changes</b> column.</li> </ul> <p>In the following example, the component Lucene had version 1.4.3 in the "compared to" version of the BOM and version 4.5 in the "current" version of the BOM:</p> <table border="1"> <tr> <td>Lucene</td> <td>4.5</td> <td>Added</td> <td>Dynamically Linked</td> <td>Apache License 2.0</td> </tr> <tr> <td>Lucene</td> <td>1.4.3</td> <td>Replaced</td> <td>Dynamically Linked</td> <td>Apache License 2.0</td> </tr> </table>	Lucene	4.5	Added	Dynamically Linked	Apache License 2.0	Lucene	1.4.3	Replaced	Dynamically Linked	Apache License 2.0
Lucene	4.5	Added	Dynamically Linked	Apache License 2.0							
Lucene	1.4.3	Replaced	Dynamically Linked	Apache License 2.0							
<b>Usage</b>	<p><a href="#">Usage</a> of the component or subproject version in the "current" version of the BOM.</p> <p>Strikeout usage text shows the usage for this component version from the "compared to" version of the BOM.</p>										
<b>License</b>	<p><a href="#">Declared license</a> of the component or subproject in use in the "current" version of the project.</p> <p>Strikeout license text shows the license for this component version from the "compared to" version of the BOM.</p>										
<b>Security Risk</b>	<p>Number of high risk (100% red), medium risk (50% red), and low risk (100% gray) vulnerabilities associated with this version of the component or with the subproject.</p> <p>The value in the <b>Security Risk</b> column indicates an increase or decrease in security risk depending on the value in the <b>Changes</b> column. If the value in the <b>Changes</b> column is:</p> <ul style="list-style-type: none"> <li><b>Removed</b> or <b>Replaced</b>. The value indicates a decrease in security risk from the "compared to" version of the BOM.</li> <li><b>New, Modified, or Added</b>. The value indicates an increase in security risk from the "compared to" version of the BOM.</li> </ul>										

## Printing a BOM

You can print a BOM.

The printout displays the BOM similar to what is shown in the UI: security, license, and operational risk graphs appear at the top of the page; component and subproject information is listed in a table.

You can filter the BOM prior to printing so that it only includes the data you wish to view. Any filters applied to the BOM are listed above the table.

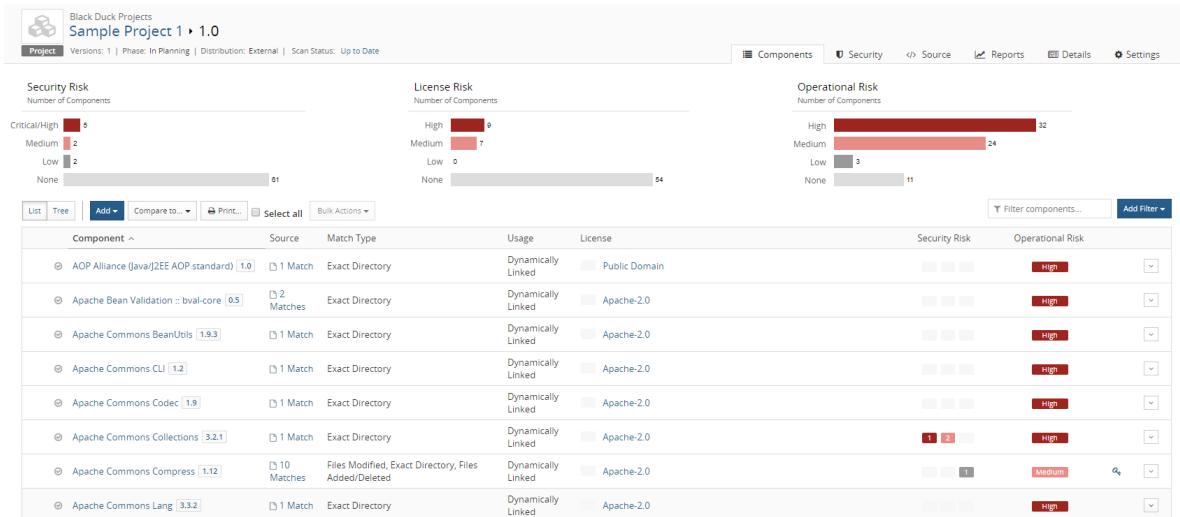
## To print a BOM

1. Select the project using the **Projects** tab on the Dashboard.

The *Project Name* page appears.

2. Select the version that you want to view.

The **Components** tab displays the BOM.



3. Optionally, filter the BOM so that the printout only shows the information you want to see.

4. Click  . A print dialog box appears.
5. Configure the print settings and print the BOM.

## Viewing issues in a project

The **Issues** tab displays the issues associated with a project version as monitored by an issue tracking product. Currently, this feature is supported using the Black Duck-JIRA plugin (version 3.3.0 and higher).

The **Issues** tab maps policy violations and optionally, security issues in JIRA (if you configured the plugin to create tickets for changes in the status of vulnerabilities) to their source in Black Duck. You can then use this tab to manage the workflow in your environment.

This tab only appears in Black Duck for a project version if the JIRA project was mapped to the Black Duck project version using the Black Duck-JIRA plugin. Once the plugin creates issues for this project, the tab appears. No additional configuration is needed. Users with the Super User [role](#) and all project members (users assigned to the project) can view the **Issues** tab for the project.

The most up-to-date status of the JIRA tickets appears in the **Issues** tab as there is bi-directional communication between Black Duck and JIRA: any changes that you make using JIRA that affect the fields

shown in Black Duck appear automatically in the **Issues** tab.

Note that the **Issues** tab does not appear if there are no issues or all issues have been deleted.

The screenshot shows the Black Duck Project Issues Management page for the project 'PSTestApp' version 1.0.0. The page displays two issues:

Component	ID	Summary	Assignee	Status	Updated
SeaMonkey 1.0.3	TEST-8	Black Duck policy violation detected on Hub project 'PSTestApp' / '1.0.0', component 'SeaMonkey' / '1.0.3' [Rule: 'No Seamonkey 1.0.3']	Not Assigned	Open	9:24 AM
SeaMonkey 1.0.3	TEST-7	Black Duck vulnerability status changes on Hub project 'PSTestApp' / '1.0.0', component 'SeaMonkey' / '1.0.3'	Not Assigned	Open	9:24 AM

Displaying 1-2 of 2

The table provides the following information for each issue:

Column	Description
Component	Component name and version affected by this ticket.
ID	Issue tracking ticket number.
Summary	Summary description for the ticket.
Assignee	User assigned to this ticket.
Status	Status of the ticket.
Updated	If updated within the past 24 hours, time when this ticket was last updated. Otherwise, date and time when this ticket was last updated.

## Viewing component versions with encryption

Open source software can use or implement cryptographic algorithms which can impact your organization from security and compliance perspective.

On the compliance side, whenever you send software out of the country – for example, on a computer, as source code, or compiled into an application that is for sale – depending upon where you live, you may be required to adhere to certain governmental regulations regarding the export of cryptography. This is especially true of strong cryptographic algorithms which may require licenses to export, however the regulations have eased in recent years.

On the security side, companies may be interested in understanding if open source is using weak cryptography or obsolete hashing mechanisms. Using a cracked (or insecure) cryptographic algorithm can add unnecessary risk to your organization, especially if well-known techniques exist to break the algorithm. Understanding algorithms in use can help companies comply with security standards.

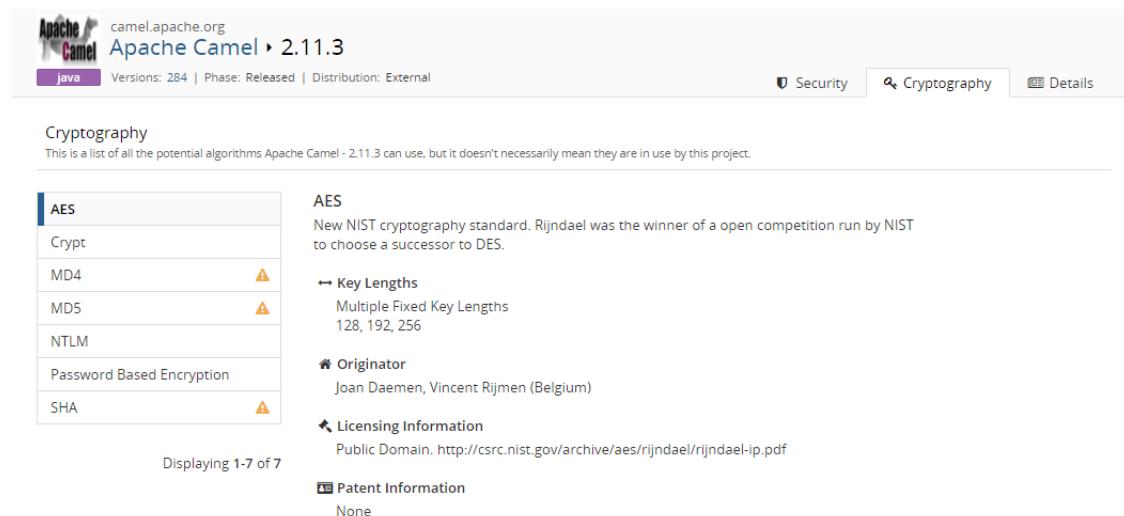
Black Duck helps you identify the component versions that have encryption algorithms.

- A cryptography filter, available in the [Component Dashboard](#) and in the component version [BOM page](#) identifies those component versions with encryption.
- A cryptography icon (

 appears in the BOM page for any component version with encryption algorithms.

Component ^	Source	Match Type	Usage	License	Security Risk	Operational Risk
Ⓐ AOP Alliance (Java/J2EE AOP standard) 1.0	3 Matches	Exact Directory	Dynamically Linked	Public Domain	Medium	High 

Select the component version to open the *Component Version* page and then select the **Cryptography** tab:



The screenshot shows the Apache Camel 2.11.3 component version page. At the top, there is a navigation bar with tabs: Security, Cryptography (which is selected), and Details. Below the navigation bar, the page title is "Apache Camel • 2.11.3". Underneath the title, it says "Versions: 284 | Phase: Released | Distribution: External". On the left side, there is a sidebar with a table listing various encryption algorithms: AES, Crypt, MD4, MD5, NTLM, Password Based Encryption, and SHA. Each algorithm has a warning symbol (⚠) next to it. The main content area is titled "Cryptography" and contains a note: "This is a list of all the potential algorithms Apache Camel - 2.11.3 can use, but it doesn't necessarily mean they are in use by this project." It then provides detailed information for each algorithm, such as descriptions, key lengths, originators, licensing, and patent information. For example, the AES entry includes a description of it being a New NIST cryptography standard, its key lengths (128, 192, 256), its originators (Joan Daemen, Vincent Rijmen), and its licensing information (Public Domain). The "Key Lengths" section also includes "Multiple Fixed Key Lengths".

- The table lists the encryption algorithms found in this component version.
- The warning symbol (⚠) indicates that this algorithm has a known weakness.
- Select an algorithm from the table to view more information, such as a description, key lengths, originator, licensing, and patent information.

Possible values for key lengths, with key length values where applicable, are:

- Single Fixed Key Length
- Multiple Fixed Key Lengths
- User Definable Key Length within a Closed Range
- User Definable Key Length Unconstrained
- No Encryption or No Key Used

Note that the **Cryptography** tab does not appear if a component version does not have encryption algorithms.

**Note:** While components added manually to existing BOMs will display cryptography information, legacy BOMs may require a rescan for cryptography data to appear.

For more information on federal regulations, visit the Bureau of Industry and Security's (BIS) website:  
<https://www.bis.doc.gov>

## About Linux distributions in Black Duck

Linux distributions combine the Linux kernel with other software, mostly open source software, to create a complete package. Black Duck reports on the vulnerabilities associated with the OSS components in these packages. However, this may lead to false positives as Linux distribution packages can be patched and these patches are not tracked by NVD.

Black Duck displays these vulnerabilities with a [remediation status](#) of "Needs Review", "Patched", or "New" (if Black Duck has verified that the vulnerability affects that version of the OSS component).

If you determine that the version of your package has been patched, you can change the remediation status to "Patched." A remediation status of "Patched" removes the CVE from the security risk calculation.

## Viewing Linux distributions in Black Duck

Black Duck shows the origin and origin ID:

- In the **Component** column when viewing details for a component on the Project Version page/**Components** tab
- In the list of components shown in the Project Version page/**Security** tab
- In the **Component** column when viewing details in the Project Version page/**Source** tab.

You can [add or edit the origin and origin ID](#) shown for a component.

Users with the appropriate [role](#) can:

- [Apply edits to all versions of a project.](#)
- [Manually add a component to a BOM.](#)
- [Exclude a component from a BOM.](#)
- [Delete a component from a BOM.](#)
- [Remove components from a BOM.](#)
- [Adjust the component and/or component version in a BOM.](#)
- [Edit an origin or origin ID.](#)
- [Ignore a component in a BOM.](#)
- [Select a different license for a component in a BOM.](#)
- [Edit license text in the BOM.](#)
- [Manage subprojects.](#)
- [Triaging snippets.](#)

## Applying edits to all versions of a project

You can select whether edits to a component apply to a specific version of a project or if edits are persistent – they apply to all versions of a project. If you select to make edits persistent then edits apply to all existing versions of a project, excluding [archived versions of projects](#) and manually added components, and will also be carried forward as additional scans are completed at the same code or Docker image.

For example, if you edit a matched component to a different component, then all other versions of the project that have that same matched component will have the match adjusted and all versions going forward will also have this match adjusted.

**Note:** There are instances when edits may not propagate to all versions. See [Persistent edit examples](#) below.

Persistent edits are enabled by default when you [create a project](#).

**Note:** Projects created prior to release 3.1.0 will have this feature disabled by default. See the examples described below as those results will apply if you enable this feature to those projects.

When you edit a component (using the BOM or Files page), a  appears in the table row to indicate that a

manual adjustment was made to this component:

Component List		Add	Compare to...	Print...	Select all	Bulk Actions	Filter components...		Add Filter
Component ^	Source	Match Type	Usage	License	Security Risk	Operational Risk			
Apache Commons Logging 1.2.0	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	High				

**Note:** A appears on the BOM page for any edits that you make to a BOM.

There is also the option of [cloning project versions](#) which enables you to baseline a project version.

## Persistent edit examples

Edits may appear to work differently than expected depending on the status of persistent edits and when the edits are made.

In the examples below, a project has several versions, none of which are archived.

Example	Final Result
<p>1. Persistent edits are enabled.</p> <p>2. An edit is made to an item in a component in one version of the project.</p> <p>For example, the license for Component A is changed in Version 1 of the project.</p> <p>The edit is propagated to all versions of the project.</p> <p>3. Persistent edits are then disabled.</p> <p>4. An edit is made to the <i>same item</i> in Component A in a version of the project.</p> <p>For example, the license for Component A is changed in Version 1 (or Version 2) of the project.</p>	Although persistent edits are disabled, the edit is propagated to <i>all versions</i> of the project as the original edit was made when persistent edits were enabled.
<p>1. Persistent edits are disabled.</p> <p>2. An edit is made to an item in a component in one version of the project.</p> <p>For example, the license for Component A is changed in Version 1 of the project.</p> <p>The edit appears in only Version 1 of the project.</p> <p>3. Persistent edits are then enabled.</p> <p>4. An edit is made to the same item in the same component in the same version.</p> <p>For example, the license for Component A is changed again in Version 1 of the project.</p>	The edit is applied to only that version of the project (Version 1 in our example). The edit does not propagate to other versions of the project as the original edit was made when persistent edits were disabled.
<p>1. Persistent edits are disabled.</p> <p>2. An adjustment is made to an item in a component in one version of the project.</p> <p>For example, the license for Component A is changed in Version 1 of the project.</p> <p>The edit appears in only Version 1 of the project.</p> <p>3. Persistent edits are then enabled.</p> <p>4. An adjustment is made to the same item in a component in a different version of the project.</p> <p>For example, the license for Component A is changed in Version 2 of the project.</p>	The edit is propagated to all versions <i>except</i> Version 1.

## Enabling or disabling persistent edits for a project

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to view the *Project Name* page.
4. Select the **Settings** tab.

Black Duck Projects  
Sample Project  
Versions: 1

Overview Settings

Project Details > Settings

Members      Project Name: Sample Project

Groups      Description:

Activity

Owner: start typing to select owner...

Tier: select tier...

Note: Archived project versions and manually added components are excluded.  
 Always maintain component adjustments to all versions of this project.

Select the attributes you'd like to clone for any new versions of this project.  
 Component Edits  
 Remediation Details  
 License Fulfillment Status

Would you like this project to be automatically matched during scans?  
 Allow automatic matching [See](#)

Save

5. Do one of the following in the **Project Details** section:
  - Select **Always maintain component adjustments to all versions of this project** to enable persistent edits.
  - Clear **Always maintain component adjustments to all versions of this project** to disable persistent edits.
6. Click **Save**.

## Manually adding a component to a BOM

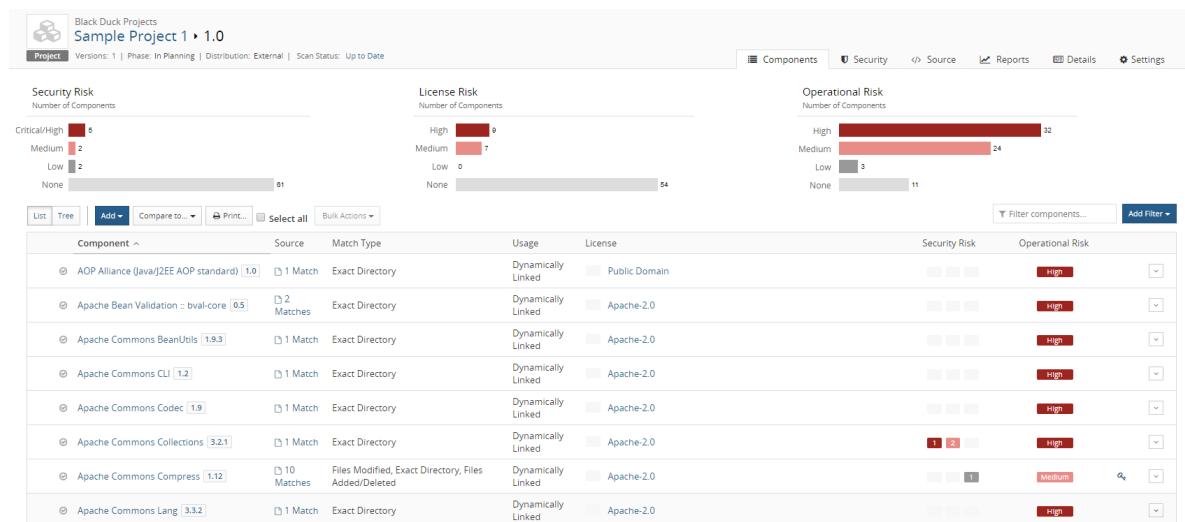
Once you have mapped a component scan to a project version, the scan results automatically populate the project version's BOM with the discovered components. Although the BOM contains all the components discovered in the mapped scan, there may be other components that you are using in that version of your project that either were not discovered in one of the mapped scans or were not scanned.

You can manually add components to the project version's BOM so that they are included in all project version information and risk calculations. You must manually add the component to the BOM of each version of the project in which you use it. You cannot manually add a component to the BOMs of multiple versions of a project at once.

**Note:** If a subsequent component scan automatically updates the project version's BOM to reflect the discovered OSS components that are included in the BOM, any OSS components that you have manually added to the BOM will be unaffected by that update. Components that are added to the BOM manually can only be [deleted from the BOM](#) manually.

### ⚙️ To manually add a component to a BOM

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the version name to open the **Components** tab.



5. Click **Add** and select **Component** to open the Add Component dialog box.
6. Enter the name of the component that you want to add.
7. Optionally, enter or select a version and an origin ID.
8. Optionally, select **Advanced Attributes** and do the following information:
  - Enter the purpose for adding this component.
  - Select **Modification** if you modified this component and optionally, enter information regarding the modification.
9. Click **Save**.
  - Black Duck adds the component to the project version's BOM. An icon appears in the row of the manually added component if you entered a purpose or you specified that you modified the component and entered information regarding the modification.
  - The **Match Type** column indicates that the component was added to the project version's BOM manually.

(Manually Added).

- All vulnerability data, license information, version age information, and project development activity information for the component that you added to the BOM is pulled from the Black Duck KB and used to update the security, license, and operational risks for this version of your project.

## Excluding a component from a BOM

A component's usage indicates how it is intended to be included in the released version of the project.

The usage statuses are:

- Dynamically Linked
- Statically Linked
- Source Code
- Separate Work
- Implementation of Standard
- Merely Aggregated
- Prerequisite
- Dev. Tool / Excluded

Click [here](#) for more information on usage.

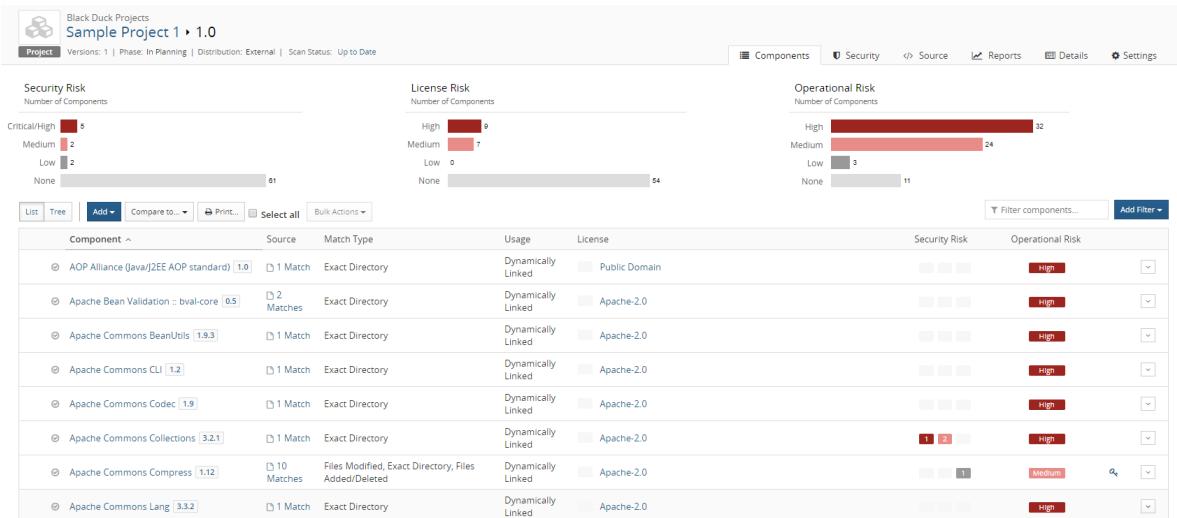
You can change a component's usage to indicate that it is not included in the project version's BOM because it is not actually being distributed with the released project version. For example, if scanning identified development tools in scanned code or a Docker image mapped to the project version, but they will not actually be included in the released version of the project, you should change their usage to exclude them from the project version's BOM.

**Note:** If you choose to exclude an automatically-added component from a project version's BOM, it will continue to be excluded even if the code or Docker image where it was discovered is rescanned and the BOM is updated.

**Important:** When you exclude a component from a project version's BOM, the license associated with that component *is not considered* when [calculating the project version's license risk](#). The security and operational risks associated with an excluded component are *still considered* when calculating the project version's security and operational risk.

### To exclude a component from a project version's BOM

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the version name to open the **Components** tab and view the BOM.



5. In the component list view of the BOM, click  and select **Edit** to open the Edit Component dialog box.
  6. Select **Dev. Tool / Excluded** from the **Usage** list,
  7. Optionally, enter a purpose for this change and/or select the **Modification** checkbox and enter information regarding this modification in the field.
  8. Click **Save**.

**Tip:** You can change the [matched component and version](#) and [license](#) at the same time as you change the OSS component's usage.

## Deleting a component from a BOM

If you added a component manually to a project version BOM, you can delete it so that it is no longer included in the project version information and risk calculations.

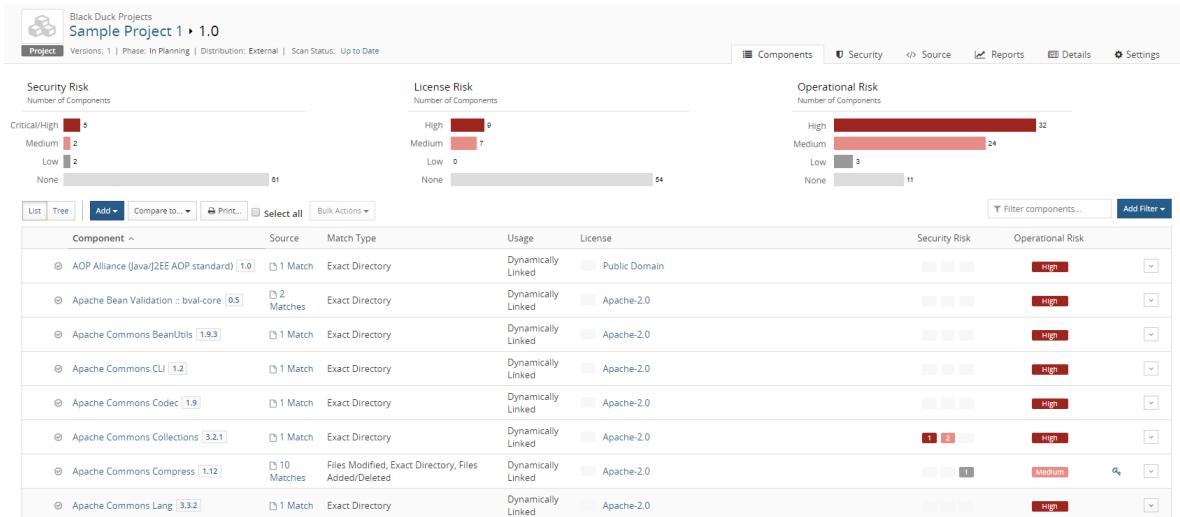
Common reasons to delete a component that was added manually include:

- The same component was discovered in a later component scan and automatically added to the BOM.
  - The component version that you selected when you added it was not the correct version.
  - You are no longer using component in that project version.

**Caution:** You cannot manually delete components that were automatically added to a project version's BOM. You can [ignore an automatically-added component in the BOM](#) so that it is not included when calculating the security, license, and operational risks for this version of your project. If you want to completely remove an automatically-added component from a project version's BOM, you must remove it from your source code or Docker image and then rescan. This will automatically update the project version's BOM to reflect only those component's that were automatically discovered in the mapped scans and manually-added to the BOM.

### To delete a component that was added manually

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the version name to open the **Components** tab.



5. In the List view of the BOM, click  and select **Delete** to open the Delete Component dialog box.
6. Click **Delete**.

The BOM is updated and the risk is recalculated.

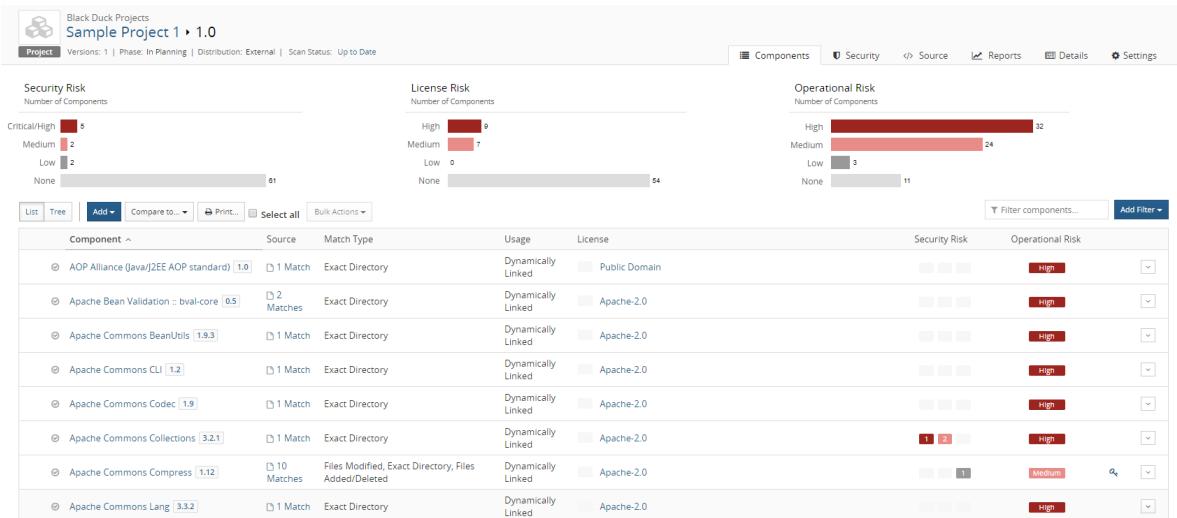
## Removing components from a BOM

The best way to remove components that were automatically added to a component version BOM is to remove the link between the component version and the scan that discovered those components.

**Note:** If you manually remove automatically-added components from a project version BOM, those components will be automatically added to the project version BOM again if the code or Docker image is rescanned.

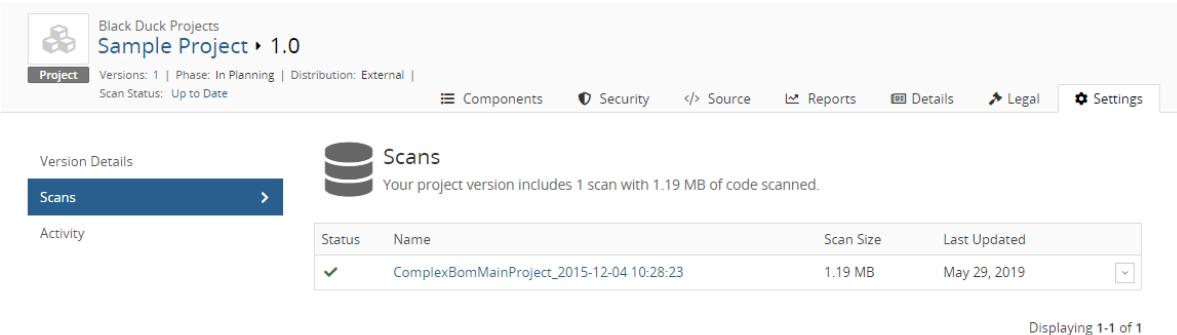
### To remove a scan from a project version to update the BOM

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the version name to open the **Components** tab and view the BOM.



## 5. Select the **Settings** tab and then select **Scans**.

Select the name of the scan to display the *Scan Name* page which provides information such as the projects and versions mapped to this scan.



## 6. Click in the row of the scan you want to remove the link (unmap) and then select **Unmap from Project**.

Black Duck removes the mapping between the scan and the project version. This removes all OSS components discovered in that scan from the BOM.

## Ignoring a component in a BOM

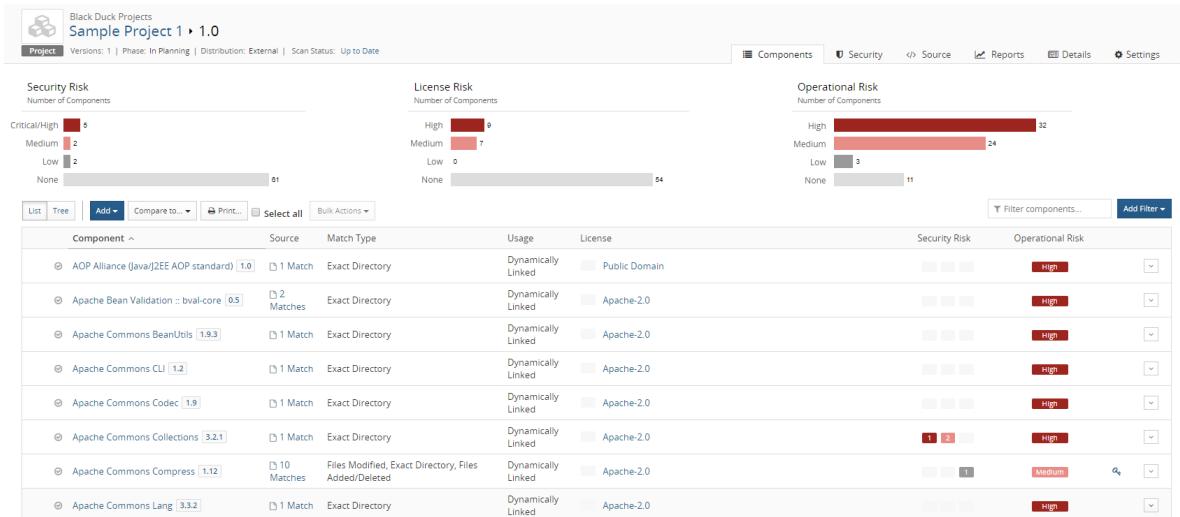
You ignore an OSS component in the BOM of a project version so that any associated risks are excluded from the risk calculations.

**Note:** If you ignore an automatically-added OSS component from a project version BOM, it will continue to be ignored even if the code where it was discovered is rescanned to update the BOM.

**Note:** You cannot ignore manually added components.

## ⚙️ To ignore a component in a project version BOM

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to open the *Project Name* page.
4. Select the version name to open the **Components** tab and view the BOM.



5. In the List view of the BOM, click and select **Ignore** to open the Ignore Component dialog box.
6. Click **Ignore**.

The component is ignored when calculating project version risk and is not displayed in the BOM.

## ⚙️ To view ignored components

1. While viewing the BOM using the component list view, select **Ignore** from the **Add filter** list.
- A list of filters appears.
2. Select **Ignored** and click **OK**.

The table displays all ignored components.

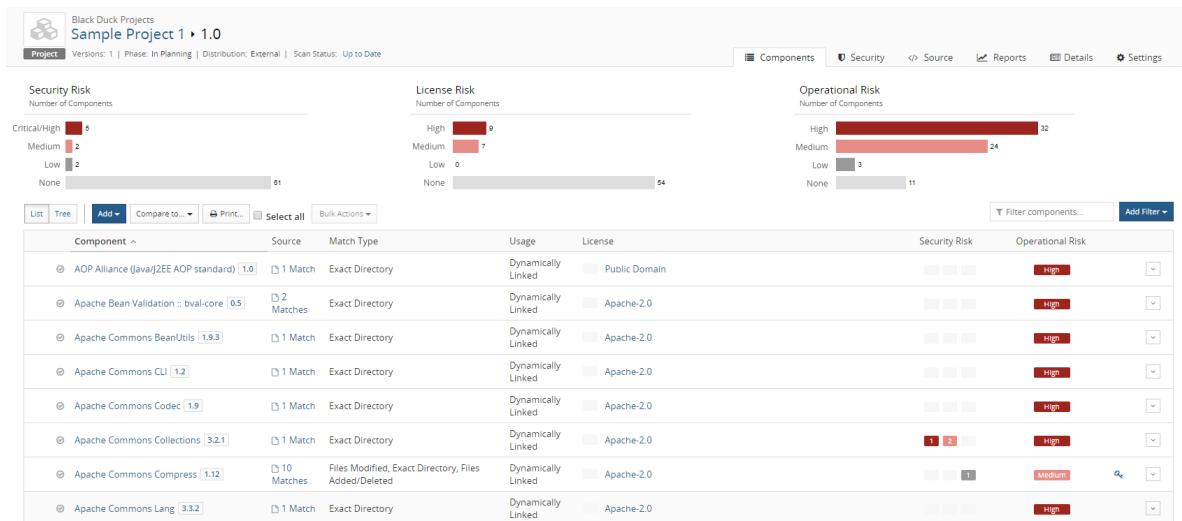
## Adjusting the component and/or component version in a BOM

Once you have mapped a component scan to a project version, the scan results automatically create the project version's BOM. Although component scanning automatically discovers the open source component and component version from most archive files by comparing them to components in the Black Duck KB, you may be using a version of the component that is not available in the Black Duck KB, or you may be using a modified version of a component. You can adjust the component and version for a component in a BOM.

- If the component/version is available in the Black Duck KB, users with the appropriate role can adjust the component or component version, as described below.
- If the component version of a component is not available in the Black Duck KB, users with the Component Manager role can create a custom version and add it to the BOM.

### To select an alternate component and/or version match for a component in a BOM

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the version name to open the **Components** tab and view the BOM.



The screenshot shows the 'Components' tab for 'Sample Project 1'. At the top, there are three risk bar charts: 'Security Risk' (Critical/High: 9, Medium: 2, Low: 2, None: 81), 'License Risk' (High: 9, Medium: 7, Low: 0, None: 54), and 'Operational Risk' (High: 32, Medium: 24, Low: 3, None: 11). Below the charts is a table listing components:

Component	Source	Match Type	Usage	License	Security Risk	Operational Risk
AOP Alliance (Java) AOP standard 1.0	1 Match	Exact Directory	Dynamically Linked	Public Domain	Medium	Medium
Apache Bean Validation : bval-core 6.5	2 Matches	Exact Directory	Dynamically Linked	Apache-2.0	Medium	Medium
Apache Commons BeanUtils 1.9.3	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	Medium	Medium
Apache Commons CLI 1.2	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	Medium	Medium
Apache Commons Codec 1.9	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	Medium	Medium
Apache Commons Collections 3.2.1	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	Medium	Medium
Apache Commons Compress 1.12	10 Matches	Files Modified, Exact Directory, Files Added/Deleted	Dynamically Linked	Apache-2.0	Medium	Medium
Apache Commons Lang 3.3.2	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	Medium	Medium

5. In the component list view of the BOM, click  and select **Edit** to open the Edit component dialog box.
6. Type the name of the OSS component in the **Component** field, and select the alternate match.
7. Select the version of the component from the **Version** list. The list contains all versions of the component that are available in the Black Duck KB.
8. Optionally, enter a purpose for this adjustment and/or select the **Modification** checkbox and optionally, enter information regarding this modification in the field.
9. Click **Save**.

The component and version for the BOM entry are updated. The BOM adjustment indicator () appears in the table row to indicate that the component and/or version were changed from the one automatically discovered in the component scan:



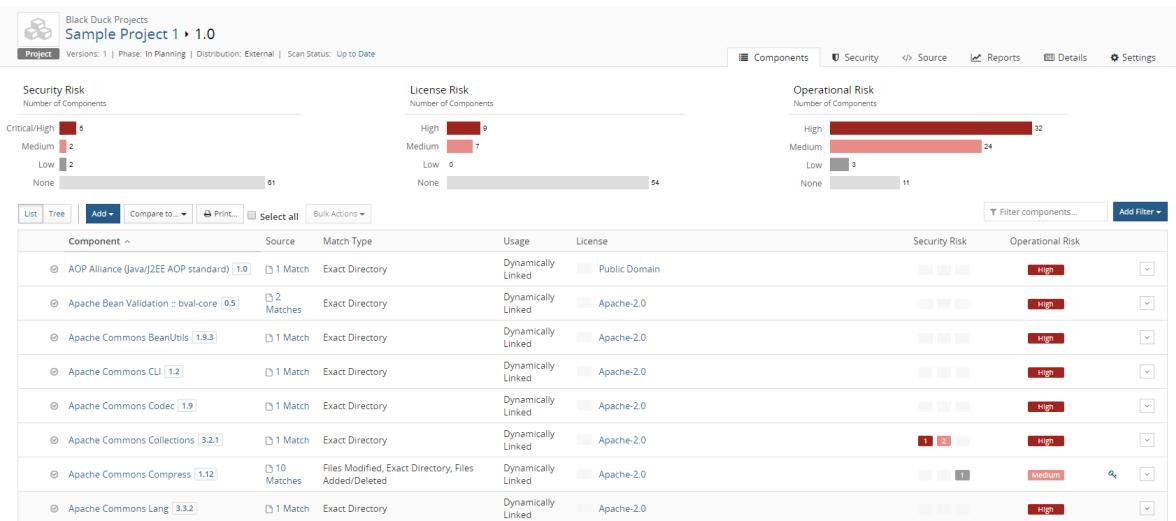
The screenshot shows the 'Components' tab for 'Sample Project 1'. A single row for 'Apache Commons Logging 1.2.0' is highlighted with a blue border. The rest of the table structure is identical to the previous screenshot.

## Editing an origin or origin ID

You can select a different origin or origin ID shown for a Linux distribution and used in a project version's BOM.

### To select a different origin or origin ID

1. Log in to Black Duck.
2. From the Dashboard, select the **Projects** tab.
3. Select the name of the project to go to the *Project Name* page.
4. Select the version name to display the **Components** tab and view the BOM.



5. In the component list view of the BOM, click  and select **Edit** to open the Edit component dialog box.
6. If the component you selected does not have a distribution, the **Origin ID** lists do not appear. If necessary, select a different component and version to display the **Origin ID** lists.
7. Select the name of the distribution and then the version from the **Origin ID** lists.

**Tip:** You can edit the matched component and version, license, and usage at the same time as you change the origin and origin ID.

8. Optionally, enter a purpose for this adjustment and/or select the **Modification** checkbox and enter information regarding this modification in the field.
9. Click **Save**.

The origin and/or origin ID is updated. If the new values carry a different type of risk than the previous one, the security risk calculations for the OSS component and for the project version are updated.

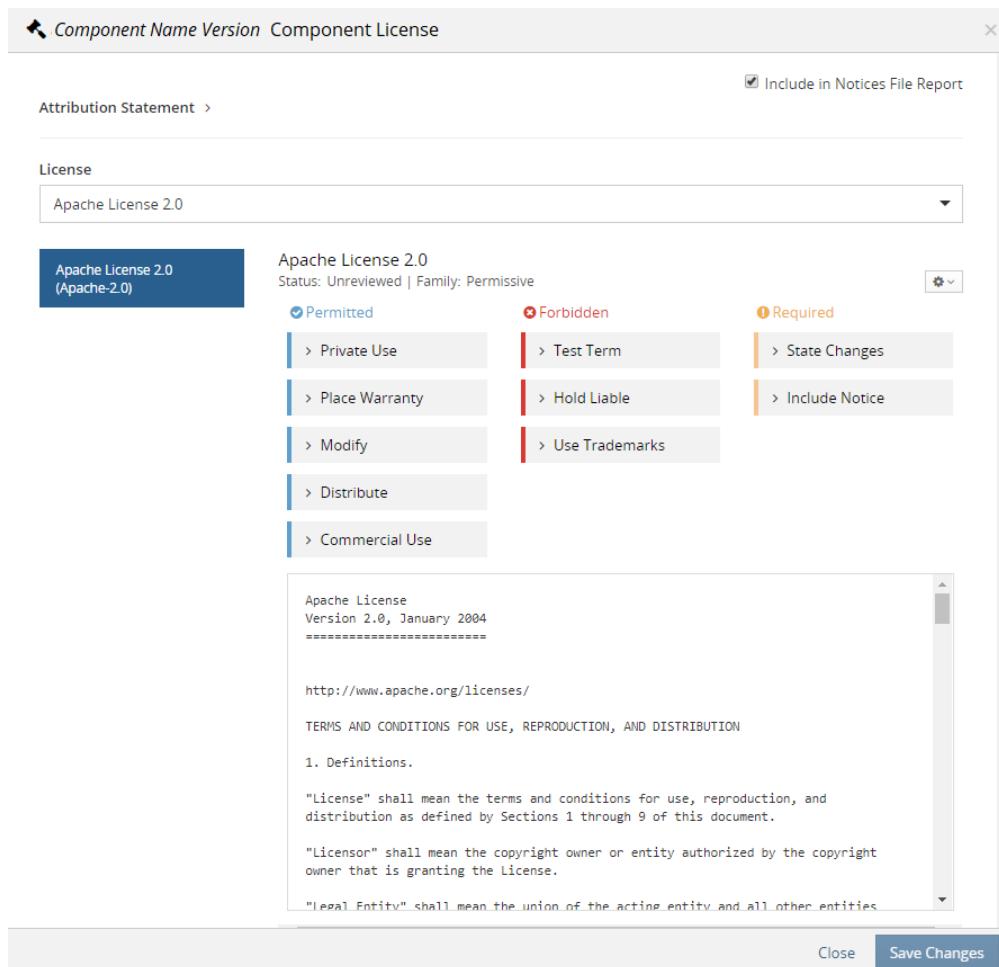
## Selecting a different license for a component in a BOM

You can select a license for a component used in a BOM that is different from the component's declared

license that is identified in the Black Duck KB.

✿ To select a different license for an OSS component in the project version's BOM

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the version name to open the **Components** tab and view the BOM.
5. Select the existing license to open the *Component Name Version Component License* dialog box.



6. Backspace to clear the field and then type the name of the license that you want to assign, and from the list of suggestions, select the one you want.
7. Click **Save Changes**.

## Selecting the license term fulfillment status

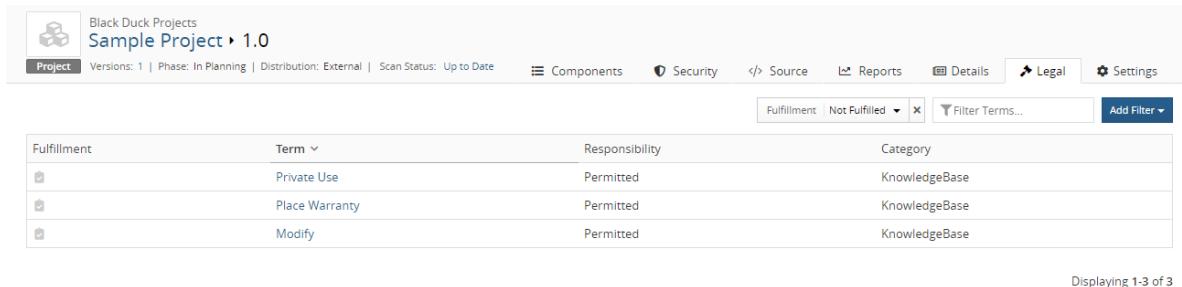
After a License Manager has defined the license terms that must be fulfilled and the system administrator has

enabled the **Legal** tab, BOM Managers, and other users with the appropriate [role](#), can denote the fulfillment status for a license term by using the *Project Version Legal* tab.

By default, the fulfillment status of a license term is unfulfilled.

 To change the fulfillment status for a license term:

- From a project version BOM, select the **Legal** tab to view a list of license terms that require fulfillment.



The screenshot shows the Black Duck Project interface with the 'Sample Project' selected. The 'Legal' tab is active. A table lists three license terms:

Fulfillment	Term	Responsibility	Category
	Private Use	Permitted	KnowledgeBase
	Place Warranty	Permitted	KnowledgeBase
	Modify	Permitted	KnowledgeBase

Displaying 1-3 of 3

By default, the **Legal** tab is filtered to show all license terms that are not fulfilled.

The tab displays the following information:

Column	Description
<b>Fulfillment</b>	Indicates fulfillment status: <ul style="list-style-type: none"> <li> indicates this license term is not fulfilled.</li> <li> indicates this license term is fulfilled.</li> </ul>
<b>Term Name</b>	License term name. Select the term to display the Term Fulfillment dialog box from which you can manage the fulfillment status for all licenses that have this term.
<b>Responsibility</b>	Indicates the responsibility for this term. Possible values are Required, Forbidden, or Permitted.
<b>Category</b>	<a href="#">Category</a> for this license term.

- Select a license term to view all licenses with this license term in this BOM which require fulfillment.

The Term Fulfillment dialog box appears.

The screenshot shows a dialog box titled "Term Fulfillment". At the top, there are four tabs: "Term", "Source", "Responsibility", and "Category", with "Term" selected. Below these are two buttons: "Mark as fulfilled" and "Mark as unfulfilled". To the right is a search bar with dropdown filters for "Fulfillment" (set to "Not Fulfilled") and "Category" (set to "KnowledgeBase"). A "Filter Components..." button and an "Add Filter" button are also present. The main area is a table listing components with columns: Fulfillment, Component, License, and Last Updated. The table contains five rows, each with a checkbox icon and a checkmark icon next to the component name. The components listed are Commons IO 1.1, Apache-Jakarta Jmeter 2.1.1, Apache-Jakarta Jmeter 2.0.3, Apache Lucene 1.4.3, and Apache Commons FileUpload 1.3.3, all under the Apache License 2.0. A message at the bottom indicates "Displaying 1-5 of 5". At the bottom right of the dialog is a "Close" button.

This dialog box lists the component name and version, license that includes this term, and the username and date that this license term was last updated.

- indicates this license term is not fulfilled.
- indicates this license term is fulfilled.

3. Select one or more checkboxes to denote the fulfillment status.

To select all terms on a page, select located at the top of the table.

4. Select **Mark as fulfilled** to indicate this license term is fulfilled or **Mark as unfulfilled** to indicate this license term is unfulfilled.

5. Click **Close**.

## Editing license text in the BOM

You may notice that the license text for some components is incomplete as the Black Duck KB may not have the full license text for some components. Since most attribution clauses in licenses usually require at a minimum that the license text be provided in any redistributions, you may need to edit the existing license text.

Note the following:

- Edits to license text only apply to the license text for that component version: edits do not apply to other components with the same license.
- If you selected to make edits persistent then edits to license text apply to all existing versions of a project

and will also be carried forward as additional scans are completed for the same code or Docker image.

- There is an option to revert to the original license text.
- The dialog box displays the first and last name and date or time the license text was edited above the license text.

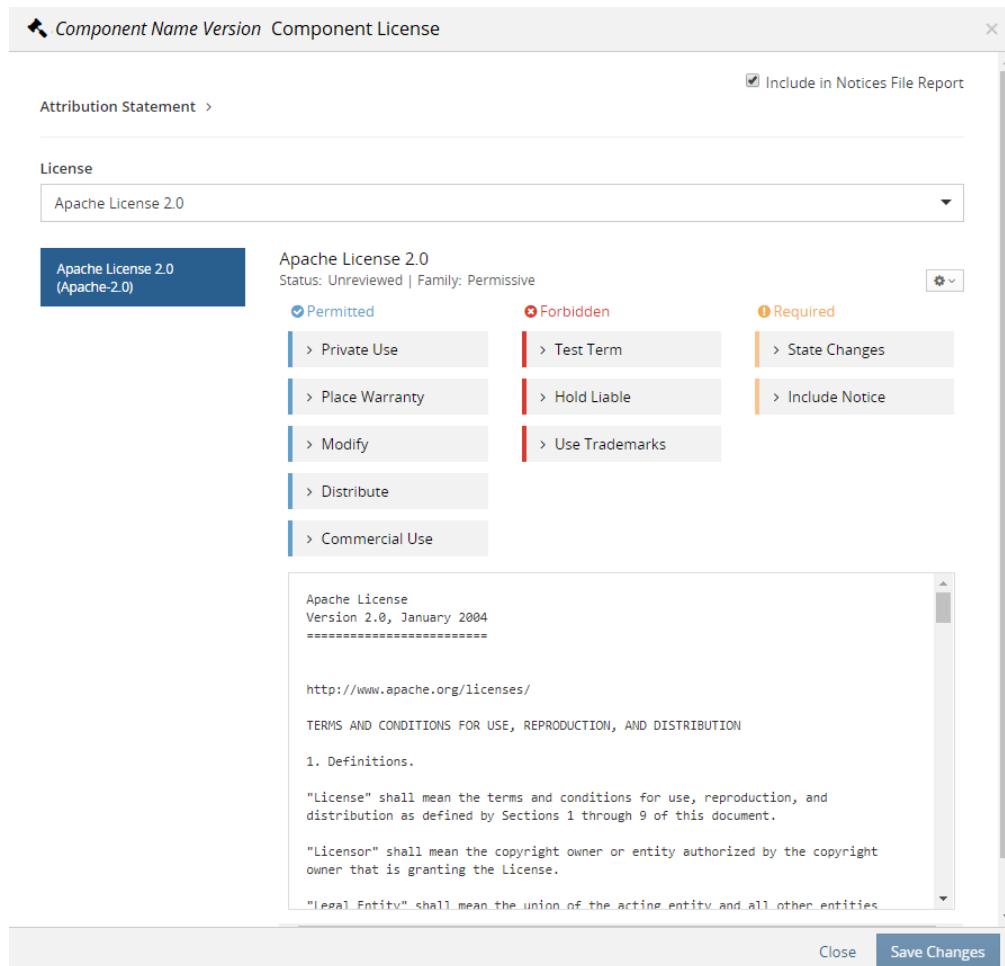
Updated by System Administrator - 11:16 AM 

This message appears for local or [global edits](#) (made by the License Manager).

- If you edited the original license, saved the changes, selected a different license, and then select the original license, your edited version of the license will appear.
- [Edits made globally to licenses](#) by the License Manager will propagate to the version used in the BOM unless the BOM Manager, Super User, or Project Manager has edited the license,

#### To edit license text

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the version name to open the **Components** tab and view the BOM.
5. Select the license name to open the *Component Name Version Component License* dialog box.



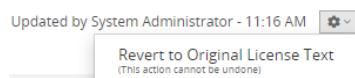
6. If there is more than one license for this component, select the license you wish to edit.

The license text appears in a field in the dialog box.

7. Edit the text directly in the field.
8. Click **Save Changes**.

#### To revert to the original license text

1. Open the *Component Name Version Component License* dialog box as described above.
2. Click located above the license text and select **Revert to Original License Text**.



## Managing subprojects

You may have applications that include code from other projects, for example, a user management module

that is included in several other applications. You can see risk information about the user management module as a project with its own BOM but may also want to see the same information in the BOM for every application that uses that module without having to re-scan the code.

Adding projects to your application's BOM gives you a complete view of this application and all associated risks, including vulnerabilities, license, and operational risk.

For these subprojects:

- You must have permission to the project to add it to the BOM.
- Users who do not have permission to the subproject will not be able to drill down to view additional data about that project version.
- Modifications made to a project outside of the BOM will propagate to the subproject in the BOM. For example, if additional scans are completed for scans mapped to this project, those changes will propagate to the subproject.

An exception to this is the subproject version license: edits made to the project version license may or may not propagate to the license shown for the subproject in the BOM:

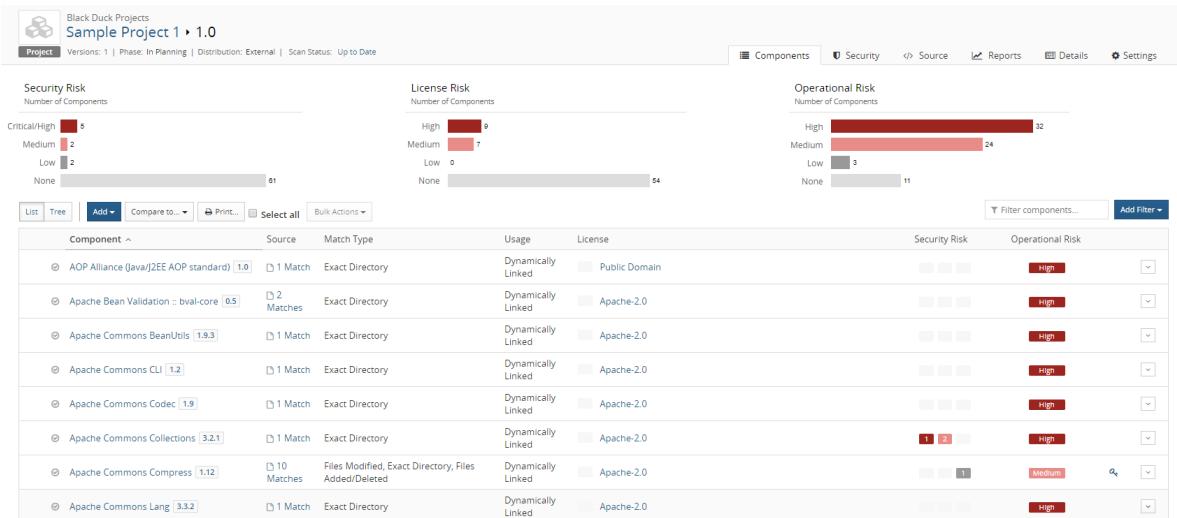
- If you modify the project version license outside of the BOM and have *not* edited the subproject license from within the BOM, the edited license will appear in the BOM for the subproject.
- If you modify the project version license outside of the BOM and have edited the subproject license from within the BOM, the license edit will *not* appear in the BOM for the subproject.

If you modify the subproject version license from within the BOM, that change is *not* propagated outside of the BOM.

- Policy violations within the subproject will not appear in the BOM. However, a policy violation will appear in the BOM for the subproject if a policy rule is violated at the project level. For example, if you specified a policy rule that triggers a violation for unknown licenses and the project is added to the BOM with an unknown license, a policy violation will be triggered for that subproject.
- Subprojects and their associated licenses are included in the Notices File report. You can [exclude](#) the subproject from the Notices File report if you have the premium offering.

## To add a project

1. Locate the project using the **Projects** tab on the Dashboard.
2. Select the name of the project to go to the *Project Name* page.
3. Select the version name to open the **Components** tab.



4. Click **Add** and select **Project** to open the Add Project dialog box.

5. Enter the name and version of the project.

**Note:** You must have permission to the project to add it to the BOM.

6. Optionally add a license for this project or modify the existing license. If you do not enter a license, "Unknown License" appears in the BOM for the license for this project.

- ## 7. Click **Save**.

Black Duck adds the selected project to the BOM.

## To edit a project

1. Select the BOM as described in the previous section



2. Click  and select **Edit** to open the Edit Component dialog box.

3. Select one or more different values and click **Update**.

### To delete a subproject from a BOM

1. Select the BOM as described in the previous section.



2. Click  and select **Delete** to open the Delete Component dialog box.

- ### **3. Click Delete.**

The BOM is updated and the risk is recalculated.

 To view where projects are included as subprojects

The **Where Used** table lists the projects where this project version is included in the BOM.

1. Locate the project using the **Projects** tab on the Dashboard by selecting the name of the project to go to the *Project Name* page.
2. Select the version name which opens the **Components** tab.
3. Select the **Details** tab to view where this project version is included as subprojects.

The screenshot shows the Black Duck Projects interface. At the top, there's a header with the project name "Test Project > 1.0" and some status information: "Versions: 1 | Phase: In Planning | Distribution: External". Below the header, there are several tabs: Components, Security, Source, Reports, Details (which is selected and highlighted in blue), and Settings. On the left, there's a sidebar titled "Where Used" with a table. The table has columns: Project, Version, Tier, Released, Distribution, and Phase. It contains one row: "Sample Projects 4" with Version "1.0", Tier "Never", Released "External", and Phase "In Planning". To the right of the table, it says "Displaying 1-1 of 1". Further down, there are sections for "Description" (No description), "Created" (Sep 4, 2018 by sysadmin), "Updated" (Sep 4, 2018 by username), "Last Scan" (Tue, Sep 4, 2018 12:55 PM), and "Last KnowledgeBase Update" (Wed, Sep 5, 2018 3:37 PM). At the bottom, there's a section for "Tags" with the message "No Tags".

The **Where Used** table lists the project name, project version, tier, release date, distribution, and phase for all projects where this project version is a subproject.

## Reviewing snippet matches

Use the **Source** tab to determine if the snippet belongs in your BOM and if so, if the snippet match is correct.

Click [here](#) for more information on using the **Source** tab.

### Snippets in the BOM

If a snippet scan has been run and snippet matches were found, a snippet badge appears next to the risk charts in the BOM indicating the number of snippets that need confirmation.



By default, the BOM does not display your unconfirmed snippet matches. Unlike reviewing a component in the BOM (which marks all instances of that component as reviewed) snippet matches are confirmed on the match level. Only after a snippet match has been confirmed will it appear unfiltered in the BOM.

You can filter the BOM to view unconfirmed snippet matches by selecting the **Unconfirmed** option for the **Match Status** filter.

The screenshot shows the Black Duck Project interface for a project named "multlinesnippet > unnamed". The top navigation bar includes tabs for Components, Security, Source, Reports, Details, and Settings. The Source tab is active. On the left, there are three risk matrices: Security Risk, License Risk, and Operational Risk, each showing the number of components across High, Medium, Low, and None levels. A prominent orange badge in the top right corner says "1 Snippet Needs Confirmation". Below the matrices is a search and filter bar with options like "Add", "Compare to...", "Print...", "Match status Unconfirmed", "Filter components...", and "Add Filter". A table below the filters lists a single component match: "Open Computer Vision Library (OpenCV) 1.0rc1" with a "1 Match" badge, "1 Snippet" type, "Source Code" usage, and "BSD-3-Clause" license. The table also includes columns for Security Risk and Operational Risk. At the bottom right, it says "Displaying 1-1 of 1".

## Viewing snippet matches in the Source tab

Selecting the badge in the BOM displays the **Source** tab filtered to show unconfirmed snippet matches:

This screenshot shows the same Black Duck Project interface as above, but with the Source tab selected. The search and filter bar now has "Match status Unconfirmed" and "Match type Snippet" selected. The main table displays a single row for "cxmatmul.c" which is identified as an "Open Computer Vision Library (OpenCV) 1.0rc1" component with a "1 Snippet" match type, BSD-3-Clause license, and Source Code usage. The table includes columns for Name, Component, Match Type, License, and Usage. The bottom right corner indicates "Displaying 1-1 of 1".

**Note:** You can also view the **Source** tab filtered to a specific match by selecting it when viewing unconfirmed matches, as described above.

- The left pane shows the top-level directory. Select the directory to view the tree structure of the files.
- The table provides information, such as the name, component, match type, license, and usage.
  - indicates a snippet match.
  - indicates there is a source file to view. This icon only appears if you [uploaded source files](#).

Clicking opens the Source Code View which displays the content of this file.

The screenshot shows the 'Source Code View' window with the file 'Cache.java' open. The code is displayed in a monospaced font. At the top, there is metadata: file:///Users/eford/Downloads/Tutorial\_Files/src\_jo's%20files/util/Cache.java, Size: 3.46 KB. Below the code, a copyright notice from the Apache Software License, Version 1.1, is visible.

```

1 --djiixNQH82k8ShdjJWdCrpN9l5evrfzC8Qm8
2 Content-Disposition: form-data; name="file"; filename="Cache.java"
3 Content-Type: text/x-java-source
4 Content-Length: 3543
5
6 /*
7 * $Id: Cache.java,v 1.7 2003/11/07 20:16:25 dfa Exp $
8 *
9 * -----
10 * The Apache Software License, Version 1.1
11 *
12 * Copyright (c) 2000 The Apache Software Foundation. All rights
13 * reserved.
14 *
15 * Redistribution and use in source and binary forms, with or without
16 * modification, are permitted provided that the following conditions
17 * are met:
18 *
19 * 1. Redistributions of source code must retain the above copyright
20 * notice, this list of conditions and the following disclaimer.
21 *
22 * 2. Redistributions in binary form must reproduce the above copyright

```

[Close](#)

- Clicking displays the Snippet View. The information shown here depends on whether you uploaded source files during the snippet scan.
  - If you uploaded source files, the Snippet View displays the source file on the left pane and the matched component on the right pane:

The screenshot shows the 'Snippet View' window. On the left, the file 'cxmatmul.c' is shown with its content. On the right, the 'Component' section displays the 'Open Computer Vision Library (OpenCV) 1.0rc1' code. Lines of code that are matched between the source file and the component are highlighted in yellow.

**cxmatmul.c**  
file:///Users/joycel/Desktop/cxmatul/cxmatmul.c  
Size: 58.77 KB

**Open Computer Vision Library (OpenCV) 1.0rc1**  
Needs confirmation  
License: BSD 3-clause "New" or "Revised" License | Release Date: Aug 18, 2006  
Snippet Match 53%

**Alternative Matches**

```

1 /* dummy */
2 /* dummy */
3 /* dummy */
4 /* dummy */
5 /* dummy */
6 /* dummy */
7 /* dummy */
8 /* dummy */
9 /* dummy */
10 // SQA START // 100-000 11L
11 #define ICV_DEF_MULTRANS_L_FUNC( flavor, srctype, dsttype, load_macro
12 static _CvStatus _CV_STDCALL
13 icvMulTransposed_#flavor( const srctype* src, int srcstep,
14 dsttype* dst, int dststep,
15 dsttype* delta, int deltastep,
16 c_CvSize size, int delta_cols, DATA_TYPE s
17 {
18     int i, j, k;
19     dsttype* tdst = dst;
20     int sstep = srcstep / sizeof(src[0]);
21     int dststep = dststep / sizeof(dst[0]);
22 }

2979     for( i = 1; i < size.width; i++ )
2980         for( j = 0; j < i; j++ )
2981             dst[dststep*i + j] = dst[dststep*j + i];
2982
2983     if( col_buf && !local_alloc )
2984         cvFree( &col_buf );
2985
2986     return CV_NO_ERR;
2987 }
2988
2989 #define ICV_DEF_MULTRANS_L_FUNC( flavor, srctype, dsttype, load_macro )
2990 static CvStatus CV_STDCALL
2991 icvMulTransposed_#flavor( const srctype* src, int srcstep,
2992 dsttype* dst, int dststep,
2993 dsttype* delta, int deltastep,
2994 c_CvSize size, int delta_cols, double scale )
2995 {
2996     int i, j, k;
2997     dsttype* tdst = dst;
2998
2999
3000

```

[Close](#)

Highlighted code indicates the lines of code that were matched in the source file to the component in the current match.

- If you did not upload source files, the matched component appears in the right pane:

Snippet View

Scanned File	Matched Component	Snippet Adjustments
samplefile1.h zip:file:%2F%2FUsers%2Fjoyce%2FDesktop%2Ftutorialfiles%2FTutorial_Files.zip#... Size: 6.32 KB	GnuPG 0.2.18 /include/ License: License Not Found   Release Date: Jul 31, 2015 Snippet Match 67% ▼Alternative Matches	⚠ Needs confirmation
	<pre>14 * GNU General Public License for more details. 15 * 16 * You should have received a copy of the GNU General Public License 17 * along with this program; if not, write to the Free Software 18 * Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA 02111 19 */ 20 21 #ifndef G10_TYPES_H 22 #define G10_TYPES_H 23 24 25 /* The AC_CHECK_SIZEOF() in configure fails for some machines. 26 * we provide some fallback values here */ 27 #if !SIZEOF_UNSIGNED_SHORT 28 #undef SIZEOF_UNSIGNED_SHORT 29 #define SIZEOF_UNSIGNED_SHORT 2 30 #endif 31 #if !SIZEOF_UNSIGNED_INT 32 #undef SIZEOF_UNSIGNED_INT 33 #define SIZEOF_UNSIGNED_INT 4 34 #endif</pre>	<a href="#">Close</a>
No files to display There is no source code for this file/component		

Highlighted text shows the lines of code of the component that were matched by the selected (current) match.

- If the file has more than one snippet match, a message appears at the bottom of the Snippet View, letting you navigate to the next snippet match.
- The Snippet View provides the following information for the current match (and any alternative matches):
  - Component name and version.
  - Component license.
  - Release date.
  - Percentage of the scanned file that matches the component file.

jodit 3.1.95 ⚠ Needs confirmation

License: GNU General Public License v2.0 or later | Release Date: May 11, 2018  
Snippet Match 41%

[▼Alternative Matches](#)

The **Alternative Matches** drop-down list shows alternative components and/or component versions which could be possible matches for the selected snippet. The match which is currently assigned to the selected snippet is the default. Selecting a match from the drop-down list displays the code for that component or component version.

- Snippet adjustments that are available are:
  - Confirm

- Needs Confirmation
- Ignore
- Unignore

#### ⚙ To review a snippet match

1. In the **Source** tab, click  for the snippet match you wish to review.

The Snippet View appears.

2. In the Snippet View:

- a. View other possible matches. Select **Alternative Matches** to view other possible matches. You can:

- Select one of possible alternative matches.
- Select to manually enter an alternative match.

Selecting this option displays fields from which you can select the component, version, and/or origin ID. After selecting the values, click **Confirm**.

- b. Use the **Snippet Adjustment** drop-down list to select one of these options:

- Confirm.
- Needs Confirmation.
- Ignore
- Unignore.

#### ⚙ To bulk edit snippet matches

1. Select more than one snippet match.
2. Click **Edit**.

The Bulk edit dialog box appears.

Bulk edit X

This adjustment will apply to all versions of tutorialfilescli - excluding archived versions.

▼ You have selected 5 items which are applicable for the edit action

File	Component	Match Type
ascii.c	Unmatched	1 Snippet <span style="float: right;">X</span>
bool.c	Unmatched	1 Snippet <span style="float: right;">X</span>
char.c	Unmatched	1 Snippet <span style="float: right;">X</span>

Show More

Component \*

Version i Select a component to list its versions

Usage

Snippet Adjustments  Adjust Snippets and Confirm

Cancel Update

3. Use this dialog box to modify the component, version, origin ID, or usage.
4. Select **Adjust Snippets and Confirm** which adjusts and automatically confirms the snippet match.
5. Click **Update**.

# Chapter 8: Managing components

Users with the [Component Manager role](#) can:

- Create and manage [custom components](#).
- Modify Black Duck [KnowledgeBase components](#).

Use the Component Management table to manage components.

## ⚙️ To view managed components

1. Log in to Black Duck with the Component Manager role.



2. Click the expanding menu icon ( ) and select **Component Management**.

The **Components** tab appears.

The screenshot shows the 'Component Management' page with the 'Components' tab selected. At the top, there's a search bar labeled 'Filter components...' and a 'Add Filter' button. Below the search bar is a table with columns: Component, License, Source, Status, and Last Modified. The table contains three rows of data:

Component	License	Source	Status	Last Modified
> Apache POI <small>1 Version</small>		KnowledgeBase	Unreviewed	Jul 30, 2019 by sysadmin
Bash		Modified KnowledgeBase	Approved	Jul 30, 2019 by sysadmin
> Sample Custom Component <small>2 Versions</small>		Custom	Unreviewed	Jul 30, 2019 by sysadmin

At the bottom right of the table, it says 'Displaying 1-3 of 3'.

The table in **Components** tab contains the following information:

Column	Description
<b>Component</b>	Name of the component. Select the component name to open the <b>Overview</b> tab of the <i>Component Name</i> page. If there are multiple versions for this component, select > to display the versions. Select a version to open the <b>Details</b> tab of the <i>Component Name &gt; Version</i> page. ⓘ indicates that there is a note for this component or component version. Hover over the icon to view the information.
<b>License</b>	License for this component.

Column	Description
<b>Source</b>	Source for this component. Possible value are: <ul style="list-style-type: none"> <li>Custom. A custom component.</li> <li>KnowledgeBase. An unmodified Black Duck KnowledgeBase component.</li> <li>Modified KnowledgeBase. A modified Black Duck KnowledgeBase component.</li> </ul>
<b>Status</b>	<u>Status</u> of this component. Possible values are: <ul style="list-style-type: none"> <li>Approved</li> <li>Deprecated</li> <li>Limited Approval</li> <li>Rejected</li> <li>Unreviewed</li> </ul>
<b>Last Modified</b>	Date this component/component version was last modified and the user who last modified it.

Select the **Component Versions** tab to view information for each component version.

Column	Description
<b>Component Version</b>	Name of the component version. Select the component name to open the <b>Overview</b> tab of the <i>Component Name</i> page. Select the version to open the <b>Details</b> tab of the <i>Component Name &gt; Version</i> page.
<b>License</b>	License for this component version.
<b>Source</b>	Source for this component version. Possible values are: <ul style="list-style-type: none"> <li>Custom. A custom component version.</li> <li>KnowledgeBase. An unmodified Black Duck KnowledgeBase component version.</li> <li>Modified KnowledgeBase. A modified Black Duck KnowledgeBase component version.</li> </ul>
<b>Status</b>	<u>Status</u> of this component version. Possible values are: <ul style="list-style-type: none"> <li>Approved</li> <li>Deprecated</li> <li>Limited Approval</li> <li>Rejected</li> <li>Unreviewed</li> </ul>
<b>Last Modified</b>	Date this component version was last modified and the user who last modified it.

## About custom components

You may want to use a component in your BOM that is not available from the Black Duck KnowledgeBase; for

example, your project uses an open source component that is not tracked by the Black Duck KB or there is a commercial component you want to add to your BOM. So that your BOM accurately reflects your project, users with the [Component Manager role](#) can create and manage custom components which can then be added to a BOM.

**Note:** Contact Black Duck Customer Support for missing versions of open source components that are managed by the Black Duck KnowledgeBase.

Black Duck provides the information Component Managers need to successfully manage their custom components. They can use the:

- *Custom Component Name Overview* tab to view the [versions for a component](#), including the status, description, and tags for this custom component.  
You can also use this tab to [create tags](#) for the custom component.
- *Custom Component Name Settings* tab to view and/or edit the [details of a component](#).  
Use this tab to delete a custom component.
- *Custom Component Name > Version Details* tab which provides details of this component version and lists the [projects used by a custom component version](#).  
Component Managers must have permission to view the projects for them to appear on this page.
- *Custom Component Name > Version Settings* tab to view and/or edit the [details of a component version](#).  
Use this tab to delete a custom component version.

Note the following:

- In the BOM:
  - The match type for a custom component added to a BOM is **Manually Added**.
  - Custom components display license risk. (Note that the license risk shown is [determined by the license](#) selected for this component.) No security risk values are shown as no security vulnerabilities are associated with the custom component and also no operational risk is shown.
- Policy Managers can create policy rules for custom components.
- You can use the search feature for custom components. A [component filter](#), **Component Source**, has the value **Black Duck Custom Component** for custom components.
- In the `component.csv` file in the [Project Version report](#), a new column, labeled **Source/Type** denotes whether the component is a custom component (value of CUSTOM\_COMPONENT) or a component that is managed by the Black Duck KnowledgeBase (value of KB\_COMPONENT).
- Custom components do not have origins.

## Managing custom components

Users with the Component Manager [role](#) can create, edit, and delete custom components.

You can:

- Create custom components.
- View custom component information.
- Edit custom components.
- Delete a component.
- [Create additional versions for a custom component.](#)
- [Add a status.](#)

## Creating custom components

1. Log in to Black Duck with the Component Manager [role](#).



2. Click the expanding menu icon ( ) and select **Component Management**.

The **Components** tab appears.

3. Click **Add > Create a component**.

The Create a Component dialog box appears.

4. Enter the component name, version, and license, which are required fields, and optionally, values for the description, URL, and release date.

5. Click **Create**.

The **Components** tab appears with the new component listed in the table; the **Component Versions** tab lists the new component and version.

## Viewing custom component information

1. Log in to Black Duck with the Component Manager [role](#).



2. Click the expanding menu icon ( ) and select **Component Management**.

The **Components** tab appears.

3. Select the component name you wish to view information.

The **Overview** tab of the *Component Name* page appears.

Version	Used count	License	Released
1.0	2	No Limit Public License	Never
2.0	0	Apache License 2.0	Never

Displaying 1-2 of 2

If provided, additional information, such as the status, description, and tags for the custom component is

shown along with the following information.

Column	Description
<b>Version</b>	Version(s) for this component. Select a version to open the <b>Details</b> tab of the <i>Component Name &gt; Version</i> which lists the projects that use this component version.
<b>Used Count</b>	Number of projects that use this component version.
<b>License</b>	License for this component version.
<b>Release</b>	Release date for this component version. <b>Never</b> is listed if a value was not entered.

## Editing custom components

1. Log in to Black Duck with the Component Manager [role](#).



2. Click the expanding menu icon ( ) and select **Component Management**.

The **Components** tab appears.

3. Select the component you wish to modify.

The *Component Name* page appears listing the versions for this component.

4. Select the **Settings** tab to add or edit the information for this component, such as a description, URL, notes, or to define a [status](#) for this component.

The screenshot shows the 'My Custom Component' settings page. At the top, there's a navigation bar with a cube icon, the text 'Custom My Custom Component', and tabs for 'Overview' and 'Settings'. The 'Settings' tab is active. Below the tabs, there are several input fields: 'Component Name' (set to 'My Custom Component'), 'Description' (empty), 'Url' (empty), 'Notes' (empty), and 'Status' (set to 'Unreviewed'). At the bottom right is a blue 'Save' button. At the very bottom of the page, there's a 'Delete component' section with a warning message and a red 'Delete Component' button.

5. Click **Save**.

## Deleting custom components

You cannot delete a custom component that is in use.

### To delete a custom component

1. Log in to Black Duck with the Component Manager [role](#).



2. Click the expanding menu icon () and select **Component Management**.

The **Components** tab appears.

3. Do one of the following:

- Click  in the row of the component that you want to delete and select **Delete**.
- Select the custom component you wish to remove to view the **Overview** tab of the *Component Name* page.

Select the **Settings** tab and click **Delete Component**.

4. Click **Delete** to confirm in the Delete Custom Component dialog box.

## Managing custom component versions

Users with the Component Manager [role](#) can:

- Create additional versions for a custom component.
- View where a custom component version is used.
- Edit version information.
- Delete a version.

### Creating additional versions for a custom component

1. Log in to Black Duck with the Component Manager [role](#).



2. Click the expanding menu icon () and select **Component Management**.

The **Components** tab appears.

3. Select the component to which you want to add versions. Note that you can also select the component from the **Component Versions** tab.

The **Overview** tab of the *Component Name* page appears listing the versions for this component.

4. Click **Create Version**.

The Create a New Version dialog box appears.

5. Enter the version and license, and optionally, select a release date for this version and click **Create**.

The **Details** tab of the *Component Name > Version* page appears for the new version.

## Viewing the projects where a version is used

1. Log in to Black Duck with the Component Manager [role](#).



2. Click the expanding menu icon ( ) and select **Component Management**.

The **Components** tab appears.

3. Select the **Component Versions** tab.

Component Version	License	Source	Status	Last Modified
Apache POI - 1.5.0	Apache License 1.1	KnowledgeBase	Approved	Jul 30, 2019 by System Administrator
Sample Custom Component - 1.0	Apache License 2.0	Custom	Unreviewed	Jul 30, 2019 by System Administrator
Sample Custom Component - 2.0	Apache License 2.0	Custom	Unreviewed	Jul 30, 2019 by System Administrator

Displaying 1-3 of 3

4. Select the version to open the **Details** tab of the *Component Name > Version* page.

**Black Duck Custom Components**  
**New Custom Component** ▶ 1.0

**Custom** Versions: 0 | Phase: In Development | Distribution: External

Project	Version	Tier	Released	Distribution	Phase
Sample Project	1.0		Never	External	In Planning

Displaying 1-1 of 1

**Description**  
New component needed for Sample Project

**Released** Released on May 9, 2018

**Licenses**  
[template] Basic MIT-Style License  
Permissive

The **Where Used** table lists the projects that use this version.

**Note:** You must have permission for a project for you to view it on this page.

From this table:

- Select the project name to view the [Project Name page](#).
- Select the project versions to view the BOM.
- Select the license to view the license text.

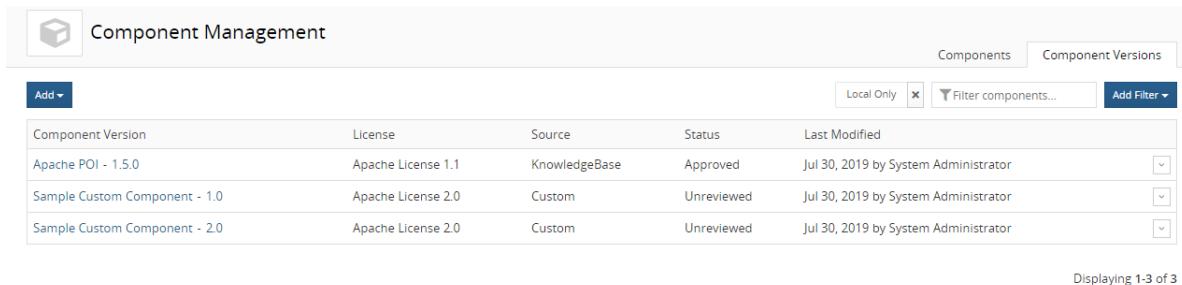
## Editing custom component versions

1. Log in to Black Duck with the Component Manager [role](#).



2. Click the expanding menu icon ( ) and select **Component Management**.

The **Components** tab appears.

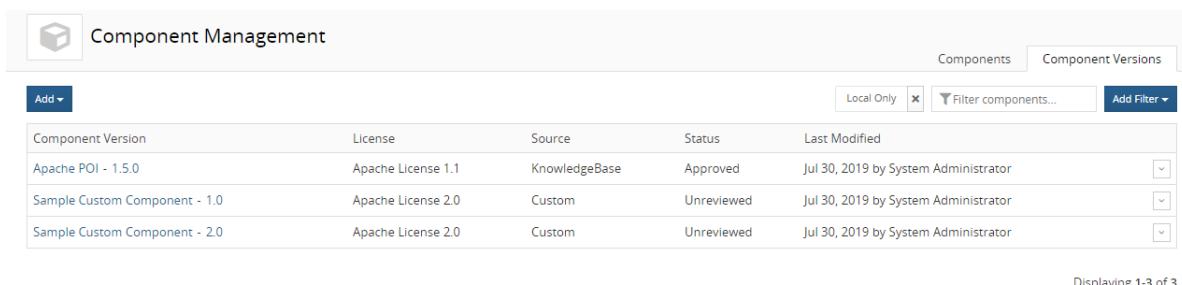


A screenshot of the Black Duck Component Management interface. The title bar says "Component Management". Below it is a toolbar with an "Add" button, a "Local Only" checkbox, a "Filter components..." search bar, and an "Add Filter" button. There are two tabs: "Components" (selected) and "Component Versions". A table below shows four rows of component data:

Component Version	License	Source	Status	Last Modified
Apache POI - 1.5.0	Apache License 1.1	KnowledgeBase	Approved	Jul 30, 2019 by System Administrator
Sample Custom Component - 1.0	Apache License 2.0	Custom	Unreviewed	Jul 30, 2019 by System Administrator
Sample Custom Component - 2.0	Apache License 2.0	Custom	Unreviewed	Jul 30, 2019 by System Administrator

At the bottom right of the table area, it says "Displaying 1-3 of 3".

3. Select the **Component Versions** tab.



A screenshot of the Black Duck Component Management interface, similar to the previous one but with the "Component Versions" tab selected. The table data is identical to the first screenshot:

Component Version	License	Source	Status	Last Modified
Apache POI - 1.5.0	Apache License 1.1	KnowledgeBase	Approved	Jul 30, 2019 by System Administrator
Sample Custom Component - 1.0	Apache License 2.0	Custom	Unreviewed	Jul 30, 2019 by System Administrator
Sample Custom Component - 2.0	Apache License 2.0	Custom	Unreviewed	Jul 30, 2019 by System Administrator

At the bottom right of the table area, it says "Displaying 1-3 of 3".

4. Select the version to open the **Details** tab of the *Component Name > Version* page.

5. Select the **Settings** tab to edit the information.

Custom  
My Custom Component ▾ 1.0

Component Details ➔

Version *	1.0
License *	No Limit Public License
Release Date	
Notes	
Status	Unreviewed

**Save**

Delete Version

Once you delete a version, you cannot restore it and you lose all information related to the version. Scans will be unmapped from the version and not deleted.

**Delete Version**

6. Click **Save**.

## Deleting a custom component version

There must be at least one version for a custom component.

You cannot delete a version that is being used in a project.

### ⚙ To delete a custom component version

1. Log in to Black Duck with the Component Manager [role](#).

2. Click the expanding menu icon (☰) and select **Component Management**.

The **Components** tab appears.

3. Select the **Component Versions** tab.

Component Management

Add ▾

Components Component Versions

Component Version	License	Source	Status	Last Modified
Apache POI - 1.5.0	Apache License 1.1	KnowledgeBase	Approved	Jul 30, 2019 by System Administrator
Sample Custom Component - 1.0	Apache License 2.0	Custom	Unreviewed	Jul 30, 2019 by System Administrator
Sample Custom Component - 2.0	Apache License 2.0	Custom	Unreviewed	Jul 30, 2019 by System Administrator

Displaying 1-3 of 3

4. Click the delete icon in the row of the custom component version you wish to remove and select **Delete**.

The Delete Custom Component dialog box appears.

5. Click **Delete**.

You can also delete a version using the **Settings** tab, as described in the previous section.

## About Black Duck KnowledgeBase components

The Black Duck® KnowledgeBase™ (the Black Duck KB) is the industry's most comprehensive database of open source component information. Since 2003, Black Duck has searched the Internet for information on open source software (OSS) components and downloadable source code. The complete version of the Black Duck KB includes more than 2 million unique components from more than 10,000 sites and contains detailed data on more than 79,000 actively traced vulnerabilities across more than 530 billion lines of code. The Black Duck KB includes detailed data for more than 2,500 unique licenses, including the full license text and dozens of encoded attributes and obligations for each license. Black Duck connects to a version of the Black Duck KB hosted in the cloud.

New OSS component versions and meta data, such as vulnerabilities, are continually added and updated to the version of the Black Duck KB that supports Black Duck.

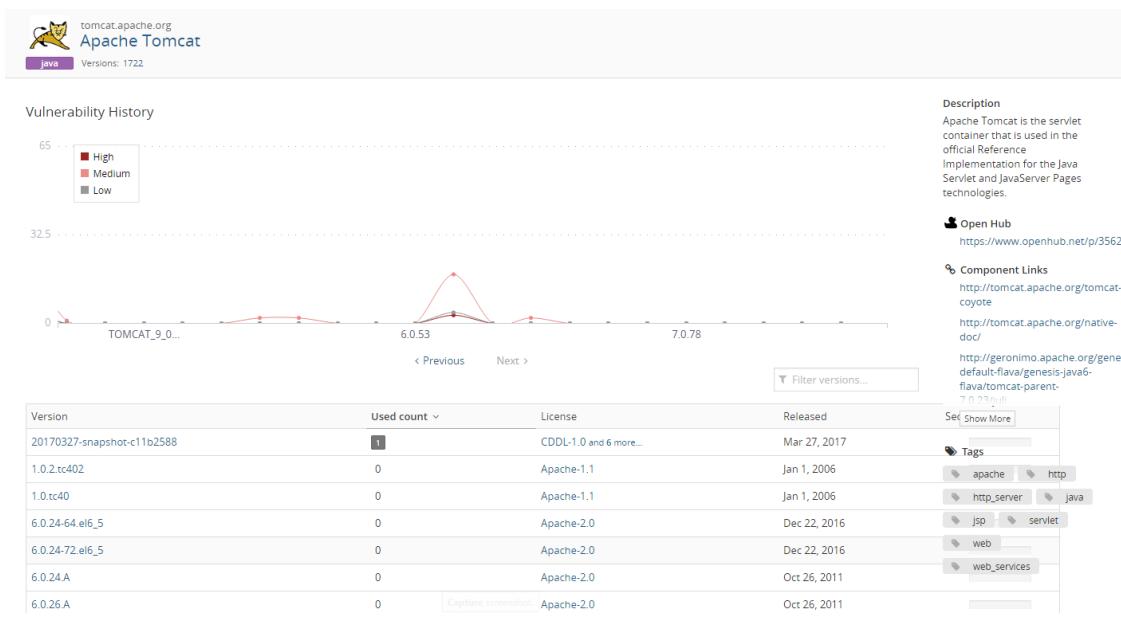
The Black Duck KB provides information about OSS components at the [component level](#) and at the [component version level](#).

So that your BOM accurately reflects your project, users with the [Component Manager role](#) can:

- [Modify](#) Black Duck KB components and/or Black Duck KB component versions.
- Undo these modifications and [reset the KB data](#) back to its original values.
- [Define a status](#) for a Black Duck KB component and/or component version to ensure that only approved components/version are included in your BOM.

## Understanding the component information available from the Black Duck KB

The Black Duck KB *Component Name* page displays information about the open source software (OSS) component, such as a description, component links, and tags, and information about each of the component versions that are available in the Black Duck KB.



A graph at the top of the page shows a history of high, medium, and low vulnerabilities for each version of this component. Use this graph to quickly view vulnerability information for component versions.

- Select **Previous** or **Next** to view older or newer versions.
- Hover over a data point in the graph to view the version, release date, and number of vulnerabilities for this version:



To view information on versions that interest you, use the filter, located above the table, to filter the versions shown in the vulnerability graph and in the table below.

The following information is available for each version:

Column	Description
Version	Release number of this version of the component.  Select the version number to display the <a href="#">Component Name &gt; Version page</a> .
Used Count	Number of project version BOMs in which this version of the OSS project is used.  <b>Tip:</b> Select the number to go to the <b>Details</b> tab for this version of the OSS component. That tab lists each project and project version in which this version of the OSS component is used.
License	Declared license of this version of the OSS component. Other license types include: <ul style="list-style-type: none"><li>• "Unknown" indicates that the OSS component version's license is not known.</li><li>• "License Not Found" indicates that although researched by Synopsys, no declared license was found for the component.</li><li>• "No License" indicates that Synopsys has found a declaration of 'No License' for the component.</li></ul> For known licenses, select the license name to view license details and license text.
Released	The date this version of the OSS component was released.
Security Risk	A graph which shows the number of high risk, medium risk, and low risk vulnerabilities associated with this version of the OSS component  Select a value in the security risk graph to display the <a href="#">Component Name &gt; Version page</a> .

## Understanding the component version information available from the Black Duck KB

On the *Component Name Version* page, the **Details** tab provides the following information:

- Description.
- Count of known security vulnerabilities.
- Associated licenses.
- Component links, if available.
- Tags, if available.
- Date this version was released.
- Number of newer versions.
- [Status](#) of this version.
- Date this component was last updated.
- Commit activity and the trend for the component over the last 12 months.
- Number of contributors for the component for the past 12 months.
- A Where Used table which lists the projects and the respective versions in which this version of the component is used.

The goal of the Apache Struts project is to encourage application architectures based on the "Model 2" approach, a variation of the classic Model-View-Controller (MVC) design paradigm. Under Model 2, a servlet (or equivalent) manages business logic execution, and presentation logic resides mainly in server pages.

Released	Newer Versions	Status	Updated
Oct 20, 2006	136	Unreviewed	May 27, 2019

**Activity**  
Last 12 Months: 335 commits, decreasing  
Last commit: May 27, 2019

**Community**  
Last 12 Months: 22 contributors

**Where Used**

Project	Version	Released	Phase
TestWP	test	Never	In Planning

Displaying 1-1 of 1

**Vulnerabilities**  
43 Vulnerabilities

**Licenses**  
Apache License 2.0

**Open Hub**  
<https://www.openhub.net/p/3569>

**Component Links**  
<http://struts.apache.org/>

**Tags**  
apache, development, framework, java, model-view-controller, mvc, mvcframework, programming, servlet, web, webapplication

The Where Used table contains the following information:

Column	Description
Project	Name of the project that uses this version of the OSS component from the Black Duck KB.  Select the project name to display the <b>Overview</b> tab of the <b>Project Name</b> page which provides information on this project.
Version	Version of the project that uses this version of the OSS component from the Black Duck KB.  Select the version to display the BOM filtered to display that component version.
Released	Date this version was released.
Phase	Development phase that this version of the project is currently in.

On the Black Duck KB Component Name Version page, the **Security** tab displays the list of vulnerabilities associated with this version of the OSS component from the Black Duck KB.

Identifier	Published	Overall Score
NVD   CVE-2011-1772	May 13, 2011	2.6 Low
VnDB   72238	May 11, 2011	2.6 Low
NVD   CVE-2016-4003	Apr 12, 2016	4.3 Medium
NVD   CVE-2015-5169	Sep 25, 2017	4.3 Medium
NVD   CVE-2011-2087	May 13, 2011	4.3 Medium
NVD   CVE-2016-2162	Apr 12, 2016	4.3 Medium
NVD   CVE-2010-1870	Aug 17, 2010	5 Medium
NVD   CVE-2016-3093	Jun 7, 2016	5 Medium

This tab contains the following information:

Column	Description
Identifier	<p>The identifier and value associated with this vulnerability.</p> <p>Select &gt; in the table next to the vulnerability to view a brief description. Depending on the identifier, select to view <a href="#">the BDSA record</a> or <a href="#">the CVE record</a>.</p>
Published	Date on which the vulnerability was published.
Overall Score	<p>Shows the Temporal score (for BDSA), or Base score (for NVD) and associated risk level. Hover over the Overall Score value to see the individual values.</p> <ul style="list-style-type: none"> <li>For BDSA, the Temporal, Base, Exploitability, and Impact scores are shown.</li> <li>For NVD, the Base, Exploitability, and Impact scores are shown.</li> </ul> <p>The Temporal score represents time-dependent qualities of a vulnerability taking into account the confirmation of the technical details of a vulnerability, the existence of any patches or workarounds, and the availability of exploit code or techniques.</p> <p>The Base score reflects the overall basic characteristics of a vulnerability that are constant over time and user environments:</p> <ul style="list-style-type: none"> <li>Access Vector (AV) - CVSS 2.0 / Attack Vector (AV) - CVSS 3.0</li> <li>Access Complexity (AC) - CVSS 2.0 / Attack Complexity (AC) - CVSS 3.0</li> <li>Authentication (Au)</li> <li>Integrity (I)</li> <li>Availability (A)</li> <li>Confidentiality (C)</li> </ul> <p><b>Note:</b> The Authentication value is not available for CVSS 3.0 scores.</p> <p>The Exploitability score measures how the vulnerability is accessed and if extra conditions are required to exploit it, taking into account access vector, complexity, and authentication.</p> <p>The Impact score reflects the possible impact of successfully exploiting the vulnerability, considering the integrity, availability, and confidentiality impacts.</p>

The **Cryptography** tab shows information on component versions that have encryption algorithms. Clear [here](#) for more information.

The **Settings** tab shows details on this component version that a [Component Manager can modify](#).

## Modifying KB components

Users with the Component Manager [role](#) can modify the information shown for a Black Duck KB component or component version.

The revised information will appear in your current BOMs and in any future BOMs that contain this component/component version. Note that local edits to a component in a BOM made by a user, such as the BOM Manager, to a BOM supersede the edits to the component/component version made by the Component

Manager.

To modify a KB component or component version:

1. Add the component and/or component version to Component Management.
2. Modify the KB component or component version.

#### ✿ To add a KB component or component version to the Component Management table

1. Log in to Black Duck with the Component Manager [role](#).



2. Click the expanding menu icon (☰) and select **Component Management**.

The **Components** tab appears.

The screenshot shows the Black Duck Component Management interface. At the top, there's a navigation bar with a cube icon, the text "Component Management", and tabs for "Components" and "Component Versions". Below the navigation bar is a search bar with a magnifying glass icon and a "Filter components..." button, along with an "Add Filter" dropdown. The main area is a table titled "Component Management" with columns: Component, License, Source, Status, and Last Modified. There are three rows of data:

Component	License	Source	Status	Last Modified
> Apache POI [1 Version]		KnowledgeBase	Unreviewed	Jul 30, 2019 by sysadmin
Bash		Modified KnowledgeBase	Approved	Jul 30, 2019 by sysadmin
> Sample Custom Component [2 Versions]		Custom	Unreviewed	Jul 30, 2019 by sysadmin

At the bottom right of the table, it says "Displaying 1-3 of 3".

3. Select **Add > Add a KnowledgeBase component** to open the Add Component dialog box.
4. Select the KB component and if adding a component version, select a version.
5. Select a status for this component.
6. Click **Save**.

The component appears in the **Components** tab with **KnowledgeBase** as the Source.

To add additional versions, repeat this process, selecting the component and versions from the Add Component dialog box.

#### ✿ To modify a KB component

1. Log in to Black Duck with the Component Manager [role](#).



2. Click the expanding menu icon (☰) and select **Component Management**.

The **Components** tab appears.

3. Select the KB component you wish to modify.

The **Overview** tab for the *Component Name* page appears.

4. Select the **Settings** tab.
5. Modify the information and click **Save**.

The Source for this component is now **Modified KnowledgeBase**.

 To modify a KB component version

1. Log in to Black Duck with the Component Manager [role](#).



2. Click the expanding menu icon () and select **Component Management**.

The **Components** tab appears.

3. Select the KB component version you wish to modify. Select the version from the **Component Versions** tab or in the **Components** tab, select > next to the KB component name to display the versions.

The **Details** tab for the *Component Name > Version* page appears.

4. Select the **Settings** tab.

5. Modify the information and click **Save**.

The Source for this component version is now **Modified KnowledgeBase**.

## Resetting a Black Duck KB component's values

If you have modified the values of a Black Duck KB component or component version, you can undo those changes and reset the KB data back to its original values.

Resetting a KB component to its original values does not change the status of the component.

**Note:** Resetting the component or component version removes all modifications.

 To reset a KB component

1. Log in to Black Duck with the Component Manager [role](#).



2. Click the expanding menu icon () and select **Component Management**.

The **Components** tab appears.

3. Do one of the following:

- Use the *Component Name* page to reset the component:

- a. Select the KB component you wish to reset.

The *Component Name* page appears.

- b. Select the **Settings** tab to view the component details.

- c. In the **Reset Component** section, click **Reset Component** to open the Reset Component dialog box.
  - d. Click **Reset** to confirm.
- Use the **Components** tab in Component Management to reset the component:
    - a. Click  in the row of the KB component you wish to reset.
    - b. Select **Restore** open the Reset Component dialog box.
    - c. Click **Reset** to confirm.

In the table on the **Components** tab, the source for this component reverts from **Modified KnowledgeBase** back to **KnowledgeBase**.

#### To reset a KB component version

1. Log in to Black Duck with the Component Manager [role](#).
2. Click the expanding menu icon () and select **Component Management**.

The **Components** tab appears.
3. Select the **Component Versions** tab.
4. Do one of the following:
  - Use the *Component Name > Version* page to reset the component version:
    - a. Select the KB component version you wish to reset.

The *Component Name > Version* page appears.
    - b. Select the **Settings** tab to view the component version details.
    - c. In the **Reset Component Version** section, click **Reset Version** to open the Reset Component Version dialog box.
    - d. Click **Reset** to confirm.
  - Use the **Component Versions** tab in Component Management to reset the component:
    - a. Click  in the row of the KB component version you wish to reset.
    - b. Select **Restore** open the Reset Component dialog box.
    - c. Click **Reset** to confirm.

In the table on the **Component Versions** tab, the source for this component version reverts from **Modified KnowledgeBase** back to **KnowledgeBase**.

## About the KnowledgeBase Feedback Service

To improve and refine the Black Duck KnowledgeBase (KB) capabilities, a feedback service has been instituted.

If you are discovering that the KB has incorrectly matched or missed matches, this service provides you with a

way to send this information back to the Black Duck KB. Feedback is sent when you make BOM adjustments to the component, version, origin, origin ID, or license of a match made by the KB. Feedback is also sent if you identify unmatched files to a component; it is not sent on manually added components that do not have files associated with them.

The Black Duck KB will use the feedback to improve the accuracy of future matches. This information also helps us to prioritize our resources so that we take a closer look at the components that are important to our customers.

**Note:** No customer-identifiable information is transmitted to the KB.

## Disabling the Black Duck KnowledgeBase feedback service

By default, the KnowledgeBase feedback service is enabled: adjustments that you make to a BOM are sent to the KnowledgeBase.

You can override the feedback service by using the `blackduck.kbfeedback.enabled` property. A value of **false** overrides the feedback service and BOM adjustments are not sent to the KnowledgeBase.

To disable the feedback service, run the following command. You may need to be a user in the docker group, a root user, or have `sudo` access to run it.

```
docker exec -it <container_id> zkCli.sh -server 127.0.0.1:2181 create /hub/config/blackduck.kbfeedback.enabled false
```

where `<container_id>` is the container id of the zookeeper container.

**Note:** Set the value to true to re-enable the feedback service.

## Setting or modify a component's status

You may want to approve versions or restrict usage in your BOM to approved Black Duck KB or custom components and/or component versions.

Users with the Component Manager [role](#) can set a review/approval status on the component or component version at the global level and then use that status in policy rules.

For example, to ensure that only approved components are included in your BOM:

1. Determine the components (from the Black Duck KB and custom components) that are approved for your BOMs.
2. Set the status for each of these components and/or component versions to "Approved".
3. [Create policy rules](#) such that any component or component version that does not have an "Approved" status triggers a policy violation.

Policy violations appear in your BOM for all components that do not have an approved status.

## Changing the status of components and/or versions

- For KB components, you set the initial status of a KB component and/or component version when you added it to Component Management.
- By default, a custom component/custom component version has a status of "Unreviewed".

Note that the status of a component is independent of the status of its versions.

### To modify the status for a component

- Log in to Black Duck with the Component Manager role.



- Click the expanding menu icon ( ) and select **Component Management**.

The **Components** tab appears.

The screenshot shows the Black Duck Component Management interface. At the top, there's a navigation bar with a cube icon, the title 'Component Management', and tabs for 'Components' and 'Component Versions'. Below the navigation bar is a search bar with 'Filter components...' and an 'Add Filter' button. A large table follows, with columns for 'Component', 'License', 'Source', 'Status', and 'Last Modified'. The table contains three rows: 'Apache POI' (KnowledgeBase, Unreviewed, Jul 30, 2019), 'Bash' (Modified KnowledgeBase, Approved, Jul 30, 2019), and 'Sample Custom Component' (Custom, Unreviewed, Jul 30, 2019). At the bottom right of the table, it says 'Displaying 1-3 of 3'.

Component	License	Source	Status	Last Modified
> Apache POI [1 Version]	KnowledgeBase	Unreviewed	Jul 30, 2019 by sysadmin	
Bash	Modified KnowledgeBase	Approved	Jul 30, 2019 by sysadmin	
> Sample Custom Component [2 Versions]	Custom	Unreviewed	Jul 30, 2019 by sysadmin	

- Do one of the following:

- Click in the row of the component that you want to change the status and select a status from the list.
- Modify the status using the **Settings** tab in the *Component Name* page:
  - Select the component you wish to modify from the **Components** tab.

The **Overview** tab of the *Component Name* page appears.

- Select the **Settings** tab.
- Select a status from the **Status** list and click **Save**.

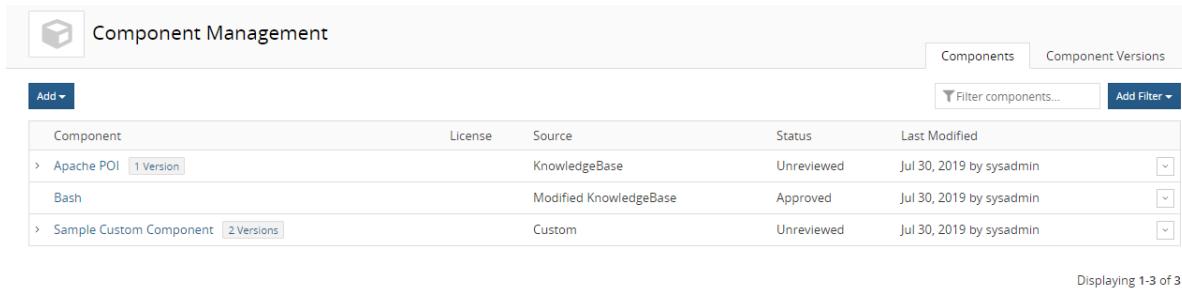
### To modify the status for a component version

- Log in to Black Duck with the Component Manager role.



- Click the expanding menu icon ( ) and select **Component Management**.

The **Components** tab appears.

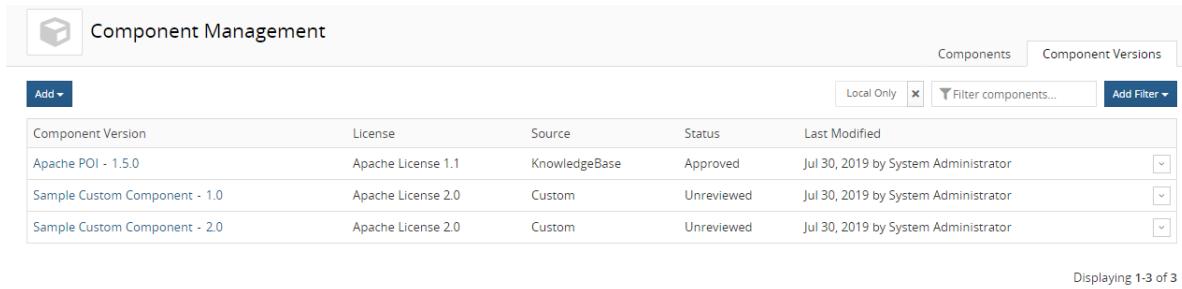


The screenshot shows the 'Component Management' interface with the 'Components' tab selected. At the top, there's a search bar labeled 'Filter components...' and a button 'Add Filter'. Below the header is a table with columns: Component, License, Source, Status, and Last Modified. The table contains three rows:

Component	License	Source	Status	Last Modified
Apache POI [1 Version]	KnowledgeBase	Unreviewed	Jul 30, 2019 by sysadmin	<input type="button" value="▼"/>
Bash	Modified KnowledgeBase	Approved	Jul 30, 2019 by sysadmin	<input type="button" value="▼"/>
Sample Custom Component [2 Versions]	Custom	Unreviewed	Jul 30, 2019 by sysadmin	<input type="button" value="▼"/>

At the bottom right of the table area, it says 'Displaying 1-3 of 3'.

3. Select the **Component Versions** tab.



The screenshot shows the 'Component Management' interface with the 'Component Versions' tab selected. At the top, there's a search bar labeled 'Filter components...', a 'Local Only' checkbox, and a 'Add Filter' button. Below the header is a table with columns: Component Version, License, Source, Status, and Last Modified. The table contains four rows:

Component Version	License	Source	Status	Last Modified
Apache POI - 1.5.0	Apache License 1.1	KnowledgeBase	Approved	Jul 30, 2019 by System Administrator
Sample Custom Component - 1.0	Apache License 2.0	Custom	Unreviewed	Jul 30, 2019 by System Administrator
Sample Custom Component - 2.0	Apache License 2.0	Custom	Unreviewed	Jul 30, 2019 by System Administrator

At the bottom right of the table area, it says 'Displaying 1-3 of 3'.

4. Do one of the following:

- Click  in the row of the component version that you want to change the status and select a status from the list.
- Modify the status using the **Settings** tab in the *Component Name > Version* page:
  - a. Select the component version you wish to modify from the **Component Versions** tab.

The **Overview** tab of the *Component Name > Version* page appears.

- b. Select the **Settings** tab.
- c. Select a status from the **Status** list and click **Save**.

# Chapter 9: Viewing risk in Black Duck

Black Duck helps you understand the type and severity of risks, at several levels of detail, across your projects. The data used to calculate risk is provided by the Black Duck KB.

Use the following pages to identify and manage risk in projects:

- Dashboard pages
- Project version page/**Components** tab
- Project version/**Security** tab

Note that the security risk values shown use CVSS 2.0 or CVSS 3.0 scores, depending on which [security risk calculation you selected](#); by default CVSS 2.0 scores are shown.

## Dashboard pages

Dashboard pages provide a high-level overview of project risk from different perspectives.

**Note:** Dashboard pages will not contain any project or component information until you [create projects](#) and then [map scans](#) to these projects or [manually add components](#) to BOMs. The risk information for the components in your project versions' BOMs will then appear on the Dashboard pages.

- Use the [Projects Dashboard](#) to view the overall risk across all projects.

 Project Dashboard

Security Risk  
Number of Projects

Risk Level	Count
Critical/High	8
Medium	2
Low	0
None	4

License Risk  
Number of Projects

Risk Level	Count
High	8
Medium	0
Low	2
None	6

Operational Risk  
Number of Projects

Risk Level	Count
High	7
Medium	0
Low	3
None	4

Project Name ^ Tier Last Updated Versions Security Risk License Risk Operational Risk

Project Name	Tier	Last Updated	Versions	Security Risk	License Risk	Operational Risk
cloning_052419		May 28, 2019	3	█	█	█
cloning_052419-2		May 28, 2019	2	█	█	█
Demo		May 28, 2019	1	█	█	█
gnutest		May 28, 2019	1	█	█	█
gnutest2		May 28, 2019	1	█	█	█
hh tutorial 5.24		May 28, 2019	1	█	█	█
HUB-Test1		May 28, 2019	1	█	█	█
nptestcvss3		May 28, 2019	1	█	█	█
oppo		May 28, 2019	1	█	█	█
Sample Project		May 28, 2019	1	█	█	█
ski		May 28, 2019	3	█	█	█
Test Comp Edit		May 28, 2019	1	█	█	█
zlib		May 28, 2019	1	█	█	█
zlibtest		May 28, 2019	1	█	█	█

Displaying 1-14 of 14

- Use the [Components Dashboard](#) to view the risk for each of the components that are used in one or more projects.

 Component Dashboard

Security Risk  
Number of Components

Risk Level	Count
Critical/High	40
Medium	38
Low	39
None	611

License Risk  
Number of Components

Risk Level	Count
High	7
Medium	8
Low	9
None	520

Operational Risk  
Number of Components

Risk Level	Count
High	374
Medium	448
Low	400
None	71

Component Name ^ Versions Used count Security Risk License Risk Operational Risk

Component Name	Versions	Used count	Security Risk	License Risk	Operational Risk
Acegi Security	3.1.2	5	█	█	█
Apache Camel	2.15.1	1	█	█	█
Apache Commons FileUpload	3 Versions	8	█	█	█
Apache Commons Logging	1.0.4	5	█	█	█
Apache Lucene	1.4.3	6	█	█	█
Apache ORO	2.0.8	6	█	█	█
Apache Struts	2.0.1	1	█	█	█
Apache log4j	2 Versions	2	█	█	█
Apache-Jakarta Jmeter	2 Versions	2	█	█	█
Async	7 Versions	7	█	█	█
BasicAuth	1.0.4	1	█	█	█
BirtBlog	3 Versions	3	█	█	█
Bluebird JS	2.11.0	1	█	█	█
Bureau Virtuel	1.0.1	1	█	█	█
CRC	3.3.0	1	█	█	█
Cardinal	2 Versions	2	█	█	█
Chai	3.5.0	1	█	█	█
Chalk	4 Versions	4	█	█	█

- Use the [Security Dashboard](#) to view the security risk associated with all the vulnerabilities that exist in your projects. This dashboard also shows the remediation status of all the vulnerabilities that exist within

the projects.

The screenshot shows the Security Dashboard with the following sections:

- Remediation Status:** A donut chart showing 1,175 items in total, with segments for New (1175), Patched (366), Remediation Required (0), Remediation Complete (0), Needs Review (0), Mitigated (0), Ignored (0), and Duplicate (0).
- Identifier:** A table listing 30 vulnerabilities, each with a link to its details. The columns include Identifier, Published, Aff. Versions, and Overall Score (with a color-coded bar).

Identifier	Published	Aff. Versions	Overall Score
BDSA-2019-2327 (CVE-2019-13638)	Jul 26, 2019	1	5 Medium
BDSA-2019-2291 (CVE-2019-1010204)	Jul 24, 2019	1	3.7 Low
BDSA-2019-2243 (CVE-2019-13115)	Jul 23, 2019	1	4.5 Medium
BDSA-2019-2266 (CVE-2019-1010238)	Jul 23, 2019	1	1.6 Low
BDSA-2019-2246 (CVE-2019-13636)	Jul 22, 2019	1	4.7 Medium
BDSA-2019-2248 (CVE-2019-13960)	Jul 22, 2019	1	3.7 Low
BDSA-2018-5020 (CVE-2019-1010025)	Jul 17, 2019	1	1 Low
BDSA-2018-5021 (CVE-2019-1010024)	Jul 17, 2019	1	1 Low
BDSA-2018-5010 (CVE-2019-1010023)	Jul 17, 2019	1	4.4 Medium
NVD CVE-2019-13272 (BDSA-2019-2164)	Jul 17, 2019	1	7.2 High
BDSA-2019-2130 (CVE-2019-8598)	Jul 16, 2019	1	4.7 Medium
BDSA-2019-2131 (CVE-2019-8600)	Jul 16, 2019	1	4 Medium
BDSA-2019-2110 (CVE-2019-5827)	Jul 16, 2019	1	3 Medium
BDSA-2019-2132 (CVE-2019-8602)	Jul 16, 2019	1	3.7 Low
BDSA-2018-5012 (CVE-2018-20852)	Jul 16, 2019	1	3.4 Low
BDSA-2017-3870 (CVE-2017-12652)	Jul 15, 2019	1	3.4 Low
BDSA-2018-5009 (CVE-2019-1010022)	Jul 15, 2019	1	1 Low
NVD CVE-2018-20852 (BDSA-2018-5012)	Jul 13, 2019	1	5 Medium
BDSA-2019-2045 (CVE-2019-13232)	Jul 11, 2019	1	3.4 Low
BDSA-2019-2075 (CVE-2019-13305)	Jul 10, 2019	1	3.4 Low
BDSA-2019-2073 (CVE-2019-13454)	Jul 10, 2019	1	3.4 Low
BDSA-2019-2074 (CVE-2019-13308)	Jul 10, 2019	1	3.4 Low

- Use the [Summary Dashboard](#) to view the overall health of the projects you have permission to view and identify areas of concern.

The screenshot shows the Summary Dashboard with the following sections:

- Top Policy Violations:** By Severity. Message: Good news! You have no policy violations.
- Project Security Risk:** A donut chart showing 100% Critical/High risk.
- Component Security Risk:** A donut chart showing 5% Critical/High risk.
- Top Components:** With Security Risk. Includes Apache Struts, Lo-Dash, qs - QS Querystring, minimatch, and jQuery.
- Statistics:** Includes counts for Projects, Versions, Vulnerabilities, Components, and Scanned Code.
- Project Policy Violations By Tier:** A line chart showing violations across Tier 1 to Tier 5 and Unknown phases.
- Projects with a critical/high vulnerability:** 5 projects.
- New vulnerable components this week:** 34 components.
- New projects created this week:** 5 projects.
- Project scanned this week:** 1 project.

**Note:** The Dashboard page that appears when you log in depends on the last main dashboard (**Projects**, **Components**, **Security**, or **Summary**) you viewed prior to previously logging out.

## Project version pages

- Use the [project version page/Components tab](#), also known as the project version BOM, to view the components, specific to that project version, that have security, license, and operational risk.

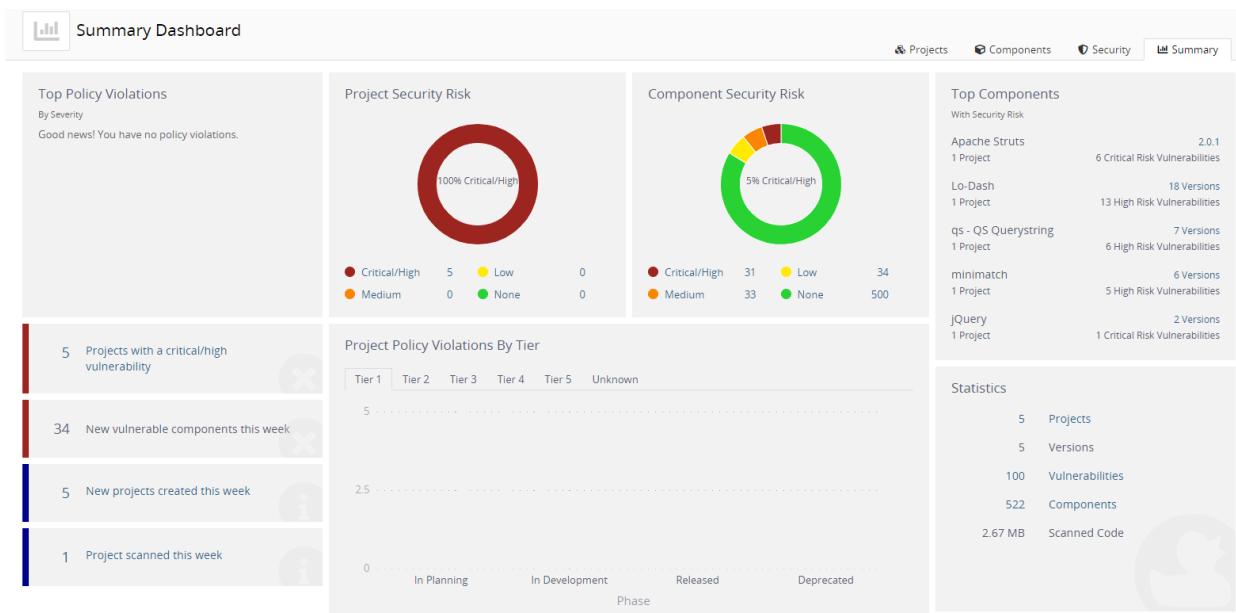
Component	Source	Match Type	Usage	License	Security Risk	Operational Risk
alpine-base 3.8.1	2 Matches	Exact File	Dynamically Linked	MIT	Low	
alpine-baselayout 3.1.0	22 Matches	Exact File, Files Modified, Exact Directory	Dynamically Linked	GPL-2.0	Medium	
alpine-keys 2.1	11 Matches	Exact File, Exact Directory	Dynamically Linked	MIT	Low	
asset v3.3.16	1 Match	Files Modified	Dynamically Linked	MIT	Low	
BusyBox 1.28.4	5 Matches	Exact File, Exact Directory	Dynamically Linked	GPL-2.0	Low	
bzip2 1.0.6	1 Match	Exact File	Dynamically Linked	Bzip2 License	High	
ca-certificates 20171114	33 Matches	Exact File, Exact Directory	Dynamically Linked	GPL-2.0+ or 1 more...	Low	
config v4.1.7	3 Matches	Exact Directory	Dynamically Linked	MIT	High	
contracts v1.0.2	9 Matches	Exact File/Exact Directory	Dynamically Linked	MIT	Low	

- Use [the project version page/ Security tab](#) to view the security vulnerabilities of each severity associated with the components used in a project version.

Identifier	Published	Overall Score ^	Status	Target date	Actual date
BDSA-2018-3897	Nov 6, 2018	1.7	Low	New	Never
NVD-CVE-2017-15396	Aug 28, 2018	4.3	Medium	New	Never
NVD-CVE-2017-15422	Aug 28, 2018	4.3	Medium	New	Never
NVD-CVE-2017-14952	Oct 16, 2017	7.5	High	New	Never
NVD-CVE-2017-17484	Dec 10, 2017	7.5	High	New	Never

## Viewing the health of your projects

Use the **Summary** tab to view the overall health of your projects and identify areas of concern. The page consists of widgets that provide business critical information which you can use to quickly assess areas where you need to focus your attention.



**Note:** The **Summary** tab only displays information for the projects you have permission to view.

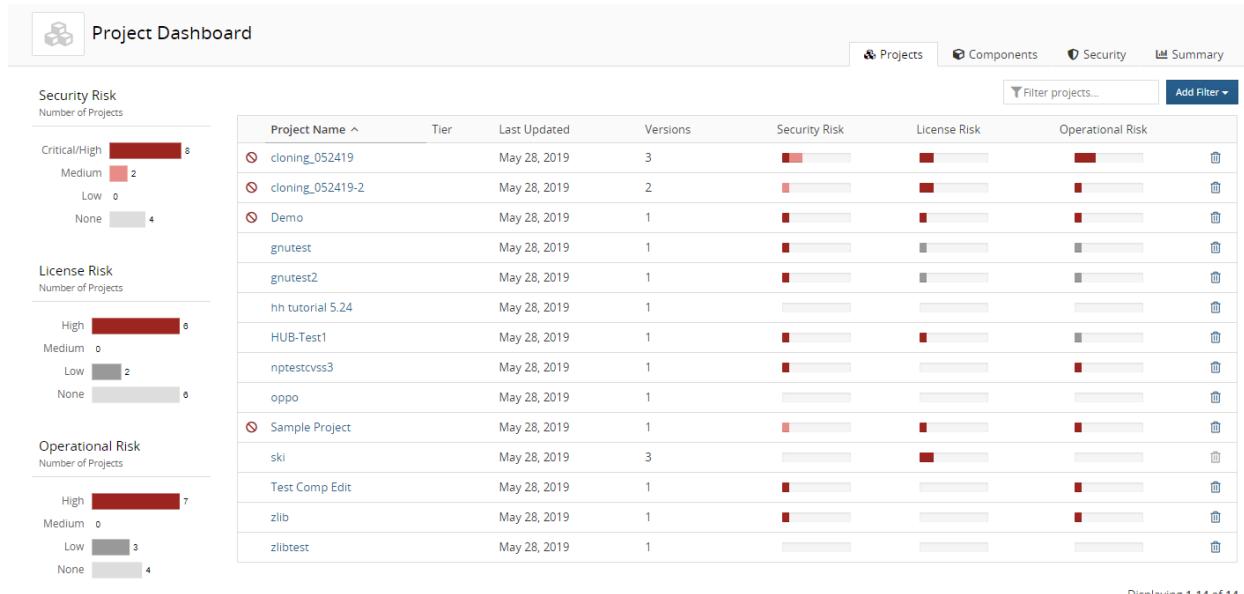
Description	More Information
<p>The <b>Top Policy Violations</b> widget displays up to the top five policy violations across all projects that you have permission to view. Policy rules are listed by severity level and then by the number of policy violations, in descending order. If policy rules do not have severity levels assigned to them, the widget displays the top five policy violations, in descending order by the number of violations.</p> <ul style="list-style-type: none"> <li>If you do not have the Policy Management module, this widget will not appear on the page.</li> <li>A message appears if you have the Policy Management module but do not have any policy rules configured or have any policy violations.</li> </ul>	Select a policy rule to view the <b>Projects</b> tab filtered to display the projects with a version that violate that policy rule.
<p>The <b>Project Security Risk</b> widget displays the number of projects you have permission to view for each level of security risk. Note that this widget counts the highest security risk level for a project, not all security levels affecting a project. For example, if a project has medium and low security risks, it is counted as a project with medium security risk; it is not included as a project with low security risks.</p>	Hover over the graph to view the number of projects with that level of security risk.
<p>The <b>Component Security Risk</b> widget displays the number of components in projects you have permission to view for each security risk level. Note that the widget counts only the highest security risk for a component. For example, if a component has medium and low security risks, it is counted as one component with a medium</p>	Hover over the graph to view the number of components with that level of security risk.

Description	More Information
security risk.	
<p>The <b>Top Components with Security Risk</b> widget displays up to the top five components used in the projects you have permission to view. The information shown for each component is:</p> <ul style="list-style-type: none"> <li>• Component name and number of versions used in your projects. If only one version is used, the specific version is listed here.</li> <li>• Number of your projects that have this component.</li> <li>• Number of security risks in this component, with the highest security risk listed here.</li> </ul> <p>Components are organized by security risk, with those components with the highest risk listed first.</p>	
<p>The <b>Projects have a critical/high vulnerability</b> widget displays the number of projects with versions that contain components with a critical and/or high security risk.</p>	Select the text to view the <b>Projects</b> tab filters to show the projects that have versions that have critical and/or high security risk.
<p>The <b>New vulnerable components this week</b> widget displays the number of components the Black Duck KB mapped a vulnerability to in the past seven days, including today.</p>	N/A.
<p>The <b>New projects created this week</b> widget displays the number of projects that you have permission to view that have been created in the past seven days, including today.</p>	Select the text to view the <b>Projects</b> tab which lists the projects created in the past week.
<p>The <b>Projects scanned this week</b> widget displays the number of projects with scans from the past seven days, including today.</p>	Select the text to view the <b>Projects</b> tab showing projects that have project versions with scans from the past week.

Description	More Information
<p>The <b>Project Policy Violations by Tier</b> widget displays the total number of projects by phase that have a policy violation, grouped by tiers.</p> <ul style="list-style-type: none"> <li>If you do not use tiers for your projects, projects are grouped in a single category called <b>Unknown</b>.</li> <li>If you do not have the Policy Management module, this widget displays <b>Projects by Tier</b>.</li> </ul>	For each tier, hover over a bar to see the number of projects in this phase and the number of projects in this phase with a policy violation.
<p>The <b>Statistics</b> widget displays the following information:</p> <ul style="list-style-type: none"> <li><b>Projects</b> lists the number of your projects.</li> <li><b>Versions</b> lists the number of project versions for your projects.</li> <li><b>Vulnerabilities</b> lists the number of vulnerabilities in your projects.</li> <li><b>Components</b> lists the number of components used in your projects, <i>including</i> ignored components.</li> <li><b>Scanned Code</b> lists the number of GBs scanned for all scans.</li> </ul>	<p>Select the projects value to view the <b>Projects</b> tab listing all projects you can view.</p> <p>Select the vulnerability value to view the <b>Security</b> tab filtered to show the vulnerabilities with a New, Needs Review, or Remediation Required status.</p> <p>Select the components value to view the <b>Components</b> tab showing all components used in the projects you can view. Note that this tab <i>excludes</i> ignored components.</p>

## Viewing overall risk for all projects

The Project Dashboard shows the overall risk across all projects where you are a project team member. On the left side of the page risk graphs show the number of projects that have each severity of security, license, and operational risk. This provides an overall view of risk across your projects. You can [use the risk graphs to filter](#) the BOM to show only the projects that have the selected severity and type of risk.



Click [here](#) for more information about using this page to understand security vulnerabilities associated with your projects.

**Tip:** To view the Project Dashboard from anywhere in the Black Duck UI, click the logo located in the top left corner of the UI.

## Understanding the types of project risk

There are three types of risk being assessed across all projects:

- **Security Risk.** Projects can have one of four categories of security risk, based on the vulnerabilities associated with the components that comprise the project.

Vulnerabilities are linked to components by the CVE numbers, as reported in the National Vulnerabilities Database (NVD) maintained by NIST or by Black Duck Security Advisories (BDSA) numbers.

Note that the security risk values shown use CVSS 2.0 or CVSS 3.0 scores, depending on which [security risk calculation you selected](#); by default CVSS 2.0 scores are shown.

Possible risk categories are:

- Critical/High. The project has critical and/or high severity vulnerabilities.
- Medium. The project has at least one component with at least one medium severity vulnerability.
- Low. The project has at least one component with at least one low severity vulnerability.
- None. All components in this project have no vulnerabilities.

- **License Risk.** Projects are assigned one of four categories of overall license risk:

- High. The project has at least one component with a high risk license.
- Medium. The project has at least one component with a medium risk license.
- Low. The project has at least one component with a low risk license.
- None. All components in this project do not have license risk.

Click [here](#) for more information on how license risk for a component is determined.

- **Operational Risk.** Operational risk is based on a combination of factors: (1) the strength of the component community, including the number of contributors and the level of commit activity; and (2) the number of newer versions of the component that are available than the one that is currently in use.

There are four categories of operational risk:

- High. The project has a version that has at least one component with high combined operational risk.
- Medium. The project has a version that has at least one component with medium combined operational risk.
- Low. The project has at least one component with low combined operational risk.
- None. All components in this project do not have operational risk.

There are three types of risk being assessed across all projects:

## Understanding the projects table

The projects table contains the following information:

Column	Description of the information displayed in the column
Project Name	Name of the project. Select the name to display the <i>Project Name</i> page. <b>Note:</b> Only projects where you are a project team member appear on this dashboard.
Tier	Importance of this project to your company. Tier 1 projects are defined as those that are most critical to the company, where Tier 5 projects are defined as least critical.
Last Updated	When this project was last updated. Hover over the value to see when a scan that was mapped to any project version was last run and when the BOM for any project version was last updated, either manually or by a new scan.
Versions	Number of versions that exist in the project. The risk numbers shown are cumulative across all components in the BOMs of all versions of your project.
Security Risk	Bar which shows the security risk for this project, across all versions.
License Risk	Bar which shows the license risk for this project, across all versions.
Operational Risk	Bar which shows the operational risk for this project, across all versions.

**Tip:** Click  to [delete the project](#) and all its associated version and BOM information from Black Duck.

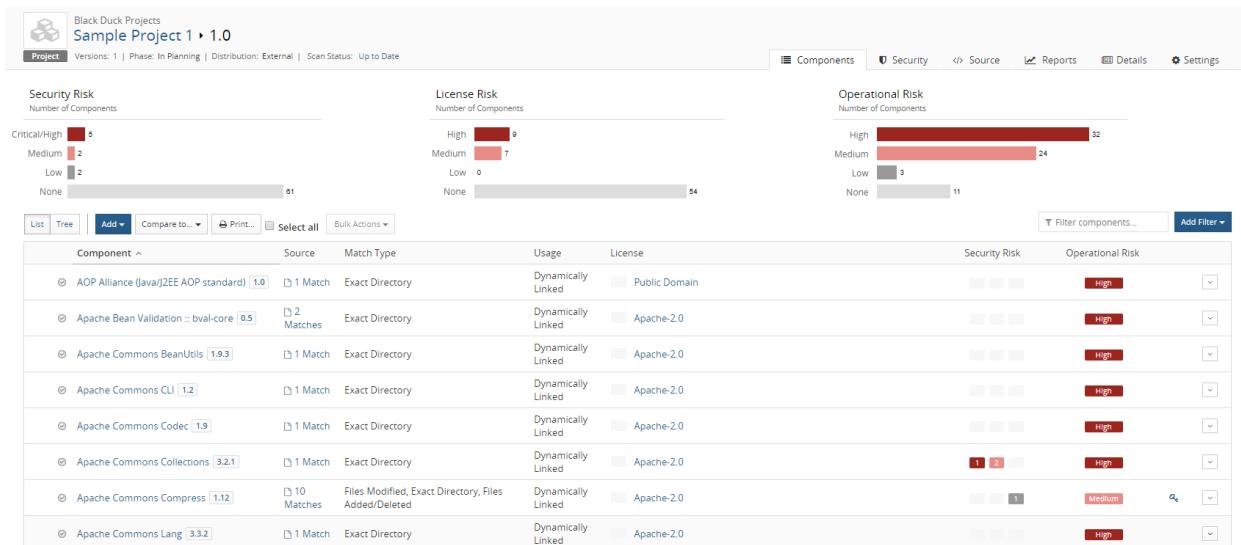
**Caution:** Once you delete a project, you cannot restore it. You can create another project with the same name, but the new project will not have any of the version or BOM information associated with the deleted project.

## Filtering the projects list by type and severity of risk

The number displayed before the risk severity bars in the risk graphs indicates the number of projects that contain at least one OSS component with that type of risk at that severity level. You can [use the risk graphs to filter](#) the projects table to show only your projects containing OSS components that have a specific type and severity of risk.

## Viewing overall risk at the project version level

Risk information for a specific project version is shown on the project version's **Components** tab.



Also known as the [BOM page](#), this tab shows all type of risks associated with each component in the project version's BOM.

There are three types of risk being assessed across all projects. As shown in the graphs at the top of the page:

- **Security Risk.** Risk values for project versions are based on the vulnerabilities associated with the components that comprise the project version's BOM.

Vulnerabilities are linked to components by the CVE numbers, as reported in the National Vulnerabilities Database (NVD) maintained by NIST or by Black Duck Security Advisories (BDSA) numbers.

Note that the security risk values shown use CVSS 2.0 or CVSS 3.0 scores, depending on which [security risk calculation you selected](#); by default CVSS 2.0 scores are shown.

Possible risk categories are:

- Critical/High. The number of components in this project version with critical and/or high severity vulnerabilities.
- Medium. The number of components in this project version's BOM with medium severity vulnerabilities.
- Low. The number of components in this project version's BOM with low severity vulnerabilities.
- None. The number of components in this project version's BOM with no vulnerabilities.

- **License Risk.** Project versions can have four levels of overall license risk:

- High. The number of components in this project version's BOM high risk license.
- Medium. The number of components in this project version's BOM with medium risk license.
- Low. The number of components in this project version's BOM with low risk license.
- None. The number of components in this project version's BOM with no license risk.

Click [here](#) for more information on how license risk for a component is determined.

- **Operational Risk.** Project versions can have four levels operational risk. Operational risk is based on a

combination of factors: (1) the strength of the component community, including the number of contributors and the level of commit activity; and (2) the number of newer versions of the component that are available than the one that is currently in use.

Project versions can have four categories of operational risk:

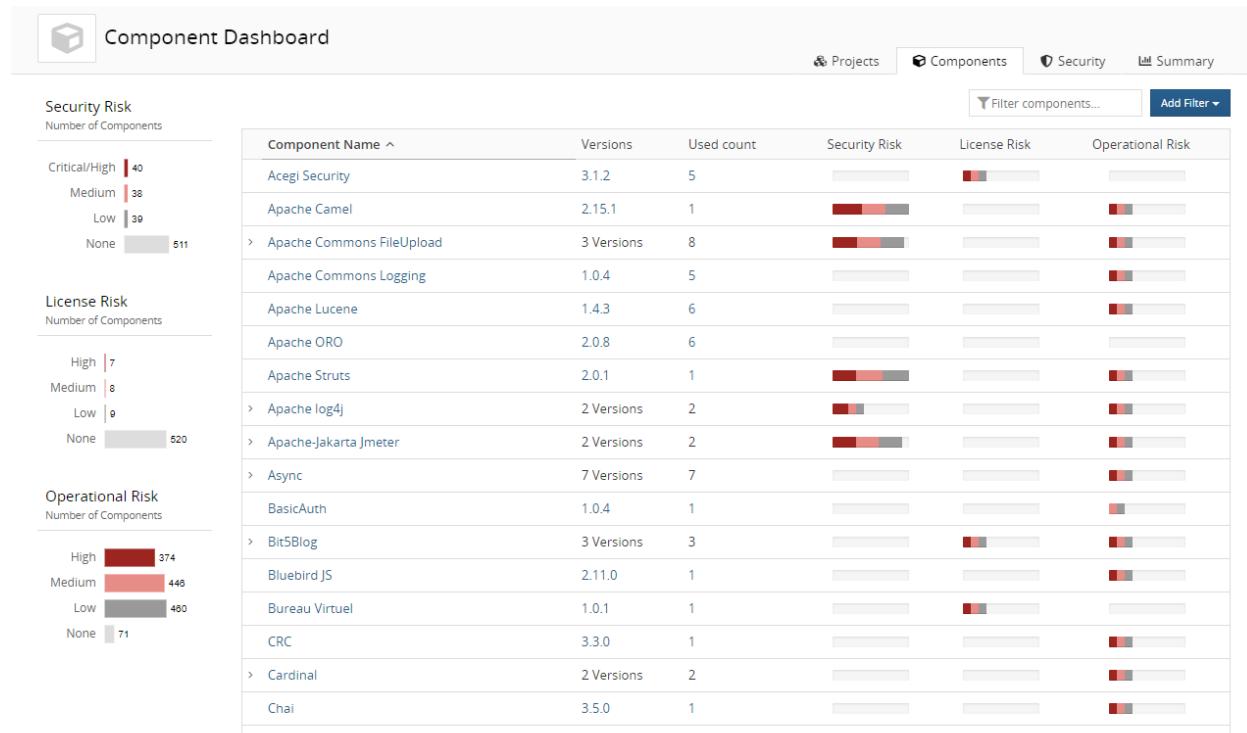
- High. The number of components in this project version's BOM with high combined operational risk.
- Medium. The number of components in this project version's BOM with medium combined operational risk.
- Low. The number of components in this project version's BOM with low combined operational risk.
- None. The number of components in this project version's BOM with no operational risk.

You can [use the risk graphs and table filters to filter](#) the BOM to show only components that have the selected severity and type of risk.

Click [here](#) for more information about understanding the BOM page.

## Viewing the risk associated with components used your projects

The Component Dashboard lists the components used in one or more of your projects. On the left side of the page risk graphs show the total number of components used in one or more of your projects and the security, license, and operational risks associated with these components. This is a focused view of risk for each of the components that comprise your projects.



**Note:** If a component is used in more than one of your projects, its associated risks in each category are counted only once in the total risk numbers shown in the risk graphs and in the components list.

Click [here](#) for more information about using this page to understand security vulnerabilities associated with your components.

## Understanding the types of component risk

There are three types of risk being assessed for components used in projects:

- **Security Risk.** Components are assigned one of four categories of security risk, based on the vulnerabilities associated with the versions in use in projects.

Vulnerabilities are linked to components by the CVE numbers, as reported in the National Vulnerabilities Database (NVD) maintained by NIST or by Black Duck Security Advisories (BDSA) numbers.

Note that the security risk values shown use CVSS 2.0 or CVSS 3.0 scores, depending on which [security risk calculation you selected](#); by default CVSS 2.0 scores are shown.

Possible risk categories are:

- Critical/High. The component has critical and/or high severity vulnerabilities.
- Medium. The component has medium severity vulnerabilities.
- Low. The component has low severity vulnerabilities.
- None. The component has no vulnerabilities.

- **License Risk.** Projects are assigned one of four categories of overall license risk:

- High. At least one version of the component in use in a project has a declared license that is high risk.
- Medium. At least one version of the component in use in a project has a declared license that is medium risk.
- Low. At least one version of the component in use in a project has a declared license that is low risk.
- None. At least one version of the component in use in a project has a declared license that is no risk.

Click [here](#) for more information on how license risk for a component is determined.

- **Operational Risk.** Operational risk is based on a combination of factors: (1) the strength of the component community, including the number of contributors and the level of commit activity; and (2) the number of newer versions of the component that are available than the one that is currently in use.

Project versions can have four categories of operational risk.

- High. The component has high combined operational risk and a version is used in at least one project.
- Medium. The component has medium combined operational risk and a version is used in at least one project.
- Low. The component has low combined operational risk and a version is used in at least one project.
- None. The component has no operational risk and a version is used in at least one project.

## Understanding the component table

The table contains the following information:

Column	Description
Component Name	Name of the component that is in use in one or more of your company's projects. Select the component name to display the <a href="#">Black Duck KB page</a> for that component. Click > for components with multiple versions to view a list of the versions used in your projects.
Versions	Number of versions of this component that are in use in one or more projects. Select the version number to display <a href="#">Black Duck KB Component Name Version page</a> .
Used Count	Number of project version BOMs that include a version this component.
Security Risk	Bar which shows the security risk for this component version.
License Risk	Bar which shows the license risk for this component version.
Operational Risk	Bar which shows the operational risk for this component version.

## Filtering the dashboard

The number displayed before the risk severity bars in the risk graphs indicates the number of components in use in at least one project that have that type of risk at that severity level. You can [use the risk graphs to filter](#) the OSS components list to show only those components that have a specific type and severity of risk.

# Chapter 10: About security risk

Black Duck helps security and development teams identify security risks across their applications.

By mapping vulnerabilities to your open source software, Black Duck can provide you with high-level overview information on security risk of your projects, along with detailed information on security vulnerabilities which you can use to investigate and remediate your security vulnerabilities.

Vulnerabilities are linked to the open source components by the Common Vulnerabilities and Exposures numbers (CVEs), as reported in the National Vulnerabilities Database (NVD) maintained by the National Institutes of Standards and Technology (NIST) and/or by (BDSA) numbers If you have licensed Black Duck Security Advisories.

## Security risk levels

NVD and BDSA use the Common Vulnerability Scoring System (CVSS) which provides a numerical score reflecting the severity of a vulnerability. The numerical score is then translated into a risk level to help you assess and prioritize security vulnerabilities.

Black Duck provides you with the option of viewing CVSS 2.0 or CVSS 3.0 scores. By default, Black Duck displays CVSS 2.0 scores.

- CVSS 2.0 scores has the following values:

- Low risk: 0.0 - 3.9
- Medium risk: 4.0 - 6.9
- High risk: 7.0-10.0

Note that Black Duck shows vulnerabilities with a 0.0 score as no risk.

Black Duck displays High risk vulnerabilities in the category labeled Critical/Hlgh.

- CVSS 3.0 scores has the following values:

- None: 0.0
- Low risk: 0.1 - 3.9
- Medium risk: 4.0 - 6.9
- High risk: 7.0 - 8.9
- Critical risk: 9.0 - 10.0

Note that if you select to view CVSS 3.0 scores, Black Duck displays Critical and High risk vulnerabilities together in one category labeled Critical/Hlgh. Use the filters to view critical or high vulnerabilities.

## Defining the default security risk calculation

Users with the system administrator role can redefine the order of security ranking that Black Duck uses to define the risk score and risk categories of security vulnerabilities. Black Duck uses the following order to calculate risk:

- If you have not licensed BDSA, the default order is:
  1. NVD 2.0
  2. NVD 3.0
- If you have BDSA licensed, the default order is:
  1. BDSA 2.0
  2. NVD 2.0
  3. BDSA 3.0
  4. NVD 3.0

As shown above, by default Black Duck defines security risk initially using CVSS 2.0 scores. You can modify the order by which Black Duck determines security risk so that CVSS 3.0 scores are used.

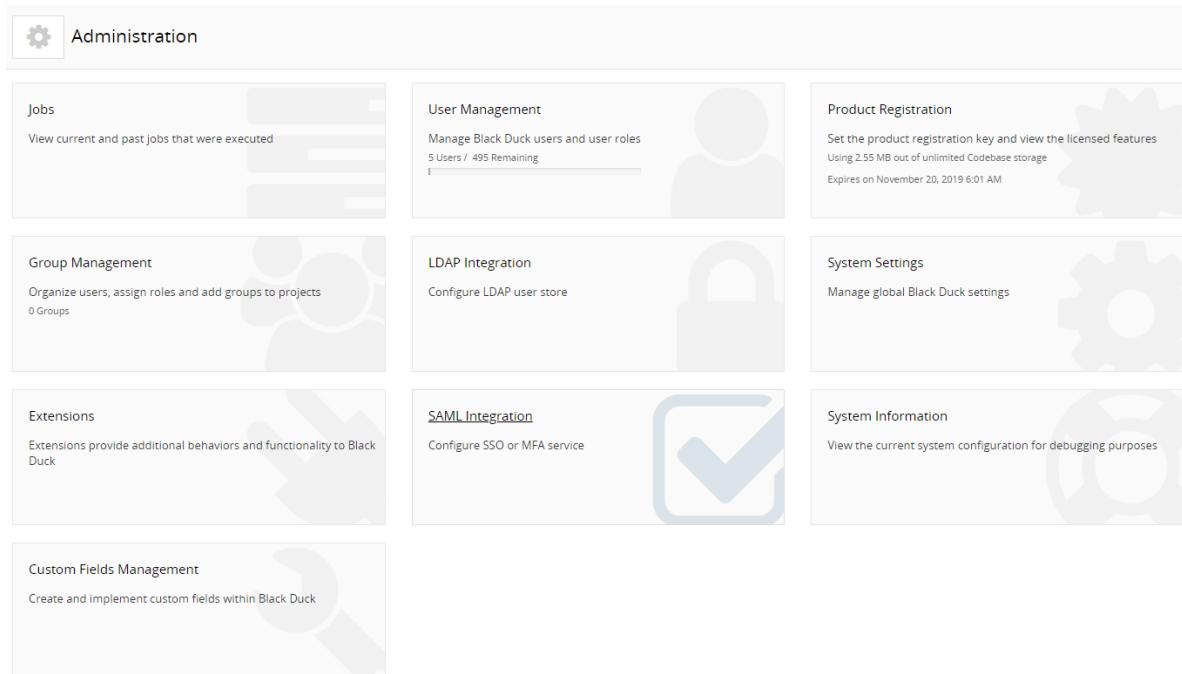
Note the following:

- Changing the order of the security risk configuration will result in revised security risk calculations for all project version BOMs and may result in new policy violations. These calculations may take a considerable amount of time to complete.
- The ability to change the security risk ranking is disabled if the security risk configuration has been reconfigured and jobs are running to recalculate security risk. Once the jobs are completed, the security risk ranking can be reconfigured.

### To configure the default security risk calculation

1. Log in to Black Duck with the System Administrator role.
2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.



### 3. Select **System Settings**.

The System Settings page appears.

The 'System Settings' page contains the following sections:

- Logo:** A logo for 'SYNOPSYS' is displayed, with a 'Upload logo' button.
- System Logs:** A link to download logs (.zip).
- Legal Tab Visibility:** A 'Enable' button.
- Security Risk Configuration Ranking:** A section for dragging and dropping tiles to rank security risk configurations. The tiles shown are 'BDSA 2.0', 'NVD 2.0', 'BDSA 3.0', and 'NVD 3.0'. A 'Save' button is located at the bottom right of this section.

4. In the **Security Risk Configuration Ranking** section, drag and drop the tiles so that the ranking is in the correct order.

5. Click **Save**.

A confirmation dialog box appears. Do one of the following:

- Click **Confirm**.

Two jobs, the VulnerabilityReprioritizationJob and the VulnerabilitySummaryFetchJob, start once you click **Confirm**.

Refresh the page to update the status of these jobs on this page. You can also view the status on the [Jobs page](#).

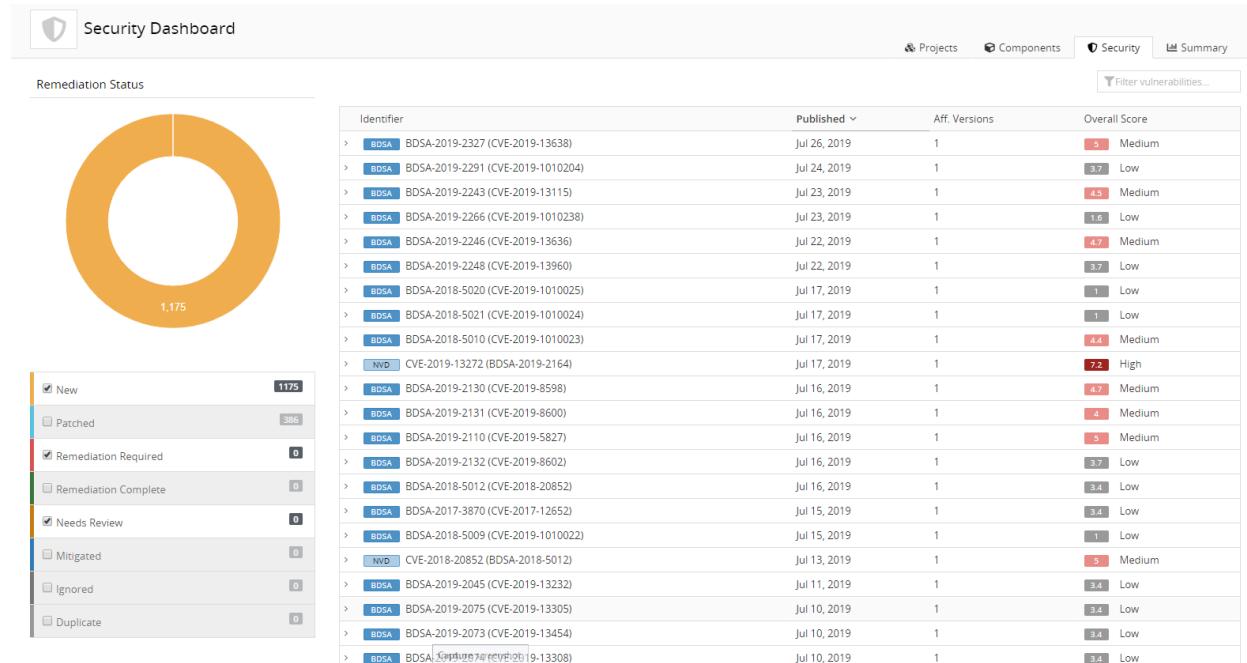
Once these jobs complete, the new security rankings appear in the Black Duck UI.

- Click **Cancel**.

The security risk configuration ranking returns to its previous order.

## Viewing all security vulnerabilities

Use the Security Dashboard to identify and manage risk. This dashboard lists all the security vulnerabilities that affect your projects.



Using the Security Dashboard is an efficient way to:

- Identify the remediation status of all the vulnerabilities in your projects.
- Review the severity of the vulnerability to determine if remediation is required.

### >To use the Security Dashboard to identify and manage risk

1. Log in to Black Duck.
2. From the Dashboard, click the **Security** tab to display the Security Dashboard.
3. You can use:
  - The table filter field to filter the vulnerabilities shown in the table by identifier.

- The **Aff. Versions** column to view the number of project versions affected by this vulnerability. Use this column to identify the vulnerabilities that are affecting the greatest number of versions of your projects.
  - The Remediation Status chart to view the remediation status of all vulnerabilities that exist within all projects and the number of vulnerabilities with each remediation status.
- By default, the chart displays all remediation statuses. Clear the check box to hide the vulnerabilities with that remediation status.
- The **Overall Score** column shows the Temporal score (for BDSA), or Base score (for NVD) and associated risk level. Hover over the **Overall Score** value to see the individual values.
    - For BDSA, the Temporal, Base, Exploitability, and Impact scores are shown.
    - For NVD, the Base, Exploitability, and Impact scores are shown.
  - The table to view more information on a vulnerability by selecting > next to the vulnerability that interests you.

Description	IBM Jazz Foundation (IBM Rational Engineering Lifecycle Manager 5.0 through 6.0.6) is vulnerable to cross-site scripting. This vulnerability allows users to embed arbitrary JavaScript code in the Web UI thus altering the intended functionality potentially leading to credentials disclosure within a trusted session. IBM X-Force ID: 152740.  <a href="#">View CVE record</a>	Base Score Metrics AV NETWORK AC MEDIUM Au SINGLE A NONE C NONE I PARTIAL	
		Published on Mar 14, 2019 Last Modified Apr 15, 2019	

**Note:** A single vulnerability can be present multiple times in the remediation status pie chart since it can have multiple different remediation types within a single BOM or across multiple project version BOMs. However, a single vulnerability is listed in only one row in the table.

## Viewing security vulnerabilities associated with your components

Use the Component Dashboard to view all components in your projects; components shown are top-level (parent) and subcomponents. The table lists the components used in one or more of your projects. On the left side of the page risk graphs show the total number of components used in one or more of your projects, which have each severity of security, license, and operational risks associated with them. From this page, you can drill down and view more information on these components and their vulnerabilities.

### ⚙️ To view vulnerabilities of components in your projects

1. Log in to Black Duck.
2. Select the **Components** tab to display the Component Dashboard.

**Component Dashboard**

Projects Components Security Summary

Filter components... Add Filter ▾

Component Name ^	Versions	Used count	Security Risk	License Risk	Operational Risk
Acegi Security	3.1.2	5	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Apache Camel	2.15.1	1	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Apache Commons FileUpload	3 Versions	8	<div style="width: 100%;">High</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Apache Commons Logging	1.0.4	5	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Apache Lucene	1.4.3	6	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Apache ORO	2.0.8	6	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Apache Struts	2.0.1	1	<div style="width: 100%;">High</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Apache log4j	2 Versions	2	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Apache-Jakarta Jmeter	2 Versions	2	<div style="width: 100%;">High</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Async	7 Versions	7	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
BasicAuth	1.0.4	1	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Bit5Blog	3 Versions	3	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Bluebird JS	2.11.0	1	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Bureau Virtuel	1.0.1	1	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
CRC	3.3.0	1	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Cardinal	2 Versions	2	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Chai	3.5.0	1	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>
Chalk	4 Versions	4	<div style="width: 10%;">Low</div>	<div style="width: 10%;">Medium</div>	<div style="width: 10%;">Low</div>

**Security Risk**  
Number of Components

Critical/High | 40  
Medium | 38  
Low | 39  
None | 511

**License Risk**  
Number of Components

High | 7  
Medium | 8  
Low | 9  
None | 520

**Operational Risk**  
Number of Components

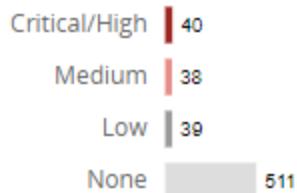
High | 374  
Medium | 448  
Low | 460  
None | 71

From this page:

- Use the **Security Risk** graph to view the total number of components, used in one or more of your projects, for each level of security risk.

### Security Risk

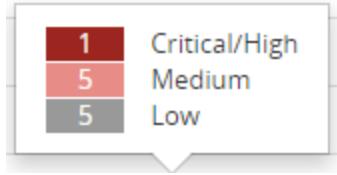
Number of Components



Select a value in the **Security Risk** graph to view the components that have that security risk level.

**Note:** This graph lists the number of components which have this level of security risk as their *highest* risk level – it is not the total number of components which have this risk level. For example, if you select to view components with a medium risk level, only those components that have medium as the highest risk level appear in the table; components that have both high *and* medium vulnerabilities are not shown.

- Select a bar in **Security Risk** column in the table to identify the components that have the greatest number of vulnerabilities.



For each version of a component, the values for each risk level are calculated as:

# of vulnerabilities \* the number of files affected by the vulnerability for each version of the project

For components that have multiple versions, the total value equals the sum of all versions.

3. Click > for components with multiple versions to view a list of the versions used in your projects.

4. Optionally, to view the vulnerabilities for a specific version of a component:

- Select a component name to view all versions of this component, along with a description:

Version	Used count	License	Released	Tags
20170327-snapshot-c11b2588	1	CDDL-1.0 and 6 more...	Mar 27, 2017	apache http
1.0.2.tc402	0	Apache-1.1	Jan 1, 2006	http_server java
1.0.tc40	0	Apache-1.1	Jan 1, 2006	jsp servlet
6.0.24-64.el6_5	0	Apache-2.0	Dec 22, 2016	web
6.0.24-72.el6_5	0	Apache-2.0	Dec 22, 2016	web_services
6.0.24.A	0	Apache-2.0	Oct 26, 2011	
6.0.26.A	0	Apache-2.0	Oct 26, 2011	

The **Used count** column shows the number of project versions that use this version of this component. A graph at the top of the page shows a history of high, medium, and low vulnerabilities for each version of this component.

- Select a component version to view a page which lists all projects and associated versions that use this version of this component. The number of vulnerabilities, a brief description, and associated licenses with this project also appear on this page.

**Description**  
The goal of the Apache Struts project is to encourage application architectures based on the "Model 2" approach, a variation of the classic Model-View-Controller (MVC) design paradigm. Under Model 2, a servlet (or equivalent) manages business logic execution, and presentation logic resides mainly in server pages.

Released	Newer Versions	Status	Updated
Oct 20, 2006	136	Unreviewed	May 27, 2019

**Activity**  
Last 12 Months: 335 commits (decreasing)  
Last commit: May 27, 2019

**Where Used**

Project	Version	Released	Phase
TestWP	test	Never	In Planning

Displaying 1-1 of 1

**43 Vulnerabilities**

**Licenses**  
Apache License 2.0

**Open Hub**  
<https://www.openhub.net/p/3569>

**Component Links**  
<http://struts.apache.org/>

**Tags**  
apache, development, framework, Java, model-view-controller, mvc, mvframework, programming, servlet, web, webapplication

Click the **Security** tab to view a list of the vulnerabilities for this version of the component.

Identifier	Published	Overall Score
> [NVD] CVE-2007-1358	May 9, 2007	2.6 Low
> [NVD] CVE-2009-2696	Aug 5, 2010	4.3 Medium
> [NVD] CVE-2013-4322	Feb 26, 2014	4.3 Medium
> [NVD] CVE-2007-2449	Jun 14, 2007	4.3 Medium
> [NVD] CVE-2014-0119	May 31, 2014	4.3 Medium
> [NVD] CVE-2006-7196	May 9, 2007	4.3 Medium
> [NVD] CVE-2014-0096	May 31, 2014	4.3 Medium
> [NVD] CVE-2013-4590	Feb 26, 2014	4.3 Medium
> [NVD] CVE-2014-0099	May 31, 2014	4.3 Medium
> [NVD] CVE-2005-4838	Dec 31, 2005	4.3 Medium
> [NVD] CVE-2000-1210	Mar 22, 2002	5 Medium
> [NVD] CVE-2014-0075	May 31, 2014	5 Medium
> [NVD] CVE-2012-5568	Nov 30, 2012	5 Medium
> [NVD] CVE-2001-0590	Aug 2, 2001	5 Medium
> [BDSA] BDSA-2016-0056	Jun 30, 2017	5 Medium
> [NVD] CVE-2008-0128	Jan 22, 2008	5 Medium
> [NVD] CVE-2013-4286	Feb 26, 2014	3.8 Medium

Click > to view more information on a vulnerability.

**Description**  
IBM Jazz Foundation (IBM Rational Engineering Lifecycle Manager 5.0 through 6.0.6) is vulnerable to cross-site scripting. This vulnerability allows users to embed arbitrary JavaScript code in the Web UI thus altering the intended functionality potentially leading to credentials disclosure within a trusted session. IBM X-Force ID: 152740.

[View CVE record](#)

**Base Score Metrics**

AV	NETWORK	A	NONE
AC	MEDIUM	C	NONE
Au	SINGLE	I	PARTIAL

Published on  
Mar 14, 2019  
Last Modified  
Apr 15, 2019

**Note:** The Authentication value is not available for CVSS 3.0 scores.

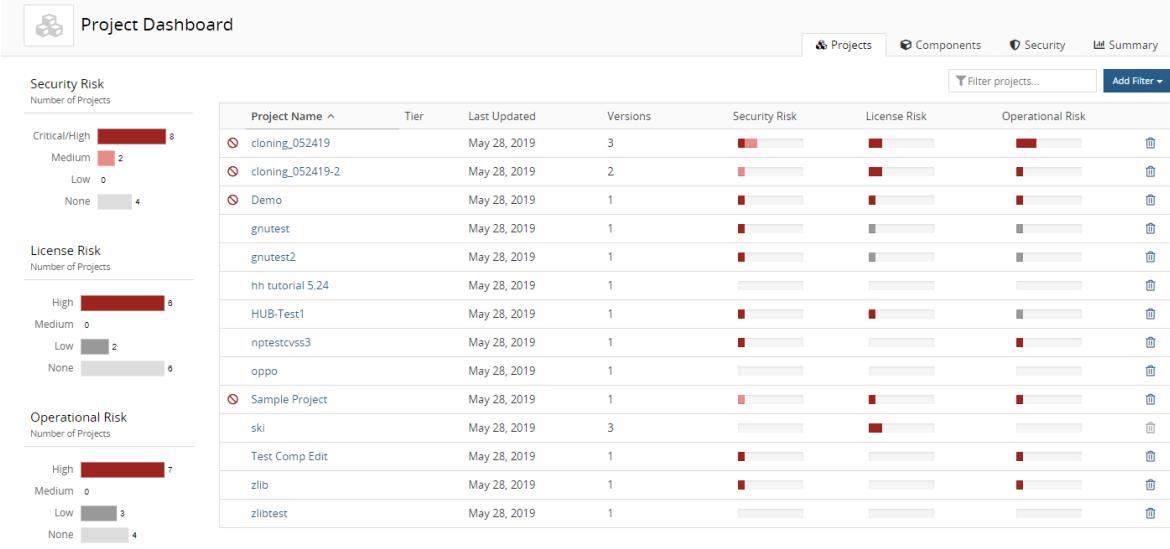
## Viewing the security vulnerabilities of your projects and project versions

Use the Project Dashboard to view the types and severity of risk that are associated with the components that

are in one or more versions of your projects. This dashboard provides an overall view of risk across all of your projects.

### To view the security vulnerabilities

1. Log in to Black Duck.
2. From the Dashboard, select the **Projects** tab to display the Project Dashboard.



The screenshot shows the Project Dashboard interface. On the left, there are three horizontal bar charts: 'Security Risk' (Critical/High: 8, Medium: 2, Low: 0, None: 4), 'License Risk' (High: 8, Medium: 0, Low: 2, None: 6), and 'Operational Risk' (High: 7, Medium: 0, Low: 3, None: 4). On the right, a table lists 14 projects with columns for Project Name, Tier, Last Updated, Versions, Security Risk, License Risk, and Operational Risk. The table includes rows for cloning\_052419, cloning\_052419-2, Demo, gnutest, gnutest2, hh tutorial 5.24, HUB-Test1, nptestcvss3, oppo, Sample Project, ski, Test Comp Edit, zlib, and zlibtest. A footer note indicates 'Displaying 1-14 of 14'.

**Tip:** You can also click the logo in the top left corner of the Black Duck UI to view the Project Dashboard.

From this page:

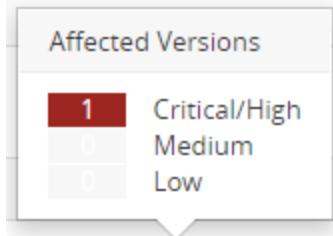
- Use the Security Risk graph to view the number of projects that have high, medium, low, or no security risk.



Select one or more values in the graph or use the filters at the top of the table to view the projects that have one or more security risk levels.

**Note:** The Security Risk graph displays the highest security risk level for a project, not all security levels affecting a project. Select a project name to open a page which lists all security risk levels for all versions of that project.

- Select a bar in **Security Risk** column in the table to see the number of versions of this project that are affected by a security risk.



Use this column to identify the vulnerabilities that are affecting the greatest number of your projects.

- Select a project name to view a page that lists all versions of this project.

Version	Phase	Last Updated	License	Security Risk	License Risk	Operational Risk
1.0	In Planning	Aug 23, 2018	Unknown License	■	■	■
2.0	In Planning	Aug 27, 2018	Unknown License	■	■	■

Displaying 1-2 of 2

Description  
No description.

Created  
Aug 13, 2018 by sysadmin

Updated  
Aug 14, 2018 by sysadmin

Tags  
No Tags

- Select a version with security risks to view a page which shows the BOM for this version of the project.

Component	Source	Match Type	Usage	License	Security Risk	Operational Risk
AOP Alliance (Java/J2EE AOP standard) 1.0	1 Match	Exact Directory	Dynamically Linked	Public Domain	■	■
Apache Bean Validation : bval-core 0.5	2 Matches	Exact Directory	Dynamically Linked	Apache-2.0	■	■
Apache Commons BeanUtils 1.9.3	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	■	■
Apache Commons CLI 1.2	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	■	■
Apache Commons Codec 1.9	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	■	■
Apache Commons Collections 3.2.1	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	■	■
Apache Commons Compress 1.1.2	10 Matches	Files Modified, Exact Directory, Files Added/Deleted	Dynamically Linked	Apache-2.0	■	■
Apache Commons Lang 3.3.2	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	■	■

- Use this page to view more information on the component and component version.

## Viewing project version vulnerabilities

Use the project version page's **Security** tab to view the security vulnerabilities associated with the components used in a project version.

The information shown uses CVSS 2.0 or CVSS 3.0 scores, depending on which [security risk calculation you selected](#); by default CVSS 2.0 scores are shown.

The screenshot shows the Black Duck Project Security page for a project named 'jade'. The top navigation bar includes tabs for Components, Security, Source, Reports, Details, and Settings. The main area features a 'Security Risks' graph on the left, which displays the number of components with critical, high, medium, low, and none vulnerabilities. Below the graph is a table of components and their vulnerabilities. On the right, there is a 'Known Vulnerability' section with two cards: one for 'ICU for Java (ICU4J) 59.1' and another for 'Most recent'. Both cards show an upgrade path from 63.2 to 64.2, both of which have no known vulnerabilities. Below these cards is a table of known vulnerabilities, each with columns for Identifier, Published date, Overall Score (CVSS 2.0), Status, Target date, and Actual date. The table lists five entries, all of which are new and have never been targeted. At the bottom right of the page, it says 'Displaying 1-5 of 5'.

Identifier	Published	Overall Score	Status	Target date	Actual date
BDSA-2018-3897	Nov 6, 2018	1.7 Low	New	Never	Never
NVD-CVE-2017-15396	Aug 28, 2018	4.3 Medium	New	Never	Never
NVD-CVE-2017-15422	Aug 28, 2018	4.3 Medium	New	Never	Never
NVD-CVE-2017-14952	Oct 16, 2017	7.5 High	New	Never	Never
NVD-CVE-2017-17484	Dec 10, 2017	7.5 High	New	Never	Never

This page has three sections:

- Security Risks graph and component/vulnerabilities table.
- Vulnerabilities table.
- Remediation guidance section, shown above the vulnerability table. Click [here](#) for more information about this feature.

The Security Risks graph shows how many vulnerabilities of each severity are associated with each component version and subproject used in this version of the project. Each component listed in the component/vulnerabilities table includes risk bars that show how many vulnerabilities of each severity exists in that component version or subproject.

In this section you can:

- Select a severity level in the Security Risks graph to view all components that share the same level of risk.

**Note:** This graph lists the number of components which have this level of security risk as their *highest* risk level – it is not the total number of components which have this risk level. For example, if you select to view components with a medium risk level, only those components that have medium as the highest risk level appear in the table; components that have both high and medium vulnerabilities are not shown.

- Use the **Filter components** field to limit the components or subprojects shown.

To view the vulnerabilities for a component, select the component in the component/vulnerabilities table; a list

of vulnerabilities appears in the vulnerabilities table. To view vulnerabilities for a subproject, if you have permission to view this project, select the subproject name in the component/vulnerabilities table, then select the link shown on the page which displays the vulnerabilities for the subproject.

The vulnerabilities table lists detailed information for each vulnerability:

Column	Description
Identifier	<p>The identifier and value associated with this vulnerability.</p> <p>Select &gt; in the table next to the vulnerability to view a brief description. Depending on the identifier, select to view <a href="#">the BDSA record</a> and/or <a href="#">the CVE record</a>.</p> <p>Users with the appropriate role can also use this section to <a href="#">remediate the vulnerability</a>.</p>
Published	Date on which the vulnerability was first published.
Overall Score	<p>Shows the Temporal score (for BDSA), or Base score (for NVD) and associated risk level. Hover over the Overall Score value to see the individual values.</p> <ul style="list-style-type: none"> <li>For BDSA, the Temporal, Base, Exploitability, and Impact scores are shown.</li> <li>For NVD, the Base, Exploitability, and Impact scores are shown.</li> </ul> <p>The Temporal score represents time-dependent qualities of a vulnerability taking into account the confirmation of the technical details of a vulnerability, the existence of any patches or workarounds, and the availability of exploit code or techniques.</p> <p>The Base score reflects the overall basic characteristics of a vulnerability that are constant over time and user environments:</p> <ul style="list-style-type: none"> <li>Access Vector (AV) - CVSS 2.0 / Attack Vector (AV) - CVSS 3.0</li> <li>Access Complexity (AC) - CVSS 2.0 / Attack Complexity (AC) - CVSS 3.0</li> <li>Authentication (Au)</li> <li>Integrity (I)</li> <li>Availability (A)</li> <li>Confidentiality (C)</li> </ul> <p><b>Note:</b> The Authentication value is not available for CVSS 3.0 scores.</p> <p>The Exploitability score measures how the vulnerability is accessed and if extra conditions are required to exploit it, taking into account access vector, complexity, and authentication.</p> <p>The Impact score reflects the possible impact of successfully exploiting the vulnerability, considering the integrity, availability, and confidentiality impacts.</p>
Status	<a href="#">Remediation status</a> of this vulnerability. Possible values are: Duplicate, Ignored, Needs Review, New, Mitigated, Patched, Remediation Complete, or Remediation Required.

Column	Description
Target date	Target date for remediating this vulnerability.
Actual date	Actual date this vulnerability was remediated.

## Viewing vulnerability details

Black Duck provides detailed information on a security vulnerability depending on whether you are viewing:

- [BDSA record](#)
- [CVE record](#)

## Black Duck Security Advisories

Black Duck Security Advisories (BDSAs) are a Black Duck-exclusive vulnerability data feed sourced and curated by our Security Research team, part of the Black Duck Centre of Open Source Research & Innovation (COSRI). BDSAs offer deeper coverage for a wider set of vulnerabilities than is available through the National Vulnerability Database (NVD), and provide detailed vulnerability insight, including severity, impact, exploitability metrics, and actionable remediation guidance.

Black Duck Security Advisory  
Apache Tomcat Vulnerable to Reflected Cross-Site Scripting (XSS) via SSI 'printenv' Debugging Command  
BDSA-2019-1661 | CVE-2019-0221 | Published May 30, 2019 | Updated May 30, 2019

Overview Affected Projects Technical CVE References

LOW 7.2 BDSA Fix Available Mar 10, 2019 Exploit Available May 26, 2019 18 Days Age

Apache Tomcat is vulnerable to reflected cross-site scripting (XSS) due to improper validation of user-supplied input in server-side includes (SSI) commands. This could allow an attacker to inject arbitrary web scripts and steal sensitive information such as authentication tokens or user cookies.

**How to fix it**

**Solution - Fix Available**

Fixed in:

- 9.0.18 by this commit,
- 8.5.40 by this commit,
- 7.0.94 by this commit.

The latest stable releases are available [here](#).

**Workaround**

If upgrading to a fixed version cannot be performed immediately, the vendor recommends disabling SSI in `conf/web.xml`.

**Common Vulnerability Scoring System (CVSS)**

Score Type	Score
2.0	7.2
3.0	7.2
NVD 2.0	8
NVD 3.0	5.8

Overall: 7.2

Detailed description: A bar chart titled 'Common Vulnerability Scoring System (CVSS)' showing scores across five categories. The y-axis ranges from 1 to 10. The 'Overall' bar is at 7.2, 'Temporal' is at 7.2, 'Base' is at 8, 'Exploitability' is at 1.6, and 'Impact' is at 5.8. Buttons for '2.0', '3.0', 'NVD 2.0', and 'NVD 3.0' are at the top of the chart area.

To view a BDSA record:

- Use the Search feature to locate BDSAs.

For example, search for BDSA-2017 to see the list of Black Duck Security Advisories from 2017.

Select a BDSA to view the record.

- Use the **Security** tab for a project version to view the vulnerabilities for a project version BOM.

The BDSA identifier (**BDSA**) indicates those vulnerabilities with a BDSA record.

Click > to view a description of the vulnerability and select **View BDSA record**.

- Use the Security Dashboard to find and select to view a BDSA record, as described above.

**Tip:** Use your browser print feature to print the information shown in a tab,

## Overview tab

By default, the **Overview** tab appears and displays the following information:

- The title bar displays the name of the vulnerability, BDSA number, CVE number (if there is a related CVE vulnerability), a published date (also known as the disclosure date), and an updated date (the last time the record was updated by NVD or BDSA).
- At the top of the page, the following information appears:



Apache Tomcat is vulnerable to reflected cross-site scripting (XSS) due to improper validation of user-supplied input in server-side includes (SSI) commands. This could allow an attacker to inject arbitrary web scripts and steal sensitive information such as authentication tokens or user cookies.

Shown here are the:

- BDSA score. Score based on analysis by Black Duck Software security analysts, who further investigated the vulnerability and provided a more detailed and accurate score. This includes the temporal score.

- Date of an available fix (if there is a fix available).
  - Whether there is an exploit for this vulnerability.
  - Age. Today's date - Disclosure date.
  - A brief description of the vulnerability.
- The **How to fix it** section describes a solution, if one is available, and a workaround.
  - The **Common Vulnerability Scoring System (CVSS)** section displays the CVSS 2.0, CVSS 3.0, and if available, NVD 2.0 and NVD 3.0 scores.

#### Common Vulnerability Scoring System (CVSS)

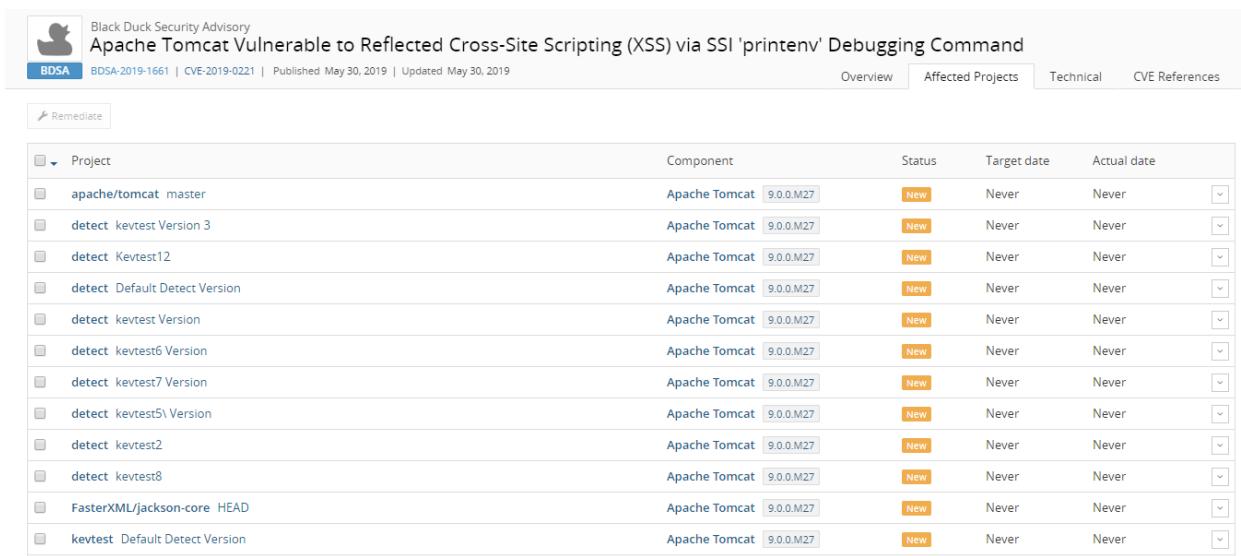


Select a value above the graph to view the information in the graph and details below.

**Note:** For more information on vulnerability metrics, visit the NVD web site: <https://nvd.nist.gov/vuln-metrics>

#### Affected Projects tab

Select this tab to see a list of your projects that are affected by this vulnerability.



The screenshot shows a table titled "Affected Projects" under the "Affected Projects" tab. The table has columns for "Project", "Component", "Status", "Target date", and "Actual date". The "Status" column contains mostly "New" entries, with one "Apache Tomcat" entry showing "9.0.0.M27". The "Target date" and "Actual date" columns are all set to "Never".

Project	Component	Status	Target date	Actual date
apache/tomcat master	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest Version 3	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_Kevtest12	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_Default Detect Version	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest Version	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest6 Version	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest7 Version	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest5\ Version	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest2	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest8	Apache Tomcat 9.0.0.M27	New	Never	Never
FasterXML/jackson-core HEAD	Apache Tomcat 9.0.0.M27	New	Never	Never
kevtest Default Detect Version	Apache Tomcat 9.0.0.M27	New	Never	Never

This tab lists all projects affected by this vulnerability:

- Project name and version affected by this vulnerability.
- Component name and version that contains this vulnerability.
- Remediation status of this vulnerability. Possible values are: New, Needs review, Mitigated, Patched, Duplicate, Remediation Required, Remediation Complete, or Ignored.
- Target date for remediating this vulnerability.
- Actual date this vulnerability was remediated.

In this tab:

- Select multiple projects to apply the same remediation status and click **Remediate** to open the Bulk remediation dialog box. Enter the remediation details and click **Update**.

To remediate a single project, select it and then select  in the row of the project version. Select **Update Remediation Plan** to display the Update Remediation Plan dialog box. Enter the remediation details and click **Update**

- Select  in the row of a project version and select:
  - **View all vulnerabilities** to view all vulnerabilities affecting this project version.
  - **View related files to view** to display the **Source** tab filtered to display the affected files.

## Technical tab

Select the **Technical** tab to view a technical description and a list of references and related links.

 Black Duck Security Advisory  
Apache Tomcat Vulnerable to Reflected Cross-Site Scripting (XSS) via SSI 'printenv' Debugging Command  
BDSA-2019-1661 | CVE-2019-0221 | Published May 30, 2019 | Updated May 30, 2019

Overview Affected Projects Technical [CVE References](#)

**Technical Description**

Tomcat does not sanitize user-supplied variables in the SSI `printenv` debugging command within the file `java/org/apache/catalina/ssi/SSIPrintenv.java`. An attacker could exploit this on a Tomcat instance which has enabled SSI (disabled by default), and enabled the `printenv` directive for debugging purposes within `ssi/printenv.shtml`. For such an instance, the attacker could craft a malicious URL to be supplied to a victim, which if accessed will result in included web scripts being executed on their system.

**References and Related Links**

 **Advisories**  
<https://lists.apache.org/thread.html/6e6e9eacf7b28fd63d249711e9d3cc4e0a83f556e324ae37be5a8c@%3Cannounce.tomcat.apache.org%3E>  
 <https://www.nightwatchcybersecurity.com/2019/05/27/xss-in-ssi-printenv-command-apache-tomcat-cve-2019-0221/>

 **Vendor Upgrade**  
<http://tomcat.apache.org/>  
<https://github.com/apache/tomcat/releases/tag/7.0.94>  
<https://github.com/apache/tomcat/releases/tag/8.5.40>  
<https://github.com/apache/tomcat/releases/tag/9.0.18>

 **Patch**  
<https://github.com/apache/tomcat/commit/15fcfd166ea2c1bb79e8541b8e1a43da9c452ceea>  
<https://github.com/apache/tomcat/commit/44ec74c44dcd05cd7e90967c04d40b51440ecd7e>  
<https://github.com/apache/tomcat/commit/4fcfd706f3ecf35912a60024289637f5acb32da>

Included in the **References and Related Links** section is a list of Key Events:

- Discovered. Date that the vulnerability was discovered.
- Vendor Notified. Date the official vendor was notified of this vulnerability.
- Vendor Fix. Date that the official vendor released a patch or upgrade to fix this vulnerability.
- Disclosure. Date the vulnerability was first publicly disclosed, whether as a bug or as a security vulnerability.
- Exploit Available. Date an exploit became publicly available for this vulnerability.

## CVE References tab

Select the **CVE References** tab to view links for additional information.

The screenshot shows a security advisory page from Black Duck. At the top, it displays the title "Apache Tomcat Vulnerable to Reflected Cross-Site Scripting (XSS) via SSI 'printenv' Debugging Command". Below the title, there's a navigation bar with tabs: Overview, Affected Projects, Technical, and CVE References. The "Overview" tab is selected. On the left, there's a sidebar with a "All" button and three other buttons: CONFIRM, FULLDISC, and MLIST. Each button has a small number next to it (1, 1, 1 respectively). The main content area is divided into sections: "CONFIRM" (with a link to a thread on the Apache mailing list), "FULLDISC" (with a link to a full disclosure post on seclists.org), and "MLIST" (with a link to a message on the Debian LTS announce mailing list).

## CVE record

Vulnerabilities are linked to components by the Common Vulnerabilities and Exposures numbers (CVEs), as reported in the National Vulnerabilities Database (NVD) maintained by the National Institutes of Standards and Technology (NIST).

The CVE record provides overview information on a vulnerability, a list of affected projects, and links to references.

### Overview tab

By default, the **Overview** tab appears and displays the following information:

- The title bar displays the CVE number, a published date (date that NVD published the CVE), and an updated date (last modified date by NVD).
- A description of the vulnerability.  
If there is a BDSA record, select the link to view this information.
- The **Common Vulnerability Scoring System (CVSS)** section displays the CVSS 2.0 and CVSS 3.0 scores.

### Common Vulnerability Scoring System (CVSS)



Select a value above the graph to view the information in the graph and details below.

**Note:** For more information on vulnerability metrics, visit the NVD web site: <https://nvd.nist.gov/vuln-metrics>

### Affected Projects tab

Select this tab to see a list of your projects that are affected by this vulnerability.

National Vulnerability Database CVE-2018-8014		Overview	Affected Projects	References
Published May 16, 2018   Updated Apr 15, 2019   <a href="https://nvd.nist.gov/vuln/detail/CVE-2018-8014">https://nvd.nist.gov/vuln/detail/CVE-2018-8014</a>				
<a href="#">Remediate</a>				
Project	Component	Status	Target date	Actual date
apache/tomcat master	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest6 Version	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest7 Version	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_Kevtest12	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest Version	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_Default Detect Version	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest5\ Version	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest2	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest Version 3	Apache Tomcat 9.0.0.M27	New	Never	Never
detect_kevtest8	Apache Tomcat 9.0.0.M27	New	Never	Never

This tab lists all projects affected by this vulnerability:

- Project name and version affected by this vulnerability.
- Component name and version that contains this vulnerability.

- Remediation status of this vulnerability. Possible values are: New, Needs review, Mitigated, Patched, Duplicate, Remediation Required, Remediation Complete, or Ignored.
- Target date for remediating this vulnerability.
- Actual date this vulnerability was remediated.

In this tab:

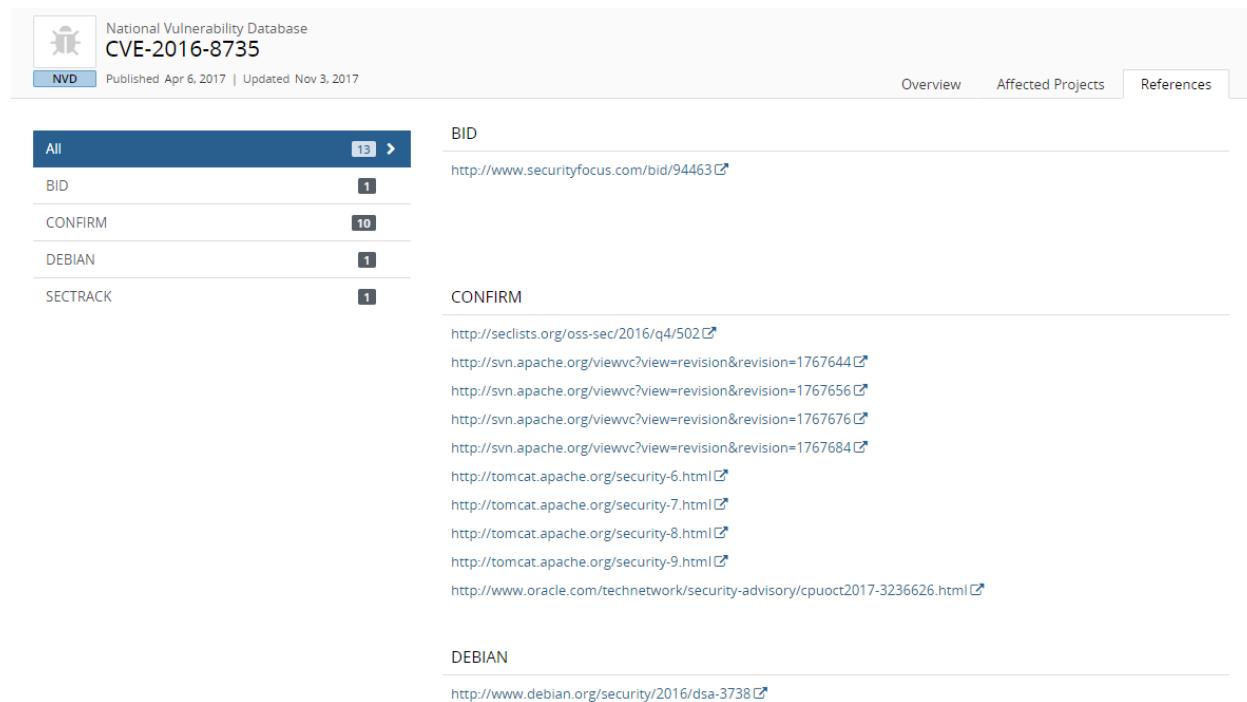
- Select multiple projects to apply the same remediation status and click **Remediate** to open the Bulk remediation dialog box. Enter the remediation details and click **Update**.

To remediate a single project, select it and then select  in the row of the project version. Select **Update Remediation Plan** to display the Update Remediation Plan dialog box. Enter the remediation details and click **Update**

- Select  in the row of a project version and select:
  - **View all vulnerabilities** to view all vulnerabilities affecting this project version.
  - **View related files to view** to display the **Source** tab filtered to display the affected files.

## References tab

Select the **References** tab to view links for additional information.



National Vulnerability Database  
CVE-2016-8735  
Published Apr 6, 2017 | Updated Nov 3, 2017

Overview Affected Projects References

All	13
BID	1
CONFIRM	10
DEBIAN	1
SECTRACK	1

**BID**  
<http://www.securityfocus.com/bid/94463>

**CONFIRM**  
<http://seclists.org/oss-sec/2016/q4/502>  
<http://svn.apache.org/viewvc?view=revision&revision=1767644>  
<http://svn.apache.org/viewvc?view=revision&revision=1767656>  
<http://svn.apache.org/viewvc?view=revision&revision=1767676>  
<http://svn.apache.org/viewvc?view=revision&revision=1767684>  
<http://tomcat.apache.org/security-6.html>  
<http://tomcat.apache.org/security-7.html>  
<http://tomcat.apache.org/security-8.html>  
<http://tomcat.apache.org/security-9.html>  
<http://www.oracle.com/technetwork/security-advisory/cpuoct2017-3236626.html>

**DEBIAN**  
<http://www.debian.org/security/2016/dsa-3738>

## Remediating security vulnerabilities

Vulnerabilities have a remediation status assigned to them. A new vulnerability can have a status of **New**, **Needs Review**, **Patched**, or **Duplicate**.

The following table describes each remediation status and whether a vulnerability with this status is included in the security risk calculations:

Remediation Status	Included in Security Risk Calculation?	Definition
New	Yes	Black Duck has determined that a vulnerability affects this component version.
Needs Review	Yes	Black Duck cannot determine if a vulnerability definitely affects this component version.  This can occur when a component version is known to contain a vulnerability, but it cannot be determined whether the patch or sub-version being used is affected by this vulnerability.
Remediation Required	Yes	Remediation is required for the component version.
Remediation Complete	No	Remediation for the vulnerability is complete.
Duplicate	No	This vulnerability is a duplicate.
Mitigated	No	The vulnerability has been mitigated.
Patched	No	The vulnerability in this version of a Linux distribution package has been patched.  Although a vulnerability has been reported on the overall component version, the vulnerability does not affect this specific matched version as the version has been patched from the source from where it came.
Ignored	No	The vulnerability has been ignored.

## Remediating a vulnerability

You may wish to change the remediation status after reviewing the severity of the vulnerability. Black Duck helps you determine which version you should use when a component has a vulnerability. Black Duck helps you to understand your options when a component has a security vulnerability.

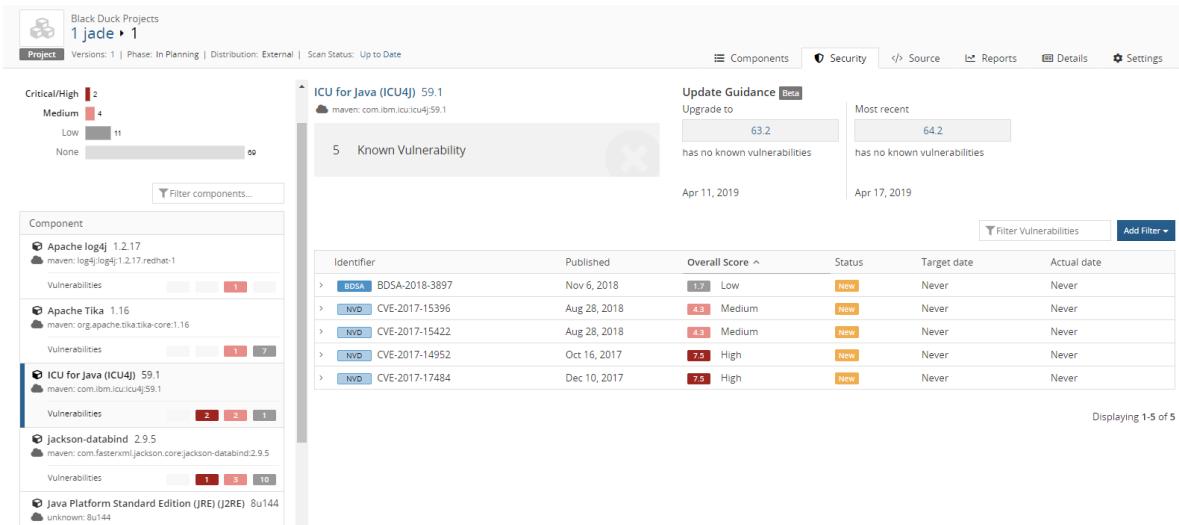
**Note:** You can select any value for the remediation status. Selecting **Remediation Complete**, **Mitigated**, **Patched**, or **Ignored** removes the vulnerability from the security risk calculations.

Only users with the appropriate [role](#) can remediate vulnerabilities.

## To remediate vulnerabilities

Use this method to identify and remediate the vulnerabilities affecting a specific project version.

1. Log in to Black Duck.
2. Select the name of the project using the **Projects** tab on the Dashboard page to go to the *Project Name* page.
3. Select the version name to open the **Components** tab and view the BOM.
4. Select the **Security** tab which lists all components and subprojects with associated security vulnerabilities for this project version.



Identifier	Published	Overall Score ^	Status	Target date	Actual date
> BDSA BDSA-2018-3897	Nov 6, 2018	1.7 Low	New	Never	Never
> NVD CVE-2017-15396	Aug 28, 2018	4.3 Medium	New	Never	Never
> NVD CVE-2017-15422	Aug 28, 2018	4.3 Medium	New	Never	Never
> NVD CVE-2017-4952	Oct 16, 2017	7.5 High	New	Never	Never
> NVD CVE-2017-17484	Dec 10, 2017	7.5 High	New	Never	Never

5. Select a component in the table on the left to view the associated vulnerabilities.

Black Duck provides [remediation guidance](#) for components with security vulnerabilities.

6. Select > in the table next to the vulnerability to view a brief description. You can also hover over the **Overall Score** value to see a breakdown of the score.

- To remediate the vulnerability: enter a different status, additional remediation details, such as the target an actual date, and click **Update**.
- To view additional information: select to view [the BDSA record](#) or [the CVE record](#).

View the projects affected by this vulnerability by selecting the **Affected Projects** tab.

Use this tab to remediate the vulnerability for this and/or additional projects:

- Select a single project, Select , select **Update Remediation Plan**, enter the remediation details, and click **Update**.
- Select multiple projects that need the same remediation status. Click **Remediate**, enter the remediation details, and click **Update**. This is also known as a bulk remediation.

## Getting remediation guidance for components with security vulnerabilities - BETA

Black Duck informs you of the vulnerabilities that impact the components in your BOMs. Detailed information is provided for each vulnerability, including a description and vulnerability scores.

After reviewing this information, you may need guidance as to what other component versions are available and whether there is a version that fixes the security vulnerability that affects the component version used in your BOM.

Black Duck provides this information: for a security vulnerability in your BOM, Black Duck displays three possible versions of the component that are available to you:

- The next current version that fixes the vulnerabilities found in the version used in your BOM. This version does not have *any* of the vulnerabilities present in the version *currently used in your BOM*.
- The next current version that does not have *any* reported vulnerabilities, if available.
- The most current version of the component by release date.

For each suggestion, select the version number to open the [Component Name Version](#) page.

Use this information to guide you in determining [how to remediate](#) a security vulnerability.

### To view guidance information

1. Select the name of the project using the **Projects** tab on the Dashboard page to go to the *Project Name* page.
2. Select the version name to open the **Components** tab and view the BOM.
3. Select the **Security** tab which lists all components and subprojects with associated security vulnerabilities for this project version.
4. Select a component from the **Component** table on the left side of the page to view a table which lists the vulnerabilities for this component and provides more information on each vulnerability. Above the table

are the suggestions of versions you can use to replace the selected component.

The screenshot shows the Black Duck Software Composition Analysis interface. At the top, it displays the project name 'jade' and its status: '1 Versions: 1 | Phase: In Planning | Distribution: External | Scan Status: Up to Date'. Below this, there's a navigation bar with links for Components, Security, Source, Reports, Details, and Settings.

On the left, a sidebar lists components with their names, versions, and vulnerability counts:

- Apache log4j 1.2.17 (maven: log4j:log4j:1.2.17.redhat-1) - 1 vulnerability
- Apache Tika 1.16 (maven: org.apache.tika:tika-core:1.16) - 7 vulnerabilities
- ICU for Java (ICU4J) 59.1 (maven: com.ibm.icu:icu4j:59.1) - 2 vulnerabilities
- Jackson-databind 2.9.5 (maven: com.fasterxml.jackson.core:jackson-databind:2.9.5) - 10 vulnerabilities
- Java Platform Standard Edition (JRE) (J2RE) 8u144 (unknown: 8u144) - 0 vulnerabilities

The main content area is focused on the 'ICU for Java (ICU4J) 59.1' component. It shows the following details:

- Update Guidance:** Upgrade to version 64.2 (Most recent).
- Known Vulnerability:** 5 known vulnerabilities, all marked as 'has no known vulnerabilities'.
- Published:** Apr 11, 2019.
- Vulnerabilities:** A table listing 5 vulnerabilities with columns: Identifier, Published, Overall Score, Status, Target date, and Actual date.

Identifier	Published	Overall Score	Status	Target date	Actual date
BDSA-2018-3897	Nov 6, 2018	1.7 Low	New	Never	Never
CVE-2017-15396	Aug 28, 2018	4.3 Medium	New	Never	Never
CVE-2017-15422	Aug 28, 2018	4.3 Medium	New	Never	Never
CVE-2017-14952	Oct 16, 2017	7.5 High	New	Never	Never
CVE-2017-17484	Dec 10, 2017	7.5 High	New	Never	Never

Displaying 1-5 of 5

# Chapter 11: Managing policies

The Policy Management feature enables you to create rules to govern your use of open source components. With policy rules, open source usage can be managed on an exception basis – as long as open source components meet the policy requirements their usage is allowed. Any open source components/versions that fail to meet your policy rules are flagged, enabling you to review and determine if the use of the component should be allowed in the particular application.

## About the policy process

To use the policy management feature:

1. [Create rules](#) that enforce your policies; a user with the Policy Manager [role](#) can create and manage policy rules. When creating policy rules determine:
  - Whether to enable the rule. BOMs will not be evaluated until the rule is enabled.
  - Whether the rule can be manually overridden.
  - The conditions for this rule.

**Note:** Rules can have multiple conditions; *all* conditions must be true for a component to be in violation of the rule.

2. View the violations and determine what to do with components that are in violation of a rule.

If you enabled the option, violations can be [manually overridden](#).

3. Optionally,
  - Create additional policies and/or [edit](#), [delete](#), or [disable or enable](#) your existing policies.
  - [View the Project Version report](#). This report includes policy violation information:
    - The component.csv and files.csv files list the policy status and override information.
    - The version.csv file indicates whether this version of the project has a policy violation.

To assist you, Black Duck provides three [default policy rules](#) that you can view, modify, enable, or delete. These policy rules are disabled by default.

## Viewing policy rules

The Policy Management page lists all your policy rules and indicates whether the rule allows manual overrides.

View this page by clicking the expanding menu icon () and selecting **Policy Management**:

The screenshot shows a "Policy Management" interface. At the top, there are buttons for "Create Policy Rule", "Edit", "Copy", and "Delete". A filter bar shows "Enabled" selected twice. Below this is a table with columns "Policy Rule", "Description", and "Severity". The table contains three rows: "test copy" (Severity: Low), "Sample Policy" (Severity: Medium), and "License Risk" (Severity: High). At the bottom right, it says "Displaying 1-3 of 3".

- The page is filtered to disable enabled rules. Modify or clear the filter to view disabled rules.
- All rules can be overridden unless noted.
- Click > to view a description, conditions, and severity of this rule.

From this page, you can view, [create](#), [edit](#), or [delete](#) policy rules.

## Viewing policy rule violations

When a component is in violation of a policy rule, the Policy Violation icon (🚫) appears in the UI.

The screenshot shows a "Black Duck Projects" dashboard for "Sample Project 1". It includes three risk charts: Security Risk (High: 5, Medium: 2, Low: 2, None: 61), License Risk (High: 9, Medium: 7, Low: 0, None: 54), and Operational Risk (High: 32, Medium: 24, Low: 3, None: 11). Below the charts is a table of components with columns: Component, Source, Match Type, Usage, License, Security Risk, and Operational Risk. Components listed include AOP Alliance, Apache Bean Validation, Apache Commons BeanUtils, Apache Commons CLI, and Apache Commons Codec. The Apache Commons Codec row has a red "High" risk indicator under both Security and Operational Risk.

The Policy Violation icon appears on the following pages:

- Source page. Icon appears next to the file name to indicate that a file in a component is in violation.
- BOM page. Icon appears next to components in violation.

In the hierarchical view of the BOM, 🚫 next to the parent component indicates that a child has a policy violation.

- **Project tab** of the main Dashboard. Icon appears next to the project name to indicate that this project has a version which has a policy violation.
- **Project Version page**. Icon appears next to the version to indicate that it has a policy violation.

Hover over the icon to view more information:

- On the project level, information such as the following appears:



This information also appears at the component/file level for users who are members of projects or have project-group privileges.

- On the component/file level, the following information appears for users with the BOM Manager, Super User, Project Manager, and Policy Violation Reviewer role:



Clicking the icon (when viewing the BOM using the List view) displays the Policy Violations dialog box from which you can override the policy violation.

## Overriding violations

If a rule was configured to allow manual overrides of violations, then you can [override a disapproved component](#) or file in that project.

When all component violations have been overridden, the Policy Violation Override icon ( ⓘ ) appears in the UI. In the hierarchical BOM, ⓘ indicates that a child's policy violation has been overridden; it appears at the parent level.

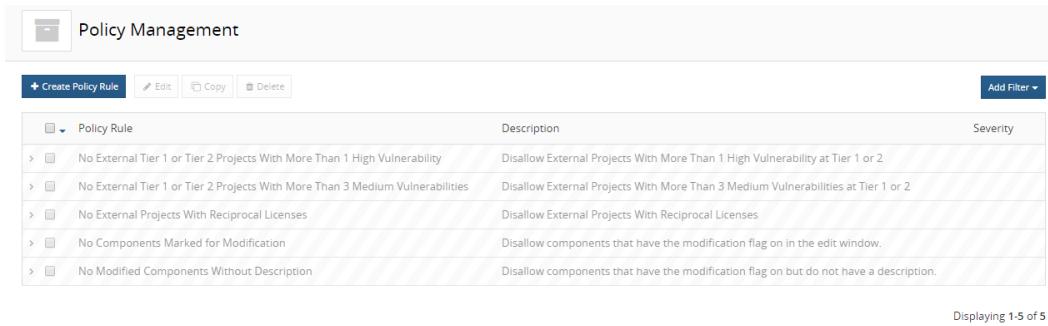
## Removing policy overrides

If a violation of a policy should not have been overridden, you can [remove](#) the override.

## Default policy rules

Black Duck provides three default policy rules which are disabled by default. Users with the Policy Manager [role](#) can [enable](#), [edit](#), or [delete](#) these rules.

View these policy rules on the Policy Management page by clicking the expanding menu icon ( ⚙ ), selecting **Policy Management**, and selecting to view disabled rules:



The screenshot shows a web-based application titled "Policy Management". At the top, there is a toolbar with buttons for "+ Create Policy Rule", "Edit", "Copy", "Delete", and "Add Filter". Below the toolbar is a table listing five default policy rules. The columns are "Policy Rule" (with a dropdown arrow), "Description", and "Severity". The rules are:

Policy Rule	Description	Severity
> No External Tier 1 or Tier 2 Projects With More Than 1 High Vulnerability	Disallow External Projects With More Than 1 High Vulnerability at Tier 1 or 2	
> No External Tier 1 or Tier 2 Projects With More Than 3 Medium Vulnerabilities	Disallow External Projects With More Than 3 Medium Vulnerabilities at Tier 1 or 2	
> No External Projects With Reciprocal Licenses	Disallow External Projects With Reciprocal Licenses	
> No Components Marked for Modification	Disallow components that have the modification flag on in the edit window.	
> No Modified Components Without Description	Disallow components that have the modification flag on but do not have a description.	

At the bottom right of the table area, it says "Displaying 1-5 of 5".

Click > to view a description and the conditions for these rules.

The default rules are:

- No External Projects With Reciprocal Licenses
- No External Tier 1 or Tier 2 Projects With More Than 1 High Vulnerability
- No External Tier 1 or Tier 2 Projects With More Than 3 Medium Vulnerabilities
- No Components Marked for Modification
- No Modified Components Without Description

## Creating a policy rule

Create rules to ensure that your projects do not have an open source component or component version that violates your policies.

You can create multiple rules and rules can have multiple conditions giving you the flexibility to create generic or highly specific global policy rules.

**Note:** Only users with the Policy Manager [role](#) can create policy rules.

## Policy conditions

Creating the condition(s) for a policy rule consists of selecting the projects that this rule applies (all or specific project attributes) and then selecting:

1. A component attribute
2. An operator (such as equals or greater than)
3. A value (depending on the option you selected)

Components that meet the condition will be disapproved.

You can create multiple conditions for a policy rule – *all* conditions must be true for a component to be in violation.

**Note:** All attributes appear for you to select, including attributes for those modules for which you are not licensed.

The table below shows the project filters you can select and the values you can specify.

## Project filters

Project Filters	Value
Project Distribution Type	Select one of the following values: <ul style="list-style-type: none"><li>• External</li><li>• Internal</li><li>• Open Source</li><li>• SaaS</li></ul>
Project Name	Begin typing to view possible values.
Project Phase	Select one of the following values: <ul style="list-style-type: none"><li>• Archived</li><li>• Deprecated</li><li>• In Development</li><li>• In Planning</li><li>• Pre-release</li><li>• Released</li></ul>
Project Tags	Enter the tag name.
Project Tier	Enter one of the following values: 1 - 5.
Project Custom Field Name	Available for Drop Down, Multiple Selections, and Single Selection field types.  Select a value.

This table shows the component attributes that you can select and the values you can specify.

## Component conditions

Component Condition	Value
Commits in the past year	Enter a number.
Component	Begin typing to view possible component values.  After selecting a component, the version field appears whereby you can enter a version number. <b>Any Version</b> is the default value if you do not enter a specific version.
Component Approval Status	Select one of the following values: <ul style="list-style-type: none"><li>• Approved</li><li>• Deprecated</li><li>• Limited Approval</li></ul>

Component Condition	Value
	<ul style="list-style-type: none"> <li>Rejected</li> <li>Unreviewed</li> </ul>
Component Modification	<p>Select either Yes or No.</p> <p>Indicates whether information was added to the <b>Modification</b> field when manually adding or editing a component.</p>
Component Modified	<p>Select either Yes or No.</p> <p>Indicates whether the <b>Modification</b> option was selected when manually adding or editing a component.</p>
Component Purpose	<p>Select either Yes or No.</p> <p>Indicates whether information was added in the <b>Purpose</b> field when manually adding or editing a component.</p>
Component Release Date	Select a date.
Component Usage	<p>Select one of the following values:</p> <ul style="list-style-type: none"> <li>Dev. tool / Excluded</li> <li>Dynamically linked</li> <li>Implementation of standard</li> <li>Merely aggregated</li> <li>Prerequisite</li> <li>Separate work</li> <li>Source code</li> <li>Statically linked</li> </ul>
Component Version Approval Status	<p>Select one of the following values:</p> <ul style="list-style-type: none"> <li>Approved</li> <li>Deprecated</li> <li>Limited Approval</li> <li>Rejected</li> <li>Unreviewed</li> </ul>
Contributors in the past year	Enter a number.
High Severity Vulnerability Count	Enter a number.
Highest Vulnerability Score	Enter a number between 0 and 10, including decimal numbers.
License	Begin typing to view possible license values.

Component Condition	Value
License Family	Select one of the following KB license families (Permissive, Reciprocal, Weak Reciprocal, AGPL, or Unknown) or a <a href="#">custom license family</a> .
License Status	Select one of the following values: <ul style="list-style-type: none"><li>• Unreviewed</li><li>• Approved</li><li>• Limited Approval</li><li>• Rejected</li></ul>
Low Severity Vulnerability Count	Enter a number.
Match Type	Select one of the following values: <ul style="list-style-type: none"><li>• Binary</li><li>• Files Added/Deleted</li><li>• File Dependency</li><li>• Direct Dependency</li><li>• Exact Directory</li><li>• Exact File</li><li>• Files Modified</li><li>• Manually Added</li><li>• Manually Identified</li><li>• Partial</li><li>• Snippet</li><li>• Transitive Dependency</li></ul>
Medium Severity Vulnerability Count	Enter a number.
Newer Versions Count	Enter a number.
Review Status	Select one of the following values: <ul style="list-style-type: none"><li>• Not Reviewed</li><li>• Reviewed</li></ul>

Component Condition	Value
Unfulfilled Fulfilled License Terms	Select <b>True</b> .  Any component that has unfulfilled license terms will trigger a policy violation.  <b>Note:</b> The <b>Legal</b> tab <a href="#">must be enabled</a> for a user to indicate that a term is fulfilled. If the <b>Legal</b> tab is disabled, a user will be unable to indicate that a term is fulfilled and policy violations cannot be cleared.
Unknown Component Version	Select <b>True</b> .  Any component that has a ? as the version will trigger a policy violation.

## Creating a policy

### ⚙️ To create a policy

1. Log in to Black Duck as a user with the Policy Manager or Sysadmin role.



2. Click the expanding menu icon ( ) and select **Policy Management**.
3. Select **Create Policy Rule** to display the Create Policy Rule dialog box.

The screenshot shows the 'Create Policy Rule' dialog box. It includes fields for Name, Description, Severity (set to Unspecified), Enabled (checked), Allow Manual Override (checked), and Projects (set to All). Below these are 'Policy Rules' settings with a dropdown menu 'Select condition...' and a 'Add Rule' button. At the bottom are 'Cancel' and 'Create' buttons.

4. Complete the following:

- **Name.** Required. Name of this policy.
- **Description.** Optional. This description appears when you select > on the Policy Management page.
- **Severity.** Optional. The severity level of this policy. You can use this option with build integrations to indicate what should happen when a policy violation occurs. For example, all policy violations with a severity of Blocker should fail the build.

Select one of the following values: **Blocker, Critical, Major, Minor, or Trivial**.

- **Enabled.** Clearing this option disables this rule. BOMs will not be evaluated until the rule is enabled.

Clear the option if you want to create draft policy rules.

You can [enable or disable](#) the rule after it is created.

- Select whether to allow manual overrides for this rule.

Users with the Policy Manager role can [override a disapproved component](#) in projects in which they are a member or have project-group privileges.

- Select whether this policy rule applies to all projects or filtered projects – projects with specific properties.

Selecting filtered projects displays the policy filters, as described above, that you can select for this policy rule. Select a project filter, an operator, and specify a value. Optionally, click + **Add Filter** to specify additional project filters.

5. Create the component condition for this policy rule: select an attribute from the **Policy Rules** list, select an operator, and specify a value.

Optionally, click + **Add Rule** to specify additional rules.

6. Click **Create**.

If the rule is enabled, existing BOMs are evaluated to determine if they are in violation of this rule. For any components that are in violation, the Policy Violation icon (  ) appears next to component name.

## Creating whitelist and blacklist policy rules

A whitelist is a list of items that are pre-approved for use while a blacklist lists items that are barred.

You can create policy rules that enforce the whitelists and blacklists you create for managing your BOM. Use a policy whitelist to pre-approve a component version in your BOM: any component version that does not match your whitelist triggers a policy violation. Use a policy blacklist to bar a component version from your BOM: a policy violation is automatically triggered for any component version that matches your blacklist.

### Whitelist examples

Suppose you want to create a whitelist whereby externally distributed projects with permissive licenses are pre-approved: any component versions that have non-permissive licenses will trigger a policy violation.

To create this policy rule, follow the instructions for [creating a policy rule](#), and set these conditions:

The screenshot shows a policy rule configuration interface. At the top, there is a condition for 'Project Distribution Type' set to 'External'. Below it, another condition for 'License Family' is set to 'not equal to' 'Permissive'. Both conditions have a delete icon to their right.

Suppose you want to create a whitelist whereby only a specific version of a component is approved: all other component versions trigger a policy violation.

To create this policy rule, follow the instructions for [creating a policy rule](#), and set these conditions:

The screenshot shows a policy rule configuration interface. The first condition is 'Component equals Apache Tomcat'. The second condition is 'Component not in' '8.0.1'. Both conditions have a delete icon to their right. A plus sign icon is located between the two conditions.

In this example, a policy violation is triggered when the Apache Tomcat version is not 8.0.1.

Suppose you want to create a whitelist whereby multiple versions of a component are approved: all other component versions trigger a policy violation.

To create this policy rule, follow the instructions for [creating a policy rule](#), and set these conditions:

The screenshot shows a policy rule configuration interface. The first condition is 'Component equals Apache Tomcat'. The second condition is 'Component not in' '8.0.1' and '8.0.3'. The '8.0.3' entry has a delete icon to its right. A plus sign icon is located between the two conditions. Below the conditions, there is a list of approved versions: 'Apache Tomcat 8.0.1' and 'Apache Tomcat 8.0.3'.

In this example, a policy violation is triggered when the Apache Tomcat version is not 8.0.1 or 8.0.3.

To create this condition:

1. Select the component, the equals operator, and the component.
2. For the second condition: select the component, the 'not in' operator, and the approved versions. To select multiple versions, select the version and click **Set selected component**. Repeat selecting approved versions and clicking **Set selected component** until all approved versions are selected.

## Blacklist example

Suppose you want to create a blacklist policy rule whereby any component versions in SaaS distributed

projects in the development or planning phase with licenses in the AGPL license family trigger a policy violation.

To create this policy rule, follow the instructions for [creating a policy rule](#), and set these conditions:

The screenshot shows the Black Duck Policy Management interface. At the top, there is a header with 'Projects' and two radio buttons: 'All' and 'Filtered'. Below this is a section for 'Policy Rules' which includes a 'License Family' filter set to 'equals AGPL'. There are also sections for 'Project Distribution Type' (set to 'SaaS') and 'Project Phase' (set to 'In Development, In Planning'). Buttons for '+ Add Filter' and '+ Add Rule' are visible.

## Editing a policy rule

Users with the Policy Manager [role](#) can edit policy rules.

After you edit a policy rule, BOMs are evaluated to determine if they are in violation of the edited rule.

### ⚙️ To edit a policy

1. Click the expanding menu icon () and select **Policy Management**.
2. Select the policy you want to edit and do one of the following to display the Edit Policy Rule dialog box:
  - Select **Edit**.
  - Click located in the **Conditions** section.
3. Edit the policy and click **Update**.

## Copying a policy rule

Users with the Policy Manager [role](#) can copy policy rules.

### ⚙️ To copy a policy

1. Click the expanding menu icon () and select **Policy Management**.
2. Select the policy you want to copy and click **Copy** to display the Copy Policy Rule dialog box:

3. Add the information for this policy and click **Create**.

The policy name is the only required field.

## Deleting a policy rule

Users with the Policy Manager [role](#) can delete policy rules.

Violations are removed for any component that was in violation of the deleted policy rule.

### To delete a policy

1. Log in to Black Duck as a user with the Policy Manager or Sysadmin role.



2. Click the expanding menu icon () and select **Policy Management**.
3. Select the policy you want to remove and click **Delete**.
4. When prompted, click **Delete** to confirm.

## Disabling or enabling a policy rule

Users with the Policy Manager [role](#) can disable or enable policy rules.

- When a rule is disabled, violations are removed for any component that was in violation of the policy rule (if the rule was previously enabled).
- When a rule is enabled, existing BOMs are immediately evaluated to determine if they are in violation of this rule.

### To disable or enable a policy



1. Click the expanding menu icon () and select **Policy Management**.
2. Select the policy you want to disable and do one of the following to display the Edit Policy Rule dialog box:
  - Click **Edit**.
  - Click
3. Do one of the following:
  - Clear the **Enabled** option to disable the rule.
  - Select the **Enabled** option to enable the rule.
4. Click **Update**.

## Overriding policy violations

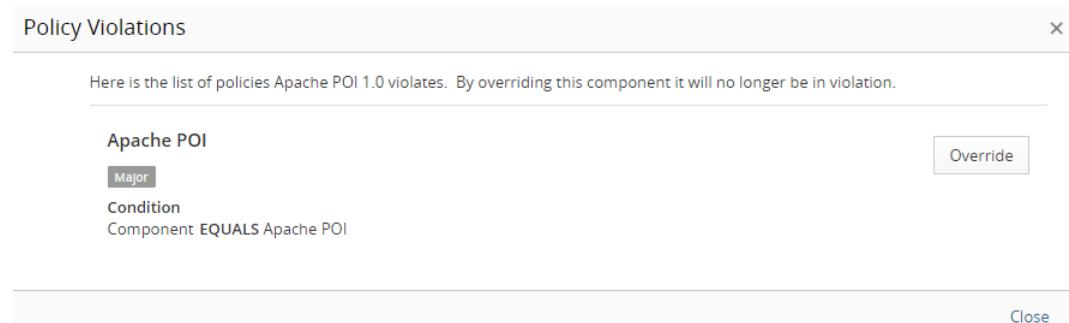
If a rule was configured to allow manual overrides of violations, then users with the appropriate [role](#) can

override a disapproved component or file in that project.

**Note:** If you override a file, the component will still be in violation if at least one file in the component is in violation of a policy.

### To override a violation

1. On the BOM page using the List view or **Source** tab, click the Policy Violation icon (Ø) of the component or file you wish to override. The Policy Violations dialog box appears.



2. Depending on whether there is one or more policy violation:

- For one policy violation, click **Override**. Optionally, enter a comment and click **Confirm**.

If you entered a comment, it appears, along with the username of the user who overrode the policy violation, in the Policy Violations dialog box.

- For multiple policy violations:

- Click **Override All** to override all policy violations. The Policy Violations dialog box displays the username of the user who overrode the policy rule.

You cannot enter a comment when using the **Override All** feature.

- Click **Override** for each policy violation you want to override. Optionally, enter a comment and click **Confirm**.

If you entered a comment, it appears, along with the username of the user who overrode the policy violation, in the Policy Violations dialog box.

3. Click **Close**.

The BOM or **Source** tab reappears and the Policy Violation Override icon (Ø) appears next to the component or file that you overrode if all policy violations were overridden. If a component has multiple policy violations and not all are overridden, then the Policy Violation icon (Ø) will still appear.

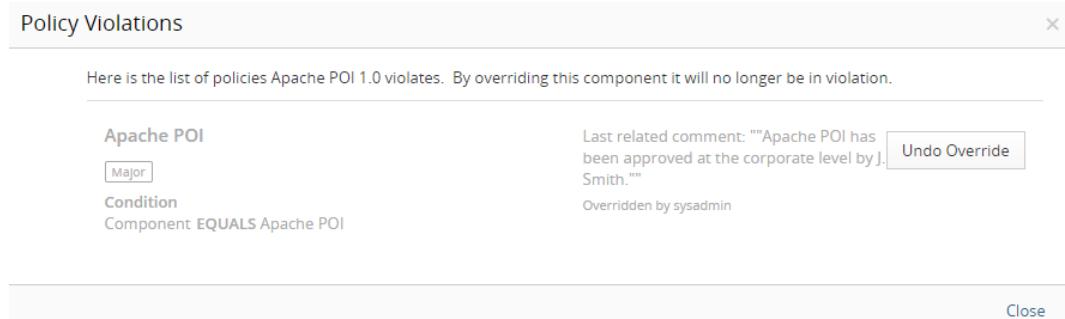
**Note:** Overrides can be [removed](#).

## Removing policy overrides

You can remove an override of a component or file that was in violation of a policy rule. Only users with the appropriate [role](#) can override a disapproved component or file in that project.

### To remove an override

1. On the BOM page using the List view or **Source** tab, click the Policy Violation Override icon ( ⓘ ) located next to the component or file. The Policy Violations dialog box appears.



2. Depending on whether there is one or more policy override to remove:

- To remove one policy override, click **Undo Override**. Optionally, enter a comment and click **Confirm**.

If you entered a comment, it appears, along with the username of the user who removed the override, in the Policy Violations dialog box.

- For multiple policy violations:

- Click **Undo All Overrides** to remove all policy overrides.

You cannot enter a comment when using the **Undo All Overrides** feature.

- Click **Undo Override** for each policy violation you want to override. Optionally, enter a comment and click **Confirm**.

If you entered a comment, it appears, along with the username of the user who removed the override, in the Policy Violations dialog box.

3. Click **Close**.

The BOM or **Source** tab appears and the Policy Violation icon ( ⓘ ) reappears.

# Chapter 12: Managing open source licenses

The use of open source software (OSS) is managed through licenses that allow you to use, modify, and/or share the software under defined terms and conditions. The conditions regarding the reuse of open source software can vary from things you can do (rights), things you cannot do (restrictions) and things you must do (obligations) in order to comply with the license.

Best practices for the redistribution of open source software include identifying all OSS content in the distribution and ensuring compliance to licensing obligations. Virtually all open source licenses contain an attribution clause as part of the licensing obligation. The attribution clause requires that the source of the software, and generally the copyright holder, be identified. Compliance with the attribution clause of these licenses generally takes the form of an attribution document, sometimes called a Notices File, which lists all OSS and the appropriate copyright and license information.

With Black Duck, you can create accurate and compliant open source notice file reports at a project/release level. Black Duck provides the actual license text for the MIT, variants of the BSD, and the ISC licenses, which are the top components in our KnowledgeBase, based upon customer usage.

For example, the following is an HTML version of the Notices File report from Black Duck:

## Sample Project ▶ 1.0 ▶ Notices File

Phase: In Planning | Distribution: External

### Components

Component	License
Apache Struts 2.2.0	Apache License 2.0

Apache License 2.0: " Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution."

### Licenses

#### Apache License 2.0

Apache Struts 2.2.0

Apache License  
Version 2.0, January 2004  
=====

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

##### 1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including

With the premium offering, you can edit and maintain the data needed to create this report. The notice files can then be included with the distribution or incorporated into documentation to satisfy the attribution obligation.

## Suggested work flow

To manage component licenses using Black Duck:

1. With the assistance of your legal counsel, determine the best combination of licenses for your company's work. This planning work can help you determine whether you need to make changes to a BOM to bring a project into compliance.
2. Use the [License Management page](#) to view licenses currently used by your company and existing [license families](#).
  - If a component uses a license that is not available from the Black Duck KnowledgeBase, users with the License Manager role can [create custom licenses](#) or [edit KnowledgeBase licenses](#).
  - If a license family does not accurately reflect your license risk, users with the License Manager role

can [create custom license families](#).

- If a license term does not accurately reflect a license obligation, users with the License Manager role can [manage license terms](#) of their custom or KnowledgeBase licenses.
3. [Create policy rules](#) (including [whitelist and/or blacklist policy rules](#)) that trigger violations when components do not comply with your license policies.
  4. Review the BOM for any license policy violations and determine what to do with components that are in violation of a rule.
  5. Review the BOM for license accuracy:
    - Research components that have Unknown License or License Not Found values.
    - Review components that have [license risk](#). Confirm that the [usage](#) of the component is correct as the combination of project distribution, usage, and license determines the license risk.
    - For components that have disjunction (OR) licenses, investigate and decide which license you plan to use.
  6. Create the [Notices File report](#). If you have the premium offering, you can make these modifications to the report contents:
    - Determine if any components or subprojects should be [excluded from the report](#).
    - [Add attribution statements](#).
    - [Edit the license text](#) if necessary.

## About license families

The use of open source software is managed through licenses that allow the software to be utilized, modified, and/or shared under defined terms and conditions. The conditions regarding the reuse of open source software can vary from things you can do (rights), things you cannot do (restrictions) and things you must do (obligations) in order to comply with the license. Black Duck tracks over 2000 open sources licenses that can range from those with few restrictions and obligations to those with many restrictions and obligations.

Depending upon the nature of these restrictions and obligations, some licenses are deemed to be riskier than others, as they require more management and care to ensure compliance with the license terms. Typically, the riskiest licenses are those that are reciprocal in nature. Reciprocal licenses, often pejoratively called "Viral Licenses", are those in which the license terms can extend beyond the open source code itself and can try to apply to other code as well. The other code could be modifications to the open source, or even simply code that uses the open source code in a way that triggers the reciprocal nature of the licenses. Once triggered, it is possible that in order to be in compliance to the license, developers who create software applications may need to treat the entire application as under the open source license and comply with all these obligations for the entire application. This could include the obligation to provide all the source code for the application (not just the open source) and allowing people who receive the application to modify and redistribute it without restrictions. This may be in conflict with a proprietary license model.

Please note, the legal aspects of managing open source licenses can be complicated and often it is best to seek legal counsel when making decisions about open source licenses and creating policies regarding their use. Legal counsel can best help determine if the license rights, restrictions, and obligations apply in a particular scenario. However, in order to help customers manage these risks in a simple and effective way, Black Duck categorizes open source licenses into license families for purposes of risk calculations and the

definition of open source policy rules. These families range from those that are highly reciprocal to those with few obligations and restrictions. These license families, called KnowledgeBase licenses are:

- Afferro General Public License (AGPL)

Licenses in the AGPL family tend to be highly reciprocal. The reciprocity can be easily triggered depending upon how the component is incorporated into the overall body of work and how much the original work is based upon the open source code. In addition, the obligations can apply when software is exposed over a network (for example, the internet). Companies who distribute software applications (either on a device or as media/downloads) or create software as a service (SaaS) applications need to pay particular attention to software under these licenses in order to ensure compliance.

- Reciprocal

Reciprocal licenses are those in which the license terms can easily apply to the overall body of work (like the AGPL) depending upon how it is used. However, typically the reciprocal nature of the license is triggered by distribution. Therefore, companies who distribute software in some fashion are generally concerned with highly managing software under these types of licenses.

- Weak Reciprocal

Licenses in this family can be reciprocal, but they are intended for open source software that is expected to be combined with other software under other licenses and therefore they tend to have a smaller reach. In this case, depending upon how the software is used, the reciprocal nature may simply cover modifications to the OSS and do not necessarily apply to the whole body of work. Companies who distribute software generally need to be keenly aware of these licenses, but tend to allow usage of components under these licenses with guidelines as to how they can be used. Staying in compliance and not triggering the reciprocity of the license tends to be easier.

- Permissive

Permissive Licenses tend to not place restrictions on the use of the open source code and generally have few obligations. Companies, for the most part, view these licenses as easy to manage and non-risky.

- Unknown

In this case, Black Duck was unable to determine the license for a component. Additional review should be done to determine the license for this component.

The following table shows the license family for the top 20 open source licenses used in open source projects:

License Family	Examples
Afferro General Public License (AGPL)	<ul style="list-style-type: none"><li>• GNU Affero General Public License v3 or later</li></ul>
Reciprocal	<ul style="list-style-type: none"><li>• GNU General Public License (GPL) 2.0 or 3.0</li><li>• Sun GPL with Classpath Exception v2.0</li></ul>
Weak Reciprocal	<ul style="list-style-type: none"><li>• Code Project Open License 1.02</li><li>• Common Development and Distribution License (CDDL) 1.0 or 1.1</li><li>• Eclipse Public License</li><li>• GNU Lesser General Public License (LGPL) 2.1 or 3.0</li></ul>

License Family	Examples
	<ul style="list-style-type: none"> <li>Microsoft Reciprocal License</li> <li>Mozilla</li> </ul>
Permissive	<ul style="list-style-type: none"> <li>Apache 2.0</li> <li>Artistic License</li> <li>BSD License 2.0 (2-clause Simplified, 3-clause, New, or Revised)</li> <li>Do What The F*ck You Want To Public License</li> <li>ISC License</li> <li>Microsoft Public License</li> <li>MIT License</li> </ul>
Unknown	N/A

## Managing license families

Users with the License Manager [role](#) can use the License Management page to manage their license families.

From this page you can view the KnowledgeBase license families or create [custom license families](#).

### ⚙️ To view the License Management page

1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon ( ) and select **License Management**.

The License Management page appears.

3. Select the **License Families** tab to view a table which lists all license families.

License Family	Source	Last Updated
Permissive	KnowledgeBase	Never
Reciprocal	KnowledgeBase	Never
Weak Reciprocal	KnowledgeBase	Never
AGPL	KnowledgeBase	Never
Unknown	KnowledgeBase	Never
Reciprocal - Modified for SaaS	Custom	Mar 18, 2019 by System Administrator

Displaying 1-6 of 6

The table provides the following information:

Column	Description																																													
<b>License Family</b>	<p>The license family for this license.</p> <p>Select a license family to view a definition and risk profile for that license family:</p> <div style="border: 1px solid #ccc; padding: 10px;"> <p><b>Reciprocal</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Description</td> <td colspan="4" style="padding: 5px;">Reciprocal licenses are those in which the license terms can easily and broadly apply to the overall body of work (including code covered under different licenses) upon distribution, depending upon how it is used within the overall body of work.</td> </tr> <tr> <td style="padding: 5px;">Risk Profile</td> <td colspan="4" style="padding: 5px;">License risk is determined by the license usage of the OSS components in the project version's bill of materials.</td> </tr> <tr> <td style="padding: 5px;"></td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">External</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">SaaS</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">Internal</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">Open Source</td> </tr> <tr> <td style="padding: 5px;">Source Code</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">High</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">Low</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> </tr> <tr> <td style="padding: 5px;">Dynamically Linked</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">High</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">Low</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> </tr> <tr> <td style="padding: 5px;">Implementation of Standard</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> </tr> <tr> <td style="padding: 5px;">Dev. Tool / Excluded</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> </tr> <tr> <td style="padding: 5px;">Separate Work</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> </tr> <tr> <td style="padding: 5px;">Statically Linked</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">High</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">Low</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> <td style="border: 1px solid #ccc; padding: 5px; text-align: center;">None</td> </tr> </table> </div> <p style="text-align: right;"><a href="#">Close</a></p>	Description	Reciprocal licenses are those in which the license terms can easily and broadly apply to the overall body of work (including code covered under different licenses) upon distribution, depending upon how it is used within the overall body of work.				Risk Profile	License risk is determined by the license usage of the OSS components in the project version's bill of materials.					External	SaaS	Internal	Open Source	Source Code	High	Low	None	None	Dynamically Linked	High	Low	None	None	Implementation of Standard	None	None	None	None	Dev. Tool / Excluded	None	None	None	None	Separate Work	None	None	None	None	Statically Linked	High	Low	None	None
Description	Reciprocal licenses are those in which the license terms can easily and broadly apply to the overall body of work (including code covered under different licenses) upon distribution, depending upon how it is used within the overall body of work.																																													
Risk Profile	License risk is determined by the license usage of the OSS components in the project version's bill of materials.																																													
	External	SaaS	Internal	Open Source																																										
Source Code	High	Low	None	None																																										
Dynamically Linked	High	Low	None	None																																										
Implementation of Standard	None	None	None	None																																										
Dev. Tool / Excluded	None	None	None	None																																										
Separate Work	None	None	None	None																																										
Statically Linked	High	Low	None	None																																										
<b>Source</b>	<p>Source for this license. Possible values are:</p> <ul style="list-style-type: none"> <li>KnowledgeBase. From the Black Duck KnowledgeBase.</li> <li>Custom. <a href="#">Custom license family</a>.</li> </ul>																																													
<b>Last Updated</b>	Date that the license family was created or last updated and the username of the user who created or last updated this license family.																																													

Use the filter to limit the information shown on this page. You can filter by:

- License Family Source: Custom or KnowledgeBase.

## About custom license families

If you discover that a KnowledgeBase license family does not accurately reflect your license risk, License Managers – users with the License Manager [role](#) – can create and manage custom license families. These custom license families can then be selected for a custom license which can then be assigned to custom components. This ensures that BOMs accurately show your license risk.

Custom license families:

- Consist of a name, a risk profile and optionally, a description.
- Can be assigned to a [custom license](#).
- Can be used to [create policy rules](#).
- Use a combination of component usage and distribution to determine [license risk](#).

License Managers can use the **License Families** tab in [License Management](#) to [create](#), [edit](#), and [delete](#) custom license families.

## Creating custom license families

Only users with the License Manager role can create [custom license families](#).

### ⚙️ To create a custom license family

1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon ( ) and select **License Management**.

The License Management page appears.

Select the **License Families** tab to display all license families.

The screenshot shows the Black Duck License Management interface. At the top, there's a header with a wrench icon, the title 'License Management', and three tabs: 'Licenses' (disabled), 'License Families' (selected and highlighted in blue), and 'License Terms'. Below the tabs is a search bar labeled 'Filter license families...' and a 'Add Filter' button. The main area is a table with columns 'License Family', 'Source', and 'Last Updated'. The table contains six rows: 'Permissive', 'Reciprocal', 'Weak Reciprocal', 'AGPL', 'Unknown', and 'Reciprocal - Modified for SaaS'. The 'Reciprocal - Modified for SaaS' row is highlighted with a yellow background. At the bottom right of the table, there's a small dropdown arrow. At the very bottom of the page, a footer note says 'Displaying 1-6 of 6'.

License Family	Source	Last Updated
Permissive	KnowledgeBase	Never
Reciprocal	KnowledgeBase	Never
Weak Reciprocal	KnowledgeBase	Never
AGPL	KnowledgeBase	Never
Unknown	KnowledgeBase	Never
Reciprocal - Modified for SaaS	Custom	Mar 18, 2019 by System Administrator

3. Click **Create License Family** to open the Create a Custom License Family dialog box.

Create a Custom License Family

Name *	<input type="text"/>			
Description	<input type="text"/>			
Risk Profile	Configure the risk based on usage and distribution. You can copy the profile from an existing family and change it.			
Copy from	<input type="text"/>			
Component Usage	External	SaaS	Internal	Open Source
Source Code	None	None	None	None
Statically Linked	None	None	None	None
Dynamically Linked	None	None	None	None
Separate Work	None	None	None	None
Implementation of Standard	None	None	None	None
Dev. Tool / Excluded	None	None	None	None

**Save**

4. Enter a name for this license family.
5. Optionally, enter a description.
6. Optionally, modify the license risk values. By default the [license risk](#) is None for all usages and distributions. You can select a license family to use as a baseline for the license risk by selecting one from the **Copy from** list. You can then use these license risk values for the custom license family or modify the values by using the drop downs to modify the license risk by usage and distribution. Possible license risk values are: none, low, medium, and high.
7. Click **Save**.

## Editing custom license families

Custom license families can be edited by users with the License Manager [role](#).

### To edit a custom license family

1. Log in to Black Duck with the License Manager role.
2. Click the expanding menu icon (



) and select License Management.

The License Management page appears.

- Select the **License family** tab to display all license families.

License Family	Source	Last Updated
Permissive	KnowledgeBase	Never
Reciprocal	KnowledgeBase	Never
Weak Reciprocal	KnowledgeBase	Never
AGPL	KnowledgeBase	Never
Unknown	KnowledgeBase	Never
<a href="#">Reciprocal - Modified for SaaS</a>	Custom	Mar 18, 2019 by System Administrator

Displaying 1-6 of 6

- Select the custom license family name or click and select **Edit** in the row of the custom license family that you want to edit to display the Edit Custom License Family dialog box.

Component Usage	External	SaaS	Internal	Open Source
Source Code	High	Medium	None	None
Statically Linked	High	Medium	None	None
Dynamically Linked	High	Medium	None	None
Separate Work	None	None	None	None
Implementation of Standard	None	None	None	None
Dev. Tool / Excluded	None	None	None	None

**Save**

- Modify the information shown for this custom license family.
- Click **Save** in the Edit Custom License dialog box. The username of the user who edited this license

family and the date appears in the **Last Updated** column.

## Deleting custom license families

You cannot delete a license family that is being used by a license in a BOM.

You also cannot delete licenses provided by the Black Duck KnowledgeBase.

1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon ( ) and select **License Management**.

The License Management page appears.

3. Select the **License Family** tab to display all license families.

A screenshot of the Black Duck License Management interface. The top navigation bar includes 'License Management' (highlighted with a gear icon), 'Licenses', 'License Families' (highlighted in blue), and 'License Terms'. Below the navigation is a search bar with 'Filter license families...' and an 'Add Filter' button. A large table lists license families with columns for 'License Family', 'Source', and 'Last Updated'. The table shows six rows: 'Permissive' (KnowledgeBase, Never), 'Reciprocal' (KnowledgeBase, Never), 'Weak Reciprocal' (KnowledgeBase, Never), 'AGPL' (KnowledgeBase, Never), 'Unknown' (KnowledgeBase, Never), and 'Reciprocal - Modified for SaaS' (Custom, Mar 18, 2019 by System Administrator). At the bottom right of the table is a dropdown menu. The status bar at the bottom right says 'Displaying 1-6 of 6'.

License Family	Source	Last Updated
Permissive	KnowledgeBase	Never
Reciprocal	KnowledgeBase	Never
Weak Reciprocal	KnowledgeBase	Never
AGPL	KnowledgeBase	Never
Unknown	KnowledgeBase	Never
Reciprocal - Modified for SaaS	Custom	Mar 18, 2019 by System Administrator

4. Click and select **Delete** in the row of the custom license family that you want to delete to display a confirmation dialog box.  
An error message appears if you try to delete a custom license family that is currently being used by a license.
5. Click **Delete** to confirm.

## Viewing licenses

The **Licenses** tab in the License Management page displays custom licenses you have created and the licenses from the Black Duck KnowledgeBase that are used in all projects in your organization.

Users with the License Manager [role](#) can use the License Management page to manage licenses.

**Note:** The License Manager role is intended to be a cross-project, enterprise role. Typically, attorneys or privileged users that have broad access to information would have this role. Therefore, License Managers can view the licenses for *all* projects, including projects in which they are not project members.

From this page, you can:

- [Create](#), [edit](#), or [delete](#) custom licenses.
- [Edit KnowledgeBase](#) licenses.
- [View the full text](#) of custom and Black Duck KnowledgeBase licenses.
- [View the number of components](#) in your projects that use a specific license.

**Note:** Edits made locally by a BOM manager, Super User, or Project Manager to the license text of a custom or KnowledgeBase license will not appear on this page.

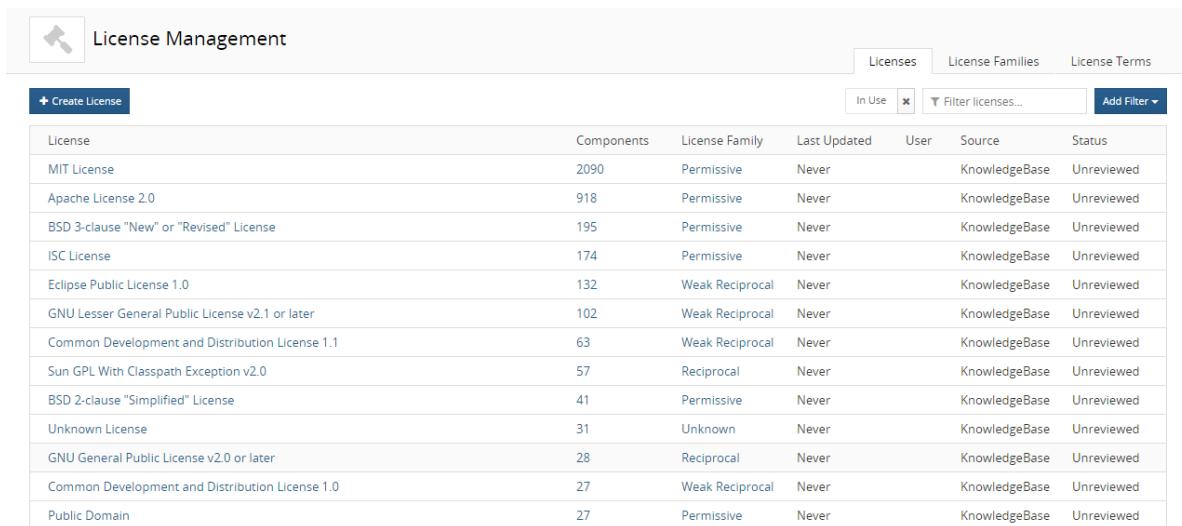
#### To view the License Management page

1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon () and select **License Management**.

The License Management page appears.



License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

Select the **Licenses** tab to view a table with the following information.

Column	Description																																							
<b>License</b>	<p>License name.</p> <p>Select the name to display the <i>License Name</i> page. Use the:</p> <ul style="list-style-type: none"> <li>• <b>Settings</b> tab to view information for this license, such as the <a href="#">license family</a> and <a href="#">license text</a>.</li> <li>• <b>License Terms</b> tab to view the terms for this license.</li> <li>• <b>Where Used</b> tab to view the component and subproject versions where this license is used.</li> </ul>																																							
<b>Components</b>	<p>Number of components or subprojects in all projects that have this license.</p> <p><b>Note:</b> The value shown here does <i>not</i> include projects assigned with this custom license.</p> <p>Select the component value to display a page which lists the component versions or subprojects <a href="#">where this license is used</a>.</p>																																							
<b>License Family</b>	<p>The <a href="#">license family</a> for this license.</p> <p>Select a license family to view a definition and risk profile for that license family:</p> <div style="border: 1px solid #ccc; padding: 10px;"> <p>Reciprocal</p> <table> <tbody> <tr> <td>Description</td> <td>Reciprocal licenses are those in which the license terms can easily and broadly apply to the overall body of work (including code covered under different licenses) upon distribution, depending upon how it is used within the overall body of work.</td> </tr> <tr> <td>Risk Profile</td> <td>License risk is determined by the license usage of the OSS components in the project version's bill of materials.</td> </tr> </tbody> </table> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Component Usage</th> <th>External</th> <th>SaaS</th> <th>Internal</th> <th>Open Source</th> </tr> </thead> <tbody> <tr> <td>Source Code</td> <td>High</td> <td>Low</td> <td>None</td> <td>None</td> </tr> <tr> <td>Dynamically Linked</td> <td>High</td> <td>Low</td> <td>None</td> <td>None</td> </tr> <tr> <td>Implementation of Standard</td> <td>None</td> <td>None</td> <td>None</td> <td>None</td> </tr> <tr> <td>Dev. Tool / Excluded</td> <td>None</td> <td>None</td> <td>None</td> <td>None</td> </tr> <tr> <td>Separate Work</td> <td>None</td> <td>None</td> <td>None</td> <td>None</td> </tr> <tr> <td>Statically Linked</td> <td>High</td> <td>Low</td> <td>None</td> <td>None</td> </tr> </tbody> </table> <p style="text-align: right;"><a href="#">Close</a></p> </div>	Description	Reciprocal licenses are those in which the license terms can easily and broadly apply to the overall body of work (including code covered under different licenses) upon distribution, depending upon how it is used within the overall body of work.	Risk Profile	License risk is determined by the license usage of the OSS components in the project version's bill of materials.	Component Usage	External	SaaS	Internal	Open Source	Source Code	High	Low	None	None	Dynamically Linked	High	Low	None	None	Implementation of Standard	None	None	None	None	Dev. Tool / Excluded	None	None	None	None	Separate Work	None	None	None	None	Statically Linked	High	Low	None	None
Description	Reciprocal licenses are those in which the license terms can easily and broadly apply to the overall body of work (including code covered under different licenses) upon distribution, depending upon how it is used within the overall body of work.																																							
Risk Profile	License risk is determined by the license usage of the OSS components in the project version's bill of materials.																																							
Component Usage	External	SaaS	Internal	Open Source																																				
Source Code	High	Low	None	None																																				
Dynamically Linked	High	Low	None	None																																				
Implementation of Standard	None	None	None	None																																				
Dev. Tool / Excluded	None	None	None	None																																				
Separate Work	None	None	None	None																																				
Statically Linked	High	Low	None	None																																				
<b>Last Updated</b>	Time, if updated today, or date that the license was last updated.																																							
<b>User</b>	<p>Username of the user who created or last updated the license.</p> <p>This field is empty for licenses from the Black Duck KnowledgeBase that have not been edited.</p>																																							

Column	Description
<b>Source</b>	Source for this license. Possible values are: <ul style="list-style-type: none"> <li>KnowledgeBase. From the Black Duck KnowledgeBase.</li> <li>Modified KnowledgeBase. An <a href="#">edited the Black Duck KnowledgeBase license</a>.</li> <li>Custom. Custom license.</li> </ul>
<b>Status</b>	The review status for the license. Possible values are: <ul style="list-style-type: none"> <li>Approved.</li> <li>Unreviewed. If Unreviewed, click the dropdown selector at the right, and select Edit.</li> <li>Rejected.</li> <li>Limited Approval.</li> </ul>

Use the filters to limit the information shown on this page. You can filter by:

- License Source: KnowledgeBase, Modified KnowledgeBase, or Custom.
- License Family: a KnowledgeBase license family (AGPL, Reciprocal, Weak Reciprocal, Permissive and/or Unknown) or a custom license family.
- License Status: Unreviewed, Conditionally Approved, Rejected, or Approved.
- In Use. Only displays those licenses associated with a component version or subproject. This filter is selected by default.

## Viewing license text

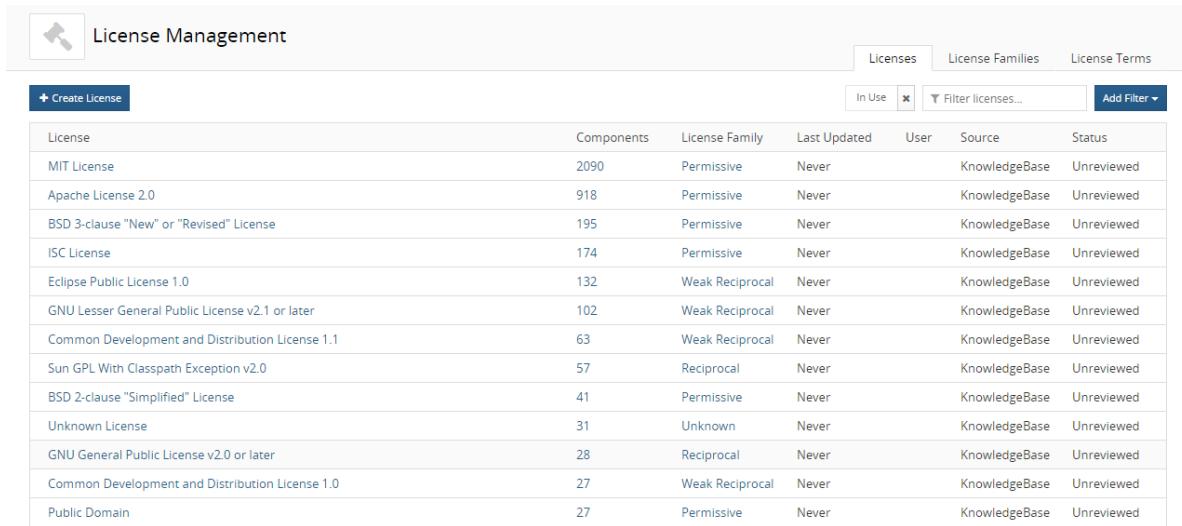
You can view the text of custom and KnowledgeBase licenses.

**Note:** For KnowledgeBase licenses, if the license is one that is modified for individual components (like the BSD or MIT license), then the template license text is shown here. However, when viewing the license text in the context of a component (such as viewing the component's license in a BOM), the actual license text for that component is shown.

### To view license text

1. Log in to Black Duck with the License Manager [role](#).
2. Click the expanding menu icon (License Management.

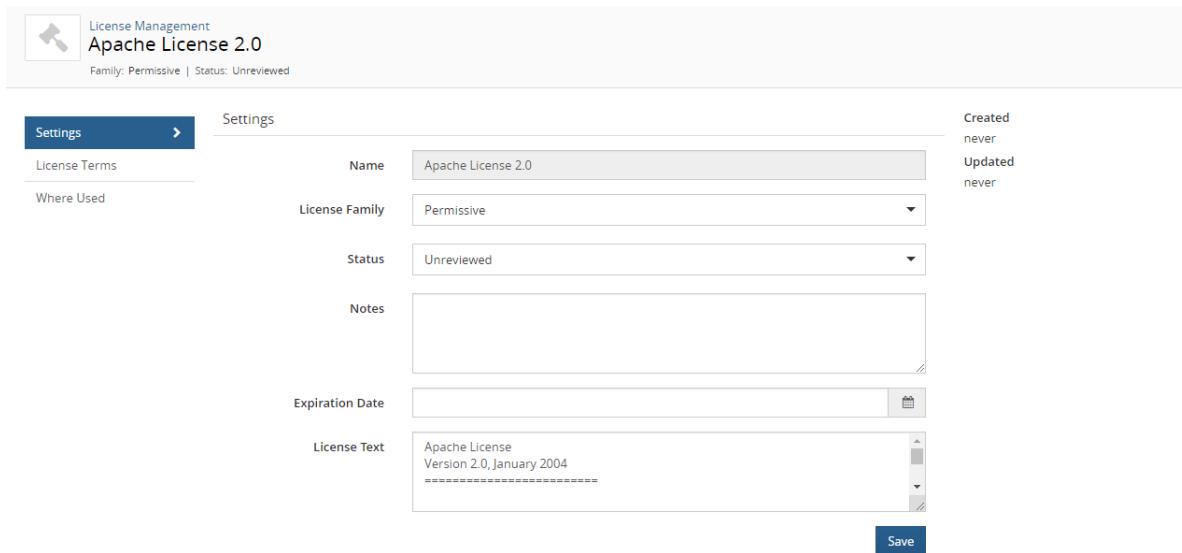
The License Management page appears.



The screenshot shows the 'License Management' interface. At the top, there's a navigation bar with tabs for 'Licenses', 'License Families', and 'License Terms'. Below the navigation bar is a search/filter bar with 'In Use' checked, a 'Filter licenses...' button, and an 'Add Filter' dropdown. A large table lists various licenses with columns for License Name, Components, License Family, Last Updated, User, Source, and Status. The table includes rows for MIT License, Apache License 2.0, BSD 3-clause "New" or "Revised" License, ISC License, Eclipse Public License 1.0, GNU Lesser General Public License v2.1 or later, Common Development and Distribution License 1.1, Sun GPL With Classpath Exception v2.0, BSD 2-clause "Simplified" License, Unknown License, GNU General Public License v2.0 or later, Common Development and Distribution License 1.0, and Public Domain.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

3. In the **Licenses** tab, select the license name to display the **License Name Settings** tab which displays the license text:



The screenshot shows the 'License Name Settings' interface for the Apache License 2.0. At the top, it displays the license name and its family and status. Below this, there are tabs for 'Settings' and 'Notes'. The 'Settings' tab contains fields for 'Name' (Apache License 2.0), 'License Family' (Permissive), 'Status' (Unreviewed), and 'Notes' (empty). The 'Notes' section has a rich text editor. The 'Expiration Date' field is empty. The 'License Text' field contains the text: 'Apache License Version 2.0, January 2004' followed by a separator line. At the bottom right is a 'Save' button.

With the appropriate [role](#), you can also [view the license text in a BOM](#).

## Viewing license use

You can view the component and subproject versions where a specific license is used.

**Note:** The information shown here lists the components and subprojects that use a license. It does not include licenses assigned to project versions.

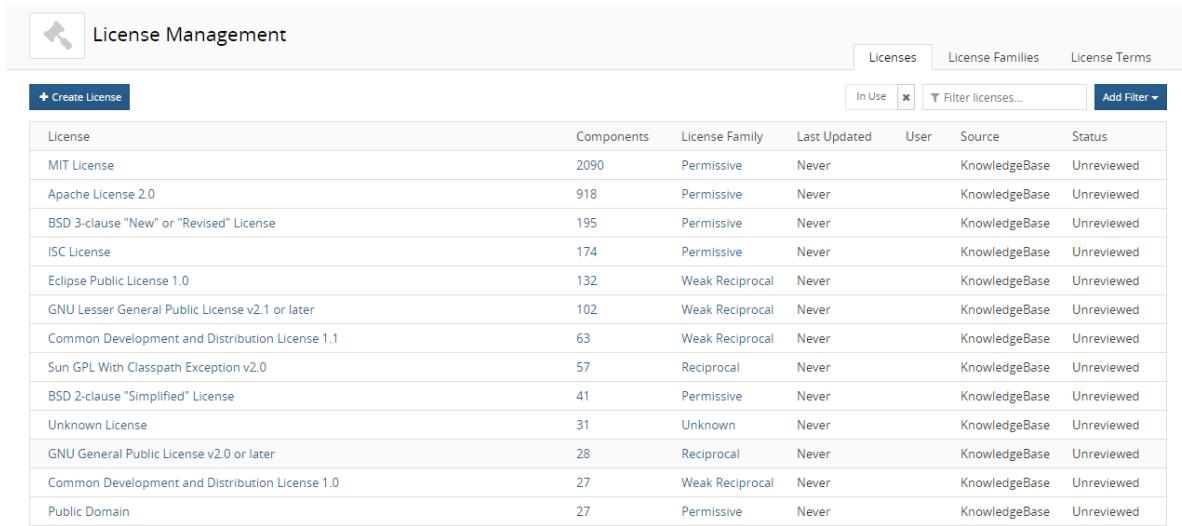
 To view where a license is used

1. Log in to Black Duck with the License Manager role.



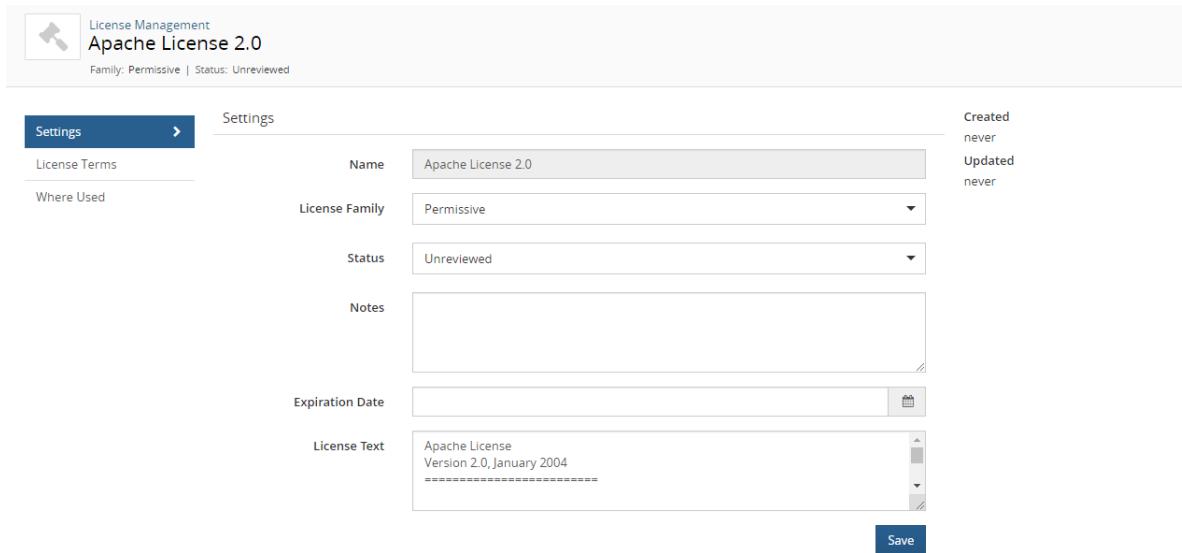
2. Click the expanding menu icon ( ) and select License Management.

The License Management page appears.



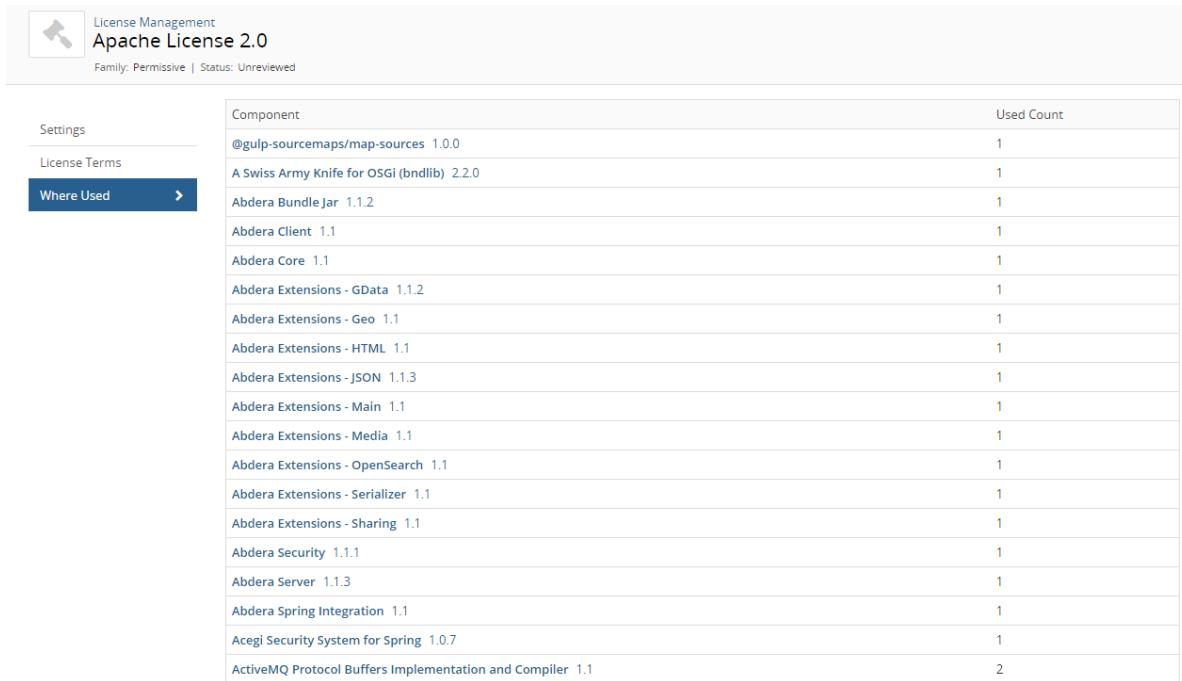
License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

3. Select the license name to display the **License Name Settings** tab.



Settings		Settings	
License Terms		Name	Apache License 2.0
Where Used		License Family	Permissive
		Status	Unreviewed
		Notes	<input type="text"/>
		Expiration Date	<input type="text"/>
		License Text	<input type="text"/> Apache License Version 2.0, January 2004 =====
<b>Save</b>			

4. Select the **Where Used** tab.



Component	Used Count
@gulp-sourcemaps/map-sources 1.0.0	1
A Swiss Army Knife for OSGi (bndlib) 2.2.0	1
Abdera Bundle Jar 1.1.2	1
Abdera Client 1.1	1
Abdera Core 1.1	1
Abdera Extensions - GData 1.1.2	1
Abdera Extensions - Geo 1.1	1
Abdera Extensions - HTML 1.1	1
Abdera Extensions - JSON 1.1.3	1
Abdera Extensions - Main 1.1	1
Abdera Extensions - Media 1.1	1
Abdera Extensions - OpenSearch 1.1	1
Abdera Extensions - Serializer 1.1	1
Abdera Extensions - Sharing 1.1	1
Abdera Security 1.1.1	1
Abdera Server 1.1.3	1
Abdera Spring Integration 1.1	1
Acegis Security System for Spring 1.0.7	1
ActiveMQ Protocol Buffers Implementation and Compiler 1.1	2

- Select the component name to display the [Black Duck KB component page](#) which displays information about the component, such as a description, component links, and tags, and information about each of the component versions that are available in the Black Duck KB.
- Select the component version to display the **Details** tab of the [Component Name Version page](#), which displays a list of the projects and project versions in which this version of the component is used.
- Select the subproject name to display the **Overview** tab of the *Project Name* page which project more information about this project.
- Select the subproject version to display the **Details** tab of the *Project Version* page to view [more information about this project version](#)

## Determining license risk

License risk is determined by the license risk of the components in the project version's BOM.

Components can have four levels of overall license risk (high, medium, low, and none), based on the [license family](#) declared by the component, the type of distribution for the project (external, internal, SaaS, or open source) and the [usage](#) (statically linked, dynamically linked, source code, dev. tool/excluded, implementation of standard, merely aggregated, prerequisite, and separate work).

**Note:** Other licenses include "Unknown" which indicates that the OSS component version's license is not known; "License Not Found" which indicates that although researched by Synopsys, no declared license was found for the component; and "No License" which indicates that Synopsys found a declaration of 'No License' for the component.

These licenses are included in the Unknown license family in the tables below.

For components with multiple licenses:

- "AND" licenses: license risk is determined by the license with the highest risk.
- "OR" licenses: license risk is determined by the license with the lowest risk.

Risk calculations assume that your project is being distributed under a proprietary license.

## Default license risk

The following tables show the license risk for the default (KnowledgeBase) license families. Users with the License Manager [role](#) can [create custom license families](#) and define the license risk by usage and distribution for those custom license families.

### License risk - by usage

#### Statically linked

The following table lists the license risk when the component's usage is **Statically Linked**.

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Affero General Public License (AGPL)	High Risk	High Risk	None	None
Reciprocal	High Risk	Low Risk	None	None
Weak Reciprocal	High Risk	Low Risk	None	None
Permissive	None	None	None	None
Unknown	High Risk	High Risk	High Risk	High Risk

#### Dynamically linked

The following table lists the license risk when the component's usage is **Dynamically Linked**.

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Affero General Public License (AGPL)	High Risk	High Risk	None	None
Reciprocal	High Risk	Low Risk	None	None
Weak Reciprocal	Medium Risk	Low Risk	None	None
Permissive	None	None	None	None
Unknown	High Risk	High Risk	High Risk	High Risk

## Source code

The following table lists the license risk when the component's usage is **Source Code**.

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Affero General Public License (AGPL)	High Risk	High Risk	None	None
Reciprocal	High Risk	Low Risk	None	None
Weak Reciprocal	High Risk	Low Risk	None	None
Permissive	None	None	None	None
Unknown	High Risk	High Risk	High Risk	High Risk

## Dev. tool / excluded

The following table lists the license risk when the component is not distributed with your product. (Usage value is **Dev. Tool / Excluded**).

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Affero General Public License (AGPL)	None	None	None	None
Reciprocal	None	None	None	None
Weak Reciprocal	None	None	None	None
Permissive	None	None	None	None
Unknown	None	None	None	None

## Implementation of Standard

The following table lists the license risk when the component usage is **Implementation of Standard**.

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Affero General Public License (AGPL)	None	None	None	None
Reciprocal	None	None	None	None
Weak Reciprocal	None	None	None	None

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Permissive	None	None	None	None
Unknown	None	None	None	None

### Separate Work

The following table lists the license risk when the component usage is **Separate Work**.

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Affero General Public License (AGPL)	None	None	None	None
Reciprocal	None	None	None	None
Weak Reciprocal	None	None	None	None
Permissive	None	None	None	None
Unknown	None	None	None	None

### Merely aggregated

The following table lists the license risk when the component's usage is **Merely aggregated**.

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Affero General Public License (AGPL)	None	None	None	None
Reciprocal	None	None	None	None
Weak Reciprocal	None	None	None	None
Permissive	None	None	None	None
Unknown	Medium	Medium	Low	Low

### Prerequisite

The following table lists the license risk when the component's usage is **Prerequisite**.

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Affero General Public License (AGPL)	Medium	None	None	None
Reciprocal	Medium	None	None	None
Weak Reciprocal	Low	None	None	None
Permissive	None	None	None	None
Unknown	Medium	Medium	Low	Low

## License risk by license family

### Affero General Public License (AGPL)

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Source Code	High Risk	High Risk	None	None
Statically Linked	High Risk	High Risk	None	None
Dynamically Linked	High Risk	High Risk	None	None
Separate Work	None	None	None	None
Merely Aggregated	None	None	None	None
Implementation of Standard	None	None	None	None
Prerequisite	Medium	None	None	None
Dev. Tool/Excluded	None	None	None	None

### Reciprocal

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Source Code	High Risk	Low Risk	None	None
Statically Linked	High Risk	Low Risk	None	None
Dynamically Linked	High Risk	Low Risk	None	None
Separate Work	None	None	None	None
Merely Aggregated	None	None	None	None

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Implementation of Standard	None	None	None	None
Prerequisite	Medium	None	None	None
Dev. Tool/Excluded	None	None	None	None

### Weak Reciprocal

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Source Code	High Risk	Low Risk	None	None
Statically Linked	High Risk	Low Risk	None	None
Dynamically Linked	Medium Risk	Low Risk	None	None
Separate Work	None	None	None	None
Merely Aggregated	None	None	None	None
Implementation of Standard	None	None	None	None
Prerequisite	Low	None	None	None
Dev. Tool/Excluded	None	None	None	None

### Permissive

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Source Code	None	None	None	None
Statically Linked	None	None	None	None
Dynamically Linked	None	None	None	None
Separate Work	None	None	None	None
Merely Aggregated	None	None	None	None
Implementation of Standard	None	None	None	None
Prerequisite	None	None	None	None
Dev. Tool/Excluded	None	None	None	None

## Unknown

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Source Code	High Risk	High Risk	High Risk	High Risk
Statically Linked	High Risk	High Risk	High Risk	High Risk
Dynamically Linked	High Risk	High Risk	High Risk	High Risk
Separate Work	None	None	None	None
Merely Aggregated	Medium	Medium	Low	Low
Implementation of Standard	None	None	None	None
Prerequisite	Medium	Medium	Low	Low
Dev. Tool/Excluded	None	None	None	None

## About custom licenses

If you discover that a license that you use for a component in your BOM is not available from the Black Duck KnowledgeBase, License Managers – users with the License Manager [role](#) – can create and manage custom licenses. These custom licenses can then be selected for a component version in a BOM to ensure that the BOMs are accurate.

**Note:** If the Black Duck KnowledgeBase is missing an open source license, instead of creating a custom license, you can contact [Black Duck Support](#) to request that this license be added to the KnowledgeBase.

Custom licenses:

- Consist of a name, a [license family](#), and license text.
- Can be used to [create policy rules](#).
- Use the same [rules to determine license risk](#) as licenses from the Black Duck KnowledgeBase.
- Can be [modified locally in a BOM](#) by users with the appropriate [role](#). That user cannot edit the name or license family but can edit the license text. The edited license text only applies to the version of the license associated with the BOM.

License Managers can use the [License Management page](#) to manage custom licenses and the Black Duck KnowledgeBase licenses used in all the projects in your organization.

## Creating custom licenses

Only users with the License Manager role can create [custom licenses](#).

1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon ( ) and select **License Management**.

The License Management page appears.

 A screenshot of the Black Duck License Management interface. The page title is "License Management". At the top right are tabs for "Licenses", "License Families", and "License Terms", with "Licenses" selected. Below the tabs are filters: "In Use" (unchecked), "Filter licenses...", and "Add Filter". A table lists various licenses with columns for Name, Components, License Family, Last Updated, User, Source, and Status. The table includes rows for MIT License, Apache License 2.0, BSD 3-clause "New" or "Revised" License, ISC License, Eclipse Public License 1.0, GNU Lesser General Public License v2.1 or later, Common Development and Distribution License 1.1, Sun GPL With Classpath Exception v2.0, BSD 2-clause "Simplified" License, Unknown License, GNU General Public License v2.0 or later, Common Development and Distribution License 1.0, and Public Domain.
 

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

3. Click **Create License** to open the Create a Custom License dialog box.

 A screenshot of the "Create a Custom License" dialog box. It has fields for Name (mandatory), License Family (set to "Nothing Selected"), License Text (empty text area), Status (empty dropdown), Notes (empty text area), and Expiration Date (empty date input). At the bottom are "Cancel" and "Create" buttons.
 

Name *	<input type="text"/>
License Family *	Nothing Selected
License Text *	<input type="text"/>
Status	Nothing Selected
Notes	<input type="text"/>
Expiration Date	<input type="text"/>
<input type="button" value="Cancel"/> <input type="button" value="Create"/>	

4. Enter the name for this custom license.

5. Select the [license family](#) for this custom license. This license family, along with the component usage, determines the [license risk](#).

You can select a KnowledgeBase or custom license family.

6. Enter the license text.

7. Optionally, select a status, enter any notes, and select an expiration date for this license.

8. Click **Create**.

## Editing a custom license

Custom licenses can be edited by users with the License Manager role and by users with the BOM Manager, Super User, or Project Manager role:

- License Managers can make *global* edits to custom licenses. The License Manager can edit any of the custom license settings.

These edits are propagated to BOMs with components using the custom license as described below.

- BOM Managers, Super Users, and Project Managers can only make *local* edits to the license text of a custom license used in a BOM.

These edits only apply to the version of the custom license used in the BOM.

When the License Manager edits a custom license:

- Edits to the license family and license name are always propagated to the custom licenses used in BOMs.
- Edits to the license text *may or may not* be propagated to the custom licenses used in BOMs:
  - If the BOM Manager/Super User/Project Manager *edited the license text*, the edits made by the License Manager *are not* propagated to the version of the custom license used in the BOM.
  - If the BOM Manager/Super User/Project Manager *did not edit the license text*, the edits made by the License Manager *are* propagated to the custom license used in the BOM.

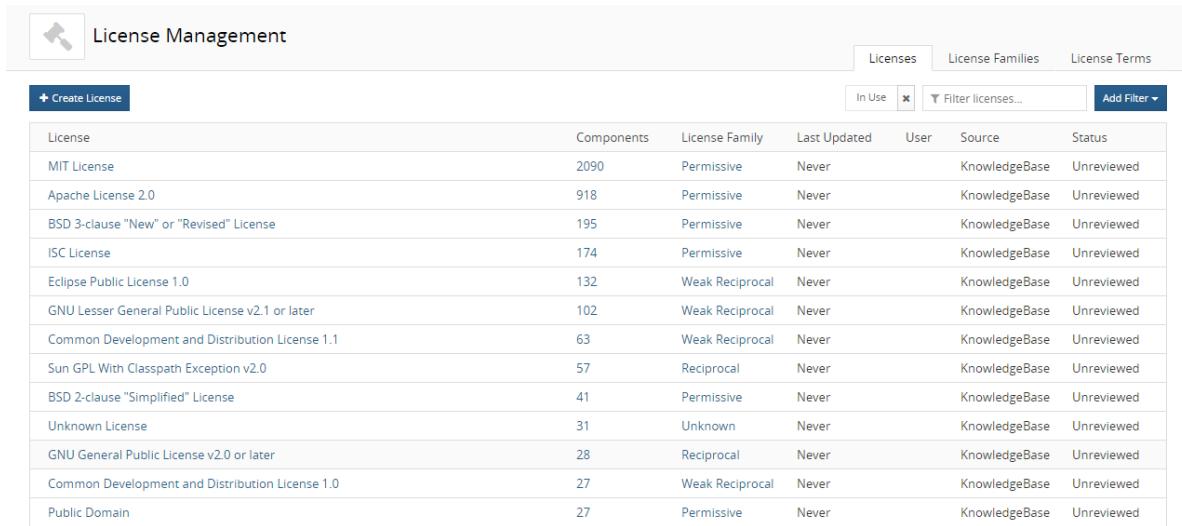
### To edit a custom license

1. Log in to Black Duck with the License Manager role.



2. Click the expanding menu icon () and select License Management.

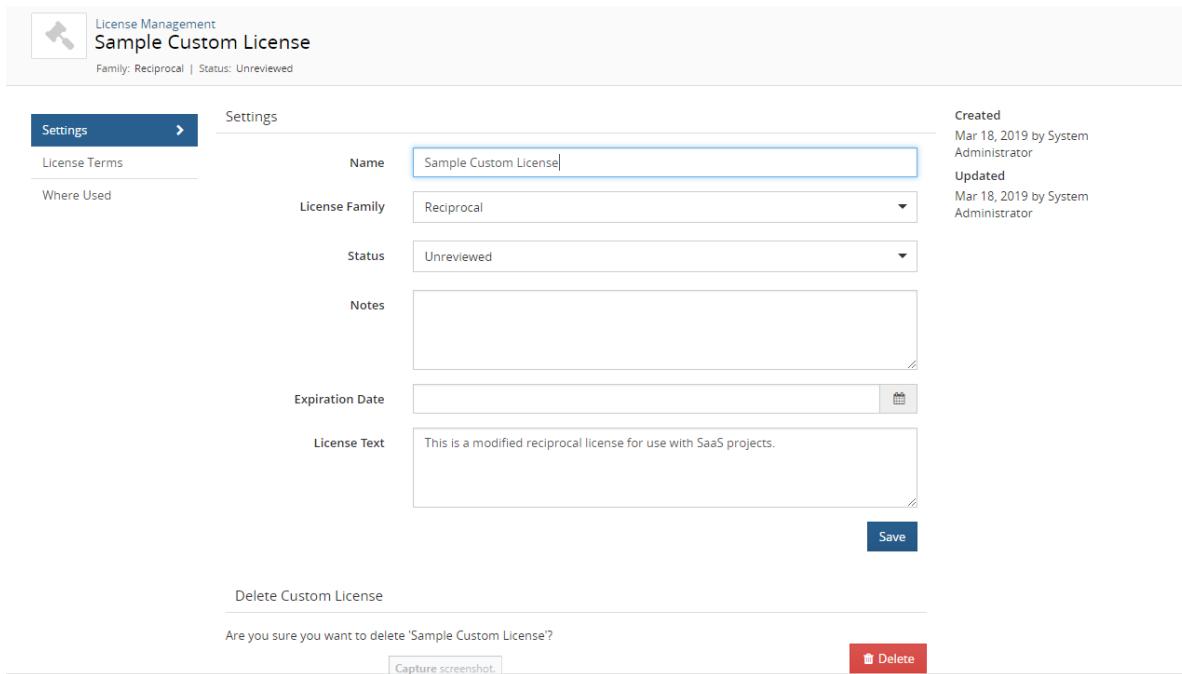
The License Management page appears.



The screenshot shows the 'License Management' interface. At the top, there are tabs for 'Licenses', 'License Families', and 'License Terms'. Below the tabs, there is a search bar with filters for 'In Use' and 'Filter licenses...', and a 'Add Filter' button. A large table lists various licenses with columns for License Name, Components, License Family, Last Updated, User, Source, and Status. Some rows have a blue background.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

3. Select the license name to display the **License Name Settings** tab.



The screenshot shows the 'Settings' tab for a 'Sample Custom License'. The left sidebar has 'Settings' and 'License Terms' sections. The main area has fields for 'Name' (Sample Custom License), 'License Family' (Reciprocal), 'Status' (Unreviewed), 'Notes' (empty), 'Expiration Date' (calendar icon), and 'License Text' (text area containing 'This is a modified reciprocal license for use with SaaS projects.'). On the right, there are 'Created' and 'Updated' logs. At the bottom, there is a 'Save' button, a 'Delete Custom License' section with a confirmation message, and a 'Delete' button.

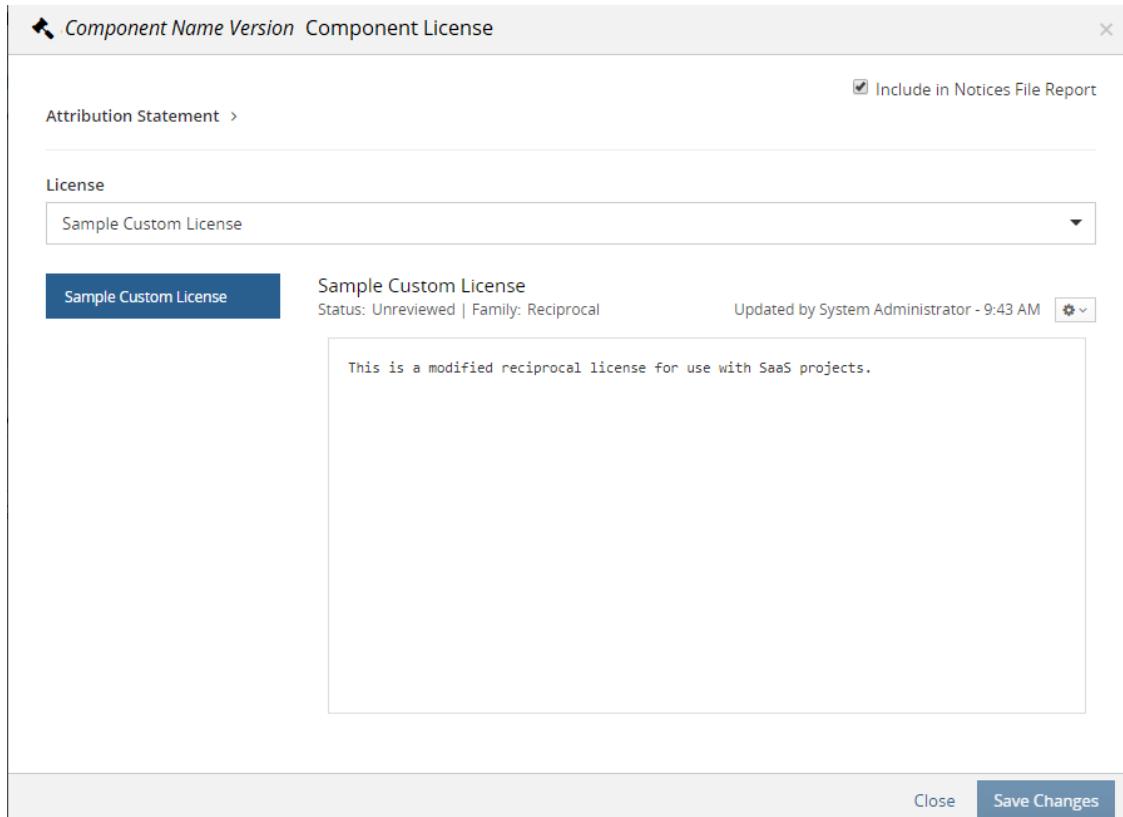
4. Modify the information shown for this custom license.

- Name:** License name. You can modify a custom license name.
- License Family:** Use the drop-down selector to choose the license family.
- Status:** Use the drop-down selector to choose the license status.
- Notes:** You can type any text in this field. Use this for additional information or helpful notes.
- Expiration Date:** Use the calendar tool to set the expiration date.

- **License Text:** The actual license as found in the component.

5. Click **Save**.

- The username of the user who edited this license appears in the **User** column and the time the license was modified appears in the **Last Updated** column in the License Management page.
- Edit information also appears in the *Component/Subproject Name Version Component License* dialog box.



## Deleting custom licenses

You cannot delete a license that is being used in a BOM.

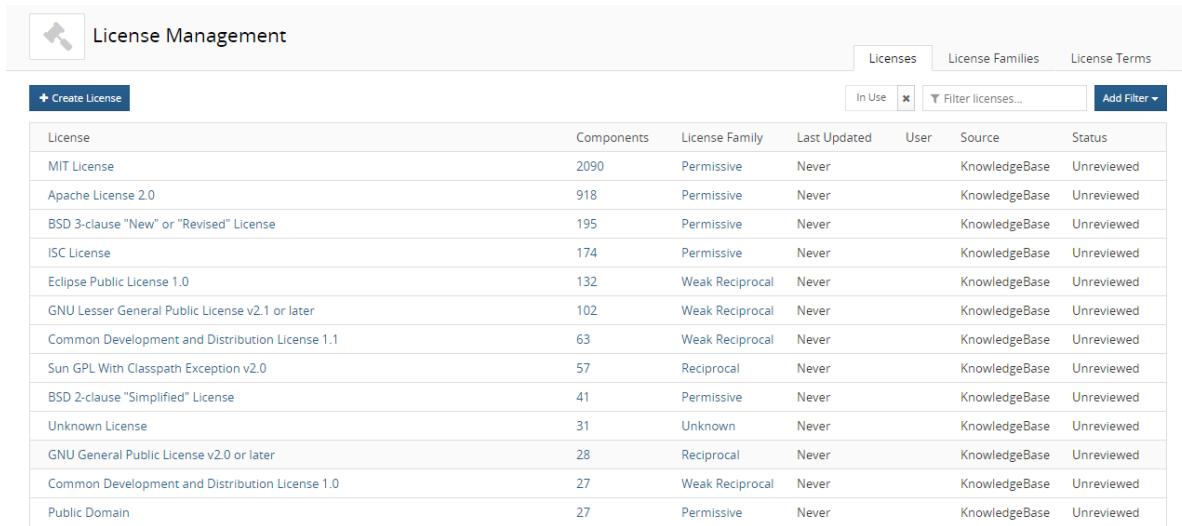
You also cannot delete licenses provided by the Black Duck KnowledgeBase.

1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon ( ) and select **License Management**.

The License Management page appears.



The screenshot shows the Black Duck License Management interface. At the top, there's a navigation bar with tabs for 'Licenses' (selected), 'License Families', and 'License Terms'. Below the navigation is a search bar with filters for 'In Use' and 'Filter licenses...', and a 'Add Filter' button. A large table lists various licenses with columns for License Name, Components, License Family, Last Updated, User, Source, and Status. Some rows are highlighted in blue, indicating they are in use.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

- Click  and select **Delete** in the row of the custom license that you want to delete to display a confirmation dialog box.

An error message appears if you try to delete a custom license that is currently being used in a BOM.

- Click **Delete** to confirm.

## About license terms

License terms are the provisions in the license which grant rights or impose restrictions on the use of the software under that license. They summarize the conditions regarding the reuse of software that is contained in the text of the license. They indicate the things you can do (permitted), the things you cannot do (forbidden) and the things you must do (required) to comply with the license. Please note that the license terms provide by the Black Duck application are just general summaries of the license and cannot be taken as legal advice.

You can create custom license terms and manage existing KnowledgeBase license terms to ensure that you meet the legal obligations associated with a license. Manage license terms to help your developers know the legal obligations associated with a license and to help you bring a project into compliance with licensing obligations.

Users with the License Manager [role](#) can:

- [Create](#), [edit](#), or [delete](#) custom license terms.
- [Associate](#) a custom or KnowledgeBase license term to one or more custom or KnowledgeBase licenses.
- [Remove](#) custom license terms from custom or KnowledgeBase licenses.
- [Remove](#) KnowledgeBase license terms from custom licenses or KnowledgeBase licenses that were not originally defined by the Black Duck KnowledgeBase.
- [Deprecate](#) custom license terms.
- [Disable](#) or [restore](#) KnowledgeBase license terms for a KnowledgeBase license.
- Determine if the license term [requires fulfillment](#).

## Suggested work flow

To manage custom and KnowledgeBase license terms:

1. With the assistance of your legal counsel, review the license terms associated with Black Duck KnowledgeBase licenses. However, please note that not all licenses will have pre-defined license terms and not every condition of use may be represented by Black Duck-provided license terms. The license terms provided by the Black Duck application are just general summaries of the license and cannot be taken as legal advice or replace a legal review.
2. Determine if there are any Black Duck KnowledgeBase terms that need to be modified to more accurately reflect your legal obligations.
  - You can disable KnowledgeBase terms associated with Black Duck KnowledgeBase licenses so that these terms are not shown to your end users.
  - Optionally, you can create new custom terms and then associate them to KnowledgeBase licenses either in addition or replacing an existing KnowledgeBase term.
3. If you created custom licenses, determine if you need to create new custom license terms or associate existing KnowledgeBase terms to the custom license.
4. Continue the review process, as you may wish to eventually deprecate a custom license term or remove a KnowledgeBase term.

## License terms process

If, after reviewing the existing terms, you determine that you need to create new license terms, do the following:

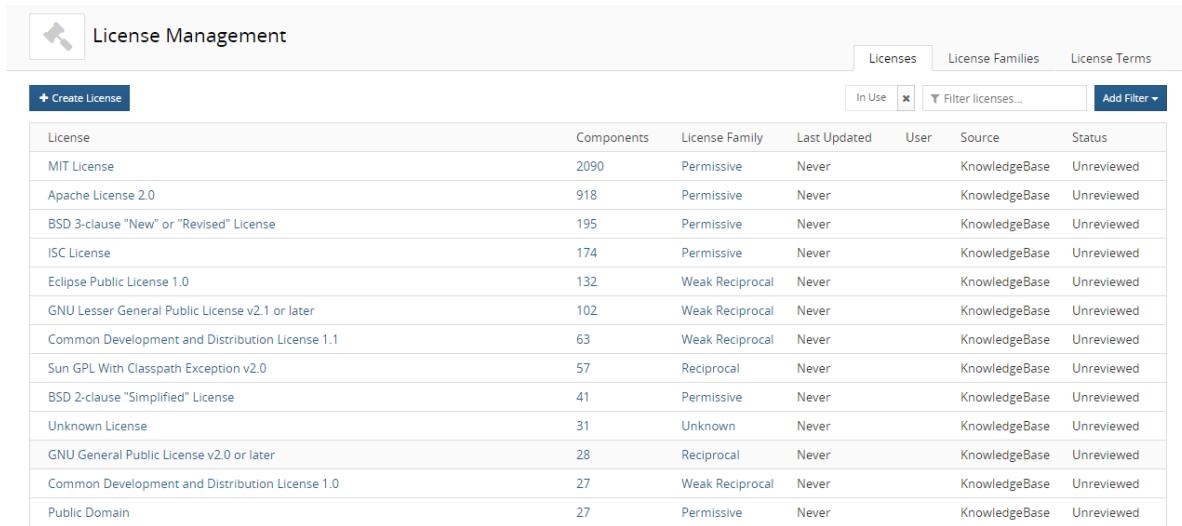
1. [Create categories](#) to manage your license terms. Categories are used to manage your license terms.  
You can also create a category while creating a license term.
2. [Create a custom license term](#).
3. [Associate](#) the new term to one or more licenses.

The **License Terms** tab shows all license terms for custom and KnowledgeBase licenses.

### To view the License Terms tab

1. Log in to Black Duck with the License Manager [role](#).
2. Click the expanding menu icon () and select **License Management**.

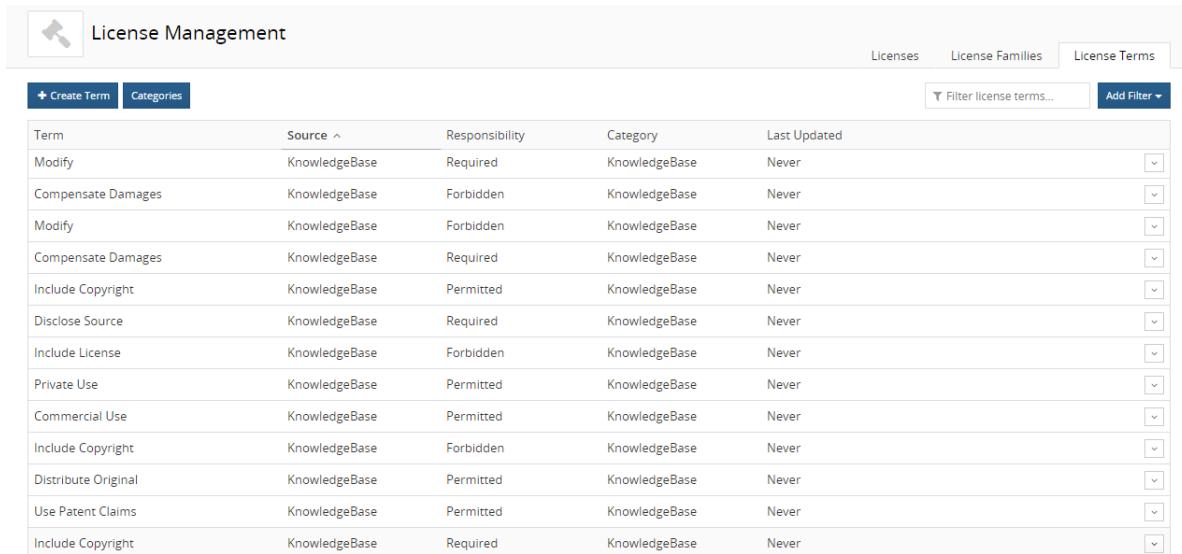
The License Management page appears.



The screenshot shows the 'License Management' interface. At the top, there are three tabs: 'Licenses' (selected), 'License Families', and 'License Terms'. Below the tabs is a search bar with filters: 'In Use' (checkbox), 'Filter licenses...', and 'Add Filter'. A large table lists various licenses with columns for License Name, Components, License Family, Last Updated, User, Source, and Status. Notable entries include MIT License, Apache License 2.0, and Sun GPL With Classpath Exception v2.0.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

### 3. Select the **License Terms** tab.



The screenshot shows the 'License Management' interface with the 'License Terms' tab selected. At the top, there are two tabs: '+ Create Term' (selected) and 'Categories'. Below the tabs is a search bar with filters: 'Filter license terms...' and 'Add Filter'. A large table lists various license terms with columns for Term, Source, Responsibility, Category, and Last Updated. Terms include Modify, Compensate Damages, and Include Copyright.

Term	Source ^	Responsibility	Category	Last Updated
Modify	KnowledgeBase	Required	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never

The table provides the following information:

Column	Description
<b>Term</b>	<p>Term name.</p> <p>Hover over the name to view the description for this term.</p> <p>For custom license terms, select the name to open the Edit a License Term dialog box.</p>
<b>Source</b>	<p>The source for this term. Possible values are:</p> <ul style="list-style-type: none"> <li>KnowledgeBase. This is a standard term from the Black Duck KnowledgeBase.</li> <li>Custom. A license term you created.</li> </ul>
<b>Responsibility</b>	<p>Responsibility for this license term. Possible values are:</p> <ul style="list-style-type: none"> <li>Permitted</li> <li>Forbidden</li> <li>Required</li> </ul>
<b>Category</b>	<p>Category for this license term.</p> <p>License terms from the Black Duck KnowledgeBase have <b>KnowledgeBase</b> as the category. Custom license terms list the category defined when adding the term.</p>
<b>Last Updated</b>	<p>Date that the license term was last updated and the username of the user who updated this term.</p> <p>The column lists <b>Never</b> for KnowledgeBase license terms.</p>

## Viewing license terms

License terms are categorized into things you are permitted to do (rights), things you are forbidden to do (restrictions), and things you are required to do (obligations) to comply with the license.

You can view license terms using the License Management page and when viewing license information in the BOM.

**Note:** License obligation will not appear in the UI, if the information is unavailable from [OpenHub](#),

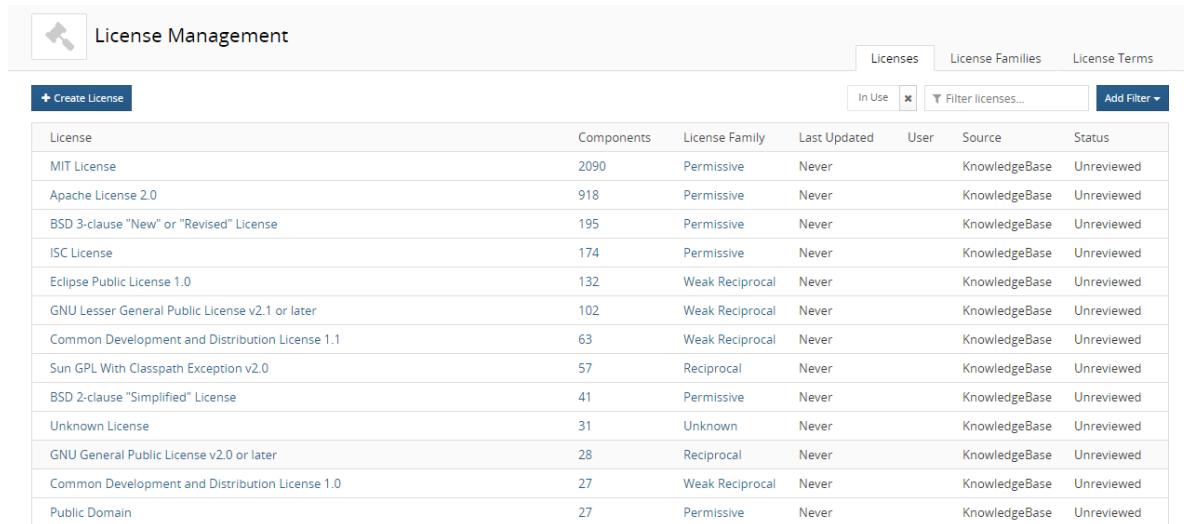
 To view the license terms from the License Management page

1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon ( ) and select **License Management**.

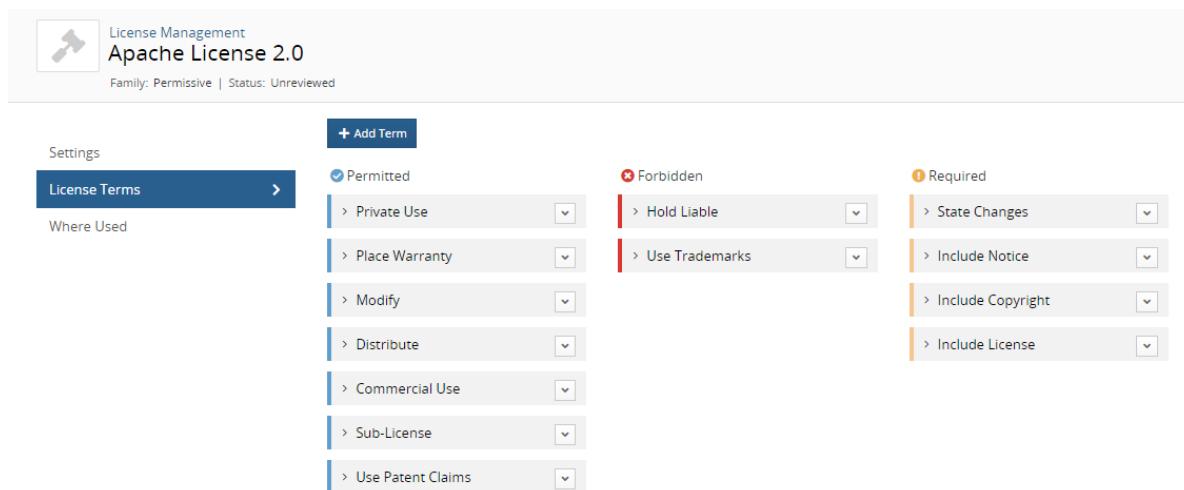
The License Management page appears.



The screenshot shows the 'License Management' interface. At the top, there are tabs for 'Licenses', 'License Families', and 'License Terms'. Below the tabs is a search bar with filters for 'In Use' and 'Filter licenses...', and a 'Add Filter' button. A large table lists various licenses with columns for License Name, Components, License Family, Last Updated, User, Source, and Status. The table includes entries like MIT License, Apache License 2.0, BSD 3-clause "New" or "Revised" License, ISC License, Eclipse Public License 1.0, and so on.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

3. Select a license from the **License** tab to display the *License Name* page.
4. Select the **License Terms** tab to view the obligations for this license.



The screenshot shows the 'Apache License 2.0' page under 'License Terms'. It includes a 'Settings' section and a 'Where Used' section. The main area displays three categories of license terms: 'Permitted' (blue), 'Forbidden' (red), and 'Required' (orange). Each category has a list of items with dropdown menus for further details.

Setting	Value
Family	Permissive
Status	Unreviewed

Category	Term
Permitted	Private Use
	Place Warranty
	Modify
	Distribute
	Commercial Use
	Sub-License
	Use Patent Claims
	> Hold Liable
Forbidden	Use Trademarks
Required	State Changes
	Include Notice
	Include Copyright
	Include License
	> > >

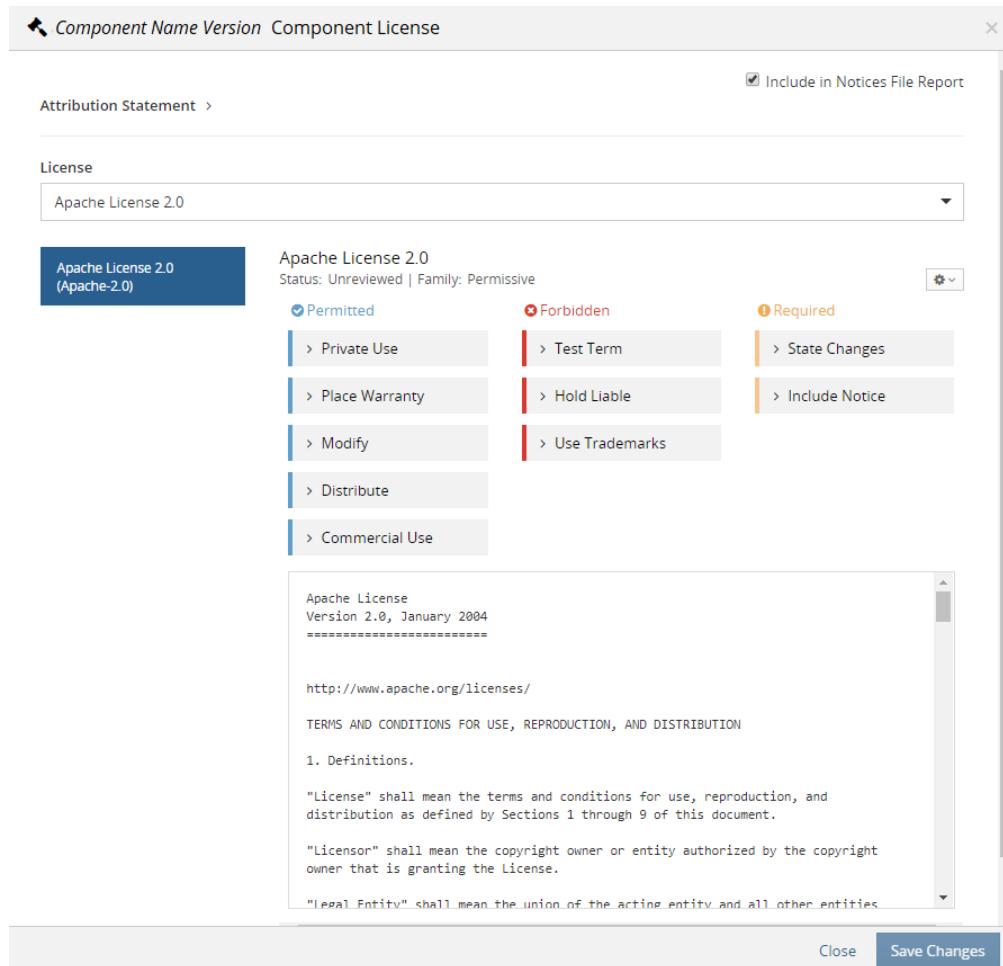
If available, select > to view additional information.

#### ⚙️ To view the license term information in a BOM

Only users with the appropriate [role](#) can view this information in the BOM.

1. Locate the project using the **Projects** tab on the Dashboard.
2. Select the name of the project to go to the *Project Name* page.
3. Select the version name to open the **Components** tab and view the BOM.

4. Select the license name to open the *Component Name Version Component License* dialog box.



5. If there is more than one license for this component, select the license you wish to view the license obligation information.

## About license term fulfillment

License Managers can define which license terms require fulfillment.

The fulfillment status of a license term is defined for a term at the license level, as not all instances of a license term may require fulfillment. This allows you to easily define the fulfillment requirements for a license term,

The work flow for license term fulfillment is:

1. License Managers determine the license terms that require fulfillment. Fulfillment can be defined when:
  - [Associating a license term](#).
  - [Viewing all terms for a specific license](#).

- [Creating a new term or adding an existing term for a specific license](#) when using the **License Name License Terms** tab.
2. The System Administrator [enables the Project Version's Legal tab](#).
  3. BOM Manager's use the *Project Version's Legal* tab to view all license terms that require fulfillment and [indicate which license terms are fulfilled](#).

Note the following:

- It may take time for license term fulfillment requirements to appear on the **Legal** tab.  
A job, LicenseTermFulfillmentJob, must complete for fulfillment requirements to appear on the **Legal** tab for all affected project version BOMs.

- Policy managers can [create a policy rule](#) that will trigger a violation when there are unfulfilled license terms.

Note that the **Legal** tab must be enabled so that a user can indicate that a term is fulfilled. If the **Legal** tab is disabled, which is the default setting, a user will be unable to indicate that a term is fulfilled and policy violations cannot be cleared.

- License term fulfillment status can be [cloned](#).
- A new project version report, `license_term_fulfillment.csv` lists the license terms and fulfillment status for a project version.

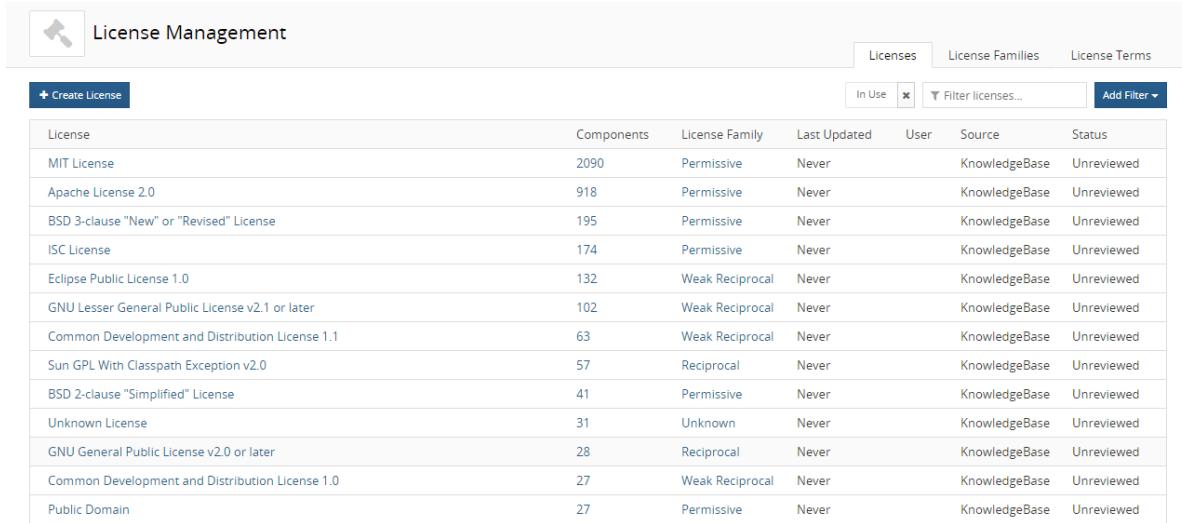
## Defining fulfillment when viewing terms for a license

License Managers can indicate a license term is required when using the **License Name License Terms** tab which shows all terms for a specific license.

### To define the fulfillment requirement when viewing a license

1. Log in to Black Duck with the License Manager [role](#).
2. Click the expanding menu icon () and select **License Management**.

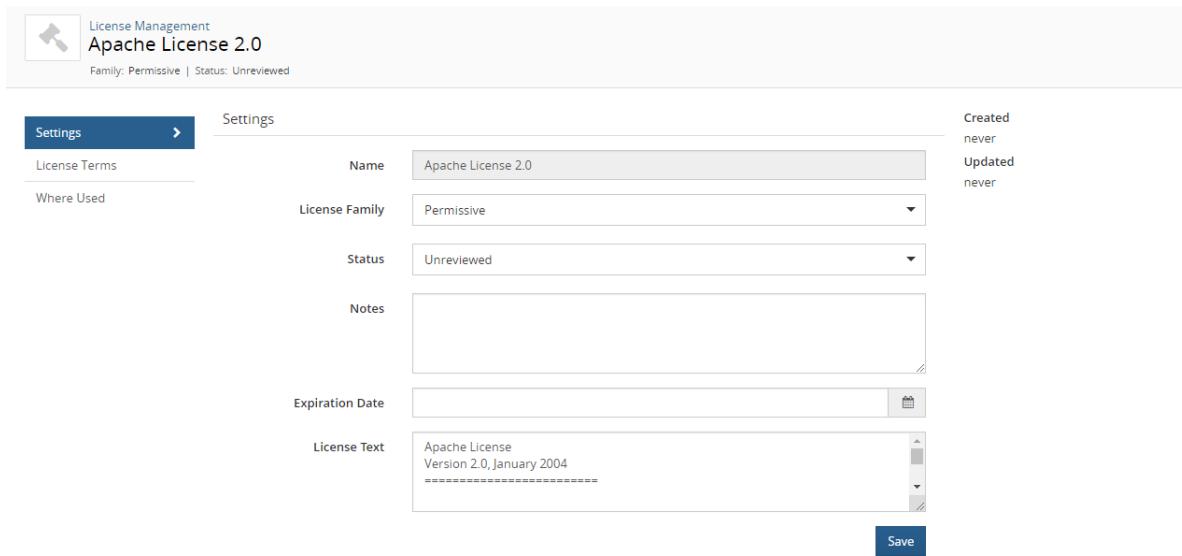
The License Management page appears.



The screenshot shows a table titled "License Management" with a "Create License" button. The columns are: License, Components, License Family, Last Updated, User, Source, and Status. The data includes various open-source licenses like MIT, Apache, BSD, and GPL, along with their respective counts and details.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

3. In the **Licenses** tab, select the license name to display the *License Name Settings* tab.



The screenshot shows the "Settings" tab for the Apache License 2.0. It includes fields for Name (Apache License 2.0), License Family (Permissive), Status (Unreviewed), Notes (empty), Expiration Date (empty), and License Text (Apache License Version 2.0, January 2004). A sidebar on the right shows creation and update details: Created never, Updated never.

Settings	
License Terms	Name: Apache License 2.0
Where Used	License Family: Permissive
	Status: Unreviewed
Notes	(Empty text area)
Expiration Date	(Empty date input field)
License Text	Apache License Version 2.0, January 2004 =====

Save

4. Select the **License Terms** tab to view the terms associated with this tab.

The screenshot shows the Apache License 2.0 settings page. At the top, it says "Family: Permissive | Status: Unreviewed". Below this, there are three main categories: "Permitted" (blue), "Forbidden" (red), and "Required" (orange). Under "Permitted", there are eight items: Private Use, Place Warranty, Modify, Distribute, Commercial Use, Sub-License, and Use Patent Claims. Under "Forbidden", there are two items: Hold Liable and Use Trademarks. Under "Required", there are four items: State Changes, Include Notice, Include Copyright, and Include License.

- Click next to the KnowledgeBase license term you wish to indicate fulfillment is required and select **Fulfillment Required**.

The Fulfillment Required icon ( ) appears to indicate this license term is required.

The screenshot shows the Apache License 2.0 settings page again. The status has changed to "Status: Limited Approval". The "Required" category now contains one item: "Include License". The other items remain in their respective categories: "Permitted" (Private Use, Place Warranty, Modify, Distribute, Commercial Use, Sub-License, Use Patent Claims), "Forbidden" (Hold Liable, Use Trademarks), and "Required" (Include License).

## Creating license terms

You can create a license term when viewing all available license terms or when viewing the terms that apply to a specific license.

Only users with the License Manager role can create license terms.

## ⚙️ To create a license term

1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon (☰) and select **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.

Term	Source	Responsibility	Category	Last Updated	Action
Modify	KnowledgeBase	Required	KnowledgeBase	Never	▼
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never	▼
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never	▼
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never	▼
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never	▼
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never	▼
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never	▼
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never	▼
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never	▼
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never	▼
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never	▼
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never	▼
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never	▼

3. Click **Create Terms** to open the Create a License Term dialog box.

The dialog box contains the following fields:

- Name \***: An input field with a placeholder character.
- Description \***: A large text area for entering a description.
- Responsibility \***: Radio buttons for **Required**, **Forbidden**, and **Permitted**.
- Category \***: A dropdown menu.
- Buttons**: **Cancel**, **Create and Associate with License** (highlighted in blue), and **Create**.

4. Complete the information in the dialog box:

- **Name.**
- **Description.**
- **Responsibility.** Select whether this responsibility is required, forbidden, or permitted.
- **Category.** Select a category for this license term. Optionally, create a new category by entering text in the field and selecting to add this new category. The new category will be automatically created.

5. Do one of the following:

- Click **Create and Associate with License**. The License Association dialog box appears. Select the licenses to associate to this license term, optionally select whether this term requires fulfillment, and click **Add**. Click [here](#) for more information about associating a term to a license.
- Click **Create**. The new license term appears in the table in the **License Terms** tab.

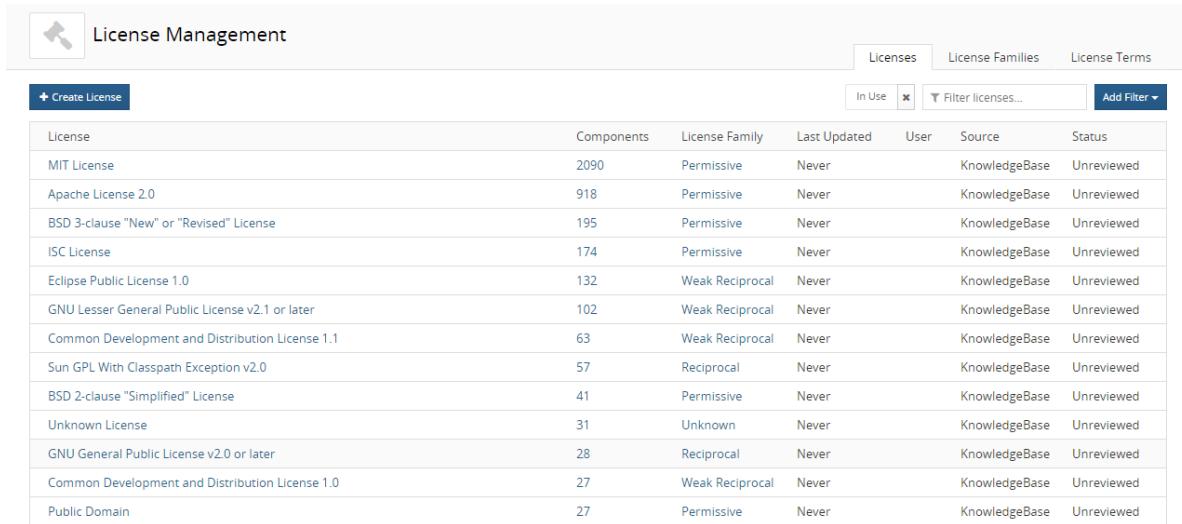
#### To create a license term for a specific license

1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon ( ) and select **License Management**.

The License Management page appears.



The screenshot shows the Black Duck License Management interface. At the top, there's a header with a wrench icon, the title "License Management", and three tabs: "Licenses" (selected), "License Families", and "License Terms". Below the header is a search bar with filters for "In Use" (unchecked) and "Filter licenses...", and a "Add Filter" button. A large table lists various licenses with columns for License Name, Components, License Family, Last Updated, User, Source, and Status. The table includes rows for MIT License, Apache License 2.0, BSD 3-clause "New" or "Revised" License, ISC License, Eclipse Public License 1.0, GNU Lesser General Public License v2.1 or later, Common Development and Distribution License 1.1, Sun GPL With Classpath Exception v2.0, BSD 2-clause "Simplified" License, Unknown License, GNU General Public License v2.0 or later, Common Development and Distribution License 1.0, and Public Domain.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

3. In the **Licenses** tab, select the license name to display the **License Name Settings** tab.

The screenshot shows the Apache License 2.0 settings page. At the top, it displays the license name "Apache License 2.0", its family "Permissive", and its status "Unreviewed". The main area contains several input fields and dropdown menus:

- License Terms**: A tab labeled "Settings" is selected. Other tabs include "Where Used" and "Notes".
- Name**: Apache License 2.0
- License Family**: Permissive
- Status**: Unreviewed
- Notes**: An empty text area.
- Expiration Date**: A date picker field.
- License Text**: A rich text editor containing the text: "Apache License Version 2.0, January 2004" followed by a separator line.

A "Save" button is located at the bottom right of the form.

4. Select the **License Terms** tab to view the terms associated with this tab.

The screenshot shows the Apache License 2.0 License Terms page. At the top, it displays the license name "Apache License 2.0", its family "Permissive", and its status "Unreviewed". The main area contains three columns of terms:

+ Add Term		
<b>Permitted</b>	<b>Forbidden</b>	<b>Required</b>
> Private Use	> Hold Liable	> State Changes
> Place Warranty	> Use Trademarks	> Include Notice
> Modify		> Include Copyright
> Distribute		> Include License
> Commercial Use		
> Sub-License		
> Use Patent Claims		

5. Select **New** to create a new term. The Add Term dialog box displays the fields you need to complete to create a new term.

Add Term

Existing  New

Name \*

Description \*

Responsibility \*  Required  Forbidden  Permitted

Category \*

Fulfillment  Required

6. Complete the information in the dialog box:

- **Name.**
- **Description.**
- **Responsibility.** Select whether this responsibility is required, forbidden, or permitted.
- **Category.** Select a category for this license term. Optionally, create a new category by entering text in the field and selecting to add this new category. The new category will be automatically created.
- **Fulfillment.** Indicate whether this term must be fulfilled.

7. Click **Add**. The new term is added to this license.

License Management  
Sample Custom License  
Family: Reciprocal | Status: Unreviewed

License Terms >  Permitted  Forbidden  Required  
Where Used  New Permitted Term

The new license term is also listed in the **License Terms** table. You can then [associate this term](#) to other licenses and specify whether the term must be fulfilled for those licenses.

## Managing license term categories

Categories help you manage and organize your license terms.

You must assign a license term to a category when you create the license term.

You can create or delete custom license term categories. License terms from the Black Duck KnowledgeBase are in the KnowledgeBase category. You cannot delete this category or add custom licenses to it.

Only users with the License Manager [role](#) can create or delete categories.

### To create a category

You can also create a category when [creating a license term](#).

1. Log in to Black Duck with the License Manager role.



2. Click the expanding menu icon ( ) and select **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.

Term	Source ^	Responsibility	Category	Last Updated	Actions
Modify	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>

3. Click **Categories**.

The License Terms Categories dialog box appears.

+ Create

There is no category yet.

Close

4. Click **Create** to display the field to enter the category name. Type the name of the new category in the field and select it (**Add Category Name**) located below the field. Click **Create** to create additional categories.
5. Click **Close** when you have finished creating categories.

#### To delete a category

You cannot delete a category that is in use.

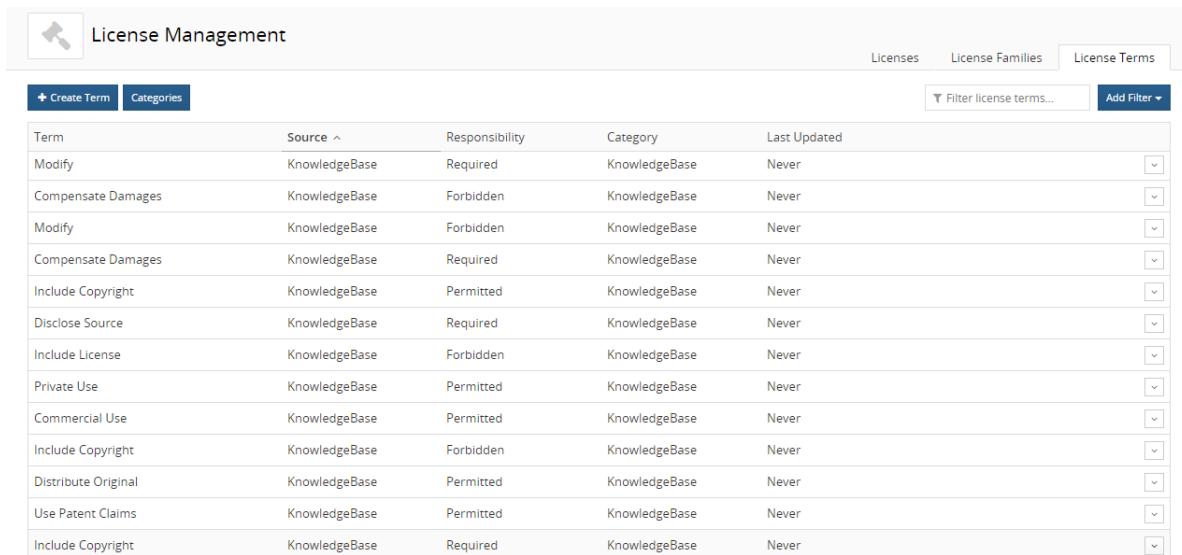
1. Log in to Black Duck with the License Manager role.



2. Click the expanding menu icon () and select **License Management**.

The License Management page appears.

Select the **Terms** tab to display all license terms.



Term	Source ^	Responsibility	Category	Last Updated	
Modify	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>

3. Click **Categories**.

The License Terms Categories dialog box appears.



4. Click  in the row of the category you want to delete.
5. Select **Delete** to confirm.

## Associating a license term to a license

You can associate a new license term you created or an existing KnowledgeBase term to one or more custom or KnowledgeBase licenses.

When a license term is associated to a license, that term will appear to users when viewing licenses terms, for example, in the BOM.

Only users with the License Manager role can associate a license term to a license.

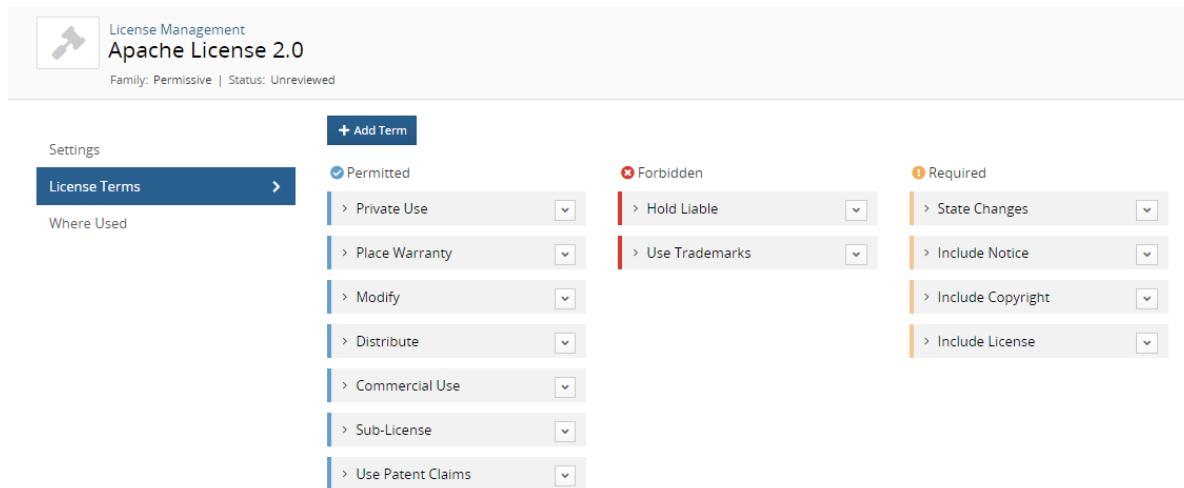
You can associate a term to a license when:

- Creating a license term. Click [here](#) for more information about creating a new term.
- Using the **License Terms** tab which lists all license terms:

The screenshot shows the "License Management" interface with the "License Terms" tab selected. At the top are buttons for "+ Create Term" and "Categories", and filters for "Filter license terms..." and "Add Filter". The main area is a table with columns: Term, Source, Responsibility, Category, and Last Updated. The table lists various license terms, each with a dropdown arrow icon to its right.

Term	Source	Responsibility	Category	Last Updated
Modify	KnowledgeBase	Required	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never

■ Using the **License Terms** tab for an individual license:



The screenshot shows the Apache License 2.0 page in the License Management interface. The 'License Terms' tab is selected. The page displays a grid of license terms categorized into three groups: Permitted (blue), Forbidden (red), and Required (orange). Each term has a dropdown menu next to it.

Category	Term
Permitted	Private Use
Permitted	Place Warranty
Permitted	Modify
Permitted	Distribute
Permitted	Commercial Use
Permitted	Sub-License
Permitted	Use Patent Claims
Forbidden	Hold Liable
Forbidden	Use Trademarks
Required	State Changes
Required	Include Notice
Required	Include Copyright
Required	Include License

 To associate a license term to one or more licenses

Use these procedures to associate a license term to one or more licenses.

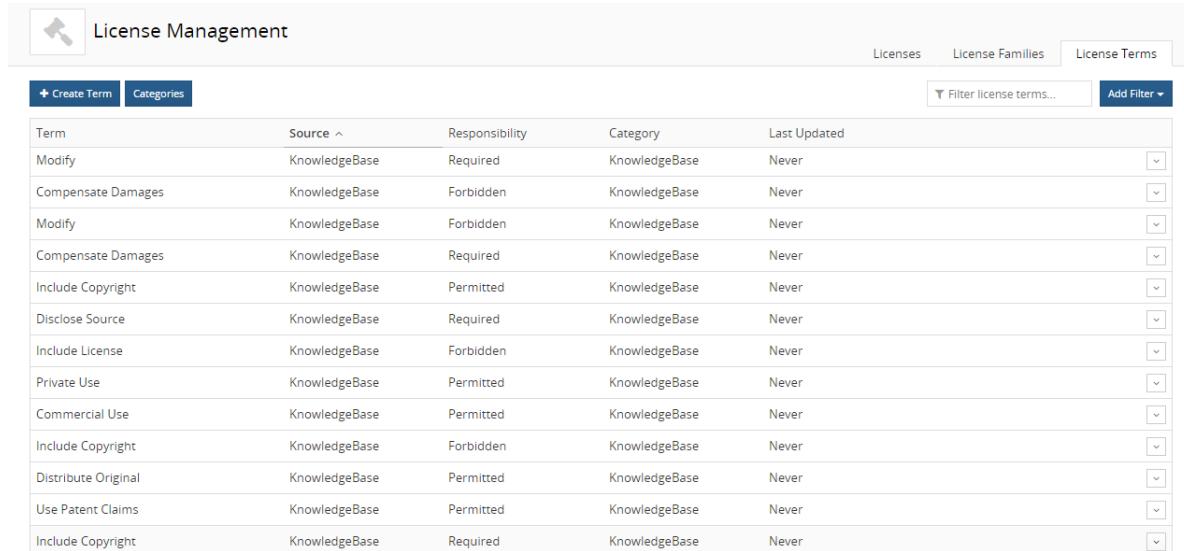
1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon () and select **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.

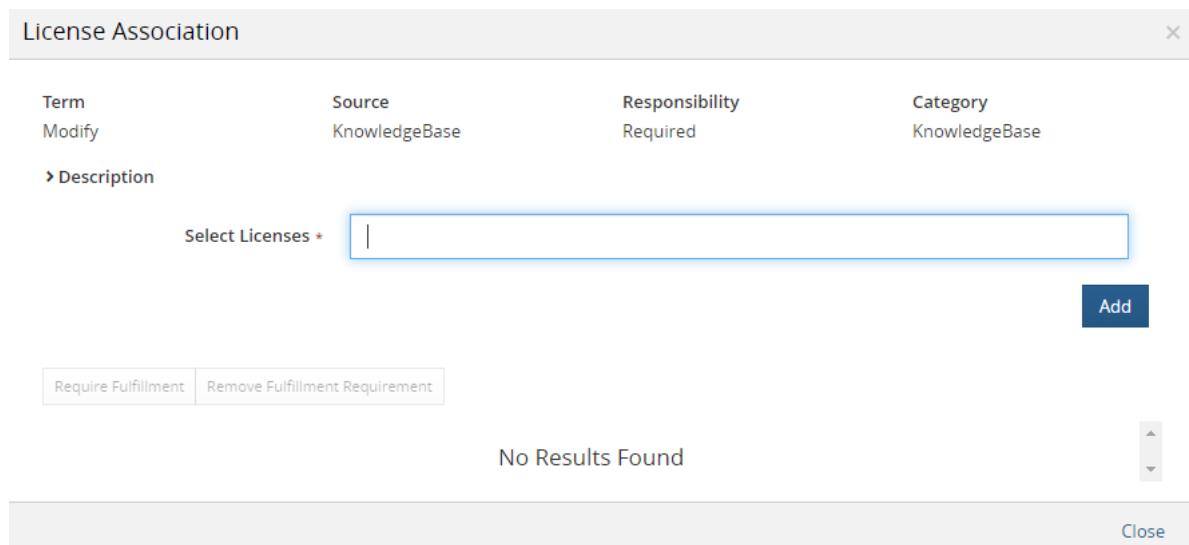


The screenshot shows the License Management page with the 'License Terms' tab selected. The page features a search bar and filter options at the top, followed by a table listing various license terms with columns for Term, Source, Responsibility, Category, and Last Updated.

Term	Source	Responsibility	Category	Last Updated
Modify	KnowledgeBase	Required	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never

3. Click  in the row of the license term and select **License Association**.

The License Association dialog box appears.



Term	Source	Responsibility	Category
Modify	KnowledgeBase	Required	KnowledgeBase

Description

Select Licenses + |

Add

Require Fulfillment Remove Fulfillment Requirement

No Results Found

Close

4. Use this dialog box to associate the term. To add a license: Begin typing the license name that you want to associate to this term. The list is type-ahead enabled, so you can see a list of available licenses that contain the text you have typed. Select the license and click **Add**.

Enter additional license names to associate the term with additional licenses.

5. Optionally, select the licenses for which this term requires fulfillment:

a. Select the check box next to the license where fulfillment of this term is required.

b. Click **Require Fulfillment**. The Fulfillment Required icon () appears in the table for the license where this term is required.

Click **Remove Fulfillment Requirement** to remove the requirement that this term must be fulfilled.

6. Click **Close**.

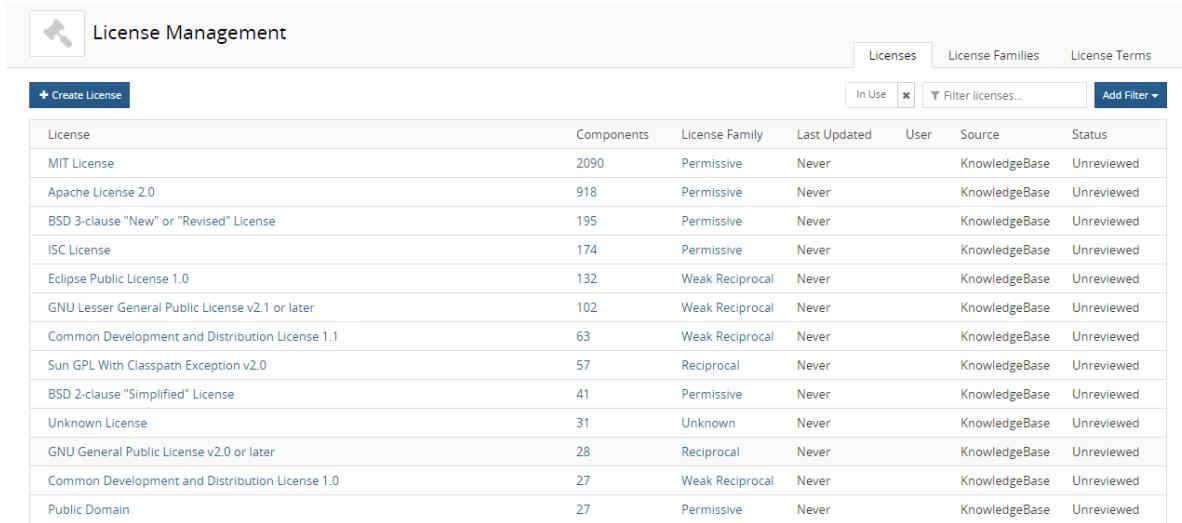
 **To associate an existing license term to a specific license**

1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon () and select **License Management**.

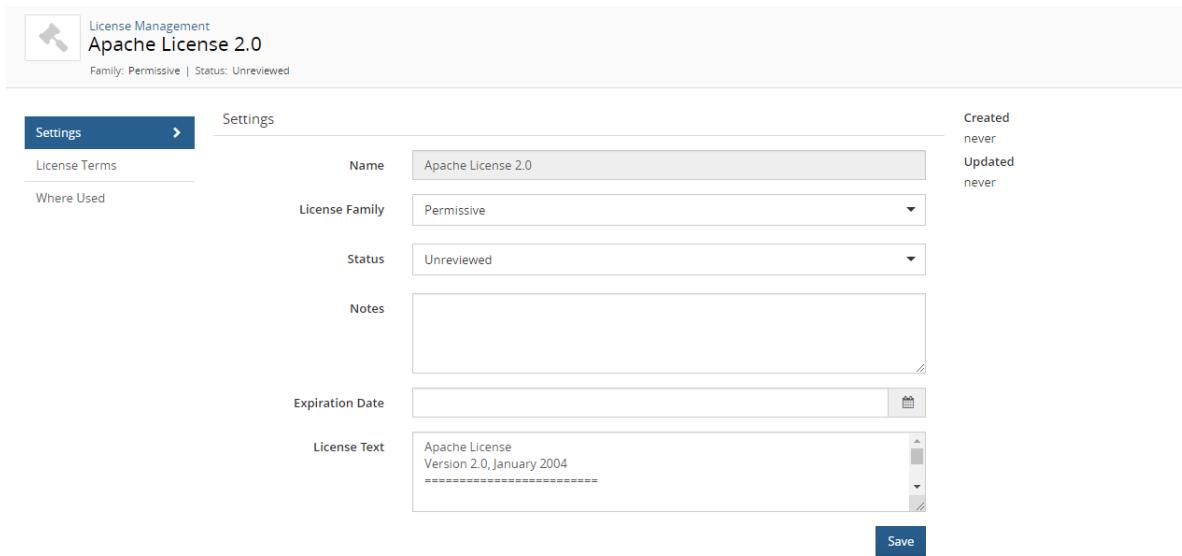
The License Management page appears.



The screenshot shows a table titled "License Management" with a "Create License" button. The columns are: License, Components, License Family, Last Updated, User, Source, and Status. The data includes various open-source licenses like MIT License, Apache License 2.0, BSD 3-clause "New" or "Revised" License, ISC License, Eclipse Public License 1.0, etc.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

3. In the **Licenses** tab, select the license name to display the *License Name Settings* tab.



The screenshot shows the "Settings" tab for the Apache License 2.0. It includes fields for Name (Apache License 2.0), License Family (Permissive), Status (Unreviewed), Notes (empty), Expiration Date (empty), and License Text (Apache License Version 2.0, January 2004). A "Save" button is at the bottom right.

4. Select the **License Terms** tab to view the terms associated with this tab.

The screenshot shows the Apache License 2.0 settings page. At the top, there's a key icon and the text "License Management Apache License 2.0". Below that, it says "Family: Permissive | Status: Unreviewed". There are three main sections: "Settings", "License Terms" (which is selected and highlighted in blue), and "Where Used". Under "License Terms", there are three tabs: "Permitted" (selected), "Forbidden", and "Required". The "Permitted" tab contains the following items: Private Use, Place Warranty, Modify, Distribute, Commercial Use, Sub-License, and Use Patent Claims. The "Forbidden" tab contains Hold Liable and Use Trademarks. The "Required" tab contains State Changes, Include Notice, Include Copyright, and Include License.

5. Click **Add Term** to open the Add Term dialog box.
6. Select **Existing** to add an existing license term.

The "Add Term" dialog box is open. It has a title bar "Add Term" and a close button "X". Inside, there are two radio buttons: "Existing" (selected) and "New". Below that is a "Name \*" field with a dropdown placeholder "Start typing to add a term...". Next is a "Description \*" field with a large text area. Then is a "Responsibility \*" section with radio buttons for "Required", "Forbidden", and "Permitted" (none selected). Below that is a "Category \*" field with a dropdown. At the bottom left is a "Fulfillment" section with a checked checkbox "Required". At the bottom right are "Cancel" and "Add" buttons, with "Add" being highlighted.

7. Begin typing the license name that you want to associate to this term. The list is type-ahead enabled, so you can see a list of available license terms that contain the text you have typed. This list displays all license terms – custom and KnowledgeBase terms.
8. Select the license term. The information for this term appears in the dialog box.

9. Optionally, select whether fulfillment is required for this term.
10. Click **Add**. The **License Terms** tab appears for this license with the new term added. The Fulfillment Required icon (checkbox) will appear for any required terms.

## Editing a custom license term

You can edit custom license terms.

Only users with the License Manager role can edit license terms.

### To edit a license term

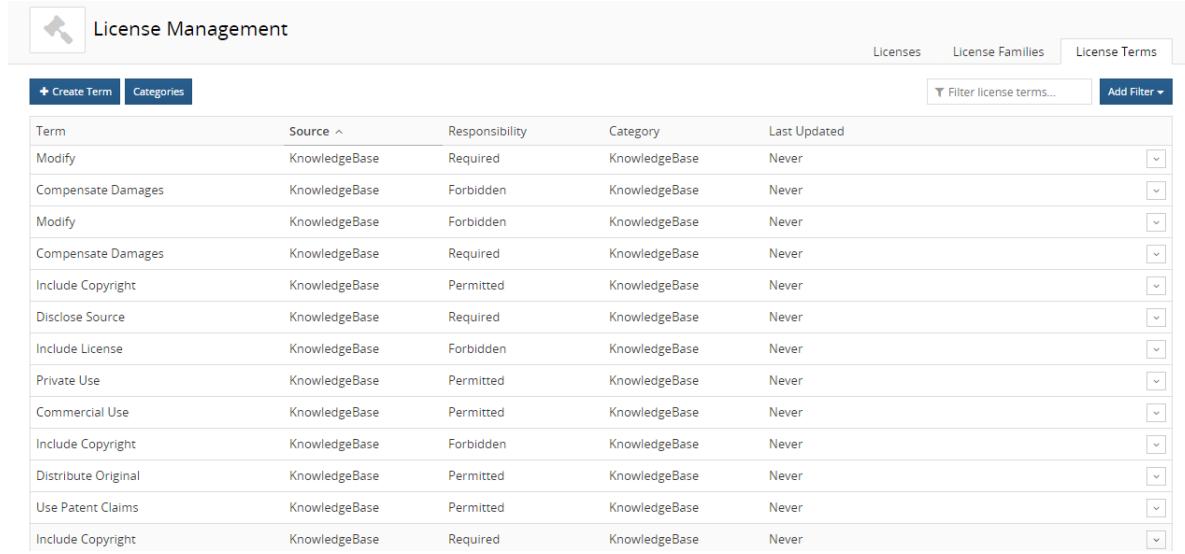
1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon (≡) and select **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.



The screenshot shows the Black Duck License Management interface. At the top, there's a navigation bar with a key icon, the title "License Management", and three tabs: "Licenses", "License Families", and "License Terms". The "License Terms" tab is currently selected. Below the tabs is a search bar with the placeholder "Filter license terms..." and a "Add Filter" button. Underneath is a table with columns: Term, Source, Responsibility, Category, and Last Updated. The table lists various license terms such as "Modify", "Compensate Damages", etc., each with its source set to "KnowledgeBase" and responsibility status. There are dropdown arrows next to each row for further configuration.

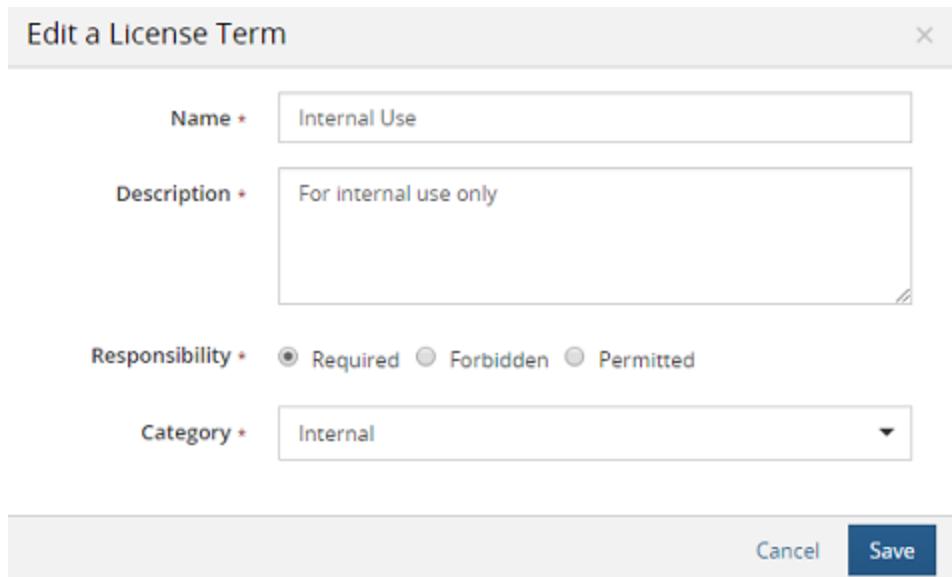
Term	Source	Responsibility	Category	Last Updated
Modify	KnowledgeBase	Required	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never

3. Select the license term to open the Edit a License Term dialog box.

Edit a License Term

Name *	Internal Use
Description *	For internal use only
Responsibility *	<input checked="" type="radio"/> Required <input type="radio"/> Forbidden <input type="radio"/> Permitted
Category *	Internal

Cancel **Save**



4. Edit the information in the dialog box and click **Save**.

## Deleting a license term

You can only delete custom license terms.

You cannot delete a KnowledgeBase license terms. Instead you can [deactivate a KnowledgeBase license term](#) so that the term does not apply to a specific license.

Only users with the License Manager role can delete license terms.

You cannot delete a custom license term that is associated to a license.

### ⚙️ To delete a license term

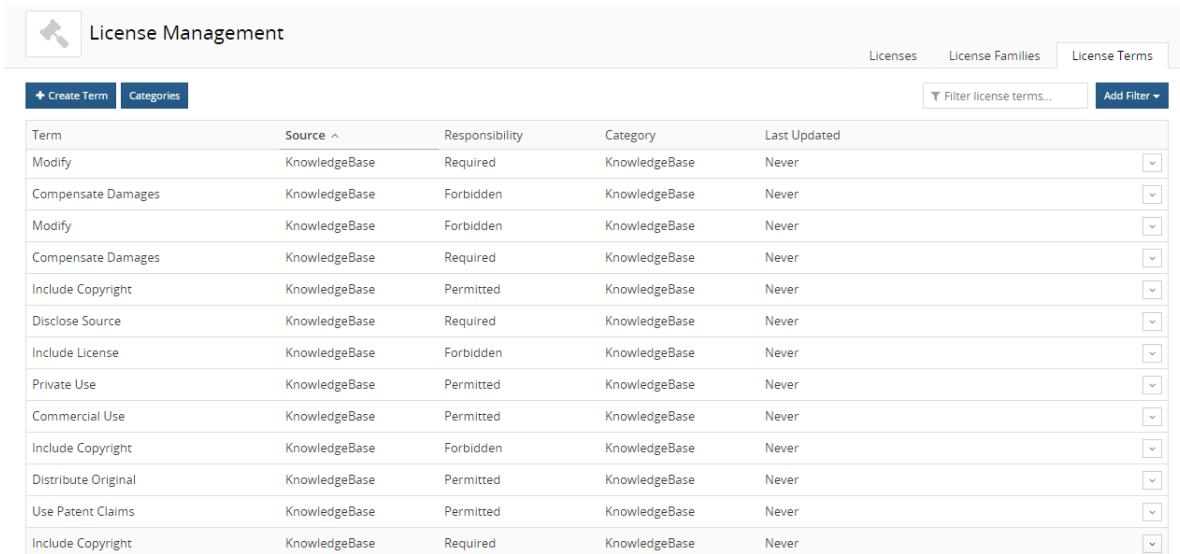
1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon (☰) and select **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.



The screenshot shows the Black Duck License Management interface. At the top, there's a navigation bar with tabs for 'Licenses', 'License Families', and 'License Terms'. Below the navigation bar is a search bar labeled 'Filter license terms...' and a 'Add Filter' button. On the left, there are two buttons: '+ Create Term' and 'Categories'. The main area is a table with columns: 'Term', 'Source ^', 'Responsibility', 'Category', and 'Last Updated'. Each row in the table represents a license term, with a small square icon containing a delete symbol in the first column.

Term	Source ^	Responsibility	Category	Last Updated
Modify	KnowledgeBase	Required	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never

- Click  in the row of the license term and select **Delete**.

The Delete a License Term dialog box appears.

- Click **Delete** to confirm.

## Deprecating or removing the deprecation status of a custom license term

You can deprecate a custom license term. Deprecating a custom license term is a global action – it applies to all licenses (custom and KnowledgeBase) that have this custom license term associated to it.

A deprecated custom license term is not available for new associations to licenses and cannot be edited. Existing licenses that have the deprecated term will still display the term to users in existing or new projects/components with no indication to these users that the term is deprecated.

Only users with the License Manager role can deprecate license terms.

### To deprecate a custom license term

Use these procedures to deprecate the term for *all* licenses that have this term associated to it.

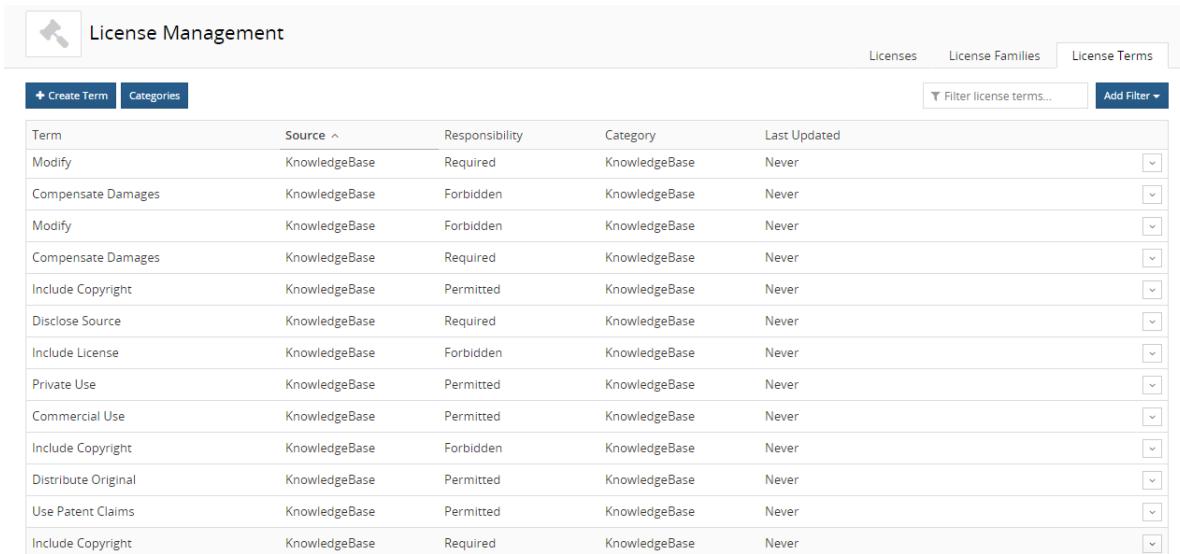
- Log in to Black Duck with the License Manager [role](#).



- Click the expanding menu icon () and select **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.



The screenshot shows the Black Duck License Management interface. At the top, there are tabs for 'Licenses', 'License Families', and 'License Terms'. Below the tabs is a search bar labeled 'Filter license terms...' and a 'Add Filter' button. The main area is a table with columns: Term, Source, Responsibility, Category, and Last Updated. The table lists various license terms such as 'Modify', 'Compensate Damages', and 'Include Copyright', each with its source (KnowledgeBase), responsibility (Required, Forbidden, Permitted), category (KnowledgeBase), and last update date (Never). There are also small dropdown arrows next to each row.

Term	Source	Responsibility	Category	Last Updated
Modify	KnowledgeBase	Required	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never

3. Click  in the row of the license term and select **Deprecate**.

The Deprecate a License Term dialog box appears.

4. Click **Deprecate** to confirm.

The date and username of the user who deprecated this term appears in the **Last Updated** column.

 **Deprecated** The  **Deprecated** label appears next to the license term where the term appears in the **License Terms** tabs in License Management.

Note that the  **Deprecated** label does not appear to the BOM manager for any licenses that have this term associated to it.

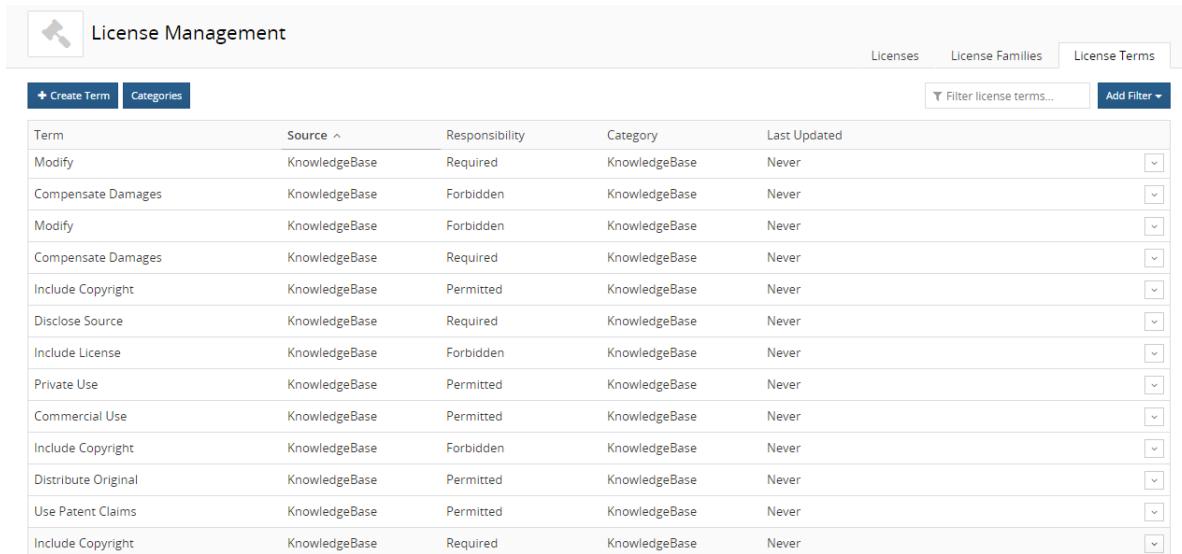
#### To undo the deprecation status of a custom license term

1. Log in to Black Duck with the License Manager [role](#).

2. Click the expanding menu icon () and select **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.



The screenshot shows a table titled "License Management" with a "Licenses" tab selected. The table has columns: Term, Source, Responsibility, Category, and Last Updated. There are 15 rows of data. Each row contains a small square icon with a downward arrow in the bottom right corner.

Term	Source	Responsibility	Category	Last Updated
Modify	KnowledgeBase	Required	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never

- Click  in the row of the license term and select **Remove Deprecated Status**.

The Deprecate a License Term dialog box appears.

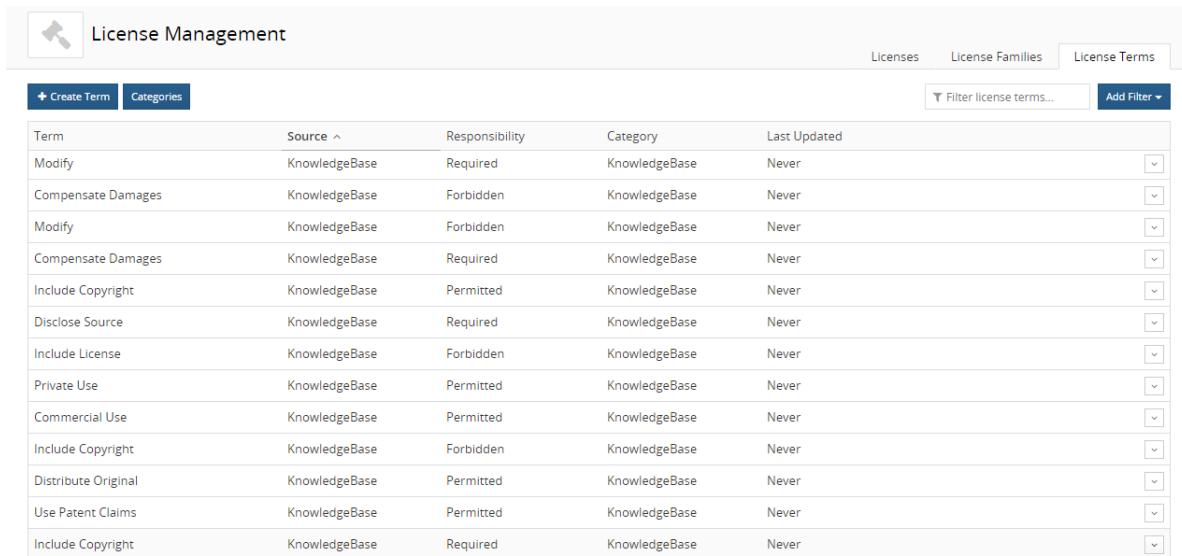
- Click **Remove Deprecated Status** to confirm. The  **Deprecated** label is removed from the license term.

## Removing a license term

Use these procedures to remove a license term that you associated to a custom license or a KnowledgeBase license. When you remove a license term from a license, the term no longer appears to users viewing license terms, for example when BOM Managers view license information in the BOM.

There are two methods you can use to remove a license term from a license:

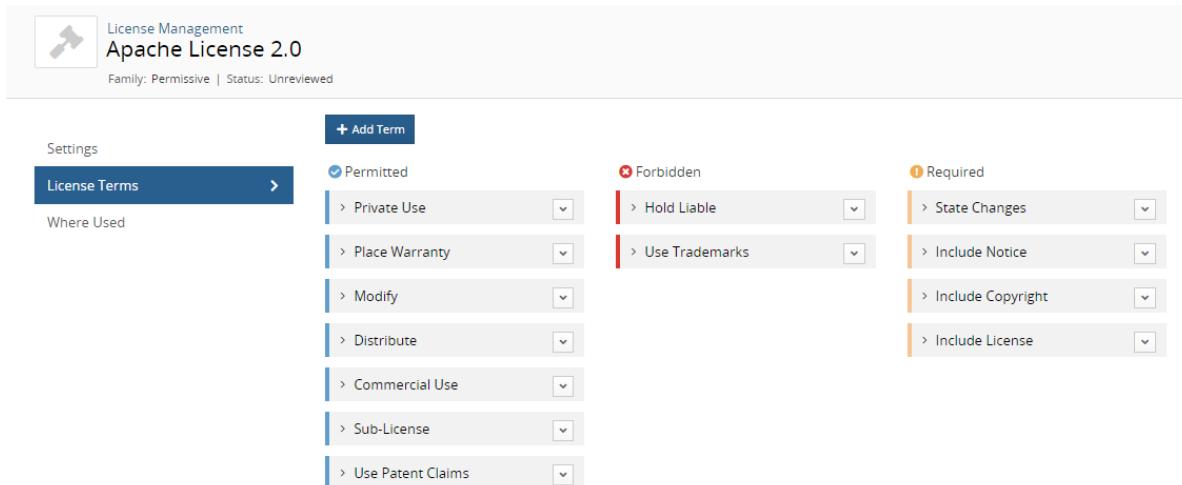
■ Using the **License Terms** tab which lists all license terms



The screenshot shows the Black Duck License Management interface. At the top, there's a navigation bar with icons for hammer and wrench, followed by the title "License Management". Below the navigation bar are three tabs: "Licenses" (selected), "License Families", and "License Terms". A search bar labeled "Filter license terms..." and a "Add Filter" button are located to the right of the tabs. Below the tabs is a table with columns: Term, Source, Responsibility, Category, and Last Updated. The table contains 15 rows of license terms, each with a dropdown arrow icon on the far right.

Term	Source	Responsibility	Category	Last Updated
Modify	KnowledgeBase	Required	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never

■ Using the **License Terms** tab for an individual license



The screenshot shows the Apache License 2.0 settings page. At the top, it displays the license name "Apache License 2.0" and its family "Permissive" and status "Unreviewed". Below the title is a "Settings" section with a "License Terms" tab selected. To the right of the tabs is a "+ Add Term" button. The main area contains three columns of terms: "Permitted" (blue), "Forbidden" (red), and "Required" (orange). Each column has a dropdown menu with several options listed.

Permitted	Forbidden	Required
Private Use	Hold Liable	State Changes
Place Warranty	Use Trademarks	Include Notice
Modify		Include Copyright
Distribute		Include License
Commercial Use		
Sub-License		
Use Patent Claims		

⚙ To remove a license term from one or more licenses

Use this method to remove a term from many licenses or if you want to view all the licenses to which this term is associated.

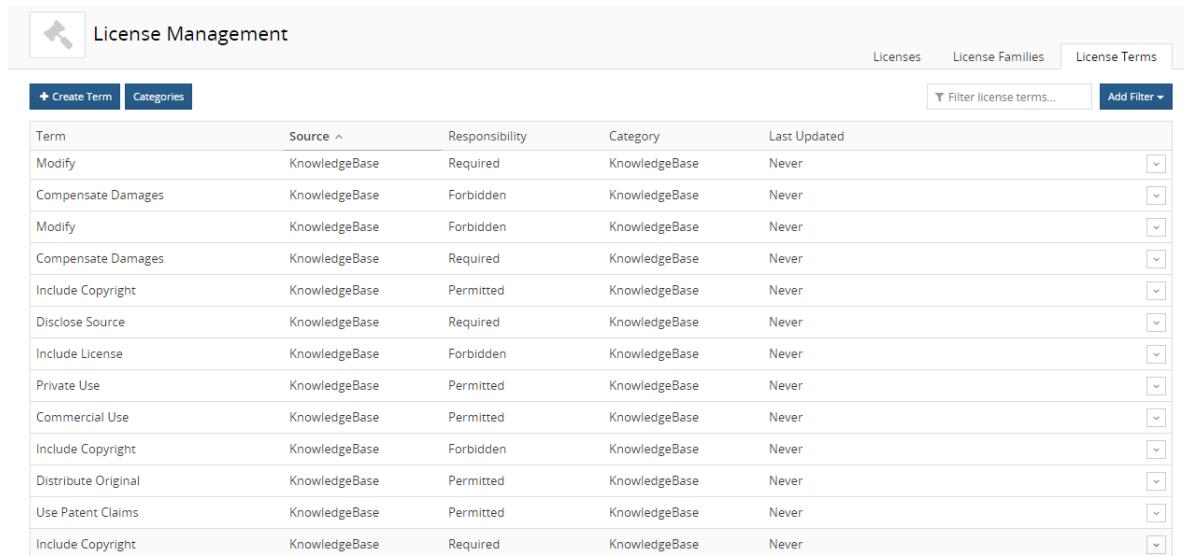
1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon () and select **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.



The screenshot shows the 'License Management' interface with the 'License Terms' tab selected. The table displays various license terms with columns for Term, Source, Responsibility, Category, and Last Updated. Each row has a dropdown arrow icon at the end of the last column.

Term	Source	Responsibility	Category	Last Updated	
Modify	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>

3. Click  in the row of the license term and select **License Association**.

The License Association dialog box appears showing all licenses that have this license terms associated to it.

### License Association

Term	Source	Responsibility	Category
Private Use	KnowledgeBase	Permitted	KnowledgeBase

**Description**

Select Licenses +

Add

Require Fulfillment | Remove Fulfillment Requirement

License	Fulfillment Required
MIT License	-
Artistic License 2.0	-
Apache License 2.0	-
Eclipse Public License 1.0	-

Displaying 1-4 of 4

**Close**

4. Click  in the row of the license you want to disassociate from this license term.
5. Select **Delete** to confirm.

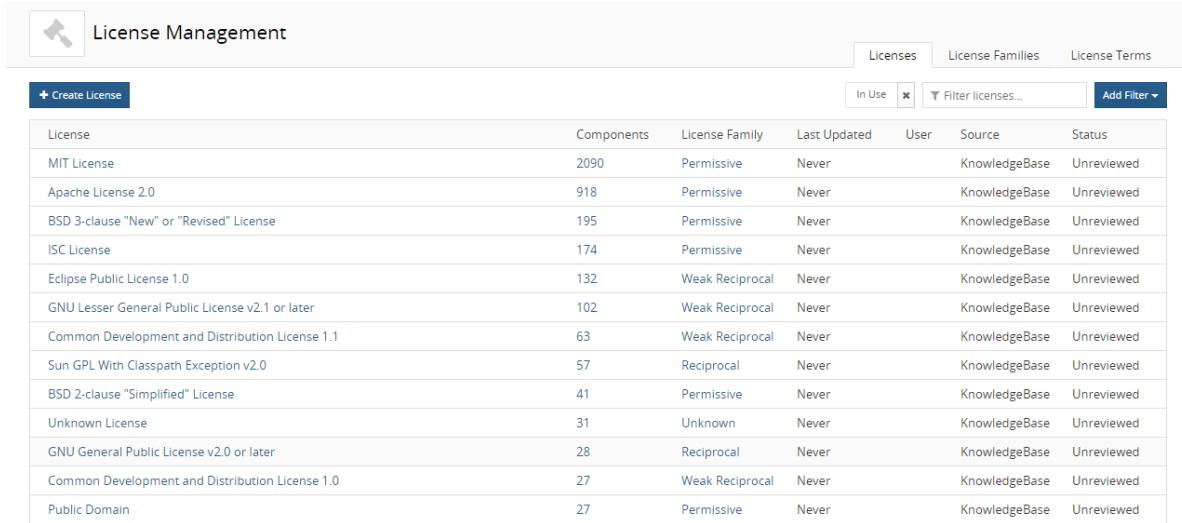
The term is removed from the license.

#### To remove a license term from a single license

Use this method to remove a license term when viewing all terms for a single license.

1. Log in to Black Duck with the License Manager [role](#).
2. Click the expanding menu icon () and select **License Management**.

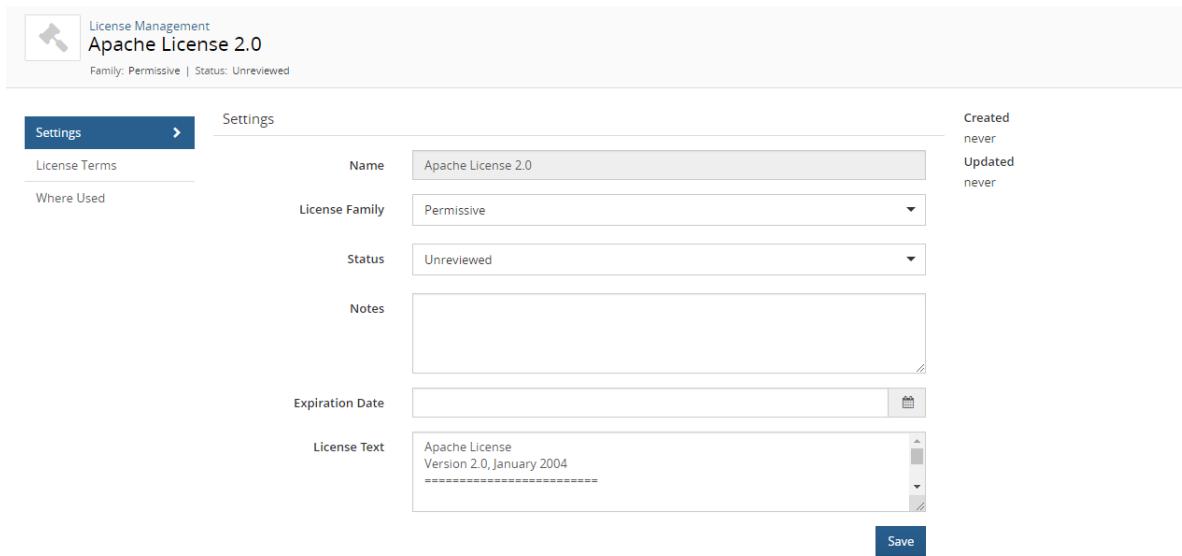
The License Management page appears.



The screenshot shows a table titled "License Management" with a "Create License" button. The columns are: License, Components, License Family, Last Updated, User, Source, and Status. The data includes various open-source licenses like MIT, Apache, BSD, and GPL, along with their respective counts and details.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

3. In the **Licenses** tab, select the license name to display the *License Name Settings* tab.



The screenshot shows the "Settings" tab for the Apache License 2.0. It includes fields for Name (Apache License 2.0), License Family (Permissive), Status (Unreviewed), Notes (empty), Expiration Date (empty), and License Text (Apache License Version 2.0, January 2004). A "Save" button is at the bottom right. On the left, there are tabs for "Settings" (selected) and "License Terms". Above the form, it says "Family: Permissive | Status: Unreviewed".

4. Select the **License Terms** tab to view the terms associated with this tab.

Category	Term
Permitted	Private Use
Permitted	Place Warranty
Permitted	Modify
Permitted	Distribute
Permitted	Commercial Use
Permitted	Sub-License
Permitted	Use Patent Claims
Forbidden	Hold Liable
Forbidden	Use Trademarks
Required	State Changes
Required	Include Notice
Required	Include Copyright
Required	Include License

5. Click in the row of the license term of the term you wish to remove and select **Remove**.

The Remove Term dialog box appears.

6. Click **Remove** to confirm.

The **License Terms** tab displays the terms for this license with the term removed.

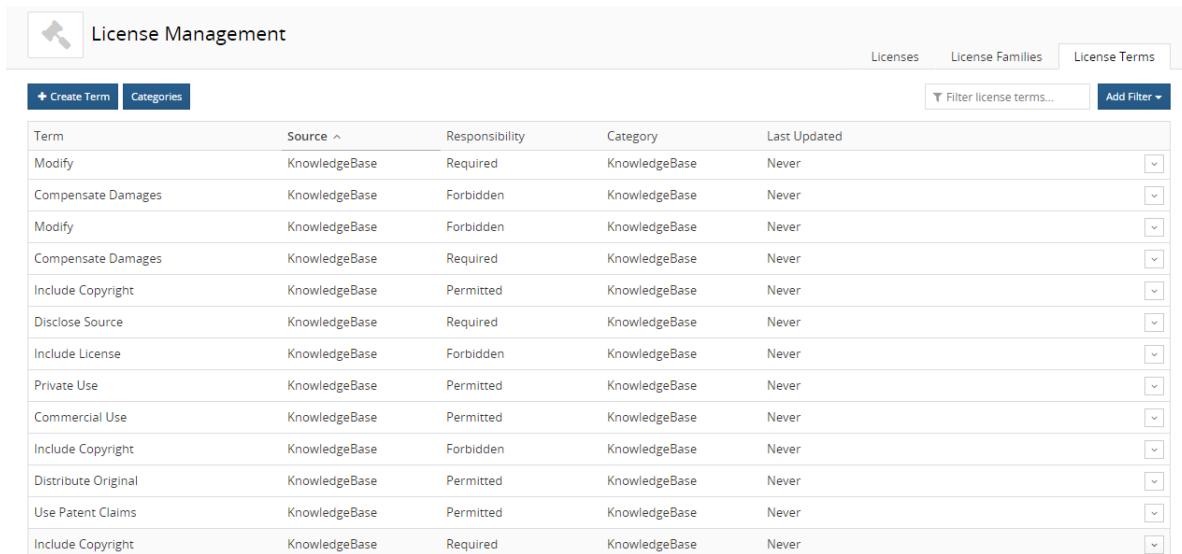
## Deactivating a KnowledgeBase term

You may decide not to show your users specific license terms that are defined by the Black Duck KnowledgeBase.

When a term is deactivated, it does not appear when users view the terms for a KnowledgeBase license; for example, when BOM Managers view the license terms in the BOM.

There are two methods you can use to deactivate a KnowledgeBase license term:

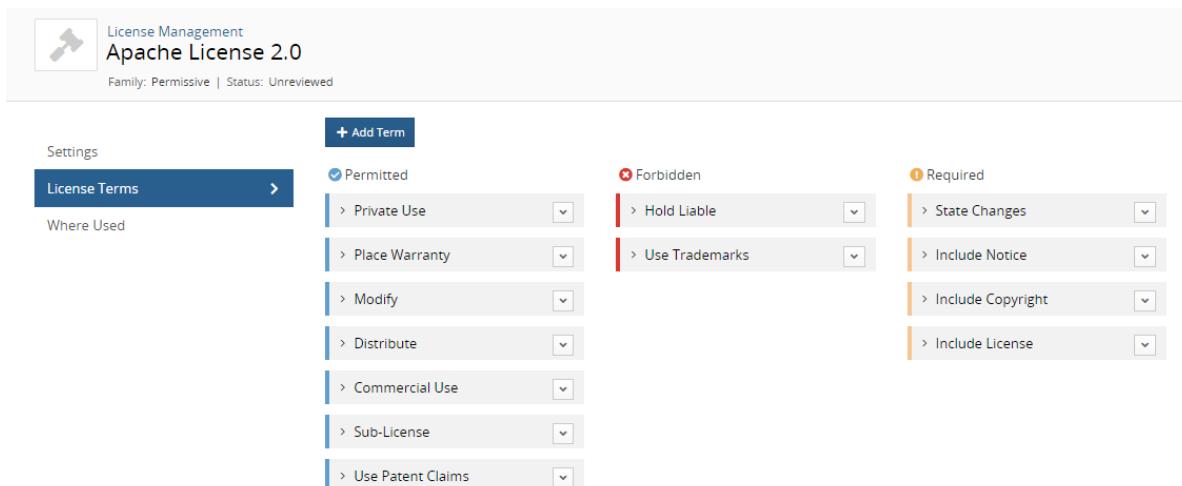
■ Using the **License Terms** tab which lists all license terms



The screenshot shows the Black Duck License Management interface. At the top, there's a navigation bar with tabs: 'Licenses' (selected), 'License Families', and 'License Terms'. Below the navigation bar is a search bar labeled 'Filter license terms...' and a 'Add Filter' button. On the left, there are buttons for '+ Create Term' and 'Categories'. The main area is a table with columns: 'Term', 'Source', 'Responsibility', 'Category', and 'Last Updated'. The table contains 15 rows of license terms, each with a dropdown arrow icon on the right.

Term	Source	Responsibility	Category	Last Updated
Modify	KnowledgeBase	Required	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never

■ Using the **License Terms** tab for an individual license



The screenshot shows the Apache License 2.0 settings page. At the top, it says 'Family: Permissive | Status: Unreviewed'. Below that is a 'Settings' section with a 'License Terms' tab selected. There are three columns of terms: 'Permitted' (blue), 'Forbidden' (red), and 'Required' (orange). Each column has a dropdown arrow icon next to it.

Setting	Value
Permitted	Private Use, Place Warranty, Modify, Distribute, Commercial Use, Sub-License, Use Patent Claims
Forbidden	Hold Liable, Use Trademarks
Required	State Changes, Include Notice, Include Copyright, Include License

Deactivated KnowledgeBase license terms can [be restored](#).

⚙ To deactivate a KnowledgeBase license term when viewing all terms

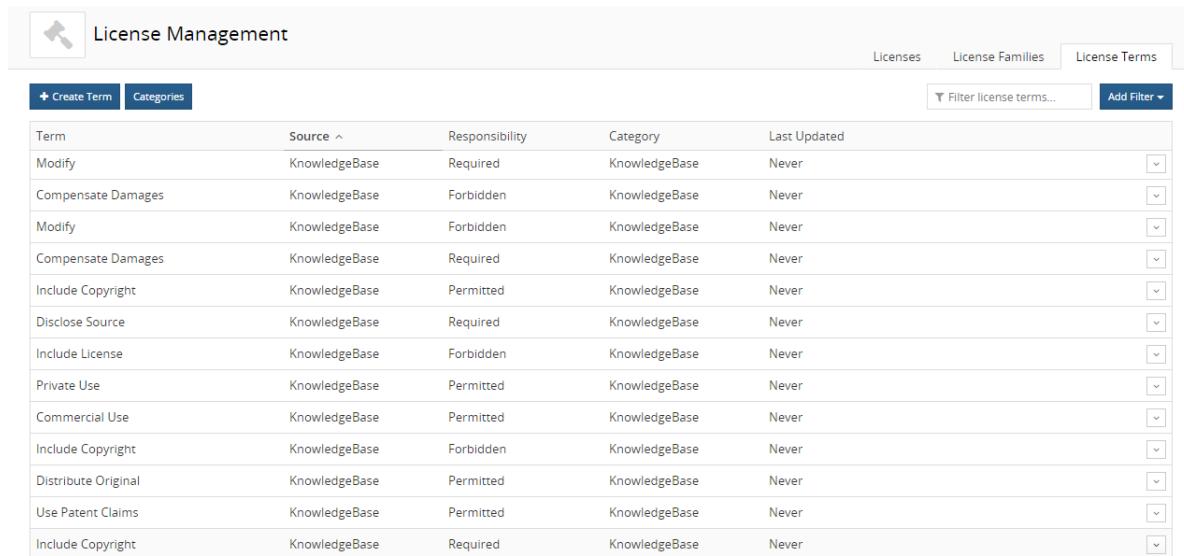
1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon ( ) and select **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.



The screenshot shows the 'License Management' interface with the 'License Terms' tab selected. The table displays various license terms with columns for Term, Source, Responsibility, Category, and Last Updated. Each row has a dropdown arrow icon at the end of the last column.

Term	Source	Responsibility	Category	Last Updated	
Modify	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>

3. Click  in the row of the license term and select **License Association**.

The License Association dialog box appears showing all licenses that have this license terms associated to it.

License Association

Term	Source	Responsibility	Category
Private Use	KnowledgeBase	Permitted	KnowledgeBase

> Description

Select Licenses \*

Add

Require Fulfillment | Remove Fulfillment Requirement

License	Fulfillment Required
MIT License	-
Artistic License 2.0	-
Apache License 2.0	-
Eclipse Public License 1.0	-

Displaying 1-4 of 4

Close

The screenshot shows a 'License Association' dialog box. At the top, there's a table with four columns: Term (Private Use), Source (KnowledgeBase), Responsibility (Permitted), and Category (KnowledgeBase). Below this is a section titled 'Description' with a 'Select Licenses \*' dropdown and an 'Add' button. Under 'Description', there are two buttons: 'Require Fulfillment' and 'Remove Fulfillment Requirement'. A table below lists five licenses: MIT License, Artistic License 2.0, Apache License 2.0, and Eclipse Public License 1.0, each with a minus sign next to it. To the right of each license is a small trash can icon. At the bottom right of the dialog is a 'Close' button.

4. Click  in the row of the license you want to disassociate to this license term.
5. Select **Deactivate** to confirm. The license term is no longer associated to that license.

License Association

Term	Source	Responsibility	Category
Private Use	KnowledgeBase	Permitted	KnowledgeBase

› Description

Select Licenses \*

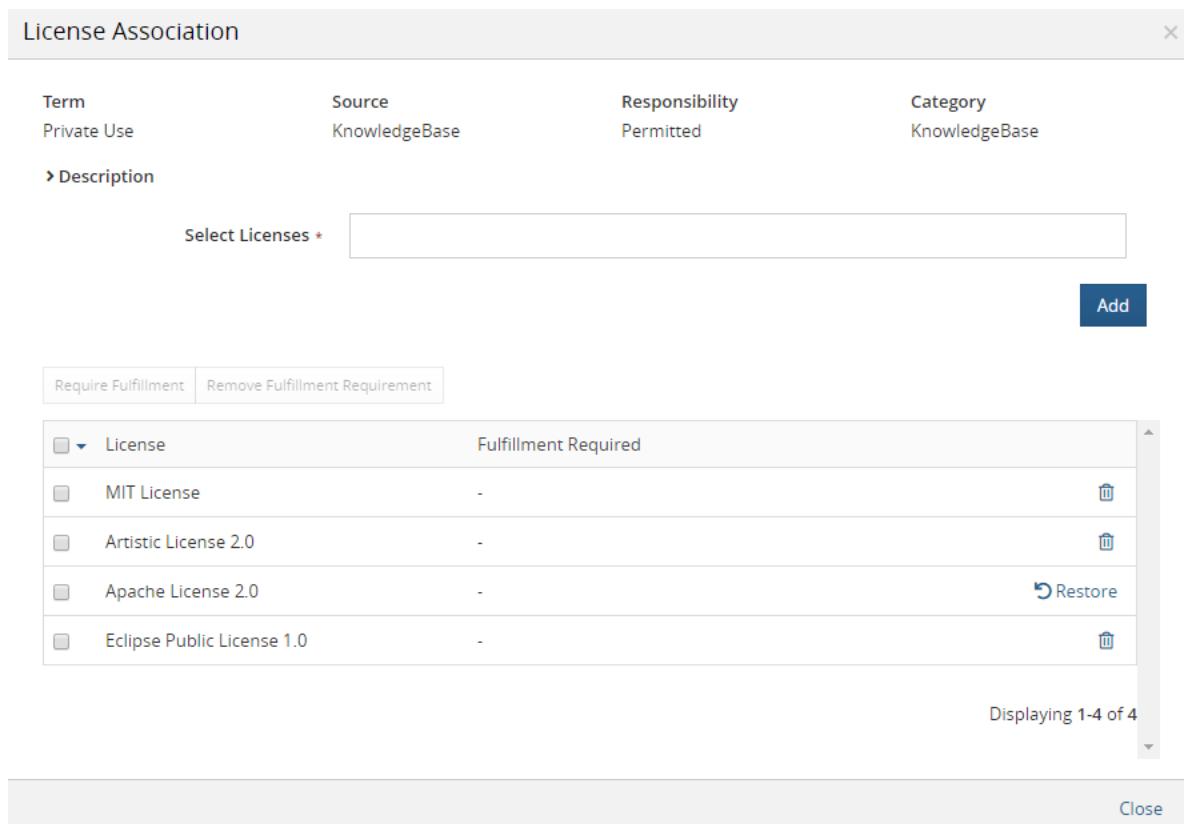
Add

Require Fulfilment | Remove Fulfilment Requirement

License	Fulfillment Required
MIT License	-
Artistic License 2.0	-
Apache License 2.0	-
Eclipse Public License 1.0	-

Displaying 1-4 of 4

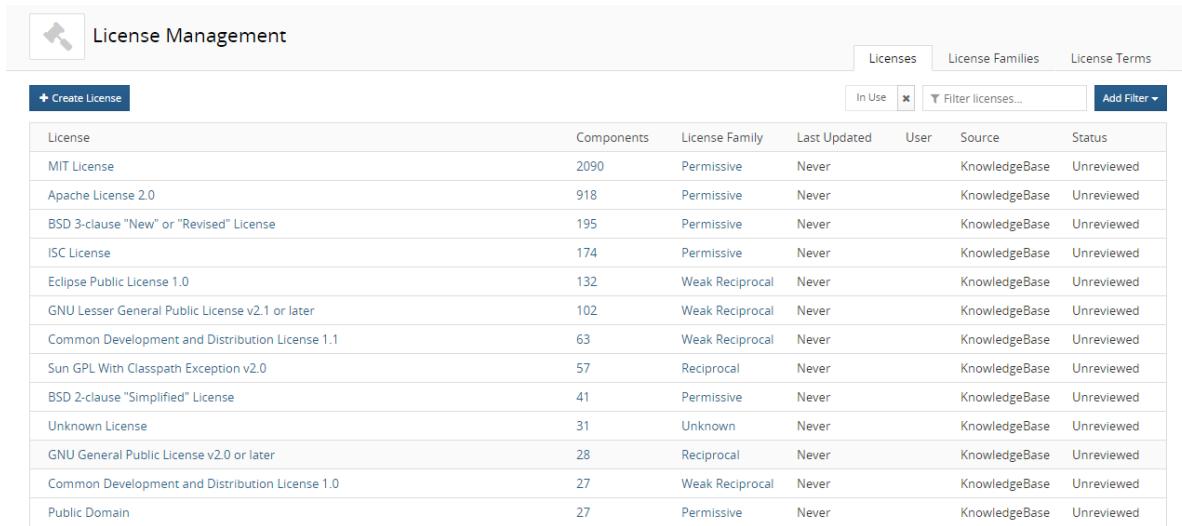
Close



⚙️ To deactivate a KnowledgeBase license term when viewing a license

1. Log in to Black Duck with the License Manager [role](#).
2. Click the expanding menu icon () and select **License Management**.

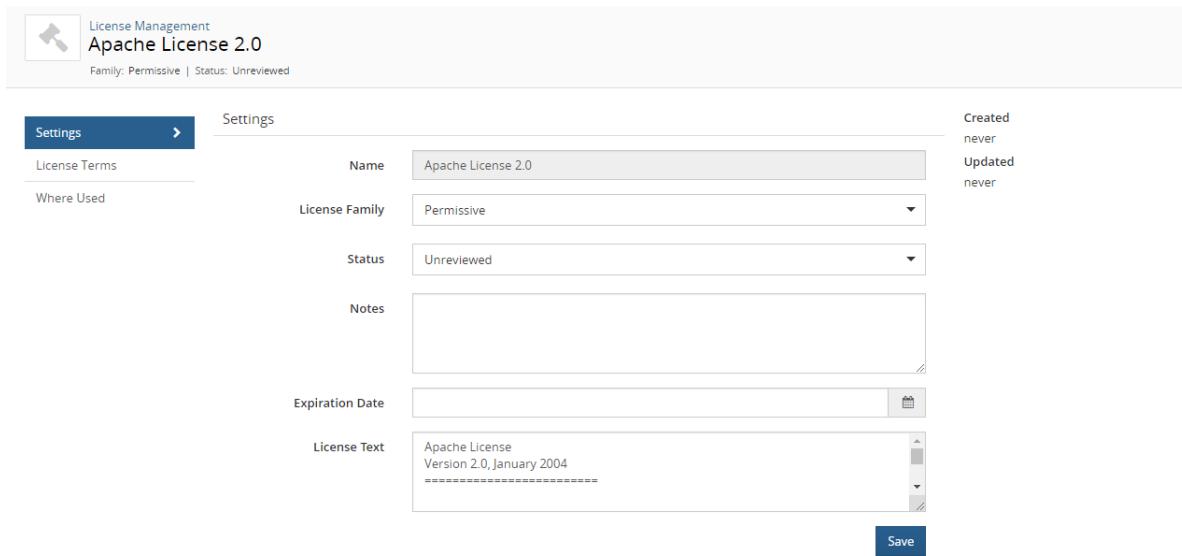
The License Management page appears.



The screenshot shows a table titled "License Management" with a "Create License" button. The columns are: License, Components, License Family, Last Updated, User, Source, and Status. The data includes various open-source licenses like MIT License, Apache License 2.0, BSD 3-clause "New" or "Revised" License, ISC License, Eclipse Public License 1.0, etc.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

3. In the **Licenses** tab, select the license name to display the *License Name Settings* tab.



The screenshot shows the "Settings" tab for the Apache License 2.0. It includes fields for Name (Apache License 2.0), License Family (Permissive), Status (Unreviewed), Notes (empty), Expiration Date (empty), and License Text (Apache License Version 2.0, January 2004). A "Save" button is at the bottom right.

4. Select the **License Terms** tab to view the terms associated with this tab.

The screenshot shows the Apache License 2.0 settings page. The 'License Terms' tab is active. Under the 'Permitted' section, 'Private Use' is selected. In the 'Forbidden' section, 'Hold Liable' and 'Use Trademarks' are selected. In the 'Required' section, 'State Changes', 'Include Notice', 'Include Copyright', and 'Include License' are selected.

5. Click next to the KnowledgeBase license term you wish to deactivate and select **Deactivate**.  
The Deactivate Term dialog box appears.
6. Click **Deactivate** to confirm.

The **License Terms** tab displays the term as deactivated.

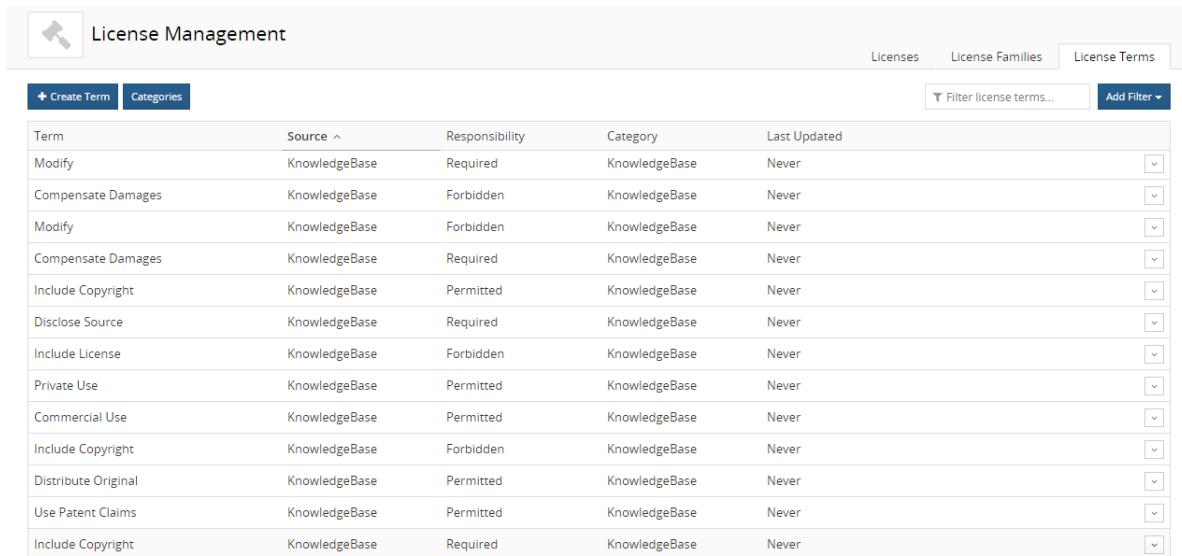
The screenshot shows the Apache License 2.0 settings page. The 'License Terms' tab is active. Under the 'Permitted' section, 'Private Use' is selected. In the 'Forbidden' section, 'Hold Liable' and 'Use Trademarks' are selected. In the 'Required' section, 'State Changes', 'Include Notice', 'Include Copyright', and 'Include License' are selected.

## Restoring a KnowledgeBase license term

Use these procedures to restore a KnowledgeBase license term that you previously [deactivated](#).

There are two methods you can use to restore a KnowledgeBase license term:

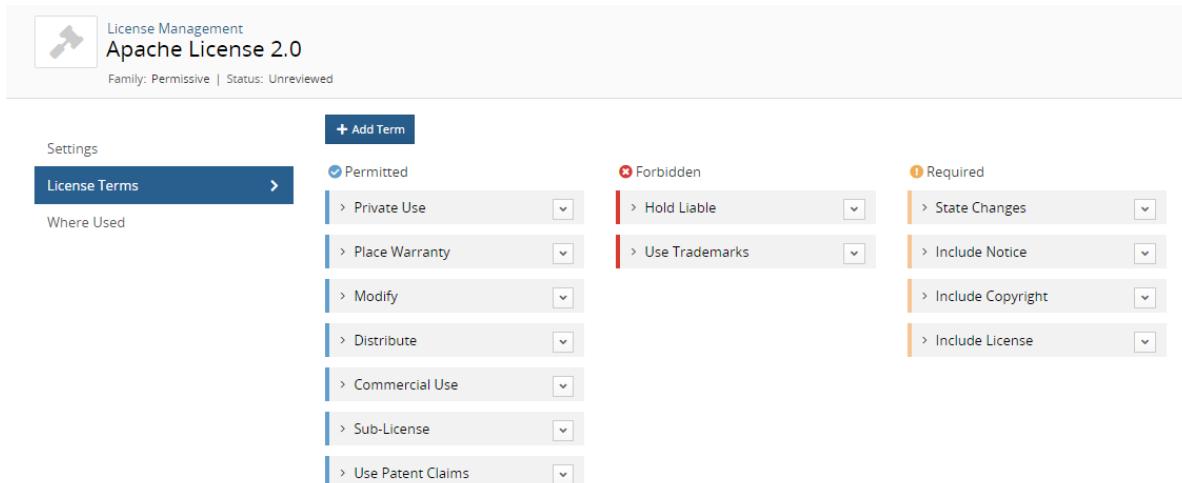
■ Using the **License Terms** tab which lists all license terms



The screenshot shows the Black Duck License Management interface. At the top, there's a navigation bar with tabs: 'Licenses' (selected), 'License Families', and 'License Terms'. Below the navigation bar is a search bar labeled 'Filter license terms...' and a 'Add Filter' button. The main area is a table with columns: 'Term', 'Source', 'Responsibility', 'Category', and 'Last Updated'. The table contains 15 rows of license terms, each with a dropdown arrow icon at the end of the last column.

Term	Source	Responsibility	Category	Last Updated
Modify	KnowledgeBase	Required	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never

■ Using the **License Terms** tab for an individual license



The screenshot shows the Apache License 2.0 settings page. At the top, it says 'Family: Permissive | Status: Unreviewed'. Below that is a 'Settings' section with a 'License Terms' tab selected. The 'License Terms' tab has a sub-section 'Where Used' with a list of terms: Private Use, Place Warranty, Modify, Distribute, Commercial Use, Sub-License, and Use Patent Claims. To the right of this are three groups of terms: 'Permitted' (Private Use, Place Warranty, Modify, Distribute, Commercial Use, Sub-License, Use Patent Claims), 'Forbidden' (Hold Liable, Use Trademarks), and 'Required' (State Changes, Include Notice, Include Copyright, Include License).

⚙ To restore a KnowledgeBase license term when viewing all terms

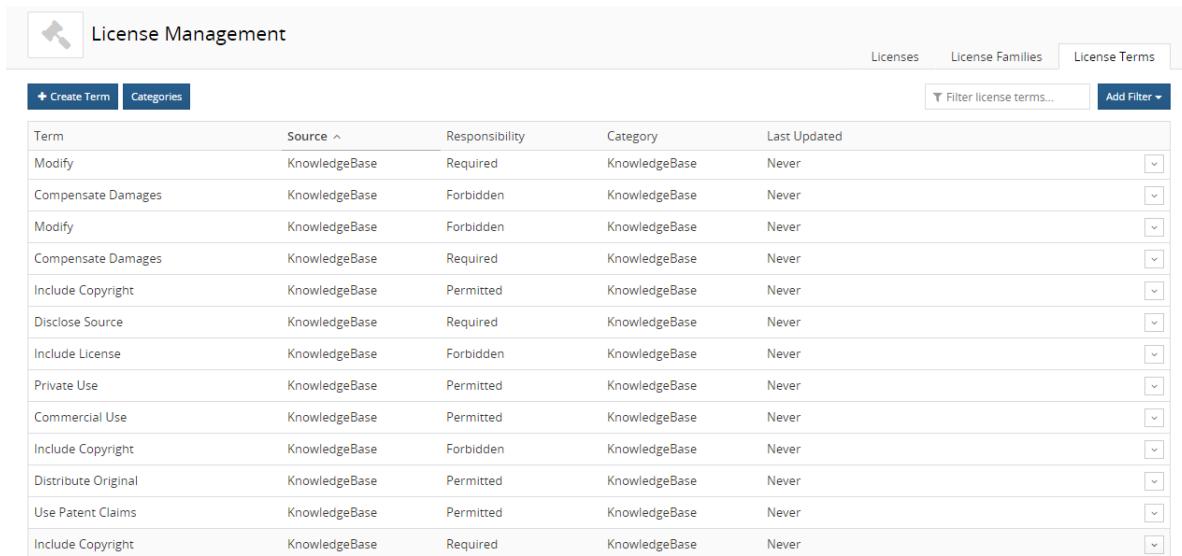
1. Log in to Black Duck with the License Manager [role](#).



2. Click the expanding menu icon () and select **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.



The screenshot shows the 'License Management' interface with the 'License Terms' tab selected. The table displays various license terms with columns for Term, Source, Responsibility, Category, and Last Updated. Each row has a dropdown arrow icon at the end of the last column.

Term	Source	Responsibility	Category	Last Updated	
Modify	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Compensate Damages	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Modify	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Compensate Damages	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Disclose Source	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include License	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Private Use	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Commercial Use	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Distribute Original	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Use Patent Claims	KnowledgeBase	Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Copyright	KnowledgeBase	Required	KnowledgeBase	Never	<input type="button" value="▼"/>

3. Click  in the row of the KnowledgeBase license term and select **License Association**.

The License Association dialog box appears showing all licenses that have this license terms associated to it.

License Association

Term	Source	Responsibility	Category
Private Use	KnowledgeBase	Permitted	KnowledgeBase

> Description

Select Licenses \*

Add

Require Fulfilment | Remove Fulfilment Requirement

License	Fulfillment Required
MIT License	-
Artistic License 2.0	-
Apache License 2.0	-
Eclipse Public License 1.0	-

Displaying 1-4 of 4

Close

The screenshot shows the 'License Association' dialog box. At the top, there are four columns: 'Term' (Private Use), 'Source' (KnowledgeBase), 'Responsibility' (Permitted), and 'Category' (KnowledgeBase). Below this is a 'Description' section with a link 'Select Licenses \*'. An 'Add' button is located to the right of the description. Under 'Description', there are two buttons: 'Require Fulfilment' and 'Remove Fulfilment Requirement'. The main area contains a table with four rows, each representing a license: MIT License, Artistic License 2.0, Apache License 2.0, and Eclipse Public License 1.0. Each row has a checkbox next to it and a trash icon to its right. To the right of the table, there is a note 'Fulfillment Required' and a 'Restore' button with a circular arrow icon. At the bottom of the dialog, it says 'Displaying 1-4 of 4' and has a 'Close' button.

- Click **Restore** in the row of the license(s) you want to restore.

The license term is enabled for this license.

License Association

Term	Source	Responsibility	Category
Private Use	KnowledgeBase	Permitted	KnowledgeBase

› Description

Select Licenses \*

Add

Require Fulfillment | Remove Fulfillment Requirement

License	Fulfillment Required
MIT License	-
Artistic License 2.0	-
Apache License 2.0	-
Eclipse Public License 1.0	-

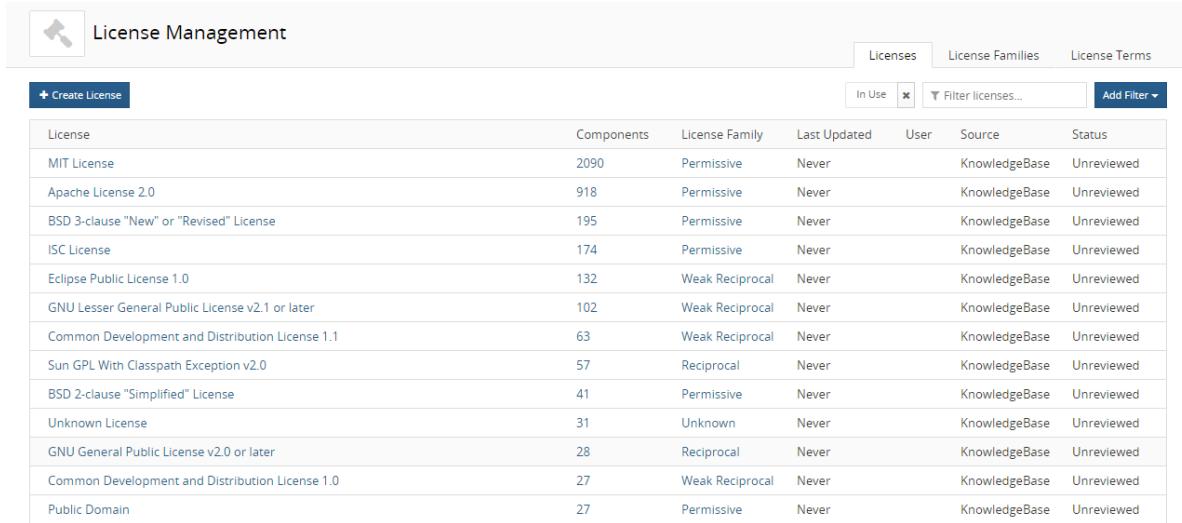
Displaying 1-4 of 4

Close

 To restore a KnowledgeBase license term when viewing a license

1. Log in to Black Duck with the License Manager [role](#).
2. Click the expanding menu icon () and select **License Management**.

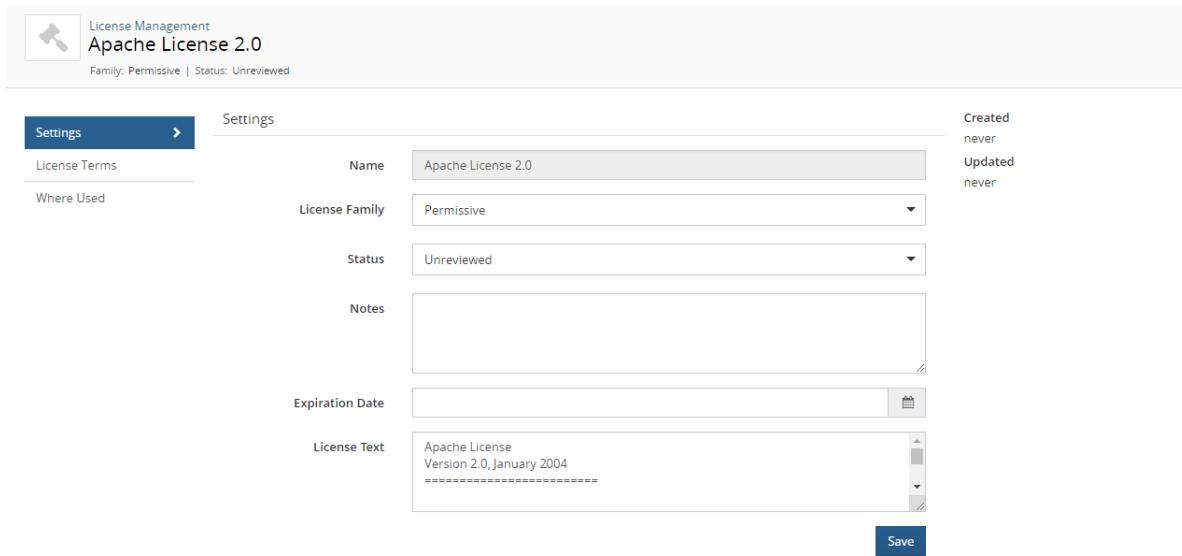
The License Management page appears.



The screenshot shows a table titled "License Management" with a "Create License" button. The columns are: License, Components, License Family, Last Updated, User, Source, and Status. The data includes various open-source licenses like MIT, Apache, BSD, and GPL, along with their respective counts and details.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

3. In the **Licenses** tab, select the license name to display the *License Name Settings* tab.



The screenshot shows the "Settings" tab for the Apache License 2.0. It includes fields for Name (Apache License 2.0), License Family (Permissive), Status (Unreviewed), Notes (empty), Expiration Date (empty), and License Text (Apache License Version 2.0, January 2004). A sidebar on the right shows creation and update details: Created never, Updated never.

Settings	
License Terms	Name: Apache License 2.0
Where Used	License Family: Permissive
	Status: Unreviewed
Notes	(Empty text area)
Expiration Date	(Empty date input field)
License Text	Apache License Version 2.0, January 2004 =====

Save

4. Select the **License Terms** tab to view the terms associated with this tab.

The screenshot shows the Apache License 2.0 settings page. The 'License Terms' tab is active. The interface is divided into three main sections: 'Permitted' (blue background), 'Forbidden' (red background), and 'Required' (orange background). Each section contains a list of terms with dropdown menus. A 'Where Used' section is located at the bottom left.

5. Click next to the KnowledgeBase license term you wish to activate and select **Restore**.

The **License Terms** tab displays the terms for this license with the term restored.

The screenshot shows the Apache License 2.0 settings page with the 'Where Used' section expanded. The 'License Terms' tab is active, displaying the same list of terms as the previous screenshot. The 'Where Used' section provides detailed information about where each term is applied across different components and BOMs.

## Editing a KnowledgeBase license

[KnowledgeBase](#) licenses can be edited by users with the License Manager role and by users with the BOM Manager, Super User, or Project Manager role:

- License Managers can make *global* edits to KnowledgeBase licenses. The License Manager can edit the license family, license text, and other license settings. License Managers can also edit the [license terms](#). The license name *cannot* be changed.

These edits are propagated to BOMs with components using the KnowledgeBase license.

- BOM Managers, Super Users, and Project Managers can only make *local* edits to the license text of a

KnowledgeBase license used in a BOM.

These edits only apply to the version of the KnowledgeBase license used in the BOM.

When the License Manager edits a KnowledgeBase license:

- Edits to the license family and license terms are always propagated to the KnowledgeBase licenses used in BOMs.
- Edits to the license text *may or may not* be propagated to the KnowledgeBase licenses used in BOMs:
  - If the BOM Manager/Super User/Project Manager *edited the license text*, the edits made by the License Manager *are not* propagated to the version of the KnowledgeBase license used in the BOM.
  - If the BOM Manager/Super User/Project Manager *did not edit* the license text, the edits made by the License Manager *are* propagated to the KnowledgeBase license used in the BOM.

**Note:** KnowledgeBase updates may modify existing KnowledgeBase licenses. However, if a KnowledgeBase license has been edited by a License Manager or BOM Manager, then modifications to a KnowledgeBase license due to KnowledgeBase updates are not propagated globally (if the License Manager has edited this license) or to the edited local version (if the BOM Manager has modified this license).

1. Log in to Black Duck with the License Manager role.



2. Click the expanding menu icon ( ) and select License Management.

The License Management page appears.

A screenshot of the Black Duck License Management interface. The top navigation bar includes 'License Management' with a wrench icon, 'Licenses' (selected), 'License Families', and 'License Terms'. Below the navigation is a search/filter bar with 'In Use' checked, a 'Filter licenses...' dropdown, and an 'Add Filter' button. A 'Create License' button is also present. The main area is a table with columns: License, Components, License Family, Last Updated, User, Source, and Status. The table lists various open source licenses such as MIT License, Apache License 2.0, BSD 3-clause "New" or "Revised" License, ISC License, Eclipse Public License 1.0, GNU Lesser General Public License v2.1 or later, Common Development and Distribution License 1.1, Sun GPL With Classpath Exception v2.0, BSD 2-clause "Simplified" License, Unknown License, GNU General Public License v2.0 or later, Common Development and Distribution License 1.0, and Public Domain. Each row shows the count of components affected, the license family (e.g., Permissive, Weak Reciprocal, Reciprocal), the last update date, the user who last updated it, the source (KnowledgeBase), and the status (Unreviewed).

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never	KnowledgeBase	Unreviewed	
Apache License 2.0	918	Permissive	Never	KnowledgeBase	Unreviewed	
BSD 3-clause "New" or "Revised" License	195	Permissive	Never	KnowledgeBase	Unreviewed	
ISC License	174	Permissive	Never	KnowledgeBase	Unreviewed	
Eclipse Public License 1.0	132	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never	KnowledgeBase	Unreviewed	
BSD 2-clause "Simplified" License	41	Permissive	Never	KnowledgeBase	Unreviewed	
Unknown License	31	Unknown	Never	KnowledgeBase	Unreviewed	
GNU General Public License v2.0 or later	28	Reciprocal	Never	KnowledgeBase	Unreviewed	
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never	KnowledgeBase	Unreviewed	
Public Domain	27	Permissive	Never	KnowledgeBase	Unreviewed	

3. Select the KnowledgeBase license name to display the *License Name Settings* tab.

The screenshot shows the 'Apache License 2.0' settings page in the License Management section. The 'Name' field is set to 'Apache License 2.0'. The 'License Family' is 'Permissive'. The 'Status' is 'Unreviewed'. The 'Notes' field is empty. The 'Expiration Date' field has a calendar icon. The 'License Text' field contains the Apache License text, starting with 'Apache License Version 2.0, January 2004'. On the right side, there are 'Created' and 'Updated' status indicators showing 'never'. A 'Save' button is located at the bottom right.

4. Modify the information:

- **Name:** License name. Note that this field is read-only.
- **License Family:** Use the drop-down selector to choose the license family.
- **Status:** Use the drop-down selector to choose the license status.
- **Notes:** You can type any text in this field. Use this for additional information or helpful notes.
- **Expiration Date:** Use the calendar tool to set the expiration date.
- **License Text:** The actual license as found in the component.

5. Click **Save**.

In the License Management page, the source for this license changes to **Modified KnowledgeBase** with the username of the user who edited this license listed in the **User** column and the time the license was modified listed in the **Last Updated** column.

KnowledgeBase licenses can be [restored](#) to their original values.

## Restoring the original text and family of a KnowledgeBase license

If a user with the License Manager role has modified the text or license family of a KnowledgeBase license, they can restore that license to its original values, as defined by the Black Duck KnowledgeBase.

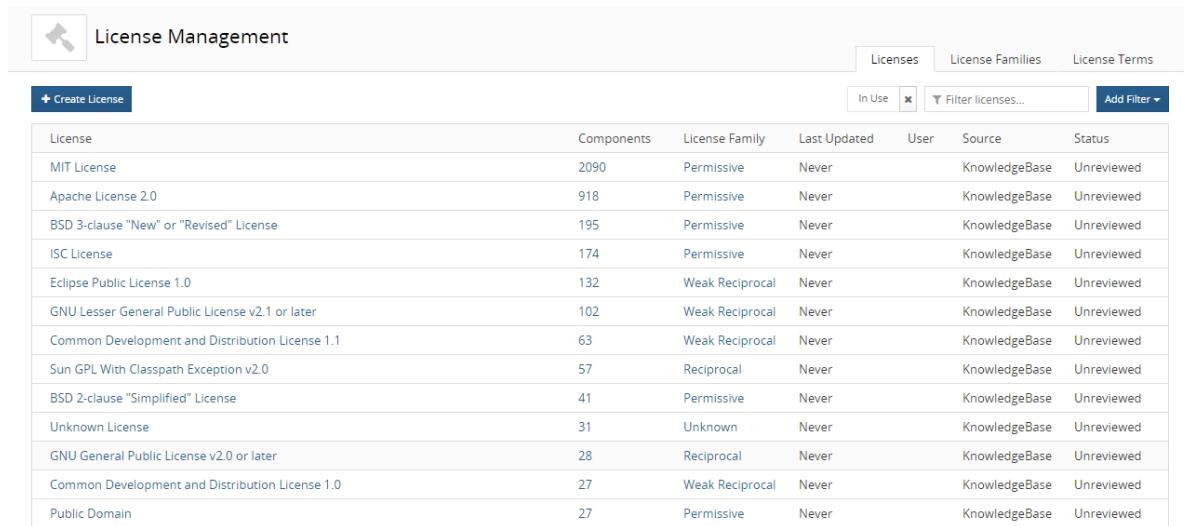
### To restore a KnowledgeBase license

1. Log in to Black Duck with the License Manager role.



2. Click the expanding menu icon ( ) and select License Management.

The License Management page appears.



The screenshot shows a table titled "License Management" with a "Create License" button. The table has columns for License, Components, License Family, Last Updated, User, Source, and Status. It lists various open source licenses like MIT License, Apache License 2.0, BSD 3-clause "New" or "Revised" License, ISC License, Eclipse Public License 1.0, etc., along with their respective details.

License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

3. Do one of the following:

- Click  and select **Restore** in the row of the KnowledgeBase license that you want to restore to display the Restore KnowledgeBase License dialog box.
- Select the KnowledgeBase license name to display the *License Name Settings* tab. In the **Restore KnowledgeBase License** section, click **Restore original**.

4. Click **Restore** in the Restore KnowledgeBase License dialog box.

In the License Management page, the source for this license reverts to **KnowledgeBase**.

- If the license family or text were the only changes made to the license (as defined on the **Settings** tab), the values in the **Last Updated** and **User** columns are removed.
- If additional changes were made (as defined on the **Settings** tab), the values in the **Last Updated** and **User** columns displays the date and username when the last of these changes occurred.

**Note:** This procedure does not restore the KnowledgeBase *license terms* to their original values. Click [here](#) for more information on restoring KnowledgeBase license terms.

## Managing attribution statements

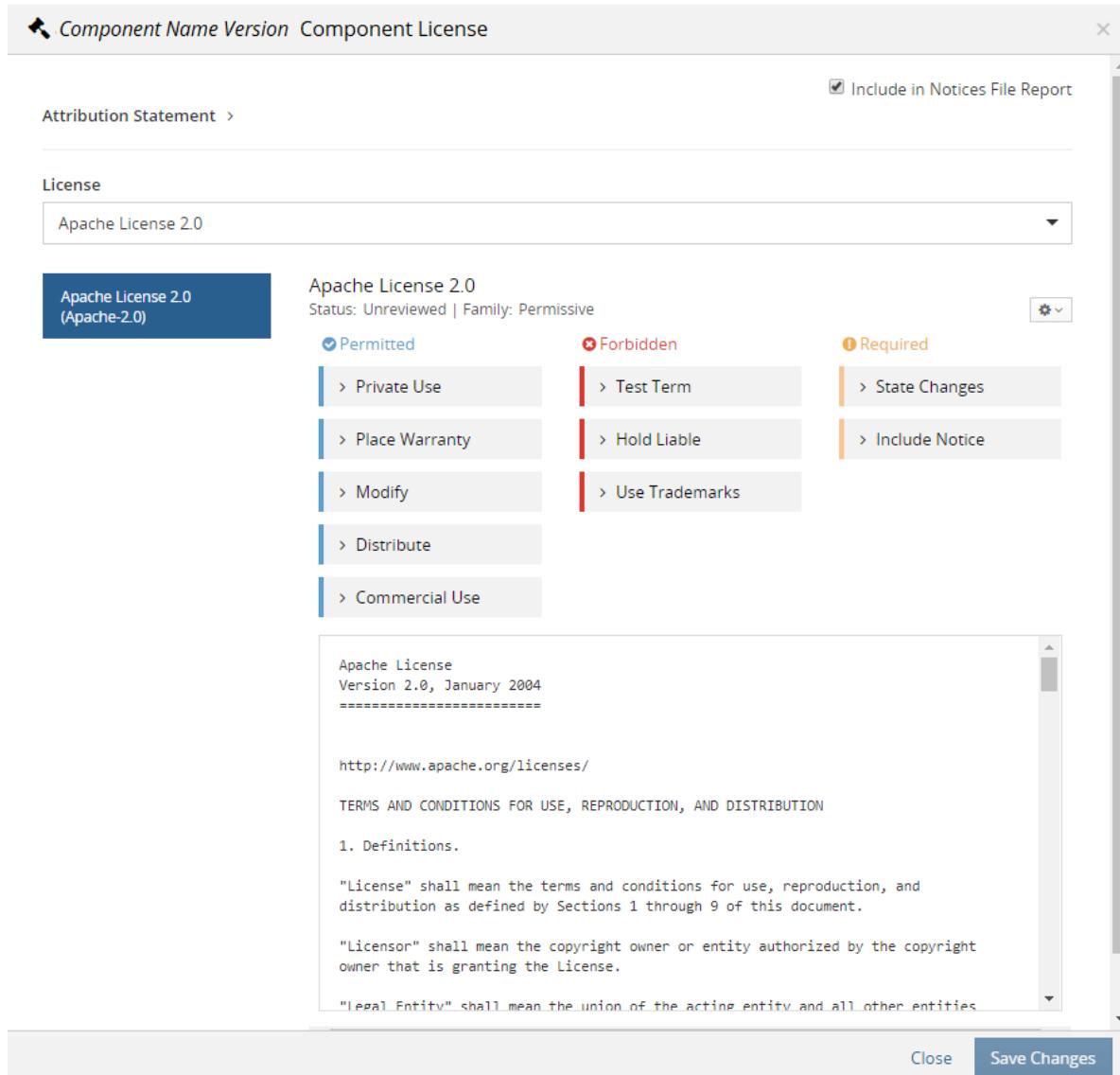
You may want to add an attribution statement to your Notices File report. An attribution statement is typically an acknowledgment to the copyright holder and is placed at the component version level.

**Note:** This feature is only available if you have the premium offering.

### To add an attribution statement

- Log in to Black Duck.
- Locate the project using the **Projects** tab on the Dashboard.

3. Select the name of the project to go to the *Project Name* page.
4. Select the version name to open the **Components** tab and view the BOM.
5. Select the license to open the *Component/Subproject Name Version Component License* dialog box.



6. Click > to open the **Attribution Statement** field and enter the text.

Delete the text in this field to remove an attribution statement.

7. Click **Save Changes**.
8. Click **Close**.

The attribution statement appears after the component name/version in the Components table in the [Notices File report](#). This example is from the HTML version of the report:

## Sample Project ▶ 1.0 ▶ Notices File

Phase: In Planning | Distribution: External

### Components

Component	License
Apache Struts 2.2.0	Apache License 2.0

Apache License 2.0: " Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution."

### Licenses

#### Apache License 2.0

Apache Struts 2.2.0

Apache License  
Version 2.0, January 2004  
=====

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including

# Chapter 13: Running a report

Black Duck provides the following reports:

- [Notices File](#)
- [Project Version](#)
- [Vulnerability Remediation](#)
- [Vulnerability Status](#)
- [Vulnerability Update](#)

These reports help you:

- View the list of components and associated license text for a project version.
- Identify the security vulnerabilities associated with all your projects.
- Track the remediation status of vulnerabilities in all your projects.
- Export and share the information of a single project version.

**Note:** Reports include subproject information *if you have permission to the [subproject](#).*

For Project Version reports:

- The archive file name is <ProjectName-ProjectVersion>\_YYYY-MM-DD-HHMMSS.zip (time stamp in UTC)
- The directory and filename are <ProjectName-ProjectVersion>\_YYYY-MM-DD-HHMMSS/<fileName>\_YYYY-MM-DD-HHMMSS.csv (same time stamps as archive file name)
- The following characters <> \ / | : \* ? + " in the project or version name are replaced with underscores (\_).

For global vulnerability reports, where <ReportType> is either remediation, status, or update:

- The archive filename is vulnerability-<ReportType>-report\_YYYY-MM-DD-HHMMSS.zip (time stamp in UTC)
- The directory and filename are: vulnerability-<ReportType>-report\_YYYY-MM-DD-HHMMSS.csv (same time stamps as archive)

## Notices File report

The Notices File report provides a list of OSS components, versions, and the associated license text. You can use this report to create an attribution report for your project release or to share BOM and license information.

This report is available as a text file or in HTML format. Each format provides the following information:

- Header information. Lists the project name, version, phase, and distribution.
- Components. Lists all components, component versions, subprojects, subproject versions, and associated licenses.  
You can [exclude a component or subproject](#) or add an [attribution statement](#) if you have the premium offering,
- Licenses. Provides the license text for all licenses listed in the **Components** section.  
You can [edit the license text](#) shown here if you have the premium offering.

The following is an example of a portion of the HTML version of the report:

## Sample Project ▶ 1.0 ▶ Notices File

Phase: In Planning | Distribution: External

### Components

Component	License
Apache Struts 2.2.0	Apache License 2.0

Apache License 2.0: "Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution."

### Licenses

Apache License 2.0

Apache Struts 2.2.0

Apache License  
Version 2.0, January 2004  
=====

<http://www.apache.org/licenses/>

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including

#### ⚙ To run a Notices File report

1. Locate the project using the **Projects** tab on the Dashboard.
2. Select the name of the project to go to the *Project Name* page.
3. Select the version of the project for which you want to run the report.
4. Select the **Reports** tab.

5. Click **+ Create Notices File** and select the format for the report:
  - Text. This is the default format for the report.
  - HTML.
6. Click **Create** to run the report.
7. A link that includes the project, version name, and date appears when the report completes. Any user who is a member of the project can access the link.
  - If you selected the text format, download the report and extract the zip file locally.
  - If you selected the HTML format, select the link to open the report in a new tab.

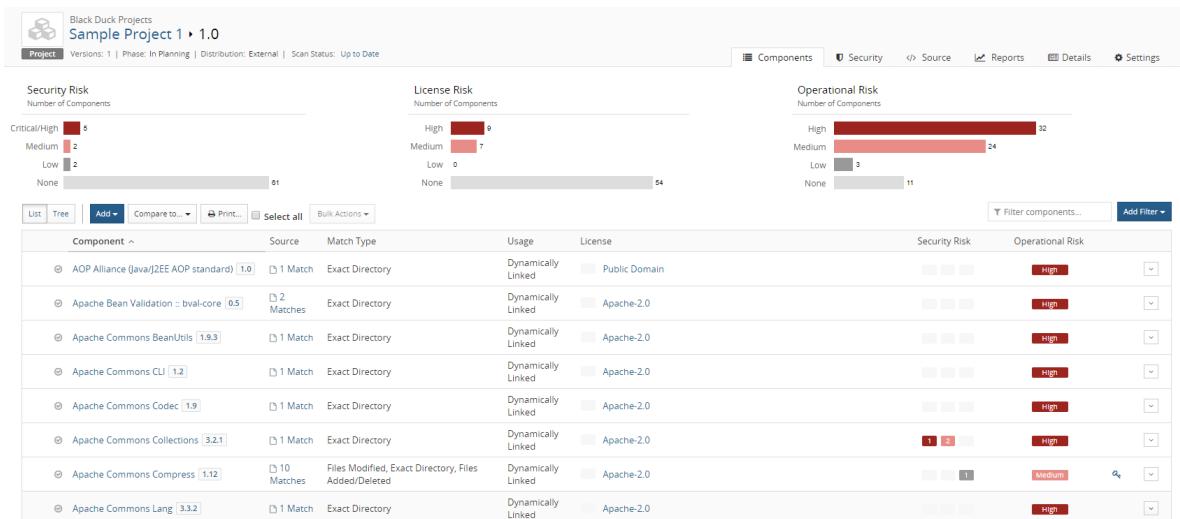
## Excluding a component or subproject from the Notices File report

By default, all components and subprojects are included in the [Notices File report](#).

If you have the premium offering, you can exclude a component or subproject from this report.

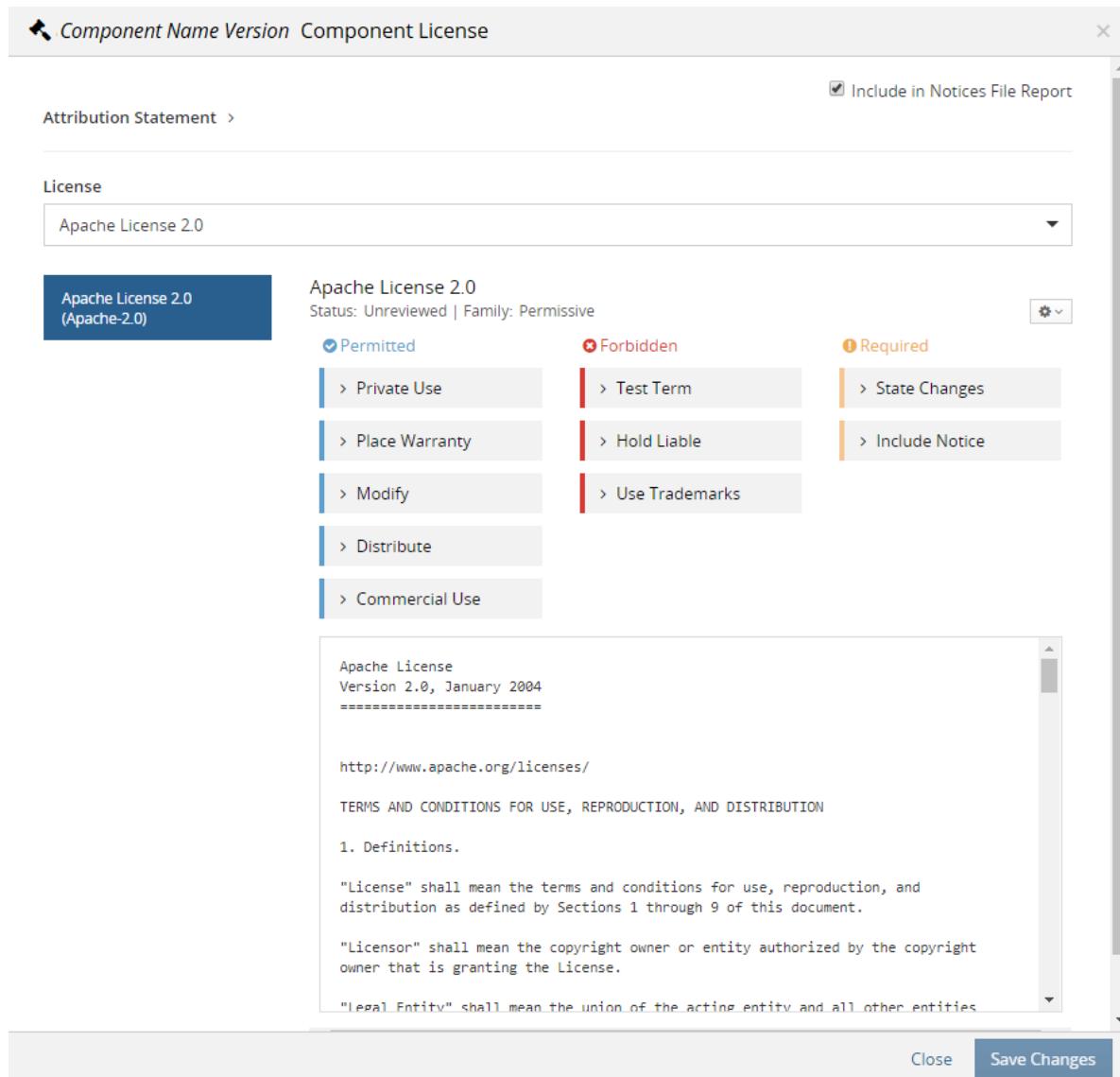
### To exclude a component in the Notices File report

1. Log in to Black Duck.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the version name to open the **Components** tab and view the BOM.



Component	Source	Match Type	Usage	License	Security Risk	Operational Risk
AOP Alliance (Java/J2EE AOP standard) 1.0	1 Match	Exact Directory	Dynamically Linked	Public Domain	<span style="background-color: #cccccc; color: black;">Low</span>	<span style="background-color: #cccccc; color: black;">Low</span>
Apache Bean Validation :: bval-core 0.5	2 Matches	Exact Directory	Dynamically Linked	Apache-2.0	<span style="background-color: #cccccc; color: black;">Low</span>	<span style="background-color: #cccccc; color: black;">Low</span>
Apache Commons BeanUtils 1.9.3	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	<span style="background-color: #cccccc; color: black;">Low</span>	<span style="background-color: #cccccc; color: black;">Low</span>
Apache Commons CLI 1.2	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	<span style="background-color: #cccccc; color: black;">Low</span>	<span style="background-color: #cccccc; color: black;">Low</span>
Apache Commons Codec 1.9	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	<span style="background-color: #cccccc; color: black;">Low</span>	<span style="background-color: #cccccc; color: black;">Low</span>
Apache Commons Collections 3.2.1	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	<span style="background-color: #cccccc; color: black;">Low</span>	<span style="background-color: #cccccc; color: black;">Low</span>
Apache Commons Compress 1.12	10 Matches	Files Modified, Exact Directory, Files Added/Deleted	Dynamically Linked	Apache-2.0	<span style="background-color: #cccccc; color: black;">Low</span>	<span style="background-color: #cccccc; color: black;">Low</span>
Apache Commons Lang 3.3.2	1 Match	Exact Directory	Dynamically Linked	Apache-2.0	<span style="background-color: #cccccc; color: black;">Low</span>	<span style="background-color: #cccccc; color: black;">Low</span>

5. Select the existing license from the **License** column to open the *Component/Subproject Name Version Component License* dialog box.



6. In the License window, clear **Include in Notices File Report** to exclude the component or subproject from the report.  
Select **Include in Notices File Report** to include the component or subproject in the report.
7. Click **Save Changes**
8. Click **Close**.

## Project Version report

The Project Version report lets you export and share the content of a single project version.

Depending on the categories you select, running a project version report creates these comma-separated files:

- `scans_date_#.csv` lists the mapped scans.
- `components_date_#.csv` lists each component in the project version, including the respective licensing, usage, match type, operation risk information, policy violation information, and review status.
- `source_date_#.csv` lists the individual files and dependencies associated with each component, including match type, usage information, and policy violation information.
- `security_date_#.csv` lists the security risk associated with each component, including the vulnerability ID and description, vulnerability scores, and remediation information.
- `version_date_#.csv` lists the name and details of the project version, including the release date, phase, method of release, and policy violation information.
- `crypto_date_#.csv` lists the cryptography information for each component in the project version, including the algorithm ID, algorithm name, key length type, and key length.
- `project_version_custom_fields_date_#.csv` lists the project version custom field labels and the values selected for this project version.
- `bom_component_custom_fields_date_#.csv` lists the same information as the `components_date_#.csv`, but also includes BOM component, component, and component version custom field labels and the values selected for this project version.
- `license_term_fulfillment_date_#.csv` lists the license terms and fulfillment status for this project version.

#### To run a Project Version report

1. Locate the project using the **Projects** tab on the Dashboard.
2. Select the name of the project to go to the *Project Name* page.
3. Select the version of the project for which you want to run the report.
4. Select the **Reports** tab.
5. Click **Create > Create Report** and select the categories you would like to include in the report:
  - Version Details
  - Scans
  - Components
  - Vulnerabilities
  - Source
  - Cryptography
  - Project Version Additional Fields
  - Component Additional Fields
  - License Terms
6. Click **Create** to run the report.

A link that includes the project and version name appears when the report completes. Any user who is a member of the project can access the link.
7. Download the report and extract the zip locally.

## Vulnerability Remediation report

Based on a specific date range, the Vulnerability Remediation report lists all the vulnerabilities that match a specific [remediation status](#).

For example, you can use this report to identify all of the vulnerabilities that require remediation or all the vulnerabilities that have been mitigated and ignored.

### ⚙️ To run a Vulnerability Remediation report



1. Log in to Black Duck and click the expanding menu icon ( ).
2. Click **Reports**.
3. Click **+ Create new report**, and from the **Report Type** list, select **Vulnerability Remediation Report**.
4. Select either **HTML** or **CSV** as the report format.

**Tip:** Use the CSV option when your data becomes too large to render and view in the browser.

5. Select the dates for this report. The date represents the day when the vulnerability was published. By default, the end date is the current date.
6. Optionally, select one or more remediation statuses.
7. Click **Confirm** to run the report.

A link that includes the report name and date appears when the report completes. Any user who is a member of the project can access the link.

8. Select the link to view the report.

If you selected CSV as the report format, download the report and extract the zip file.

**Note:** You can use the native print functionality of your web browser to print the HTML version of the report.

## Vulnerability Status report

The Vulnerability Status report includes all the vulnerabilities that are associated with the projects and project versions to which you have access.

For example, you can use this report to identify which projects are secure and which projects and project versions contain security risks.

### ⚙️ To run a Vulnerability Status report



1. Log in to Black Duck and click the expanding menu icon ( ).
2. Click **Reports**.

3. Click **+ Create new report**, and from the **Report Type** list, select **Vulnerability Status Report**.
4. Select either **HTML** or **CSV** as the report format.

**Tip:** Use the CSV option when your data becomes too large to render and view in the browser.

5. Click **Confirm** to run the report.

A link that includes the report name and date appears when the report completes. Any user who is a member of the project can access the link.

6. Select the link to view the report.

If you selected CSV as the report format, download the report and extract the zip file.

**Note:** You can use the native print functionality of your web browser to print the HTML version of the report.

## Vulnerability Update report

Based on a specific date range, the Vulnerability Update report includes the following information for projects to which you have access:

- New vulnerabilities.

For example, you can use this report to identify new vulnerabilities after code or a Docker image has been rescanned.

- Updates to the [remediation status](#) of existing vulnerabilities.

For example, you can use this report to track the progress of a remediation effort.

- Updates to any of the data that is associated with vulnerabilities.

For example, you can use this report to identify if the risk scores associated with existing vulnerabilities have changed.

### To run a Vulnerability Update report



1. Log in to Black Duck and click the expanding menu icon ( ).
2. Click **Reports**.
3. Click **+ Create new report**, and from the **Report Type** list, select **Vulnerability Update Report**.
4. Select either **HTML** or **CSV** as the report format.

**Tip:** Use the CSV option when your data becomes too large to render and view in the browser.

5. Select the dates for this report. The date represents the day on which the vulnerability was added to a project version or the information associated with the vulnerability was updated. By default, the end date

is the current date.

6. Click **Confirm** to run the report.

A link that includes the report name and date appears when the report completes. Any user who is a member of the project can access the link.

7. Select the link to view the report.

If you selected CSV as the report format, download the report and extract the zip file.

**Note:** You can use the native print functionality of your web browser to print the HTML version of the report.

## Deleting reports

### To delete a report

1. Click  in the row of the report you wish to delete.
2. Click **Delete** in the confirmation dialog box.

Note the following:

- Reports older than 30 days are automatically deleted.
- The system retains up to 20 reports, per user, across all project versions. If a user creates more than 20 reports, the system automatically deletes the oldest reports and retains the 20 newest reports.

# Chapter 14: Managing Black Duck user accounts

There are two ways to manage user accounts in Black Duck:

1. Managing user accounts manually. A user with the [Super User role](#) can:

- [Add a new user account](#)
- [Inactivate a user account](#)
- [Change user account information](#)
- [Change a user's password](#)
- [View a user's groups](#)
- [Manage user roles](#)

2. [Enabling and configuring LDAP to manage user accounts.](#)

After you configure LDAP to manage user accounts for Black Duck, new user accounts will be automatically created the first-time users attempt to log in. Your LDAP server will then manage passwords and account details for those user accounts in Black Duck.

**Tip:** If you are using LDAP to manage most of your user accounts in Black Duck, you can still manually manage those user accounts that do not also exist in your LDAP directory, such as a default system administrator account.

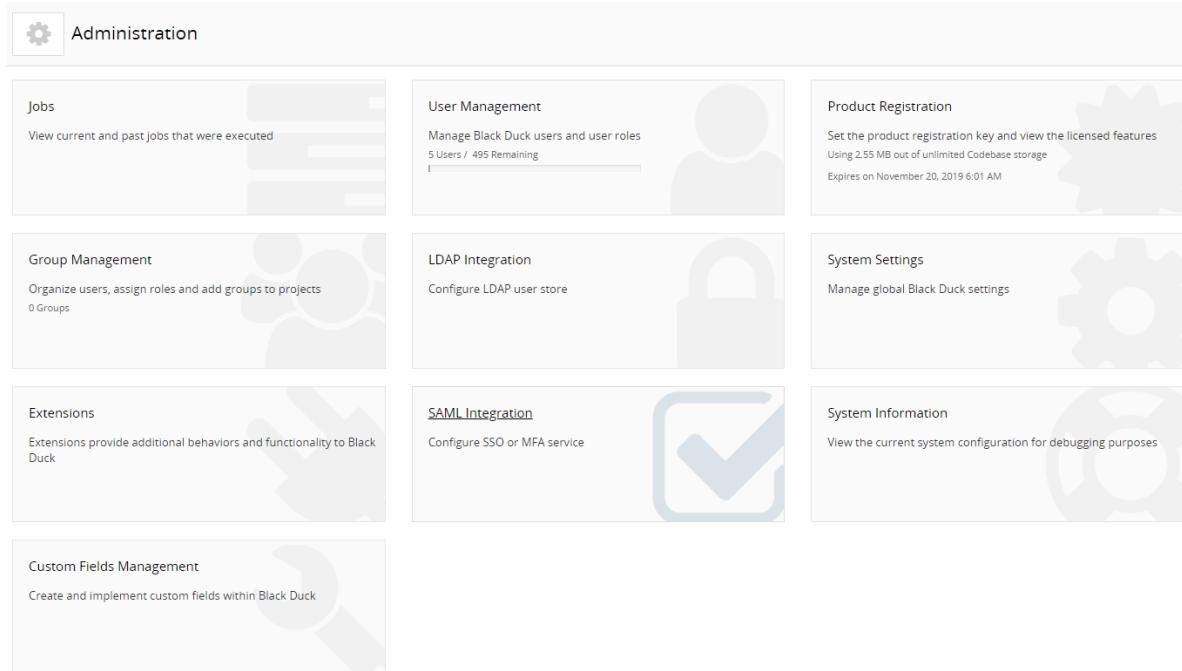
## Creating a user account

**Note:** If you have enabled LDAP, you should create users on your LDAP server instead of in Black Duck. Black Duck will authenticate user IDs against the LDAP server, and if the username and password are valid, will copy the user ID to Black Duck database.

### To create a user account

1. Log in to Black Duck.
2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.



3. Select **User Management** to display the User Management page.

The screenshot shows the 'User Management' page. It features a table with columns for Username, First Name, Last Name, Email, Roles, and Status. The table contains three rows of data. At the bottom right, it says 'Displaying 1-3 of 3'.

Username	First Name	Last Name	Email	Roles	Status
sysadmin	System	Administrator	noreply@blackducksoftware.com	Component Manager, Global Code Scanner, License Manager, Policy Manager, Project Creator, Super User, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23			Active

4. Click **+ Create User**.
5. In the Create a New User dialog box, type the basic information for the user.
  - Username
  - Email: This must be a valid email address. Black Duck validates this when you create the user account.
  - First Name
  - Last Name
  - Password
  - Confirm password: This must match the password you entered. Black Duck validates this when you create the user account.
6. Select whether this user is active or inactive. Clearing this check box inactivates this user.

## 7. Click **Create**.

Black Duck creates the user account with the password you specified.

After creating a user, [assign roles to this user](#).

## Disabling a user

**Note:** If you have enabled LDAP, you should manage user records in the LDAP server. If you delete a record in Black Duck and do not delete the user from the LDAP server, the next time the user attempts to log in to Black Duck, their user record will be recreated with data from the LDAP server.

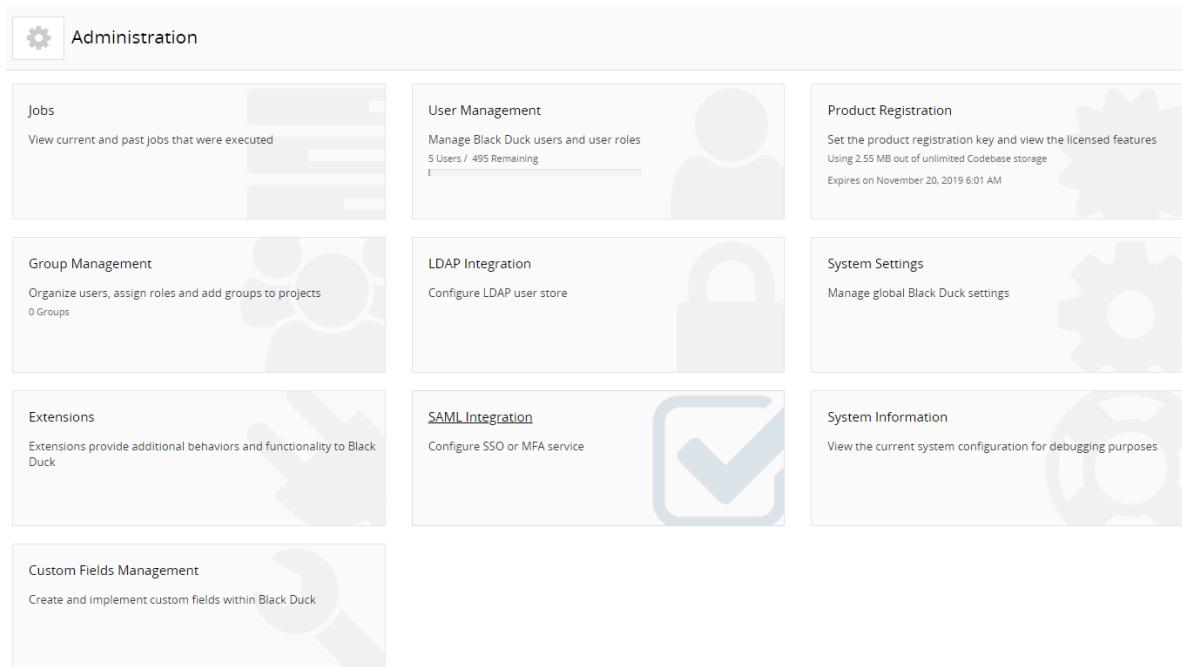
### To disable a user account

1. Log in to Black Duck.



2. Click the expanding menu icon ( ) and select **Administration**.

The Administration page appears.



A screenshot of the Black Duck Administration page. The page has a header with a gear icon and the word "Administration". Below the header is a grid of nine cards, each representing a different administrative function:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. It shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. It shows Using 2.55 MB out of unlimited Codebase storage, Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. It shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck.

3. Select **User Management** to display the User Management page.

The screenshot shows the 'User Management' page in the Black Duck Administration interface. At the top, there is a header with a user icon, the text 'Administration' and 'User Management', a 'Create User' button, and filter options for 'User Status' (set to 'Active'), 'Filter user list...', and 'Add Filter'. Below this is a table listing three users:

Username	First Name	Last Name	Email	Roles	Status
sysadmin	System	Administrator	noreply@blackducksoftware.com	Component Manager, Global Code Scanner, License Manager, Policy Manager, Project Creator, Super User, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23			Active

At the bottom right of the table area, it says 'Displaying 1-3 of 3'.

4. Find the user you want to deactivate:

- Filter the users that appear on the page.
- Sort the list of users by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.
- Use the pagination bar at the bottom of the list to go to the appropriate page if there are more users than are listed on this page.

5. Select the user to display the *Username* page.

The screenshot shows the 'Administration / User Management' section for a 'Sample User'. The 'User Details' tab is active, displaying the following information:

Username *	SampleUser
First Name *	Sample
Last Name *	User
Email	[Empty]

A checked checkbox labeled 'Active user' is present. A 'Save' button is located at the bottom right.

The 'Overall Roles' section lists several roles with descriptions:

- Component Manager**  
This role can create, update and delete custom components
- Global Code Scanner**  
This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project
- Global Project Viewer**  
This role has read only access to all projects
- License Manager**  
Ability to create/modify/delete licenses
- Policy Manager**  
Ability to create/modify/delete policies
- Project Creator**  
This role can create projects/versions and edit their settings.
- Super User**  
This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**  
This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application

The 'User Groups' section contains a '+ Add group' button and displays the message 'No Results Found'.

The 'Project Access' section contains a '+ Add project' button and displays the message 'No Results Found'.

6. Clear the **Active user** check box in the **User Details** section and click **Save**.

## Viewing a user's groups

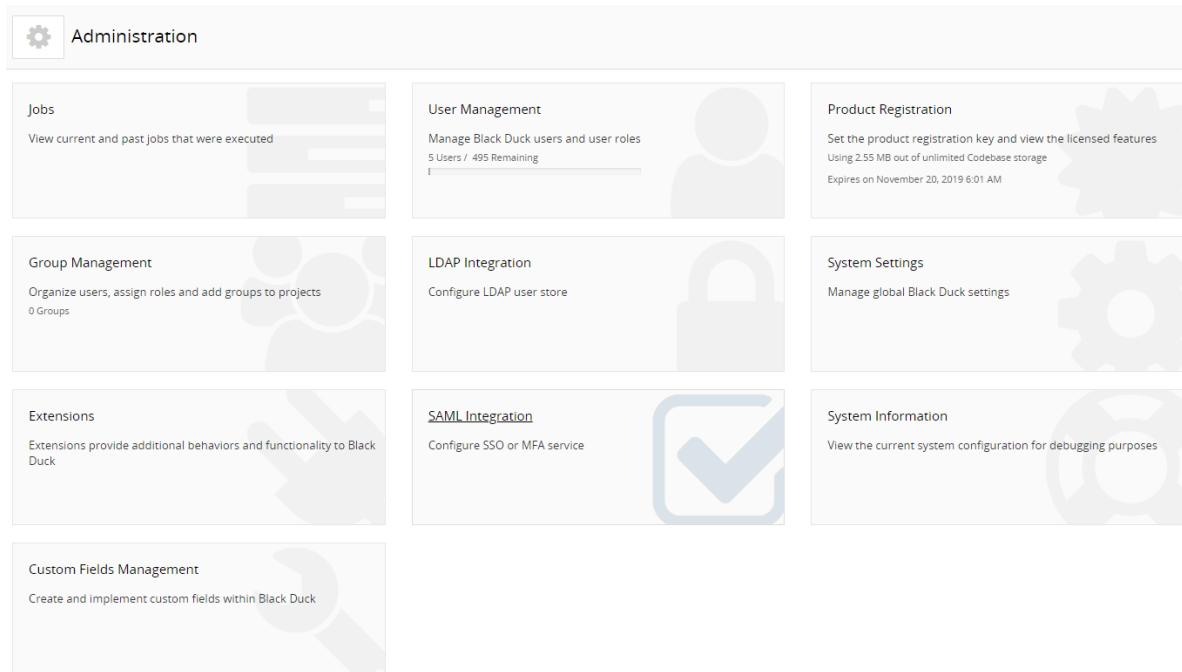
You can view the groups a user belongs to, and the source, status, and roles associated with that group.

### To view a user's groups

1. Log in to Black Duck.

2. Click the expanding menu icon () and select **Administration**.

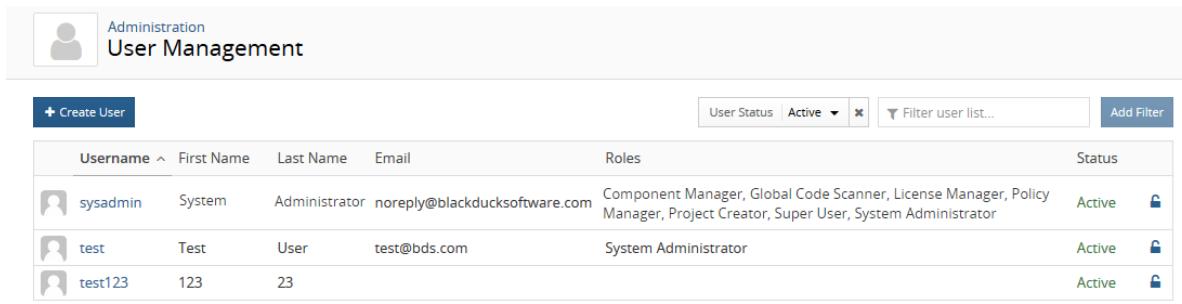
The Administration page appears.



The screenshot shows the Black Duck Administration interface. At the top left is a gear icon labeled "Administration". Below it are several cards representing different administrative functions:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. It shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. It shows Using 2.55 MB out of unlimited Codebase storage, Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. Shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck.

3. Select **User Management** to display the User Management page.



The screenshot shows the User Management page. At the top left is a user icon labeled "User Management". Below it is a "Create User" button. At the top right are filters for "User Status" (set to "Active"), "Filter user list...", and "Add Filter". The main area is a table listing users:

Username	First Name	Last Name	Email	Roles	Status
sysadmin	System	Administrator	noreply@blackducksoftware.com	Component Manager, Global Code Scanner, License Manager, Policy Manager, Project Creator, Super User, System Administrator	Active 
test	Test	User	test@bds.com	System Administrator	Active 
test123	123	23			Active 

Displaying 1-3 of 3

4. Find the user you want to find:

- Select the **Display Inactive Users** check box to include inactive users. Clearing this check box hides all inactive users.
- Filter the users that appear on the page.
- Sort the list of users by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.
- Use the pagination bar at the bottom of the list to go to the appropriate page if there are more users than are listed on this page.

5. Select the user to display the *Username* page.

The screenshot shows the 'Administration / User Management' section for a 'Sample User'. The top navigation bar includes a user icon, the title 'Administration / User Management', and the specific user name 'Sample User'. A 'Reset Password for User' button is visible.

**User Details**

Username *	SampleUser
First Name *	Sample
Last Name *	User
Email	[Empty]

Active user

**Save**

**Overall Roles**

- Component Manager**  
This role can create, update and delete custom components
- Global Code Scanner**  
This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project
- Global Project Viewer**  
This role has read only access to all projects
- License Manager**  
Ability to create/modify/delete licenses
- Policy Manager**  
Ability to create/modify/delete policies
- Project Creator**  
This role can create projects/versions and edit their settings.
- Super User**  
This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**  
This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application

**User Groups**

**+ Add group**

No Results Found

**Project Access**

**+ Add project**

No Results Found

6. The **User Groups** section lists the groups to which this user belongs. In this section, you can also:
- Select a group name to view the *Group Name* page from which you can [manage group information](#), [group roles](#) and [group membership](#).
  - [Add this user to one or more groups](#).
  - [Remove this user from a group](#) by clicking  in the row of the group. Select **Remove** in the Remove User from Group dialog box to confirm.

[Users can view the groups that they belong to](#) by using the My Profile page.

## Viewing a user's projects

You can view the projects a user belongs to, and whether the user was added individually or as a member of a group.

### To view a user's projects

1. Log in to Black Duck.



2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.

The screenshot shows the Black Duck Administration page with the following sections:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. It shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. It shows Using 2.55 MB out of unlimited Codebase storage and Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. It shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service. It shows a checkmark icon.
- System Information**: View the current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck.

3. Select **User Management** to display the User Management page.

The screenshot shows the 'User Management' page in the Black Duck Administration interface. At the top, there's a header with the title 'User Management'. Below the header is a search bar with filters for 'User Status' set to 'Active' and a 'Filter user list...' button. There's also a 'Create User' button. The main area is a table listing three users:

Username	First Name	Last Name	Email	Roles	Status
sysadmin	System	Administrator	noreply@blackducksoftware.com	Component Manager, Global Code Scanner, License Manager, Policy Manager, Project Creator, Super User, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23			Active

At the bottom right of the table, it says 'Displaying 1-3 of 3'.

4. Find the user you want to find:
  - Filter the users that appear on the page.
  - Sort the list of users by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.
  - Use the pagination bar at the bottom of the list to go to the appropriate page if there are more users than are listed on this page.
5. Select the user to display the *Username* page.
6. The **Project Access** section lists the projects to which this user belongs. For each project, it lists whether the user is a direct member (the user was added individually and not as part of a group), and the groups the user is a member of that have access to the project. You can:
  - Select a project name to view the *Project Name* page from which you can view project versions and [manage project details, members, and groups](#).
  - Add this user to projects: click **Add project**, enter the name of one or more projects, select the roles for this user for this project, and click **Add**.
  - Remove members that were directly added to a project: click **Remove** and then confirm removal of this user.

## Changing your Black Duck password

**Note:** If your system administrator has enabled LDAP on the Black Duck server, user account information and passwords are managed by LDAP. You cannot change your password in Black Duck.

If your Black Duck server does not use LDAP to manage user accounts, your username and initial password were created by your Black Duck administrator. You can change your password on your profile page.

**Tip:** If you forget your password, a user with the Super User role can change it for you.

### To change your password:

1. Log in to Black Duck.
2. Select your username or the user profile icon:



3. Select **My Profile**.
4. Type your current password in the **Current Password** field.
5. Type your new password in the **New Password** field.
6. Type the same new password in the **Confirm Password** field.
7. Click **Save**.

## Changing user account information

**Note:** If you have enabled LDAP, you should manage user account information on the LDAP server. Any changes you make to user account information in Black Duck will be overwritten the next time user information is synchronized with the data on the LDAP server.

⚙ To change user account information:

1. Log in to Black Duck.
2. Click the expanding menu icon ( ) and select **Administration**.

The Administration page appears.

The screenshot shows the Black Duck Administration page with the following sections:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. It shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. It shows Using 2.55 MB out of unlimited Codebase storage and Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. It shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service. It shows a checkmark icon.
- System Information**: View the current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck.

3. Select **User Management** to display the User Management page.

The screenshot shows the 'User Management' page under the 'Administration' section. At the top, there is a 'Create User' button and a search/filter bar with dropdowns for 'User Status' (set to 'Active') and 'Filter user list...'. Below the search bar is a 'Add Filter' button. The main area is a table listing three users:

Username	First Name	Last Name	Email	Roles	Status
sysadmin	System	Administrator	noreply@blackducksoftware.com	Component Manager, Global Code Scanner, License Manager, Policy Manager, Project Creator, Super User, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23			Active

At the bottom right of the table, it says 'Displaying 1-3 of 3'.

4. Find the user whose information you want to change:
  - Filter the users that appear on the page.
  - Sort the list of users by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.
  - Use the pagination bar at the bottom of the list to go to the appropriate page if there are more users than are listed on this page.
5. Select the username to open the *Username* page.

The screenshot shows the Black Duck User Management interface for managing user accounts. The top navigation bar includes links for Administration, User Management, and Sample User. A 'Reset Password for User' button is visible. The main section is titled 'User Details' and contains fields for Username (SampleUser), First Name (Sample), Last Name (User), and Email (empty). An 'Active user' checkbox is checked. A 'Save' button is located at the bottom right of this section. Below this is the 'Overall Roles' section, which lists various roles with descriptions: Component Manager, Global Code Scanner, Global Project Viewer, License Manager, Policy Manager, Project Creator, Super User, and System Administrator. The 'Super User' role is described as having access to all user and project data, including the ability to create, modify, and delete projects and users. The 'System Administrator' role is described as being for IT personnel who manage system settings like registration keys and LDAP. The 'User Groups' section shows a '+ Add group' button. The 'Project Access' section shows a '+ Add project' button. Both sections display the message 'No Results Found'.

6. Enter the new user information in the **User Details** section.

7. Click **Save**.

## Changing a user's password

**Note:** If you have enabled LDAP authentication, user account passwords are managed by LDAP. You will not be able to change passwords in Black Duck.

### To change a user's password

1. Log in to Black Duck.



2. Click the expanding menu icon ( ) and select **Administration**.

The Administration page appears.

The screenshot shows the Black Duck Administration page with the following sections:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. It shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. It shows Using 2.55 MB out of unlimited Codebase storage and Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. It shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck.

3. Select **User Management** to display the User Management page.

The screenshot shows the 'User Management' page in the Black Duck Administration interface. At the top, there's a header with a user icon and the title 'User Management'. Below the header is a toolbar with a 'Create User' button, a dropdown for 'User Status' set to 'Active', a search bar labeled 'Filter user list...', and a 'Add Filter' button. The main area is a table listing three users:

Username	First Name	Last Name	Email	Roles	Status
sysadmin	System	Administrator	noreply@blackducksoftware.com	Component Manager, Global Code Scanner, License Manager, Policy Manager, Project Creator, Super User, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23			Active

At the bottom right of the table, it says 'Displaying 1-3 of 3'.

4. Find the name of the user whose password you want to reset:
  - Filter the users that appear on the page.
  - Sort the list of users by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.
  - Use the pagination bar at the bottom of the list to go to the appropriate page if there are more users than are listed on this page.
5. Do one of the following:
  - Click the password reset ( lock icon) for that user.
  - Select the username to open the *Username* page and click **Reset Password for User**.
6. In the Reset Password for User dialog box, type the new password in the **Password** field.
7. Type the same password in the **Confirm Password** field.
8. Click **Save**.

## Understanding roles

Black Duck provides global and project roles which helps you control access and capabilities without impeding productivity. Roles define the tasks users can perform and the information users can view. Project-level roles provide the flexibility so that you can assign users individual roles per project - the roles only apply to the projects a user is assigned.

- You can assign roles to either [individual user accounts](#) or to [groups](#).
- If you assign a role to a group, the entire group membership inherits the role and its permissions.
- If you do not assign a role, users have read-only access to Black Duck.

For more information on the tasks that can be performed for each role, refer to the [Black Duck user role matrix](#).

## Global roles

The following global roles are available:

- Component Manager

The Component Manager is responsible for creating, editing, and/or deleting custom components.

This role is often assigned to a centralized group responsible for the management of custom components. In smaller organizations, this role can be given to subject matter experts (SMEs) or development managers.

- **Global Code Scanner**

The Global Code Scanner has access to all scans in Black Duck and can map or delete scans for any project within the system.

This role is often assigned to a user account used for continuous integration (CI) builds and sometimes, in smaller organizations, given to a release/build engineer who manages all builds for a company.

- **Global Project Viewer**

The Global Project Viewer can view *all* projects. Users with this role can view all BOMs, but cannot edit the BOM; they can only add or edit comments.

When you assign a user this role, they automatically have read-only access to all projects – you do not have to assign the users to the projects.

This role is often assigned to executives and users in the Legal department.

- **License Manager**

The License Manager is responsible for approving and/or rejecting licenses and managing the licenses that can be used in applications.

This role is often assigned to someone within the Legal department.

- **Policy Manager**

The Policy Manager can create, edit, or delete global policy rules.

The Policy Manager role should be assigned to users who are responsible for defining and managing all your OSS company policies. Often, these users are from the Legal/Compliance department or the IT/Security department. This user can also be the CTO overseeing all technology/development or the CISO who is responsible for all security practices.

- **Project Creator**

The Project Creator can create projects and project versions and can edit project and version settings.

The Project Creator role is often assigned to the Global Code Scanner or the Project Code scanner if that user needs to create new projects. The Global Code Scanner should almost always have the Project Creator role as well unless your organization has a centrally managed system for setting up new applications company wide.

- **Super User**

The Super User can manage (create/modify/delete) all projects and versions, create/modify/deactivate users, and assign/remove users to/from projects in Black Duck.

This role could be assigned to anyone from a VP of engineering who is responsible for a development

organization or a program manager who is responsible for company-wide OSS security/compliance.

- **System Administrator**

The System Administrator role can configure system settings.

The System Administrator role is geared primarily to the user that installs, sets up, and configures the Black Duck application. Most of the time, this will be an IT person responsible for registering the product, configuring LDAP and SSO, and so on.

## Project roles

The following project roles are available:

- **BOM Manager**

The BOM Manager can modify the BOM for projects in which they are members or have project-group privileges.

This role is often assigned to a lead developer or developer manager for a project.

- **Project Code Scanner**

The Project Code Scanner only has access to specific project scans in Black Duck and can map or delete scans for that project within the system. Unlike the Global Code Scanner, the Project Code Scanner only has code scanning capability for a set of projects – users are restricted from all other projects. The Project Code Scanner can create project versions of projects they have access to, but cannot create projects.

This role is often used in larger enterprises where multiple groups are responsible for builds/releases. This role could be assigned to a release engineer for a specific business unit or for a CI account for that business unit.

- **Project Manager**

The Project Manager has complete access to a specific Black Duck project. Project Managers can create/modify/delete versions for projects in which they are members or have project-group privileges but cannot create projects. Project Managers can assign users to the project, run reports, modify BOM entries, override policies, and remediate security vulnerabilities.

In smaller organizations this role is often assigned to the development manager or team lead and in larger enterprises this role could be assigned to the Director of engineering.

- **Project Viewer**

The Project Viewer role provides read-only access to individual projects. This is the lowest level of access and is often assigned to users who need to view information and access reports but should not be allowed to change anything. Project Viewers can add comments to a BOM.

This role is assigned to users by default if no other role is assigned to the user: a user without any project roles (no other project roles selected), will be a project viewer. This role is not shown as a selectable option.

- **Policy Violation Reviewer**

The Policy Violation Reviewer can override policies in projects in which they are members or have project-group privileges.

In smaller organizations this role is often assigned to a development manager, Director or VP of engineering, or even a program manager. In larger enterprises this role is often assigned to users who manage the OSS policies across the entire system. These users verify that what was needed to obtain approval for an override was completed as well as vet the validity of the override for each instance.

- Security Manager

This Security Manager can modify remediation for vulnerabilities associated with components.

In smaller organizations this role is often assigned to the development manager while in larger enterprises this is commonly assigned to someone in the security group reporting to the CISO.

## Managing user roles

Once you have created a user account, you can add [overall roles](#) to the user account. These overall roles specify what actions the user is able to perform and what information the user can view in Black Duck. Click [here](#) for more information on the tasks that can be performed for each role.

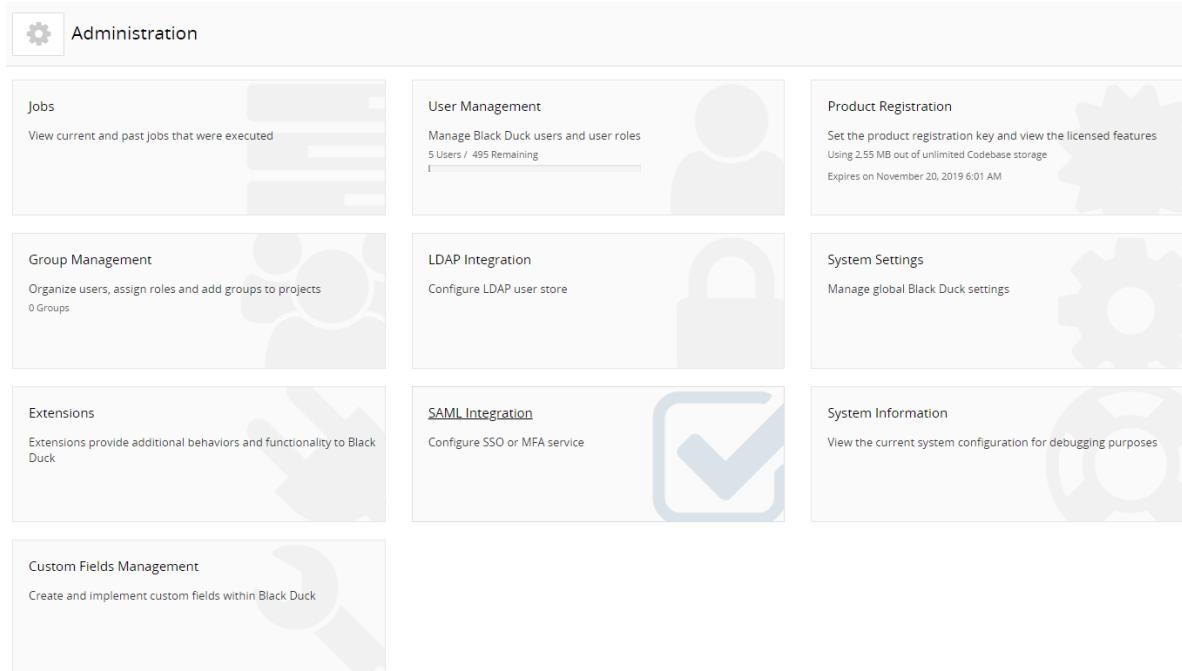
**Note:** If you do not assign a role to a user, that user has read-only access to Black Duck; this user cannot create projects. However, if a user with no roles is added as a project member, that user will be able to update and delete that project.

### ⚙️ To assign an overall role to a user



1. Click the expanding menu icon ( ) and select **Administration**.

The Administration page appears.



2. Select **User Management** to display the User Management page.

The 'User Management' page displays a table of users. The columns are: Username, First Name, Last Name, Email, Roles, and Status. The table contains three rows of data:

Username	First Name	Last Name	Email	Roles	Status
sysadmin	System	Administrator	noreply@blackducksoftware.com	Component Manager, Global Code Scanner, License Manager, Policy Manager, Project Creator, Super User, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23			Active

At the bottom right, it says 'Displaying 1-3 of 3'.

The roles assigned to each user appears on the page.

3. Find the user to whom you want to assign a role:

- Select the **Display Inactive Users** check box to include inactive users. Clearing this check box hides all inactive users.
- Filter the users that appear on the page.
- Sort the list of users by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.
- Use the pagination bar at the bottom of the list to go to the appropriate page if there are more users than are listed on this page.

4. Select the username to display the *Username* page.

The screenshot shows the 'Administration / User Management' section for a 'Sample User'. The 'User Details' tab is active, displaying the following information:

Username *	SampleUser
First Name *	Sample
Last Name *	User
Email	[Empty]

A checked checkbox labeled 'Active user' is present. A 'Save' button is located at the bottom right.

The 'Overall Roles' section lists several roles with descriptions:

- Component Manager**  
This role can create, update and delete custom components
- Global Code Scanner**  
This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project
- Global Project Viewer**  
This role has read only access to all projects
- License Manager**  
Ability to create/modify/delete licenses
- Policy Manager**  
Ability to create/modify/delete policies
- Project Creator**  
This role can create projects/versions and edit their settings.
- Super User**  
This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**  
This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application

The 'User Groups' section contains a '+ Add group' button and displays the message 'No Results Found'.

The 'Project Access' section contains a '+ Add project' button and displays the message 'No Results Found'.

5. In the **Overall Roles** section, select the global [roles](#) that you want to assign to this user account. Deselect any roles that you want to remove from this user account.
- The role is automatically assigned or removed. You do not have to save your configuration information.

## Viewing your roles

Use the My Profile page to view the roles assigned to your user account.

 To view your roles:

1. Log in to Black Duck.
2. Select **My Profile** from the user profile icon () to display the My Profile page.

The roles assigned to your user account are listed in the **Overall Roles** section.

**Note:** Users with the Super User role can view the roles assigned to a user account by selecting the username in the User Management page.

## Black Duck user role matrix

The roles assigned to a user or group determine the tasks that can be performed. You can assign multiple roles (or no roles) to a user or group.

Roles are also assigned to a user when a user is assigned as a member of a project.

### Global Roles

Task	Super User role	Component Manager role	Global Code Scanner role	Global Project Viewer	License Manager role	Policy Manager role	Project Creator role	System Administrator role
Manage code scans/Protex BOM files: <ul style="list-style-type: none"><li>• Scan code.</li><li>• Upload scans to Black Duck.</li><li>• Map or unmap scans to projects.</li></ul>			✓					
Create, edit, delete projects.	✓						✓	
Manage projects: <ul style="list-style-type: none"><li>• Create, edit,</li></ul>	✓						✓	

Task	Super User role	Component Manager role	Global Code Scanner role	Global Project Viewer	License Manager role	Policy Manager role	Project Creator role	System Administrator role
<ul style="list-style-type: none"> <li>delete project versions.</li> <li>Edit project or version settings, including tags.</li> </ul>							Only applies when user is member of project and assigned the Project Manager role (default value).	
Manage custom components.		✓						
Manage licenses:					✓ Can manage licenses for all projects.			
View BOMs: <ul style="list-style-type: none"> <li>View BOM.</li> <li>Add/edit/view comments.</li> <li>Print BOM.</li> <li>Compare BOMs.</li> </ul>	✓			✓ Can view all projects.				
Manage BOMs:	✓							
<ul style="list-style-type: none"> <li>Manually add components; delete manually</li> </ul>								

Task	Super User role	Component Manager role	Global Code Scanner role	Global Project Viewer	License Manager role	Policy Manager role	Project Creator role	System Administrator role
<ul style="list-style-type: none"> <li>added components.</li> <li>Ignore components.</li> <li>Review components.</li> <li>Remediate security vulnerabilities.</li> <li>Override policy violations.</li> <li>Remove override of policy violations.</li> <li>Edit licenses, including excluding license from Notices File report, adding an attribution statement, or selecting a different license for a component version.</li> <li>Indicate license term fulfillment status.</li> </ul>								
Manage policy rules: create, edit, or delete policy rules.						✓		
Run project vulnerability reports from the Report menu.	✓	✓	✓	✓	✓	✓	✓	✓

Task	Super User role	Component Manager role	Global Code Scanner role	Global Project Viewer	License Manager role	Policy Manager role	Project Creator role	System Administrator role
Project version reports: <ul style="list-style-type: none"><li>• Run Notices File report.</li><li>• Run Project Version report.</li></ul>	√			√				
View information in Dashboard pages (Project, Components, Security, and Summary).	√			√				
Access the Tools page from which user can: <ul style="list-style-type: none"><li>• Download the scanner.</li><li>• Access API documentation.</li></ul>	√	√	√	√	√	√	√	√
Search	√	√	√	√	√	√	√	√
Administer Black Duck. Use the Administration menu to: <ul style="list-style-type: none"><li>• View jobs.</li><li>• Register Black Duck.</li><li>• Configure LDAP.</li><li>• Configure SAML.</li><li>• Manage system settings.</li><li>• Manage custom fields.</li></ul>								√
Administer users and groups. Use the	√							

Task	Super User role	Component Manager role	Global Code Scanner role	Global Project Viewer	License Manager role	Policy Manager role	Project Creator role	System Administrator role
Administration menu to:								
<ul style="list-style-type: none"> <li>Manage users, including resetting passwords.</li> <li>Manage groups.</li> </ul>								
Manage snippets	✓							

## Project roles

These project-level roles only apply to the projects a user is assigned.

Task	Project Manager role	Security Manager role	BOM Manager role	Project Code Scanner role	Policy Violation Reviewer	No roles/Project Viewer
Manage code scans/Protex BOM files:	✓ Can unmap scans from their projects.			✓ Can map or unmap a code scan to/from projects for which they have access.		When a user is created and assigned to a project they have a read only/project viewer role.
Create, edit, delete projects.						
Manage projects:	✓ <ul style="list-style-type: none"> <li>Create, edit, delete project versions.</li> <li>Edit project or version settings,</li> </ul>			✓ Can only create project versions.		

Task	Project Manager role	Security Manager role	BOM Manager role	Project Code Scanner role	Policy Violation Reviewer	No roles/Project Viewer
including tags.						
Manage custom licenses: <ul style="list-style-type: none"><li>• Create, edit, delete custom licenses.</li></ul>			✓  Can only edit custom license text in BOM.			
View BOMs: <ul style="list-style-type: none"><li>• View BOM.</li><li>• View notifications.</li><li>• Add/edit/view comments.</li><li>• Print BOM.</li><li>• Compare BOMs.</li></ul>	✓	✓	✓	✓	✓	✓

Task	Project Manager role	Security Manager role	BOM Manager role	Project Code Scanner role	Policy Violation Reviewer	No roles/Project Viewer
Manage BOMs:	√	√	√ Can only modify remediation for vulnerabilities associated with components.	Cannot remediate security vulnerabilities. Cannot override policy violations or remove override of policy violations.	√ Can only override policies or remove overrides.	When a user is created and assigned to a project they have a read only/project viewer role.
Manage policy rules: create, edit, or delete policy rules.						
Run project vulnerability reports from the Report menu.	√	√	√	√	√	√
Project version reports:	√	√	√	√	√	√

Task	Project Manager role	Security Manager role	BOM Manager role	Project Code Scanner role	Policy Violation Reviewer	No roles/Project Viewer
<ul style="list-style-type: none"> <li>Run Notices File report.</li> <li>Run Project Version report.</li> </ul>						When a user is created and assigned to a project they have a read only/project viewer role.
View information in Dashboard pages (Project, Components, Security, and Summary).	√	√	√	√	√	√
Access the Tools page from which user can: <ul style="list-style-type: none"> <li>Download the scanner.</li> <li>Download Protex BOM Tool.</li> <li>Access API documentation.</li> </ul>	√	√	√	√	√	√
Search	√	√	√	√	√	√
Administer Black Duck. Use the Administration menu to: <ul style="list-style-type: none"> <li>View jobs.</li> <li>Register Black Duck.</li> <li>Configure LDAP.</li> <li>Configure SAML.</li> <li>Manage system settings.</li> </ul>						

Task	Project Manager role	Security Manager role	BOM Manager role	Project Code Scanner role	Policy Violation Reviewer	No roles/Project Viewer
Administer users and groups. Use the Administration menu to: <ul style="list-style-type: none"><li>• Manage users, including resetting passwords.</li><li>• Manage groups.</li></ul>						When a user is created and assigned to a project they have a read only/project viewer role.
Manage snippets	√		√			

## Authenticating users with LDAP

Authenticating users through an existing LDAP corporate directory helps to facilitate:

- The creation of user accounts. If the user account does not exist, upon successful authentication, the Black Duck user account is created.
- Centralized management of user account details. Each time a user logs in to Black Duck, Black Duck synchronizes with the directory server. If changes were made to mapped attributes, Black Duck updates the user account information.
- (Optional) The creation of groups. If a user is a member of an LDAP group, upon successful authentication, a Black Duck user account, as well as a Black Duck group, is created. The group is populated with the new user.

**Note:** Note: If the Black Duck group already exists, the Black Duck user account is created and the group is populated.

### To authenticate users with LDAP

1. Contact your LDAP administrator and gather the following information:

#### LDAP server details

This is the information that Black Duck uses to connect to the directory server.

- (required) The host name or IP address of the directory server, including the protocol scheme and port, on which the instance is listening.

**Example:** `ldap://<server_name>.<domain_name>.com:339`

Click [here](#) for more information on configuring secure LDAP.

- (optional) If your organization does not use anonymous authentication, and requires credentials for LDAP access, the password and either the LDAP name or the absolute LDAP distinguished name (DN) of a user that has permission to read the directory server.

**Example of an absolute LDAP DN:** `uid=ldapmanager,ou=employees,dc=company,dc=com`

**Example of an LDAP name:** `jdoe`

- (optional) If credentials are required for LDAP access, the authentication type to use: simple or digest-MD5.

### LDAP users attributes and LDAP attribute mappings

This is the information that the Black Duck uses to locate users in the directory server:

- (required) The absolute base DN under which users can be located.

**Example:** `dc=example,dc=com`

- (required) The attribute used to match a specific, unique user. The value of this attribute personalizes the user profile icon with the name of the user.

**Example:** `uid={0}`

- (optional). If some of your users are not located under the absolute base DN for the user search, the user DN pattern is used to match a specific, unique user.

**Example:** `cn={0},ou=contractors`

- (optional) The attributes that map to the first name, last name, and email address of users.

### LDAP groups

If you are enabling LDAP group synchronization, this is the required information that Black Duck uses to locate user groups in the directory server:

- (required) The absolute base DN under which groups can be located.

**Example:** `ou=groups,dc=example,dc=com`

- (required) The attribute used to match a unique user member within a given group.

**Example:** `uniqueMember={0}`

- (required) The attribute that identifies a specific, unique group name.

**Example:** `cn`

2. Log in to Black Duck as a system administrator.



3. Click the expanding menu icon ( ) and select **Administration**.

The Administration page appears.

A screenshot of the Black Duck Administration page. The page has a header with a gear icon and the word "Administration". Below the header are nine management options arranged in a grid:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. It shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. It shows Using 2.55 MB out of unlimited Codebase storage and Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. It shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service. It shows a checkmark icon.
- System Information**: View the current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck. It shows a magnifying glass icon.

4. Select **LDAP Integration** to open the LDAP Integration page.

The screenshot shows the 'Administration - LDAP Integration' page. It includes sections for 'LDAP Server Details', 'LDAP User Attributes', 'LDAP Attribute Mappings', 'LDAP Groups', and a 'Test Connection' section.

- LDAP Server Details:**
  - Enable LDAP:
  - Server URL: ldap://mamba-vm.blackducksoftware.com:389
  - Authentication Type: Simple
  - Manager DN: cn=BlackDuck Manager,ou=blackduck,ou=People,dc=blackducksoftware,dc=com
  - Manager Password: Password already set, click to change it.
- LDAP User Attributes:**
  - User Search Base: ou=People,dc=blackducksoftware,dc=com
  - User Search Filter: cn=(O)
  - User DN Pattern:
- LDAP Attribute Mappings:**
  - First Name: givenName
  - Last Name: Initials
  - Email: jmail
- LDAP Groups:**
  - Synchronize LDAP groups:
  - Group search base: ou=groups,dc=blackducksoftware,dc=com
  - Group filter: uniquemember=(O)
  - Group name attribute: cn
- Test Connection, User Authentication and Field Mapping:**

Tests ability to connect. Also tests ability to authenticate test-user and shows result of mapping test-user's meta-data. Note: test-user credentials are not saved.

  - Test Username:
  - Test Password:
  - Test Connection:

5. In the **LDAP Server Details** section:
  - Select **Enable LDAP**.
  - Enter the server connection and authentication details that Black Duck is to use to connect to the directory server.
6. In the **LDAP User Attributes** section, enter the user attributes values Black Duck is to use to locate users.
7. (Optional) Enter the attributes that map to user-specific information in the **LDAP Attribute Mappings** section.
8. (Optional) Select **Synchronize LDAP groups** and enter the group attribute values Black Duck is to use to locate groups in the **LDAP Groups** section.
9. (Optional) Enter user credentials in the **Test Connection, User Authentication and Field Mapping**

section and click **Test Connection** to test the connection to the directory server.

If the LDAP group synchronization is enabled and configured, the user's first name, last name, email address, and user's LDAP groups are displayed for successful connections.

10. Click **Save**.

## Configuring secure LDAP

If you see certificate issues when connecting your secure LDAP server to Black Duck, the most likely reason is that the Black Duck server has not set up a trust connection to the secure LDAP server. This usually occurs if you are using a self-signed certificate.

To set up a trust connection to the secure LDAP server, import the server certificate into the local Black Duck LDAP truststore by:

1. Obtaining your LDAP information.
2. Using the Black Duck UI to import the server certificate.

**Note:** All hosted customers should secure access to their Black Duck application by leveraging our out-of-the-box support for single sign on (SSO) via SAML or LDAP. Information on how to enable and configure these security features can be found in the installation guides. In addition, we encourage customers that are using a SAML SSO provider that offers two-factor authorization to also enable and leverage that technology to further secure access to their Black Duck application.

### Obtaining your LDAP information

Contact your LDAP administrator and gather the following information:

#### LDAP Server Details

This is the information that Black Duck uses to connect to the directory server.

- (required) The host name or IP address of the directory server, including the protocol scheme and port, on which the instance is listening.

**Example:** ldaps://<server\_name>.<domain\_name>.com:339

- (optional) If your organization does not use anonymous authentication, and requires credentials for LDAP access, the password and either the LDAP name or the absolute LDAP distinguished name (DN) of a user that has permission to read the directory server.

**Example of an absolute LDAP DN:** uid=ldapmanager,ou=employees,dc=company,dc=com

**Example of an LDAP name:** jdoe

- (optional) If credentials are required for LDAP access, the authentication type to use: simple or digest-MD5.

#### LDAP Users Attributes

This is the information that Black Duck uses to locate users in the directory server:

- (required) The absolute base DN under which users can be located.

**Example:** dc=example,dc=com

- (required) The attribute used to match a specific, unique user. The value of this attribute personalizes the user profile icon with the name of the user.

**Example:** uid={0}

### Test Username and Password

- (required) The user credentials to test the connection to the directory server.

## Importing the server certificate

### To import the server certificate

1. Log in to Black Duck as a system administrator.



2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.

3. Select **LDAP integration** to display the LDAP Integration page.

The screenshot shows the 'Administration' section with 'LDAP Integration' selected. The page is divided into several sections:

- LDAP Server Details:** Contains fields for 'Enable LDAP' (checkbox), 'Server URL' (text input), 'Authentication Type' (dropdown menu showing 'Nothing Selected'), 'Manager DN' (text input), and 'Manager Password' (text input).
- LDAP User Attributes:** Contains fields for 'User Search Base' (text input), 'User Search Filter' (text input), and 'User DN Pattern' (text input).
- LDAP Attribute Mappings:** Contains fields for 'First Name' (text input), 'Last Name' (text input), and 'Email' (text input).
- LDAP Groups:** Contains fields for 'Synchronize LDAP groups' (checkbox), 'Group search base' (text input), 'Group filter' (text input), and 'Group name attribute' (text input).
- Test Connection, User Authentication and Field Mapping:** Contains fields for 'Test Username' (text input) and 'Test Password' (text input). Below these is a 'Test Connection' button.

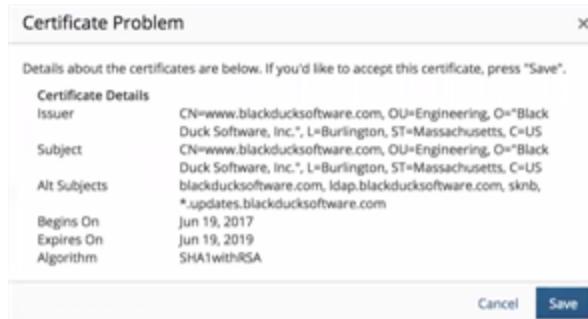
A 'Save' button is located at the bottom right of the main configuration area.

4. Select the **Enable LDAP** option and complete the information in the **LDAP Server Details** and **LDAP User Attributes** sections, as described above. In the **Server URL** field, ensure that you have configured the secure LDAP server: the protocol scheme is ldaps://.
5. Enter the user credentials in the **Test Connection, User Authentication and Field Mapping** section and click **Test Connection**.
6. If there are no issues with the certificate, it is automatically imported and the "Connection Test Succeeded" message appears:

Test Connection, User Authentication and Field Mapping  
Tests ability to connect. Also tests ability to authenticate test-user and shows result of mapping test-user's meta-data. Note: test-user credentials are not saved.

Test Username *	<input type="text" value="flast"/>
Test Password *	<input type="password" value="*****"/>
Test Connection	<span style="color: #0070C0;">⚠ Test Connection</span> <span style="color: #0070C0;">✓ Connection Test Succeeded</span>
<span style="color: #0070C0;">✓ First Name:</span> First <span style="color: #0070C0;">✓ Last Name:</span> Last <span style="color: #0070C0;">✓ Email:</span> flast@company.com	

7. If there is an issue with the certificate, a dialog box listing details about the certificate appears:



Do one of the following:

- Click **Cancel** to fix the certificate issues.

Once fixed, retest the connection to verify that the certificate issues have been fixed and the certificate has been imported. If successful, the "Connection Test Succeeded" message appears.

- Click **Save** to import this certificate.

Verify that the certificate has been imported by clicking **Test Connection**. If successful, the "Connection Test Succeeded" message appears.

## About locked out user accounts

A user will be locked out of their account for 10 minutes if they fail to enter the correct password after 10 attempts. After the 10th failed attempt, a message will appear on the login page notifying the user that their account is locked.

Log files contain information by username on successful logins, unsuccessful logins, and account lockouts.

**Note:** This lockout feature does not apply to users logging in using SAML or LDAP.

# Chapter 15: Managing groups in Black Duck

You can use groups in Black Duck to manage overall roles and project team membership for several user accounts at once instead of managing that information at the individual user account level. You can:

- [Create a group](#)
- [Manage group information](#) such as the group name or status
- [Manage group roles](#)
- [Add a member to a group](#)
- [Remove a member from a group](#)
- [Delete a group](#)

**Note:** If you are using an external LDAP directory server to authenticate users and have enabled LDAP group synchronization, the Group Management page uses the **Source** column to identify groups that were created in Black Duck (**Internal**) and groups that were created because of LDAP authentication (**LDAP**).

## Viewing your groups

You can view the groups you belong to, and the source, status, and roles associated with each group.

 To view your groups:

1. Log in to Black Duck.
2. Select your username or the user profile icon: .
3. Select **My Profile**.

The screenshot shows the 'My Profile' section for a user named 'SamplePolicyMgr'. It includes tabs for 'Profile', 'Change Password', and 'Overall Roles'. Under 'Profile', fields are shown for Username (SamplePolicyMgr), Email (PolicyMgr@companyname.com), First Name (First), and Last Name (Last). Under 'Change Password', fields are for Current Password, New Password, and Confirm Password. Under 'Overall Roles', 'Policy Manager' is listed with the ability to 'Ability to create/modify/delete policies'. The 'User Groups' section lists a group named 'Sample Group' with Source 'Internal' and Status 'Active', and assigned Role 'Policy Manager'. The 'Extensions' section shows an 'Email Extension' entry with a note: 'A Hub Extension to send emails based on notifications.' A large plug icon is displayed next to the heading 'User Extension Configuration'. Below it, a note says: 'Information saved for extensions can be read by that extension. Please make sure you trust the extension to handle your information correctly.' A 'Capture screenshot' button is also present.

#### 4. View your groups in the **User Groups** section.

You can [view the groups associated with a particular user](#).

## Creating groups

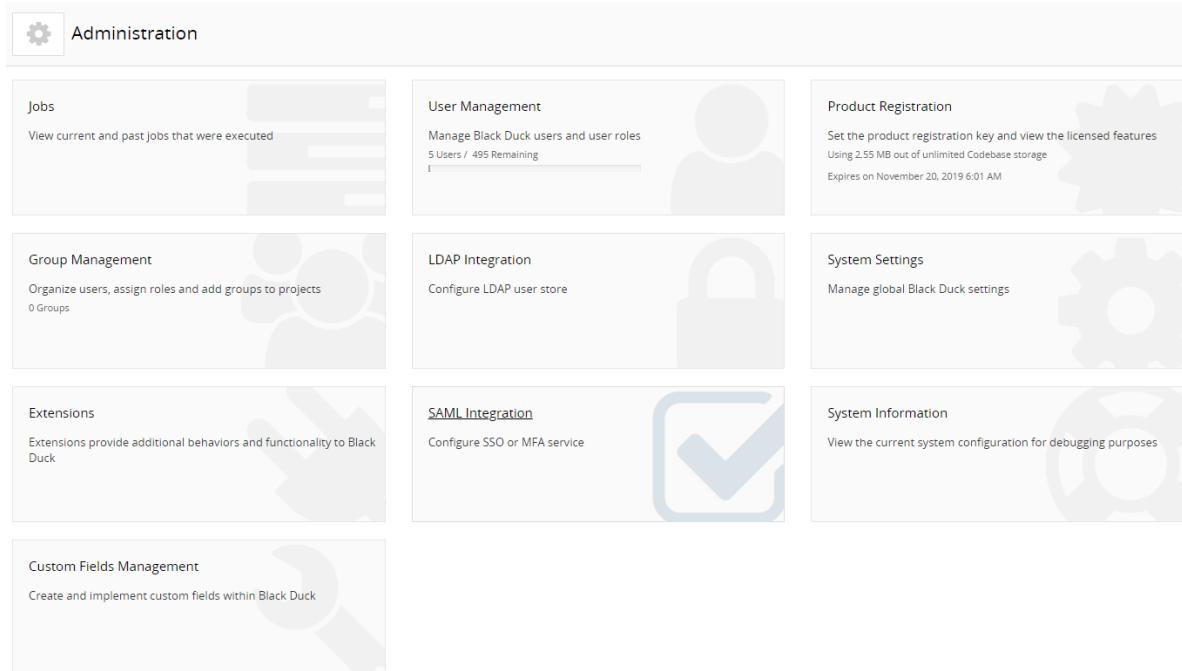
You can create and configure a group with specific roles that will be granted to all members of the group.

### To create a group

#### 1. Log in to Black Duck.

#### 2. Click the expanding menu icon ( ) and select **Administration**.

The Administration page appears.



### 3. Select **Group Management** to display the Group Management page.

The screenshot shows the 'Group Management' page. It features a table listing groups with columns for 'Group Name', 'Source', and 'Status'. A 'Create Group' button is located at the top left, and a 'Display Inactive Groups' checkbox is at the top right.

Group Name	Source	Status
Sample Group	Internal	Active
sysadmingroup	Internal	Active
Test 123	Internal	Active
Test Group 1	Internal	Active
Test Group 10	Internal	Active
Test Group 12	Internal	Active
Test Group 2	Internal	Active
Test Group 3	Internal	Active
Test Group 4	Internal	Active
Test Group 5	Internal	Active
Test Group 6	Internal	Active
Test Group 7	Internal	Active
Test Group 8	Internal	Active
Test Group 9	Internal	Active

### 4. Click **+ Create Group** to display the Create a New Group dialog box.

### 5. Type the name of the group in the **Group Name** field, select whether this group is active or inactive, and click **Create Group**. The Group Management page updates to display the new group.

You can now:

- [Add members](#) to the group.
- [Assign roles](#) to the group.

## Managing group information

After you have created a group, you can change the group name and/or status (active/inactive) if needed.

### To manage group information

1. Log in to Black Duck.



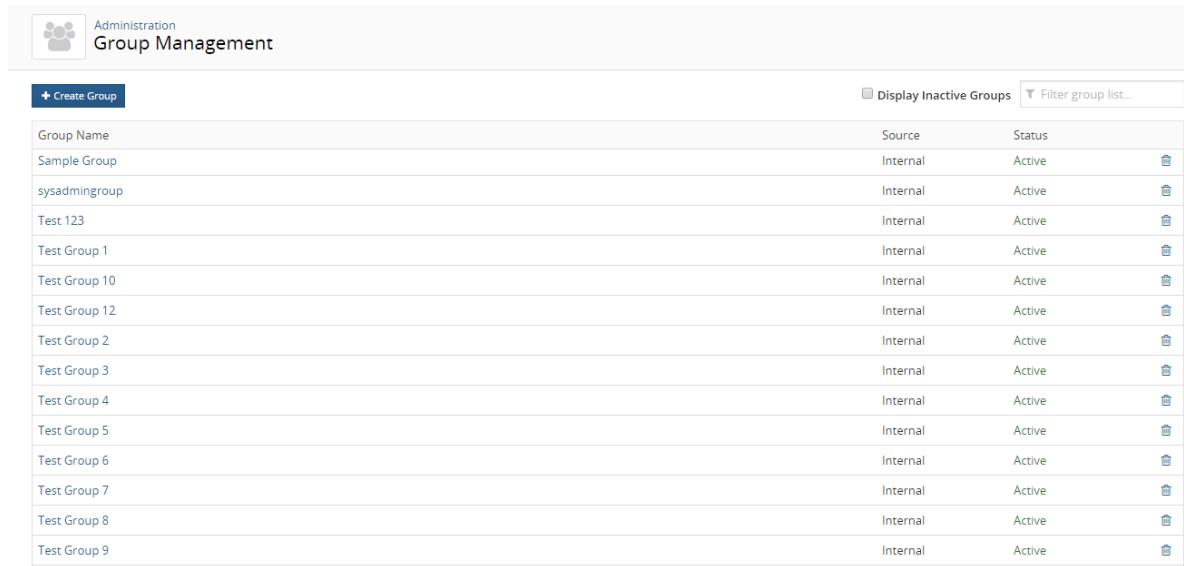
2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.

The screenshot shows the Black Duck Administration page with a grid of nine items:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. Shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. Shows Using 2.55 MB out of unlimited Codebase storage. Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. Shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service. Shows a checkmark icon.
- System Information**: View the current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck. Shows a magnifying glass icon.

3. Select **Group Management** to display the Group Management page.



The screenshot shows the 'Group Management' page under the 'Administration' section. At the top, there is a 'Create Group' button and two filter options: 'Display Inactive Groups' (unchecked) and 'Filter group list...' (unchecked). The main area is a table with three columns: 'Group Name', 'Source', and 'Status'. The 'Group Name' column lists various groups, some of which have arrows indicating they can be sorted. The 'Source' and 'Status' columns show that most groups are 'Internal' and 'Active'. There are also icons for deleting each group.

Group Name	Source	Status
Sample Group	Internal	Active
sysadmingroup	Internal	Active
Test 123	Internal	Active
Test Group 1	Internal	Active
Test Group 10	Internal	Active
Test Group 12	Internal	Active
Test Group 2	Internal	Active
Test Group 3	Internal	Active
Test Group 4	Internal	Active
Test Group 5	Internal	Active
Test Group 6	Internal	Active
Test Group 7	Internal	Active
Test Group 8	Internal	Active
Test Group 9	Internal	Active

4. Find the name of the group whose name you want to modify:
  - Select the **Display Inactive Groups** check box to include inactive groups. Clearing this check box hides all inactive groups.
  - Filter the groups that appear on the page.
  - Sort the list of group names by selecting the column. An arrow next to the column name indicates the direction the list is sorted.
  - Use the pagination bar at the bottom of the list to go to the appropriate page if there are more groups than are listed on this page.
5. Select the group name you want to edit to display the *Group Name* page.

The screenshot shows the 'Group Details' section of the Black Duck Group Management interface. The group name is 'Sample Group 1'. The 'Active Group' checkbox is checked. Below the form are sections for 'Overall Roles' (listing various system roles like Component Manager, Global Code Scanner, etc.) and 'Group Members' (showing a button to add members). The 'Group Projects' section is also shown.

Group Details

Group Name \* Sample Group 1

Active Group  Active Group

[Delete Group](#) [Save](#)

Overall Roles

- Component Manager**  
This role can create, update and delete custom components
- Global Code Scanner**  
This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project
- Global Project Viewer**  
This role has read only access to all projects
- License Manager**  
Ability to create/modify/delete licenses
- Policy Manager**  
Ability to create/modify/delete policies
- Project Creator**  
This role can create projects/versions and edit their settings.
- Super User**  
This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**  
This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application

Group Members

[+ Add Member](#)

No Results Found

Group Projects

[+ Add Project](#)

No Results Found

6. In the **Group Details** section, type the new group name and/or change the status of this group.
7. Click **Save** to save the changed information.

Black Duck saves the group name and status.
8. Use the other sections on this page to:
  - [Manage group roles](#).
  - [Add or remove](#) group members.
  - [Add or remove](#) projects.

## Managing group projects

You can manage the projects assigned to a group using the **Group Name** page.

### To assign a project to a group

1. Log in to Black Duck.



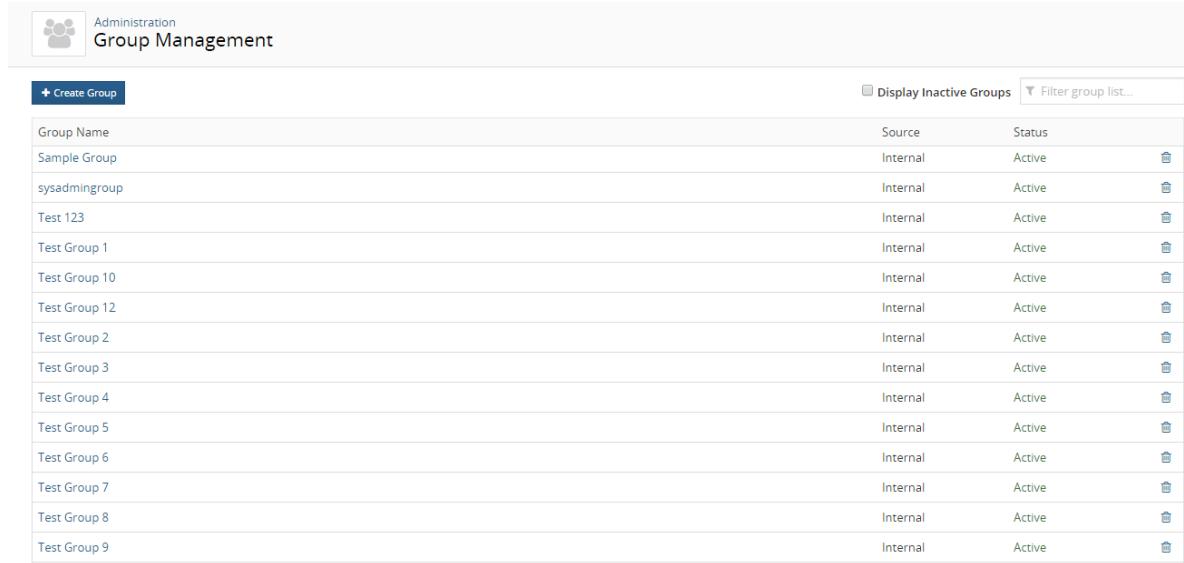
2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.

The screenshot shows the Black Duck Administration page with a grid of nine items:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. (5 Users / 495 Remaining)
- Product Registration**: Set the product registration key and view the licensed features. (Using 2.55 MB out of unlimited Codebase storage. Expires on November 20, 2019 6:01 AM)
- Group Management**: Organize users, assign roles and add groups to projects. (0 Groups)
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service. (Checkmark icon)
- System Information**: View the current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck. (Search icon)

3. Select **Group Management** to display the Group Management page.



Group Name	Source	Status
Sample Group	Internal	Active
sysadmingroup	Internal	Active
Test 123	Internal	Active
Test Group 1	Internal	Active
Test Group 10	Internal	Active
Test Group 12	Internal	Active
Test Group 2	Internal	Active
Test Group 3	Internal	Active
Test Group 4	Internal	Active
Test Group 5	Internal	Active
Test Group 6	Internal	Active
Test Group 7	Internal	Active
Test Group 8	Internal	Active
Test Group 9	Internal	Active

4. Select the name of the group to display the *Group Name* page.

The screenshot shows the 'Group Details' section for 'Sample Group 1'. It includes fields for 'Group Name \*' (set to 'Sample Group 1'), an 'Active Group' checkbox (unchecked), and buttons for 'Delete Group' (red) and 'Save' (blue). Below this is the 'Overall Roles' section, which lists eight roles with descriptions:

- Component Manager**  
This role can create, update and delete custom components
- Global Code Scanner**  
This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project
- Global Project Viewer**  
This role has read only access to all projects
- License Manager**  
Ability to create/modify/delete licenses
- Policy Manager**  
Ability to create/modify/delete policies
- Project Creator**  
This role can create projects/versions and edit their settings.
- Super User**  
This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**  
This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application

The 'Group Members' section shows a button to '+ Add Member' and displays the message 'No Results Found'. The 'Group Projects' section shows a button to '+ Add Project' and displays the message 'No Results Found'.

5. Click **Add Project** in the **Group Projects** section to display the Add Project dialog box.

6. Enter one or more projects and click **Add**.

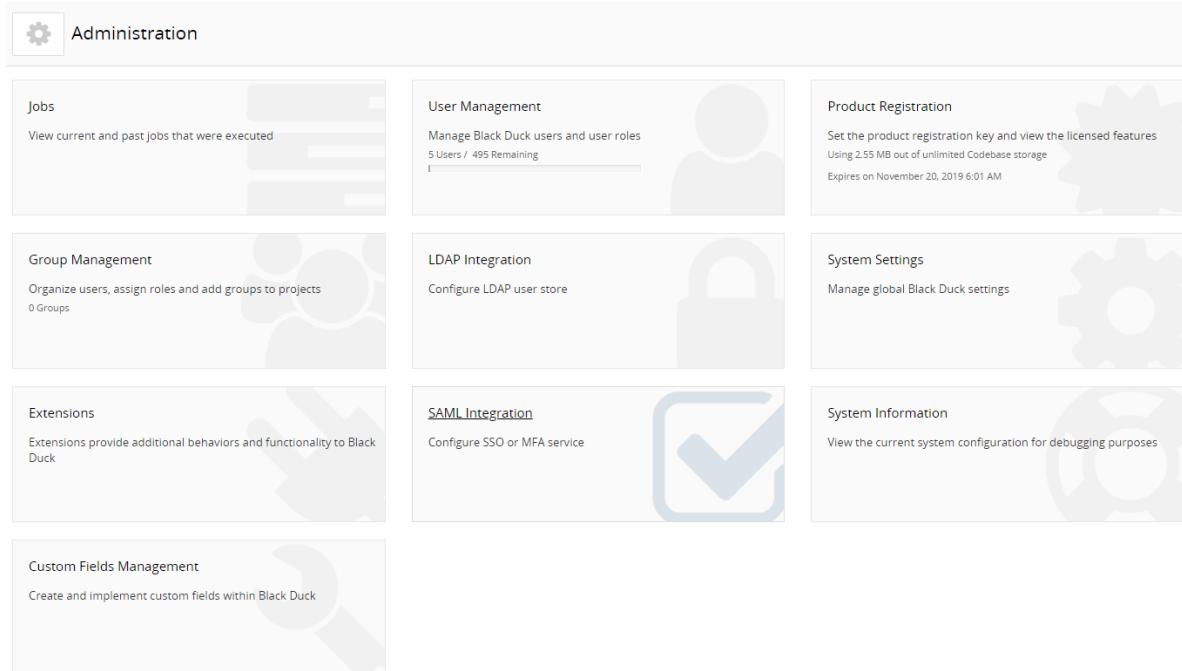
 **To remove a project from a group**

1. Log in to Black Duck.



2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.



### 3. Select **Group Management** to display the Group Management page.

The Group Management page displays a list of groups:

Group Name	Source	Status	Action
Sample Group	Internal	Active	
sysadmingroup	Internal	Active	
Test 123	Internal	Active	
Test Group 1	Internal	Active	
Test Group 10	Internal	Active	
Test Group 12	Internal	Active	
Test Group 2	Internal	Active	
Test Group 3	Internal	Active	
Test Group 4	Internal	Active	
Test Group 5	Internal	Active	
Test Group 6	Internal	Active	
Test Group 7	Internal	Active	
Test Group 8	Internal	Active	
Test Group 9	Internal	Active	

### 4. Select the name of the group you want to remove.

The screenshot shows the Black Duck Group Management interface. At the top, there's a navigation bar with a user icon and the text "Administration / Group Management". Below it, the group name "Sample Group 1" is displayed. The interface is divided into several sections:

- Group Details:** Shows the group name "Sample Group 1" and an "Active Group" checkbox which is checked.
- Overall Roles:** A list of roles with descriptions:
  - Component Manager:** This role can create, update and delete custom components.
  - Global Code Scanner:** This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project.
  - Global Project Viewer:** This role has read only access to all projects.
  - License Manager:** Ability to create/modify/delete licenses.
  - Policy Manager:** Ability to create/modify/delete policies.
  - Project Creator:** This role can create projects/versions and edit their settings.
  - Super User:** This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
  - System Administrator:** This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application.
- Group Members:** A section with a "+ Add Member" button. It displays the message "No Results Found".
- Group Projects:** A section with a "+ Add Project" button. It displays the message "No Results Found".

5. Click in the row of the group you want to remove in the **Group Projects** section.
6. Click **Remove** to confirm.

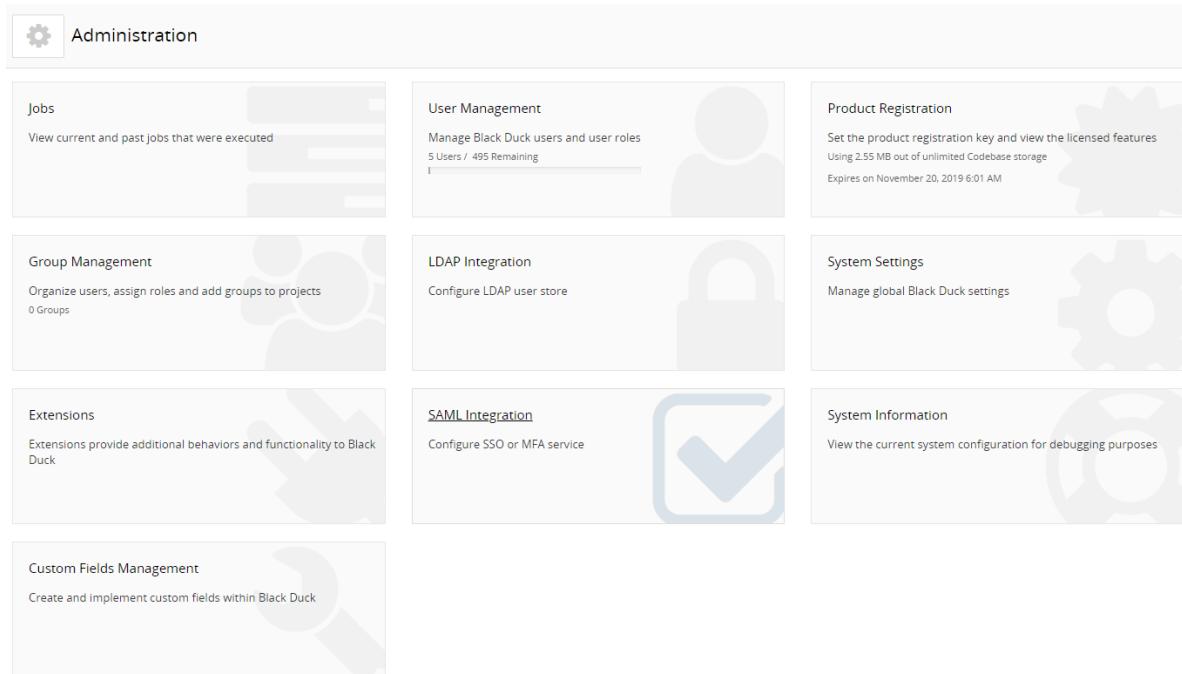
## Managing group roles

Once you have added [overall roles](#) to a group, you can add users to the group, then assign that group to one or more projects. These users will have the overall roles assigned to the group and will be members of all project teams to which the group has been added.

## ⚙️ To manage group roles

1. Click the expanding menu icon () and select **Administration**.

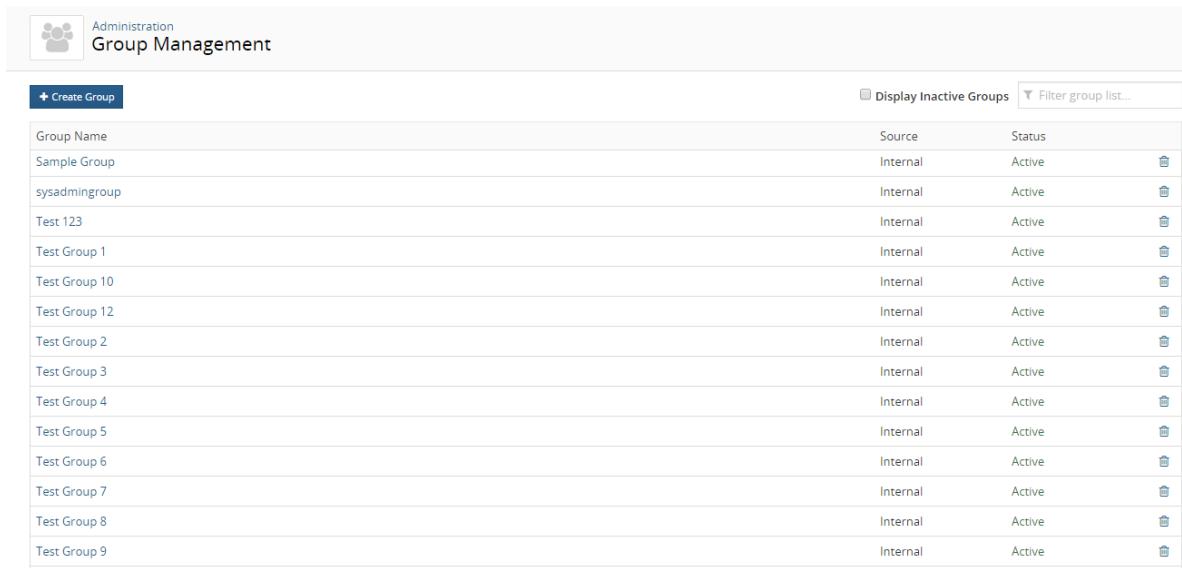
The Administration page appears.



The screenshot shows the Black Duck Administration page with the following sections:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. Shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. Shows Using 2.55 MB out of unlimited Codebase storage. Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. Shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service. Shows a checked checkbox icon.
- System Information**: View the current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck.

2. Select **Group Management** to display the Group Management page.



The screenshot shows the Black Duck Group Management page with the following interface elements:

- Create Group** button.
- Display Inactive Groups** checkbox.
- Filter group list...** input field.
- Group Name** column: Sample Group, sysadmingroup, Test 123, Test Group 1, Test Group 10, Test Group 12, Test Group 2, Test Group 3, Test Group 4, Test Group 5, Test Group 6, Test Group 7, Test Group 8, Test Group 9.
- Source** column: Internal, Internal.
- Status** column: Active, Active.
- Action** column: Delete icons.

3. Find the name of the group for which you want to manage roles to display the **Group Name** page:

- Select the **Display Inactive Groups** check box to include inactive groups. Clearing this check box hides all inactive groups.
- Filter the groups that appear on the page.
- Sort the list of groups by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.
- Use the pagination bar at the bottom of the list to go to the appropriate page if there are more groups than are listed on this page.

4. Select the name of a group to display the *Group Name* page.

The screenshot shows the 'Administration / Group Management' interface. A sidebar on the left displays a user icon and the text 'Administration / Group Management'. The main content area is titled 'Sample Group 1'. It contains three sections: 'Group Details', 'Overall Roles', and 'Group Members'.

**Group Details:** This section includes a 'Group Name' field set to 'Sample Group 1', an 'Active Group' checkbox checked, and two buttons: 'Delete Group' (red) and 'Save' (blue).

**Overall Roles:** This section lists several roles with descriptions:

- Component Manager**: This role can create, update and delete custom components.
- Global Code Scanner**: This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project.
- Global Project Viewer**: This role has read only access to all projects.
- License Manager**: Ability to create/modify/delete licenses.
- Policy Manager**: Ability to create/modify/delete policies.
- Project Creator**: This role can create projects/versions and edit their settings.
- Super User**: This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**: This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application.

**Group Members:** This section features a '+ Add Member' button and the message 'No Results Found'.

**Group Projects:** This section features a '+ Add Project' button and the message 'No Results Found'.

5. In the **Roles** section, select the roles that you want to assign to all members of this group. Deselect any

roles that you want to remove from this group.

The role is automatically assigned to the group. You do not have to save your configuration information.

## Adding a member to a group

You can add members to a group by:

- Managing a group and adding members to the group
- Managing a user and adding the user to groups

### To add a member to a group by managing a group

1. Log in to Black Duck.



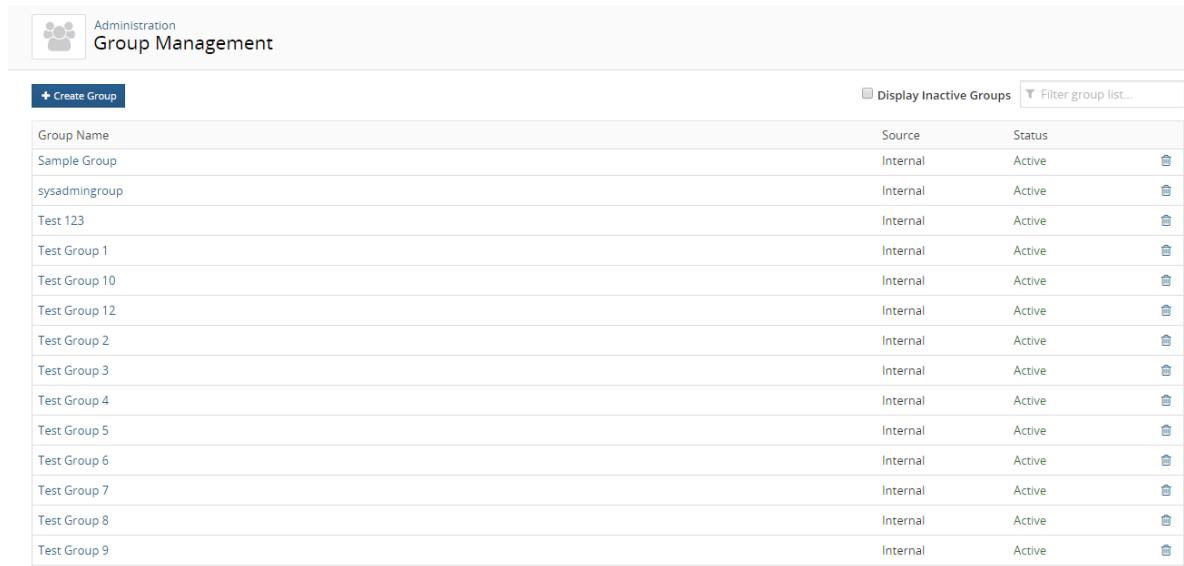
2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.

The screenshot shows the Black Duck Administration page with a grid of nine items:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. Shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. Shows Using 2.55 MB out of unlimited Codebase storage and Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. Shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service. Shows a checkmark icon.
- System Information**: View the current system configuration for debugging purposes.

3. Select **Group Management** to display the Group Management page.



The screenshot shows the 'Group Management' page under the 'Administration' section. At the top, there is a 'Create Group' button and two checkboxes: 'Display Inactive Groups' (unchecked) and 'Filter group list...' (unchecked). The main area is a table with columns: 'Group Name', 'Source', and 'Status'. The 'Group Name' column lists various groups: Sample Group, sysadmingroup, Test 123, Test Group 1, Test Group 10, Test Group 12, Test Group 2, Test Group 3, Test Group 4, Test Group 5, Test Group 6, Test Group 7, Test Group 8, and Test Group 9. The 'Source' column shows all entries as 'Internal'. The 'Status' column shows all entries as 'Active'. Each row has a small trash can icon in the bottom right corner.

Group Name	Source	Status
Sample Group	Internal	Active
sysadmingroup	Internal	Active
Test 123	Internal	Active
Test Group 1	Internal	Active
Test Group 10	Internal	Active
Test Group 12	Internal	Active
Test Group 2	Internal	Active
Test Group 3	Internal	Active
Test Group 4	Internal	Active
Test Group 5	Internal	Active
Test Group 6	Internal	Active
Test Group 7	Internal	Active
Test Group 8	Internal	Active
Test Group 9	Internal	Active

4. Find the name of the group for which you want to manage membership:

- Select the **Display Inactive Groups** check box to include inactive groups. Clearing this check box hides all inactive groups.
- Filter the groups that appear on the page.
- Sort the list of groups by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.
- Use the pagination bar at the bottom of the list to go to the appropriate page if there are more groups than are listed on this page.

5. Select a group to display the *Group Name* page.

The screenshot shows the Black Duck Group Management interface. At the top left is a user icon and the navigation path "Administration / Group Management". The main title is "Sample Group 1".

**Group Details:**

- Group Name \*: Sample Group 1
- Active Group:  Active Group

**Delete Group** **Save**

**Overall Roles:**

- Component Manager**  
This role can create, update and delete custom components
- Global Code Scanner**  
This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project
- Global Project Viewer**  
This role has read only access to all projects
- License Manager**  
Ability to create/modify/delete licenses
- Policy Manager**  
Ability to create/modify/delete policies
- Project Creator**  
This role can create projects/versions and edit their settings.
- Super User**  
This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**  
This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application

**Group Members:**

**+ Add Member**

No Results Found

**Group Projects:**

**+ Add Project**

No Results Found

6. Click **+ Add Member** in the **Group Members** section to display the Add a Group Member dialog box.
7. Begin typing the user name of the user that you want to add to the project team. The list is type-ahead enabled, so you can see a list of available user names that contain the text you have typed.
8. Select the username that you want to add to the group.
9. Click **Add**.

The group member list updates to show the newly-added member.

## To add a member to a group by managing a user

1. Log in to Black Duck.



2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.

The screenshot shows the Black Duck Administration interface. It features a grid of nine cards:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. Shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. Shows Using 2.55 MB out of unlimited Codebase storage, Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. Shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service. Shows a large checkmark icon.
- System Information**: View the current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck.

3. Select **User Management** to display the User Management page.

The screenshot shows the User Management page. At the top, there is a search bar with filters for User Status (Active), a filter button, and an add filter button. Below the search bar is a table displaying user information:

Username	First Name	Last Name	Email	Roles	Status
sysadmin	System	Administrator	noreply@blackducksoftware.com	Component Manager, Global Code Scanner, License Manager, Policy Manager, Project Creator, Super User, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23			Active

At the bottom right of the table, it says "Displaying 1-3 of 3".

4. Find the user you want to find:

- Select the **Display Inactive Users** check box to include inactive users. Clearing this check box hides all inactive users.
- Filter the users that appear on the page.

- Sort the list of users by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.
  - Use the pagination bar at the bottom of the list to go to the appropriate page if there are more users than are listed on this page.
5. Select the user to display the *Username* page.

The screenshot shows the 'Administration / User Management' section for a 'Sample User'. The user details are as follows:

Username *	SampleUser
First Name *	Sample
Last Name *	User
Email	[Empty]

A checkbox labeled 'Active user' is checked. A 'Save' button is located to the right of the form.

The 'Overall Roles' section lists several roles with descriptions:

- Component Manager**  
This role can create, update and delete custom components
- Global Code Scanner**  
This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project
- Global Project Viewer**  
This role has read only access to all projects
- License Manager**  
Ability to create/modify/delete licenses
- Policy Manager**  
Ability to create/modify/delete policies
- Project Creator**  
This role can create projects/versions and edit their settings.
- Super User**  
This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**  
This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application

The 'User Groups' section contains a '+ Add group' button and displays the message 'No Results Found'.

The 'Project Access' section contains a '+ Add project' button and displays the message 'No Results Found'.

6. Click **Add group** in the **User Groups** section to display the Add Group dialog box.
7. Begin typing the group name. The list is type-ahead enabled, so you can see a list of available group names that contain the text you have typed.
8. Select the groups you want this user to join.
9. Click **Add**.

The group table updates to display the newly-added group(s).

Note that the roles assigned to this user are [determined by the group](#).

## Removing a member from a group

You can remove members from a group by:

- Managing a group and removing members from the group
- Managing a user and removing group membership from the user

### To remove a member from a group when managing a group

1. Log in to Black Duck.



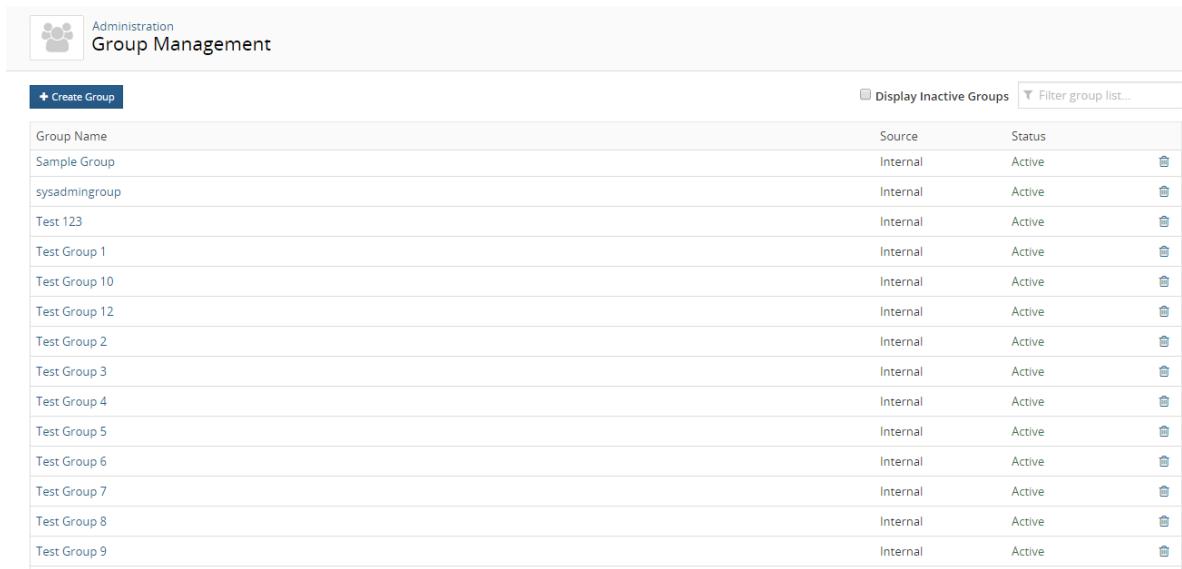
2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.

The screenshot shows the Black Duck Administration page with the following sections:

- Administration**: The main title.
- Jobs**: View current and past jobs.
- User Management**: Manage Black Duck users and user roles. Shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view licensed features. Shows Using 2.55 MB out of unlimited Codebase storage, Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles, and add groups to projects. Shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service. Shows a checkmark icon.
- System Information**: View current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck.

3. Select **Group Management** to display the Group Management page.



The screenshot shows the 'Group Management' page under the 'Administration' section. At the top, there is a 'Create Group' button and a 'Display Inactive Groups' checkbox which is checked. Below the table, there is a 'Filter group list...' input field. The table lists 14 groups: Sample Group, sysadmingroup, Test 123, Test Group 1, Test Group 10, Test Group 12, Test Group 2, Test Group 3, Test Group 4, Test Group 5, Test Group 6, Test Group 7, Test Group 8, and Test Group 9. The columns are 'Group Name', 'Source', and 'Status'. All groups listed are Internal and Active, with a trash icon in the status column.

Group Name	Source	Status
Sample Group	Internal	Active
sysadmingroup	Internal	Active
Test 123	Internal	Active
Test Group 1	Internal	Active
Test Group 10	Internal	Active
Test Group 12	Internal	Active
Test Group 2	Internal	Active
Test Group 3	Internal	Active
Test Group 4	Internal	Active
Test Group 5	Internal	Active
Test Group 6	Internal	Active
Test Group 7	Internal	Active
Test Group 8	Internal	Active
Test Group 9	Internal	Active

4. Find the name of the group for which you want to manage membership:

- Select the **Display Inactive Groups** check box to include inactive groups. Clearing this check box hides all inactive groups.
- Filter the groups that appear on the page.
- Sort the list of group names by selecting the column. An arrow next to the column name indicates the direction the list is sorted.
- Use the pagination bar at the bottom of the list to go to the appropriate page if there are more groups than are listed on this page.

5. Select the group to display the *Group Name* page.

The screenshot shows the Black Duck Group Management interface. At the top left is a navigation icon and the path "Administration / Group Management". The main title is "Sample Group 1".

**Group Details:**

- Group Name \*: Sample Group 1
- Active Group:  Active Group

**Delete Group** (red button) and **Save** (blue button) buttons.

**Overall Roles:**

- Component Manager**: This role can create, update and delete custom components
- Global Code Scanner**: This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project
- Global Project Viewer**: This role has read only access to all projects
- License Manager**: Ability to create/modify/delete licenses
- Policy Manager**: Ability to create/modify/delete policies
- Project Creator**: This role can create projects/versions and edit their settings.
- Super User**: This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**: This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application

**Group Members:**

**+ Add Member** button. Below it, the message "No Results Found".

**Group Projects:**

**+ Add Project** button. Below it, the message "No Results Found".

6. In the **Group Members** section, click  in the row of the name of the member you want to remove.
7. In the Remove User from Group dialog box, click **Remove**.

The group member list updates to reflect the updated group membership.

 **To remove a member from a group while managing a user**

1. Log in to Black Duck.
2. Click the expanding menu icon (



) and select **Administration**.

The Administration page appears.

The screenshot shows the Black Duck Administration interface. At the top left is a gear icon labeled "Administration". Below it are several cards representing different management functions:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. It shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view licensed features. It shows 2.55 MB out of unlimited Codebase storage, with an expiration date of November 20, 2019, 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. Shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck.

### 3. Select **User Management** to display the User Management page.

The screenshot shows the User Management page. At the top left is a user icon labeled "User Management". Below it is a table listing users:

Username	First Name	Last Name	Email	Roles	Status
sysadmin	System	Administrator	noreply@blackducksoftware.com	Component Manager, Global Code Scanner, License Manager, Policy Manager, Project Creator, Super User, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23			Active

At the bottom right of the table, it says "Displaying 1-3 of 3".

### 4. Find the user you want to find:

- Select the **Display Inactive Users** check box to include inactive users. Clearing this check box hides all inactive users.
- Filter the users that appear on the page.
- Sort the list of users by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.

- Use the pagination bar at the bottom of the list to go to the appropriate page if there are more users than are listed on this page.
5. Select the user to display the *Username* page.

The screenshot shows the 'Administration / User Management' section for a 'Sample User'. The user details are as follows:

Username *	SampleUser
First Name *	Sample
Last Name *	User
Email	[Empty]

A checked checkbox indicates the user is 'Active user'. A 'Save' button is present.

The 'Overall Roles' section lists several roles with descriptions:

- Component Manager**  
This role can create, update and delete custom components
- Global Code Scanner**  
This user can run code scans and map them to projects. When assigned globally, they can scan and map to any project
- Global Project Viewer**  
This role has read only access to all projects
- License Manager**  
Ability to create/modify/delete licenses
- Policy Manager**  
Ability to create/modify/delete policies
- Project Creator**  
This role can create projects/versions and edit their settings.
- Super User**  
This role has access to all user and project data. This person can create/modify/delete projects and versions, create/modify/deactivate users, etc.
- System Administrator**  
This role will have access to system settings like the registration key, jobs page, LDAP, SSO, etc. This is more for an IT person who installs and manages the hosting of the application

The 'User Groups' section contains a '+ Add group' button and displays 'No Results Found'.

The 'Project Access' section contains a '+ Add project' button and displays 'No Results Found'.

6. Find the group you want to remove for this user in the **User Groups** section and click .
7. In the Remove User from Group dialog box, click **Remove**.

The group list updates to reflect the updated group membership.

## Deleting groups

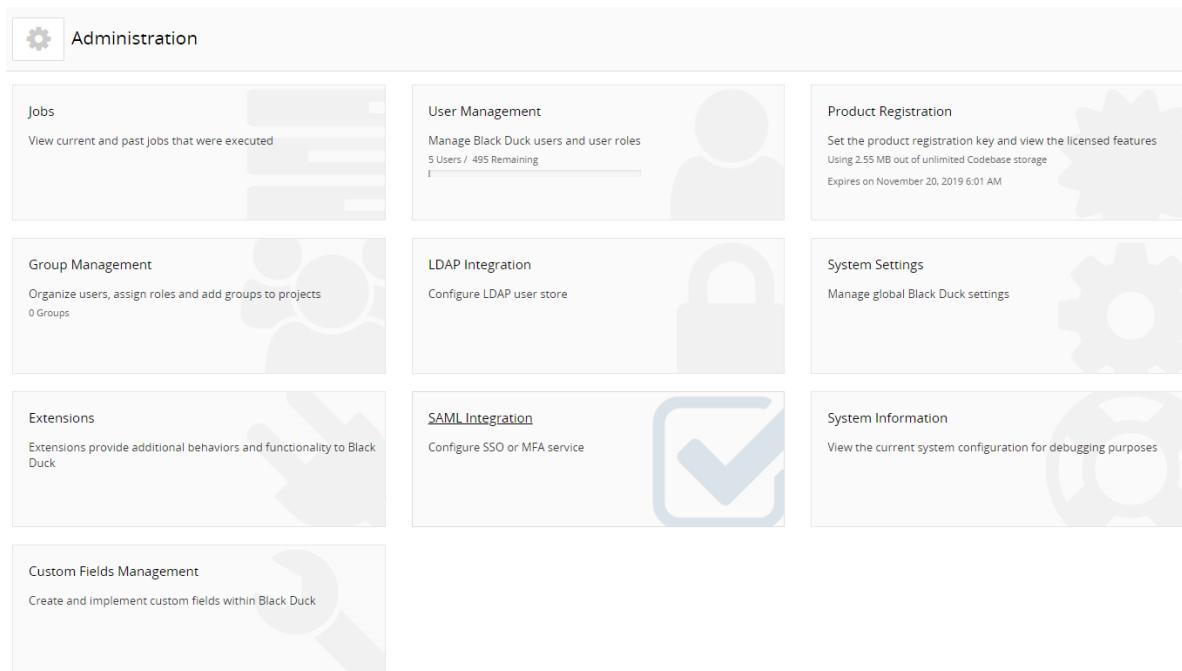
You do not need to remove members from a user group to delete it. When you delete the group, the group membership and permissions are removed from the user's records.

### To delete a user group

1. Log in to Black Duck.

2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.



The screenshot shows the Black Duck Administration page with a grid of nine cards:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. Shows 5 Users / 495 Remaining.
- Product Registration**: Set the product registration key and view the licensed features. Shows Using 2.55 MB out of unlimited Codebase storage. Expires on November 20, 2019 6:01 AM.
- Group Management**: Organize users, assign roles and add groups to projects. Shows 0 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service. Shows a checkmark icon.
- System Information**: View the current system configuration for debugging purposes.

3. Select **Group Management** to display the Group Management page.

Group Name	Source	Status
Sample Group	Internal	Active
sysadmingroup	Internal	Active
Test 123	Internal	Active
Test Group 1	Internal	Active
Test Group 10	Internal	Active
Test Group 12	Internal	Active
Test Group 2	Internal	Active
Test Group 3	Internal	Active
Test Group 4	Internal	Active
Test Group 5	Internal	Active
Test Group 6	Internal	Active
Test Group 7	Internal	Active
Test Group 8	Internal	Active
Test Group 9	Internal	Active

4. Find the group you want to delete:

- Select the **Display Inactive Groups** check box to include inactive groups. Clearing this check box hides all inactive groups.
- Filter the groups that appear on the page.
- Sort the list of group names by selecting the column. An arrow next to the column name indicates the direction the list is sorted.
- Use the pagination bar at the bottom of the list to go to the appropriate page if there are more groups than are listed on this page.

5. Click in the row of the group that you want to delete.

6. In the Delete Group dialog box, click **Delete**.

The group is deleted from Black Duck. Users who were assigned to the deleted group no longer have any overall roles that were associated with belonging to that group and no longer have membership on project teams granted through that group.

# Chapter 16: About Custom Fields

Custom fields provide you with a way to include additional information to help you manage open source software in your company or organize large projects. For example, to help you organize your development teams, you may want your projects to include the responsible business unit.

You can create custom fields for:

- BOM Components
- Components
- Component Versions
- Projects
- Project versions

A custom field is a system-wide property that will apply to all BOM components, components, component versions, projects, or project versions.

Users with the system administrator [role](#) can:

- [Create](#) or [edit](#) custom fields.
- [Activate or deactivate](#) a custom field. By default, a custom field is inactive and not shown to users.
- [Determine the order](#) of the custom fields as shown in the UI.

Note the following:

- All custom fields are optional.
- A custom field option is available for the [Project Version report](#). Selecting this option lists the project version custom field labels and values.
- You cannot change the type of custom field once it has been created. For example, suppose you created a multiple choice custom field. If, after you created the field, you want to change that custom field to a single choice custom field, you must create a new custom field.
- Custom field information is not available from the reporting database.
- You can create a policy rule for project custom fields that use the Drop Down, Multiple Selections, or Single Selection field type.

## Viewing custom field information in the Black Duck UI

- BOM Component custom field information appears when viewing the details of a component in the BOM.

The Custom Field icon ( ⓘ ) indicates that there are custom fields for this component.

To add information:

1. Click ⏺ and select **Edit** in the component version row to display the Edit Component dialog box.
2. Select **Additional Fields** and enter the information for the custom fields.
3. Click **Update**.

**Note:** If you manually adjust this component, the ⓘ adjustment icon appears in the component version row; the ⓘ adjustment icon which indicates a modified component overrides the Custom Field icon ( ⓘ ).

Click ⓘ or ⓘ to open the Component Details dialog box which displays the information.

- Component custom field information is shown in the **Additional Fields** section of the **Component Name Settings** tab:

OpenSSL http://www.openssl.org/  
OpenSSL Versions: 360

Component Name: OpenSSL

Description: OpenSSL is a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. It is also a general-purpose cryptography library.  
OpenSSL is licensed under an Apache-style license, which basically means that you are free to get and use it for

Url: http://www.openssl.org/

Notes:

Status: Unreviewed

Additional Fields

Select the team who originally added this component:

Architecture  
 Maintenance  
 New Development

Save

- Component version custom field information is shown in the **Additional Fields** section of the **Component Name Version Name Settings** tab:

The screenshot shows the 'Component Details' tab for the OpenSSL 0.8.1b version. It includes fields for Version (0.8.1b), License (SSLeay License), Release Date (12/21/1998), Notes (empty), and Status (Unreviewed). There are tabs for Security, Cryptography, Details, and Settings at the top.

**Additional Fields**

Select the team who originally added this component version:

- Architecture
- Maintenance
- New Development

**Save**

Once you have selected values for the custom fields, the information appears on the **Component Name Version Name Details** tab:

The screenshot shows the 'Details' tab for the OpenSSL 0.8.1b version. It includes sections for Description, Activity, Where Used, and a sidebar with Licenses, Open Hub, Component Links, Tags, and Vulnerabilities (76).

**Description**

OpenSSL is a robust, commercial-grade, and full-featured toolkit for the Transport Layer Security (TLS) and Secure Sockets Layer (SSL) protocols. It is also a general-purpose cryptography library. OpenSSL is licensed under an Apache-style license, which basically means that you are free to get and use it for commercial and non-commercial purposes subject to some simple license conditions.

Released	Newer Versions	Status	Updated
Dec 21, 1998	357	Unreviewed	Apr 18, 2019

Activity	Community
Last 12 Months: 1618 commits — stable Last commit: Apr 18, 2019	Last 12 Months: 126 contributors

**Where Used**

Project	Version	Released	Phase
TutorialFiles	1.0.0	Never	In Development

Displaying 1-1 of 1

**Licenses**  
SSLeay License

**Open Hub**  
<https://www.openhub.net/p/3728>

**Component Links**  
<http://www.openssl.org>

**Tags**

- c
- cryptography
- decryption
- openssl
- perl
- secure
- secureconnection
- security
- sockets
- ssl
- tls
- transport
- win32
- windows
- x64

Select the team who originally added this component version:  
Architecture

- Project custom field information is shown in the **Additional Fields** section of the **Project Name Settings** tab:

The screenshot shows the 'Custom Sample' project settings page. It includes sections for 'Additional Fields' and 'Application ID'. In the 'Additional Fields' section, there is a text input field labeled 'Has the attorney approved all licenses?' and a radio button group for selecting the team responsible for the project. In the 'Application ID' section, there is a text input field and a 'Save' button.

Once you have selected values for the custom fields, the information appears on the **Project Name Overview** tab:

The screenshot shows the 'Sample Project 4' project overview page. It displays a table of versions, a filter bar, and various project details. On the right side, it shows custom field values: 'Description' (No description), 'Created' (Mar 21, 2019 by sysadmin), 'Updated' (Mar 21, 2019 by sysadmin), 'Tags' (No Tags), 'Has Legal approved this project?' (true), and 'Select the team responsible for this project' (Architecture).

- Project version custom field information is shown in the **Additional Fields** section of the **Project Name Version Name Settings** tab:

The screenshot shows a project management interface for 'Custom Sample > 1.0'. At the top, there are tabs for 'Components', 'Security', 'Source', 'Reports', 'Details' (which is selected), and 'Settings'. Below the tabs, there's a section for 'Version Details' with fields for 'Version' (1.0), 'License' (Unknown License), 'Notes', 'Nickname', 'Release Date' (with a calendar icon), 'Phase' (In Planning), and 'Distribution' (External). A 'Save' button is located at the bottom right of this section. Below this, there's a section for 'Additional Fields' with a field for 'Enter the GA date for this project version' containing '02/14/2019', another 'Save' button, and a 'Scans' link.

## Creating a custom field

The process to create a custom field consists of:

1. Creating the field as described below.
2. [Activating the field](#).
3. [Determining the location of the custom field](#) when shown in the UI.

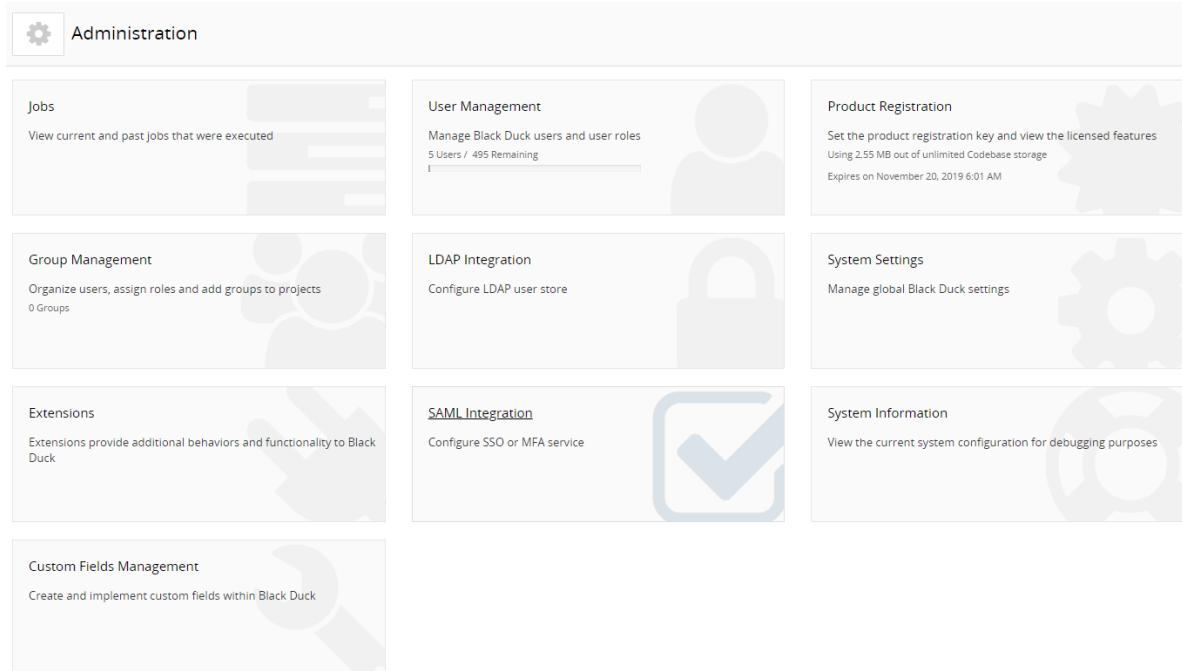
You must have the System Administrator role to create and manage custom fields.

### To create a custom field



1. Click the expanding menu icon ( ) and select **Administration**.

The Administration page appears.

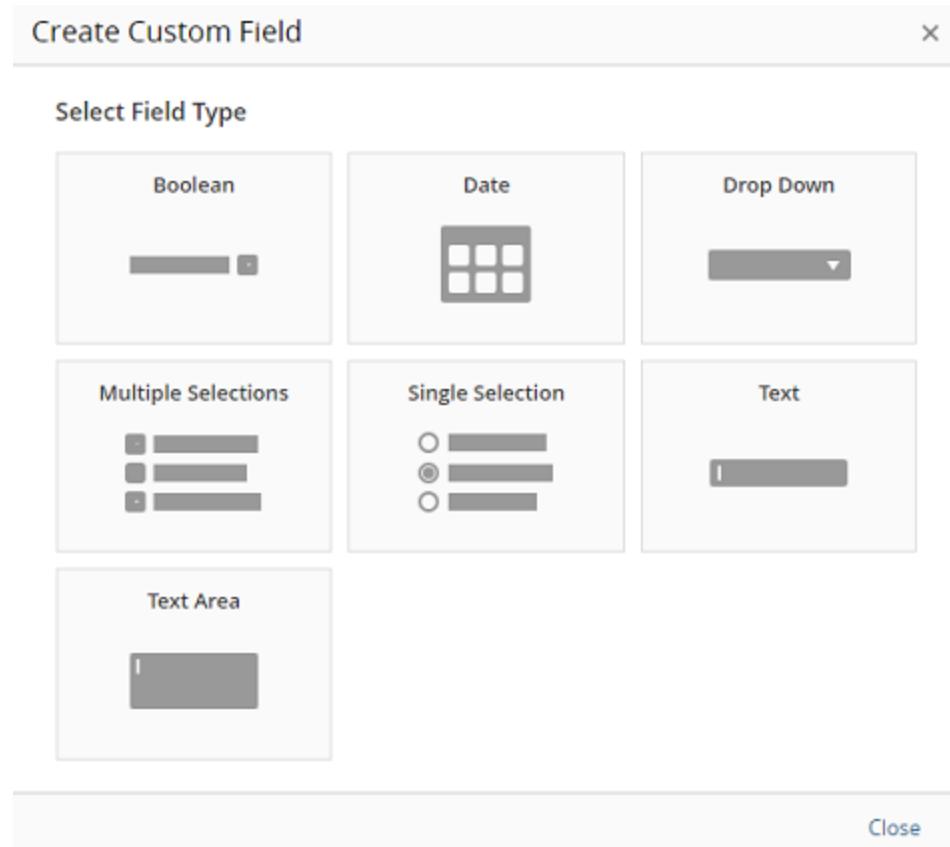


2. Select **Custom Fields Management** to display the Custom Fields Management page.

A screenshot of the Custom Fields Management page. At the top left is a wrench icon labeled "Administration" and below it "Custom Fields Management". A navigation bar at the top has tabs: "BOM Component" (selected), "Component", "Component Version", "Project", and "Project Version". To the right of the tabs is a button "+ Create". The main area shows a message "No Results Found" and a sub-message "It looks like you haven't created custom fields yet.".

By default, the **BOM Component** tab is selected.

3. Select the type of custom field you wish to create (such as for a component or project) and click **Create** to display the Create Custom Field dialog box.



4. Select the type of custom field. The types of custom fields are:

- **Boolean.** A check box will appear to the user from which they can select or clear the option.
- **Date.** A calendar will appear to the user from which they can select a date. The default is the user's current date.
- **Drop Down.** A drop-down list appears to the user, from which they must select an option.
- **Multiple Selections.** A list appears to the user, from which they can select one or more options.
- **Single Selection.** A list appears to the user, from which they can select only one option.
- **Text.** A field which you can enter text. There is no limit to the number of characters you can enter for this field.
- **Text Area.** A field which you can enter a large amount of text. There is no limit to the number of characters you can enter for this area.

The Create Custom Field dialog box reappears with the required fields for the custom field type you selected.

5. Regardless of the type of field you selected, all custom fields in the Create Custom Field dialog box have these fields and options:
  - **Label.** Enter a label for this custom field. This label will appear to the user when viewing the settings for the project or project version. This field is required. Note that there is no limit to the number of characters for the label.

- **Description.** Optionally, enter a description for this custom field. This description will appear to the user when viewing the settings for the project or project version. Note that there is no limit to number of characters for the description.
  - Click **Change Field Type** to return to the previous dialog box, as shown in step 3. If you select this option, you will lose the information you entered in this dialog box.
6. For the Drop Down, Multiple Selections, and Single Selection custom field types, use the **Field Options** section to define the options for the user to select.
- Enter text in the **Value** field. This is the text that the user sees when viewing the options.
  - By default, the dialog box shows only one value. Click **Add Option** to display an additional option. There is no limit to the number of options you can add.
  - Click  to remove the list item. If there is only one value, you cannot delete it.
  - To rearrange the order that these options appear to your users, use  , located to the left of the value, to drag and drop the option to the correct location.
7. Click **Save**.

The field appears at the top of the table on the Custom Fields Management page.

## Activating or deactivating a custom field

By default, a custom field is deactivated when it is first created. A deactivated field will not appear in the UI to your users. For a custom field to appear to your users, you must activate it.

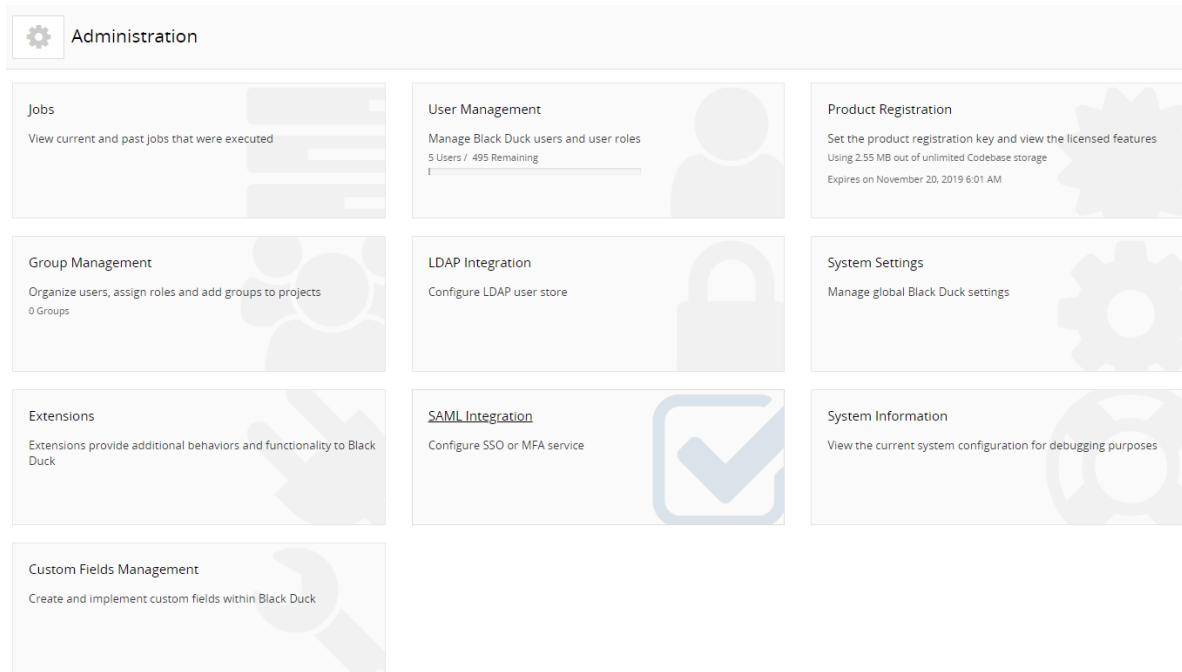
You must have the System Administrator role to activate or deactivate a custom field.

Note that you can deactivate a custom field at any time. If that custom field contained data (your users entered information for that custom field), it is retained; if you reactivate the field, the data for that custom field will reappear in the UI.

### To activate or deactivate a custom field

1. Click the expanding menu icon () and select **Administration**.

The Administration page appears.



## 2. Select **Custom Fields Management** to display the Custom Fields Management page.

Custom Fields Management					
BOM Component	+ Create		Type	Last Modified	Active
Component			Text	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>
Component Version			Boolean	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>
Project			Single Selection	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>
Project Version					

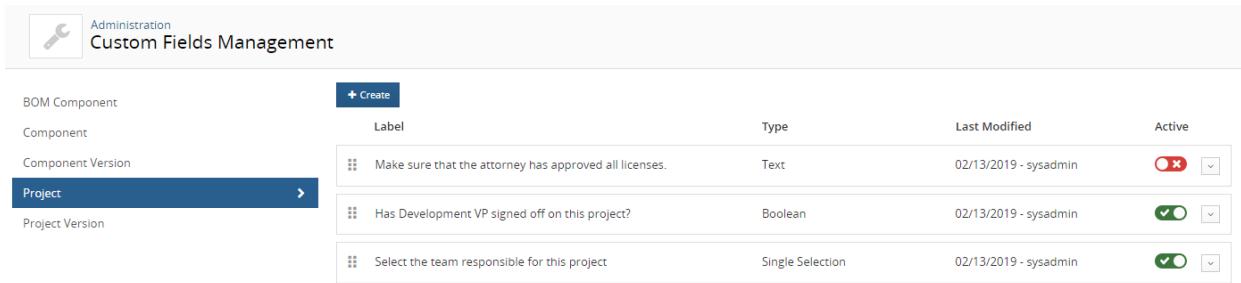
## 3. Select the tab which contains the custom field.

## 4. In the row of the custom field, select the **Active** switch:

- indicates the custom field is active.
- indicates the custom field is inactive.

## Determining the order of custom fields shown in the UI

Custom fields appear in a specific order when shown in the UI, such as in the **Additional Fields** section in the project or project version **Settings** tab. This location is determined by the tables shown in the Custom Fields Management page – the order of the custom fields shown here defines the order of custom fields shown in the UI.



BOM Component	+ Create	Label	Type	Last Modified	Active
Component		Make sure that the attorney has approved all licenses.	Text	02/13/2019 - sysadmin	
Component Version		Has Development VP signed off on this project?	Boolean	02/13/2019 - sysadmin	
<b>Project</b>	▶	Select the team responsible for this project	Single Selection	02/13/2019 - sysadmin	
Project Version					

By default, when you create a new custom field, it appears on the top of the table on the Custom Fields

Management page. To rearrange the order of the custom field, use  , located to the left of the custom field, to drag and drop it to the correct location.

You can change the order of a custom field at any time.

## Editing a custom field

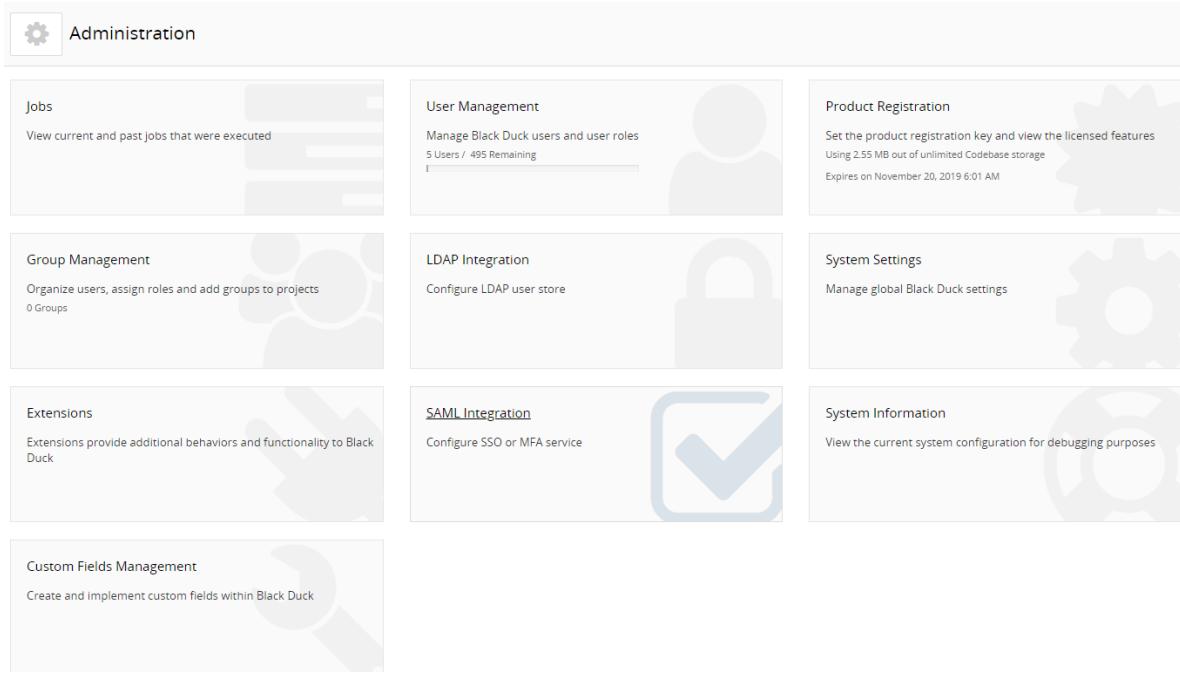
You can only edit the label or description of an existing custom field.

You cannot change the type of custom field once it has been created. For example, suppose you created a multiple choice custom field. If, after you created the field, you want to change that custom field to a single choice custom field, you must create a new custom field.

### To edit a custom field

1. Click the expanding menu icon () and select **Administration**.

The Administration page appears.



2. Select **Custom Fields Management** to display the Custom Fields Management page.

The screenshot shows the 'Custom Fields Management' page for a 'Project'. It lists three custom fields:

BOM Component	+ Create	Label	Type	Last Modified	Active
Component		Make sure that the attorney has approved all licenses.	Text	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>
Component Version		Has Development VP signed off on this project?	Boolean	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>
Project	>	Select the team responsible for this project	Single Selection	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>
Project Version					

3. Select the tab which contains the custom field you want to edit.

4. Click and select **Edit** in the row of the custom field.
5. In the Edit Custom Field dialog box, modify the label and/or the description, and click **Save**.

# Chapter 17: Other administrative tasks

This chapter describes:

- [Viewing project and project version audit information](#)
- [Viewing product registration information.](#)
- [Managing code size limits.](#)
- [Managing user access tokens.](#)
- [Customizing the logo.](#)
- [Viewing jobs.](#)
- [Accessing log files.](#)

## Viewing project and project version audit information

Black Duck tracks and displays all updates and changes that affect a project and/or project version. Use this information to understand who made changes or the events that caused changes to a project or project version. With this audit trail, you can determine, for example:

- who made changes to the BOM, such as who reviewed a component, added a comment, or ignored a component
- what changes occurred due to a scan, such as what components were added or deleted and what changes occurred due to those components (for example, the vulnerabilities that were added)
- who created or deleted a project version
- when was a policy violation triggered or when was a component no longer in violation,
- when was a policy violation overridden or when was the override reversed
- when did a component in your BOM introduce a new vulnerability
- when was remediation information updated for a vulnerability on a component in your project
- when did someone add or remove users from a project
- when was a snippet match confirmed or ignored

Black Duck provides the following information:

- The object that affected the project or project version, such as a component, vulnerability, or scan
- The type of event, such as vulnerability was found or a component was edited
- Who caused the event in the format User: *username*. If the Black Duck system caused the event (for example components or vulnerabilities found during a scan or an update to the Black Duck

KnowledgeBase that changed a vulnerability), the column shows User: blackduck\_system.

- Date and time this event occurred.

The following is an example of a new project and project version created during a scan:

Object	Event	Cause	Date and Time
> Project: 1.0	Project Created	User: sysadmin	Mon, Mar 4, 2019 12:29 PM
> User: sysadmin	User Role Added	User: sysadmin	Mon, Mar 4, 2019 12:29 PM
> User: sysadmin	User Role Added	User: sysadmin	Mon, Mar 4, 2019 12:29 PM
> Project Version: Sample Audit Project	Version Created	User: sysadmin	Mon, Mar 4, 2019 12:29 PM

Displaying 1-4 of 4

Note the following:

- Information is shown for the past 24 hours with the most recent changes appearing at the top of the table. Use the date filter to view information for different periods of time.
- While the deletion of a project version appears at the project level, deletion of a project will not appear here.

#### To view audit information

Audit information appears on the **Settings** tab of the project or project version.

1. Log in to Black Duck.
2. Select the name of the project from the **Projects** tab to view to the *Project Name* page.
3. Do one of the following:
  - To view *project* level audit information, select the **Settings** tab and then select **Activity**.

Object	Event	Cause	Date and Time
> Project: 1.0	Project Created	User: sysadmin	Mon, Mar 4, 2019 12:29 PM
> User: sysadmin	User Role Added	User: sysadmin	Mon, Mar 4, 2019 12:29 PM
> User: sysadmin	User Role Added	User: sysadmin	Mon, Mar 4, 2019 12:29 PM
> Project Version: Sample Audit Project	Version Created	User: sysadmin	Mon, Mar 4, 2019 12:29 PM

Displaying 1-4 of 4

- To view *project version* level audit information, select the version, select the **Settings** tab, and then select **Activity**.

The screenshot shows the Black Duck Project interface for a 'Sample Audit Project'. The 'Activity' tab is selected. A table lists component additions:

Object	Event	Cause	Date and Time
> Component: Java API for XML Processing	Component Added	User: blackduck_system	Mon, Mar 4, 2019 12:33 PM
> Component: gradle-one-jar	Component Added	User: blackduck_system	Mon, Mar 4, 2019 12:33 PM
> Component: AspectJ weaver	Component Added	User: blackduck_system	Mon, Mar 4, 2019 12:33 PM
> Component: Apache Commons Codec	Component Added	User: blackduck_system	Mon, Mar 4, 2019 12:33 PM
> Component: SLF4J LOG4J-12 Binding	Component Added	User: blackduck_system	Mon, Mar 4, 2019 12:33 PM
> Component: swagger-annotations	Component Added	User: blackduck_system	Mon, Mar 4, 2019 12:33 PM
> Component: jakarta.jws API	Component Added	User: blackduck_system	Mon, Mar 4, 2019 12:33 PM

#### 4. From this page:

- Click > located to the left of the object name to view details of this event.

The screenshot shows the details for a specific component addition. The component is 'Component: Java API for XML Processing'. The change history table includes:

Source	KnowledgeBase
Type	Component
Version	1.4
Is Modified	false
Origin External Namespace	maven
Origin External Id	jaxp.xml:jaxp-api:1.4
Origin Id	1.4

- Filter the table to view specific information, such as activity during a specific date range or a specific type of event.

## Viewing product registration information

The Product Registration page lists:

- Your registration ID
- Status and expiration date and time
- Registration features
  - Number of users
  - Number of projects
  - Number of project versions
  - Number of codebase KBs/MBs/GBs
  - Number of scans
- Licensed modules. Available modules are:
  - Black Duck Security Advisory
  - Black Duck Binary Analysis
  - Cryptography
  - Component Scanning
  - License Management
  - Notifications

- OpsSight
- OSS Notices Report
- Policy Management
- Risk Management
- Snippets

## Updating your product registration

Your Black Duck license may restrict the number of users, projects, and/or project versions. If you need more capacity, you can purchase a new license. Once you receive a new license from Black Duck Software, enter the new registration ID information in Black Duck to activate your newly-licensed capacity.

1. Log in to Black Duck as a system administrator.



2. Click the expanding menu icon and select **Administration**.

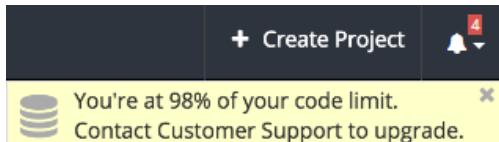
The Administration page appears.

A screenshot of the Black Duck Administration page. The page has a header with a gear icon and the word "Administration". Below the header are nine cards arranged in a grid. The cards are: "Jobs" (View current and past jobs that were executed), "User Management" (Manage Black Duck users and user roles, 5 Users / 495 Remaining), "Product Registration" (Set the product registration key and view the licensed features, Using 2.55 MB out of unlimited Codebase storage, Expires on November 20, 2019 6:01 AM), "Group Management" (Organize users, assign roles and add groups to projects, 0 Groups), "LDAP Integration" (Configure LDAP user store), "System Settings" (Manage global Black Duck settings), "Extensions" (Extensions provide additional behaviors and functionality to Black Duck), "SAML Integration" (Configure SSO or MFA service), and "System Information" (View the current system configuration for debugging purposes). Each card has a small icon and a brief description.

3. Select **Product Registration** to open the Product Registration page.
4. Type your new registration key in the **Registration ID** field. Be sure that you accept the terms of the End User License Agreement.
5. Click **Save**.

## Managing your code size limits

Black Duck will notify you when you are approaching your code size limit (as declared in your license). A notification, such as the following, appears in the UI when you are at 80% or higher of your code size limit:



If you exceed your code size limit, an error message appears when trying to scan (for example, shown in log files in Jenkins or on the screen in Synopsys Detect (Desktop)) or when uploading scans to Black Duck. You will not be able to scan or upload scans if you exceed your code size limit.

When receiving this notification, you can:

- Contact Customer Support to upgrade your service.
- View the scan size for a project version:
  1. Select the name of the project using the **Projects** tab on the Dashboard. The *Project Name* page appears.
  2. Select the version name to open the **Components** tab.
  3. Select the **Settings** tab.
  4. Select **Scans** to view the scans mapped to this project version.

The screenshot shows the Black Duck Project page for "Sample Project". The "Scans" tab is selected. The page displays a table of one scan entry:

Status	Name	Scan Size	Last Updated
✓	ComplexBomMainProject_2015-12-04 10:28:23	1.19 MB	May 29, 2019

Displaying 1-1 of 1

The scan size appears above the list of scans.

- [Delete existing scans](#) to free up space.

To determine the size of a scan:

1. Select **Scans** from the expanding menu ( ) to display the Scans page.
2. Select the path of the scan that you want to view the results to open the *Scan Name* page.

The screenshot shows the Black Duck Scans interface. At the top, there's a header with a cylinder icon, the text "Scans bds00992#/C:/Scan36/", and a "Scan" button. Below the header, it says "Host: bds00992 | Scan Initiated By: sysadmin | Updated: 2:23 PM". The main content area has a section titled "Scan Details - for the last completed scan" with a table of statistics:

	Host	bds00992	Path	/C:/Scan36/	Created on	Aug 2, 2017	Match Count	399	Files	59,172	Folders	3,574	Scan Size	195.24 MB
Delete Scan														

Below this is a "Map Scan to Project Version" section with buttons for "Map to Project" and "Create Project". It also states "This scan is not mapped to any versions." There's a "Scan History" section with a table:

Status	Components	Host	Path	Scan Size	Last Updated	Scan Initiated By
Complete	399 Matches	bds00992	/C:/Scan36/	195.24 MB	2:23 PM	

The **Scan Details** sections lists the scan size.

**Note:** You can view your current usage versus your limit on the Scans page. Values appear in the upper right corner of the page.

## Managing user access tokens

Black Duck provides the ability for you to generate one or more “tokens” for accessing Black Duck APIs. These tokens are intended to replace the use of username/password credentials in integration configurations, such as Jenkins or for the Scan Client CLI. With access tokens, if a security breach occurs, the user’s credentials (which might be their SSO or LDAP credentials) are not directly compromised.

Note the following:

- Access tokens can only be created by the current user.
- Access tokens are tied to a user’s account; therefore, an access token has the same role as the user who created the token.
- A user can have multiple tokens. Each token must have a unique name.
- Access tokens do not expire.
- If a user is inactivated, their tokens are invalidated.

Refer to the Getting Started with the SDK for information on using the API keys.

### To generate an access token

1. Log in to Black Duck.
2. Select your username or the user profile icon:



3. Select **My Profile**.

The My Profile page appears.

4. Enter a name, description (optional), and select the scope for this token (read and/or write access) in the **User Access Token** section. You can select one or more access for a token.

User Access Token

User access tokens can be used instead of a username and password or to authenticate to the API over Basic Authentication. Once your key is generated, you will get a message showing you the key. For security reasons, this will be the only time your key is presented to you, so be sure to save it. You also have the option to regenerate a new key with the same Name and Description at any time.

Name \*

Description

Scopes \*  Read Access  
 Write Access

**Generate**

There are currently no user access tokens. Feel free to generate one above!

5. Click **Generate**.

The Access Token Name dialog box appears with the access token.

6. Copy the access token shown in the dialog box. This token can only be viewed here at this time. Once you close the dialog box, you cannot view the value of this token.
7. Click **Close**.

To edit an access token

You can edit the name and description of an access token. You cannot edit the scope (read and/or write access) of a token.

1. Log in to Black Duck.



2. Select your username or the user profile icon: .
3. Select **My Profile**.

The My Profile page appears.

4. In the **User Access Token** section, click in the row of the token you want to revise and select **Edit**.

The Edit User Access Token dialog box appears.

5. Edit the name or description and click **Update**.

#### To regenerate an access token

You can regenerate a new access token which provides a different key for the same name, description, and access.

1. Log in to Black Duck.



2. Select your username or the user profile icon: .
3. Select **My Profile**.

The My Profile page appears.

4. In the **User Access Token** section, click  in the row of the token you want to regenerate and select **Regenerate**.

The Regenerate User Access Token dialog box appears.

5. Click **Regenerate** to confirm.

The Access Token Name dialog box appears with the new access token.

6. Copy the access token shown in the dialog box. This token can only be viewed here at this time. Once you close the dialog box, you cannot view the value of this token.
7. Click **Close**.

#### To delete an access token

1. Log in to Black Duck.



2. Select your username or the user profile icon: .
3. Select **My Profile**.

The My Profile page appears.

4. In the **User Access Token** section, click  in the row of the token you want to remove and select **Delete**.

The Delete User Access Token dialog box appears.

5. Click **Delete** to confirm.

## Enabling license term fulfillment

BOM Managers, and other users with the appropriate [role](#), manage the fulfillment status for a license term

using the *Project Version's Legal tab*.

By default, this tab is disabled. System Administrators must enable this tab for it to appear to these users.

**Note:** Enabling the **Legal** tab is a global setting. Once enabled, all project versions will display a **Legal** tab.

### To display the Legal tab

1. Log into Black Duck with the System Administrator role.



2. Click the expanding menu icon () and select **Administration**.

The Administration page appears.

The screenshot shows the 'Administration' page with the following sections:

- Jobs**: View current and past jobs that were executed.
- User Management**: Manage Black Duck users and user roles. (5 Users / 495 Remaining)
- Product Registration**: Set the product registration key and view the licensed features. (Using 2.55 MB out of unlimited Codebase storage. Expires on November 20, 2019 6:01 AM)
- Group Management**: Organize users, assign roles and add groups to projects. (0 Groups)
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- Extensions**: Extensions provide additional behaviors and functionality to Black Duck.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- Custom Fields Management**: Create and implement custom fields within Black Duck.

3. Select **System Settings**.

The screenshot shows the 'System Settings' page under 'Administration'. The 'Logo' section displays the current logo ('SYNOPSYS') and provides instructions for uploading a new one. The 'System Logs' section allows downloading a zip file of current logs. The 'Legal Tab Visibility' section contains an 'Enable' button. The 'Security Risk Configuration Ranking' section shows a drag-and-drop interface for reordering risk configurations (BDSA 2.0, NVD 2.0, BDSA 3.0, NVD 3.0) and includes a warning about policy violations. A 'Save' button is located at the bottom right.

4. Click **Enable** located in the **Legal Tab Visibility** section to display the tab.

Click **Disable** to remove the **Legal** tab.

## Customizing the logo

You can replace the logo that appears in the header of the user interface:



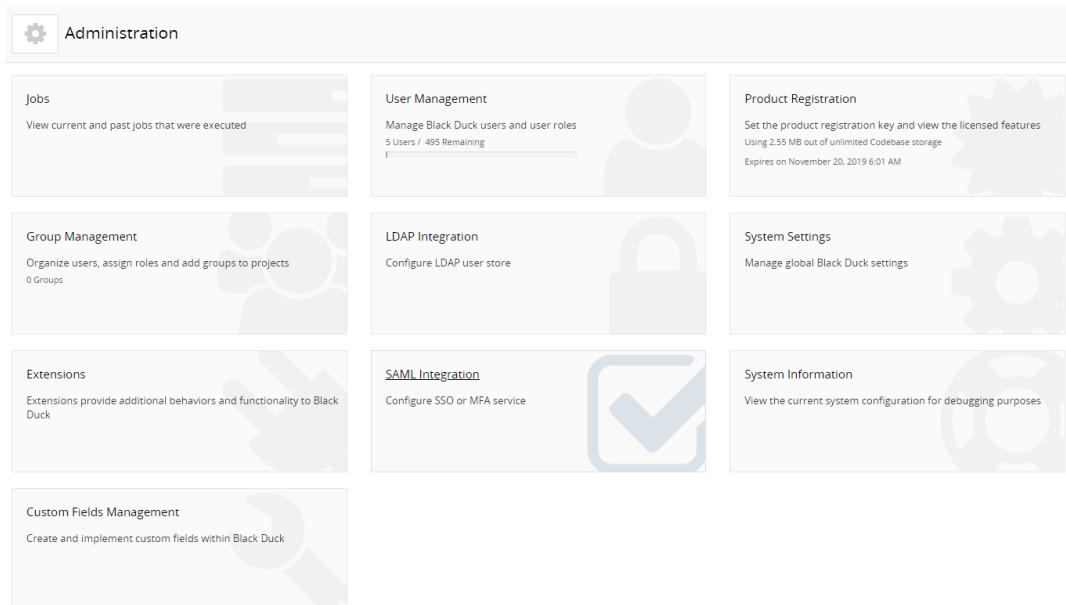
The maximum height for a logo is 37px.

### ⚙️ To change the logo

1. Log in to Black Duck with the System Administrator role.

2. Click the expanding menu icon (☰) and select **Administration**.

The Administration page appears.



3. Select **System Settings** to display the System Settings page.

**Logo**  
The dimensions will be constrained to a height of 34px and a maximum width of 150px.  
 [Upload logo](#)

**System Logs**  
If you need to troubleshoot any issues, you can start by downloading a zip file containing the current logs. [Download Logs \(.zip\)](#)

**Legal Tab Visibility**  
Enables the Legal tab in Project Versions so project team members can check off license terms as fulfilled as part of their workflow. Note: this also requires license administrators to indicate which terms require fulfillment. See the documentation on license terms for more information. [Enable](#)

**Security Risk Configuration Ranking**  
Drag and drop the security risk configuration priority order  
Warning: Changing the order of the security risk configuration will result in revised security risk calculations for all project version BOMs and may result in new policy violations. These calculations may take a considerable amount of time to complete.  
 BDSA 2.0  
 NVD 2.0  
 BDSA 3.0  
 NVD 3.0 [Save](#)

4. Click **Upload logo** and select the file.

The new logo appears in the header.

**Note:** To redisplay the Synopsys logo, select **Restore to original**. This link appears on the page after you customize the logo.

## Accessing log files

You may need to troubleshoot an issue or provide log files to Customer Support.

Users with the System Administrator role can download a zipped file that contains the current log files.

## ⚙️ To download the log files from the Black Duck UI

1. Log in to Black Duck with the System Administrator role.



2. Click the expanding menu icon (☰) and select **Administration**.

The Administration page appears.

3. Select **System Settings**.

The System Settings page appears.

The screenshot shows the 'System Settings' page under the 'Administration' menu. At the top, there is a 'Logo' section with a placeholder for a logo image and a 'Upload logo' button. Below it is a 'System Logs' section with a 'Download Logs (.zip)' button. Further down is a 'Legal Tab Visibility' section with an 'Enable' button. The bottom half of the page is titled 'Security Risk Configuration Ranking' and contains a list of items with priority icons: BDSA 2.0, NVD 2.0, BDSA 3.0, and NVD 3.0. A 'Save' button is located at the bottom right of this section.

4. Click **Download Logs (.zip)**.

It may take a few minutes to prepare the log files.

## Viewing jobs

You can view all the jobs in the system if you need to troubleshoot an issue and determine if a process ran.

Note that any job older than 30 days is purged from the list.

Possible jobs are:

Job Name	Description
BomAggregatePurgeOrphansJob	Deletes any BOM data not associated with a project version.
BomVulnerabilityNotificationJob	Creates vulnerability notifications for users.
CodeLocationDeletionJob	Deletes scan and code location matches for a code location.
ComponentDashboardRefreshJob	Refreshes the information shown on the Component Dashboard.
HierarchicalVersionBomJob	Creates and updates the <a href="#"><u>hierarchical version BOM</u></a> .

Job Name	Description
KbComponentUpdateJob	Processes any changes made to the Black Duck KB related to a particular component and determines if any components on the local Black Duck server are affected by the changes. If they are, the changes will be applied locally and vulnerability data for that component is recomputed if necessary.
KbReleaseUpdateJob	Processes any changes made to the Black Duck KB related to particular component version releases and determines if any component versions on the local Black Duck server are affected by the changes. If they are, the changes

Job Name	Description
	will be applied locally and vulnerability data for that component version is recomputed if necessary.
KbVulnerabilityUpdateJob	Updates Black Duck with the latest Black Duck KB NVD vulnerability data.
KbVulnerabilityBdsUpdateJob	Updates Black Duck with the latest Black Duck KB BDSA vulnerability data.
LicenseTermDataPopulatorJob	Updates Black Duck with the latest Black Duck KB license term data.
LicenseTermFulfillmentJob	Applies license term fulfillment requirements to all BOMs.
PolicyRuleModificationBomComputationJob	Computes version BOMs affected by changes to policy rules.
ReportingDatabaseTransferJob	Migrates Black Duck

Job Name	Description
	data to the Black Duck reporting warehouse.
ScanAutoBomJob	Manages matching and BOM computation for a scan.
ScanGraphJob	Processes incoming scans and prepares them for a BOM computation.
ScanPurgeJob	Deletes old scans or BOM imports.
SnippetScanAutoBomJob	Manages the matching process for snippet scan signatures.
VersionBomComputationJob	Manages version BOM computation.
VersionLicenseReportJob	Creates the <a href="#">Notices File report</a> .
VersionReportJob	Creates the <a href="#">Project Version report</a> .
VulnerabilityRemediationReportJob	Creates the <a href="#">Vulnerability Remediation Report</a> .

Job Name	Description
VulnerabilityReprioritizationJob	Recomputes all BOMs with the new vulnerability priority setting.
VulnerabilityStatusReportJob	Creates the <a href="#">Vulnerability Status Report</a> .
VulnerabilitySummaryFetchJob	Locates missing CVSS 3.0 data.
VulnerabilityUpdateReportJob	Creates the <a href="#">Vulnerability Update Report</a> .

### To view jobs

1. Log in to Black Duck with the System Administrator role.



2. Click the expanding menu icon ( ) and select **Administration**.

The Administration page appears.

3. Select **Jobs** to display the Jobs page which is divided into a **Summary** and **Details** section.

- The **Summary** section lists the number of successes, failures, and jobs in progress for each job for the number of days you are retaining logs (30 days by default).
- The **Details** sections lists each job and provides information on the status, duration and start time for the job.

Use the **Related to** column to select links for some jobs so that you can view what a job is related to.

- Use the filter to view the information in the table by job status or job type.

# Appendix A: Understanding how to search in Black Duck

You can search for the following categories of information within Black Duck:

- **Projects:** These are projects that your company's developers are coding. For Black Duck to index basic project information for search, someone must create one or more projects. Once Black Duck has finished indexing the new project information, that project will be available for search.

**Note:** The information that you provide about your projects is not shared outside of your company.

- **Components:** Components that comprise projects can be searched by the component name. The Black Duck KB contains over a decade's worth of data, including programming languages used, lines of code, user ratings from the Open Hub community, and data about commits and committers.
- **Vulnerabilities:** Security vulnerabilities that impact components and, as a result, which may impact your projects can be searched for by BDSA number, CVE number, or another identifier, for example, "CVE-2014-0160", "Heartbleed", and so on.

When you execute a search using the **Search** field, Black Duck returns results that may include projects, components, and vulnerabilities.

The screenshot shows the Black Duck search results interface. At the top, there is a search bar with a magnifying glass icon and the text "Search Results". Below the search bar, there is a "Search Types" section with three categories: "Projects" (0 results), "Components" (451 results), and "Vulnerabilities" (207 results). The "Components" section is expanded, showing a list for "Apache Tomcat". The entry for Apache Tomcat includes the component name, version (1696), and usage across 12 projects. It also lists associated tags: web, java, http\_server, servlet, http, jsp, web\_services, and apache. A link "View all 451 results ..." is provided. Below this, the "Vulnerabilities" section is shown, listing a specific entry for "BDSA-2017-1201 (CVE-2017-12616)". This entry includes the BDSA number, publication date (Oct 5, 2017), severity (Medium), and a brief description: "Apache Tomcat is prone to information disclosure vulnerability due to improper security constraints implementation." To the right of this entry, there are three performance metrics: Base (5.0), Exploitability (10.0), and Impact (2.9).

A list of entries for each category is displayed, along with a count of the results within each category. You can further refine these results to only include information of a particular type, such as vulnerabilities, by clicking the **View All # Results** link within a category or by using the search facets in the left pane of the search results.

## Searching for projects and components

You can search for:

- Your company's own projects that have been added to Black Duck
- [The Black Duck KnowledgeBase \(KB\)](#), which is a comprehensive database of open source software (OSS) components.

To search for a project or component in Black Duck, type your search terms in the Search field and press **Enter**.

Black Duck displays results that meet your search criteria.

The screenshot shows the Black Duck search interface. At the top, there is a search bar with a magnifying glass icon and the placeholder text "Search Results". Below the search bar, there is a sidebar titled "Search Types" with three categories: "Projects" (0), "Components" (451), and "Vulnerabilities" (207). The "Components" category is selected. The main content area displays a search result for "Apache Tomcat". The result includes the component name, version (1696), and the number of projects it is used in (12). It also lists its binary distribution (Binary distribution of Apache Tomcat) and tags (web, java, http\_server, servlet, http, jsp, web\_services, apache). Below this, there is a link to "View all 451 results ...". Another section below shows "207 Vulnerabilities found" for the same component, listing one specific advisory (BDSA-2017-1201) with details like publication date (Oct 5, 2017), severity (Medium), and impact scores (Base 5.0, Exploitability 10.0, Impact 2.9). There is also a link to "View all 207 results ...".

**Tip:** If your results include components, the description indicates the number of projects that use this component. indicates that there are known vulnerabilities associated with versions of this component,

## How project/component searching works

Black Duck conducts a literal text string search of:

- **Project/component names:** Black Duck matches exact project/component names and matches segments of project/component names. For components in the Black Duck KB, component descriptions are part of the information provided by the Black Duck Open Hub. For projects in Black Duck, project descriptions are part of the basic project information maintained by project team members.

During search indexing of project/component names, Black Duck uses white space and non-alphanumeric characters to determine segments. This means that `New_Project`, `New-Project`, and `New! Project` are all treated as two words (two separate searchable segments), but `NewProject` is treated as a single word (a single searchable segment) and would not be returned when you search for the word "new."

## Specifying your search terms

Because Black Duck searches for the literal text string, you cannot use wildcards (\*) in your search. Black Duck assumes that if special characters are used in a project name they may have special meaning, so:

- If you put quotation marks (" ") around your search terms, Black Duck searches include your search terms inside the quotation marks.
- If you use an asterisk (\*) in your search terms, Black Duck searches for an asterisk in exactly the same location in the project names.
- If you include a percent sign (%) in your search terms, Black Duck searches for the % symbol in project names.

**Tip:** If your search query returns many results, for example a large number of Black Duck KB project results, and you want to see only projects, [use the search filters](#) to narrow your results.

## Filtering your search results

The search results initially list all the projects, components, and vulnerabilities that matched your search terms. You can progressively narrow your search results using the available filters.

As you apply filters to the search results, the numbers that appear on the filters reflect the number of items that still meet your filtered search criteria. This can help you determine when you have reached a manageable number of results to use.

### Project and components filters

- **Component Source:** This filter defines the source of this components. Possible values are **Black Duck Custom Component** or **Black Duck KB**.
- **Primary Language:** This information is only available for projects or components in the Black Duck KB. This is the primary language in which the project or component is written. This information is provided by the Black Duck Open Hub.

**Tip:** This filter displays the list of available languages in descending order of frequency of use in the projects or components. The list is type-ahead enabled, so you can quickly narrow the list by typing the first few characters of the language name you want to filter on.
- **Commit Activity:** This information is only available for projects or components in the Black Duck KB and represents the trending commit activity level for the OSS project or component over time. This information is provided by the Black Duck Open Hub.
- **Tags:** This information is available for all projects or components that have tags applied to them to provide additional metadata about the project or component. For projects in the Black Duck KB, this information is provided by the Black Duck Open Hub. For projects in Black Duck, these [tags are assigned and managed](#) by project team members.

**Tip:** This filter displays the list of available tags in descending order of frequency of use in the projects or components. The list is type-ahead enabled, so you can quickly narrow the list by typing the first few characters of the tag name you want to filter on.

## Vulnerabilities filters

The filters available for vulnerabilities are:

- Base score
- Exploitability score
- Impact score
- Published year

## Searching for security vulnerabilities

You can search Black Duck for published security vulnerabilities. Searching by vulnerability is an efficient way to:

- Identify if a new or existing security vulnerability affects a component that is included in your projects.
- Review the severity of the security vulnerability to determine if remediation is required.
- Enter [remediation details for one or more of your projects](#).

### To search for security vulnerabilities

1. Log in to Black Duck.
2. Enter vulnerability search criteria into the **Search** field. Common search criteria include the following:
  - The Common Vulnerabilities Exposure (CVE) name.
  - The CVE or BDSA identifier.
  - Keywords that are included in the vulnerability description.
3. Press **Enter**.

Black Duck displays results that meet your search criteria. By default, the search returns results from the National Vulnerability Database (NVD). If you are licensed for BDSA, the search returns results from both NVD and BDSA. You can refine the results by using the facets on the left: select **Vulnerabilities** under **Search Types** to view facets/filters you can select to [narrow the search results](#).

**Note:** The user interface displays only those facets that are applicable to the information that is available to you. If the components returned by the search do not contain a specific attribute or classification, the respective facet does not appear.

4. Select its title to view more information on the vulnerability. You can view National Vulnerability Database (NVD) information by selecting the [CVE number](#) or view Black Duck Security Advisory (BDSA) information by selecting the [BDSA number](#).

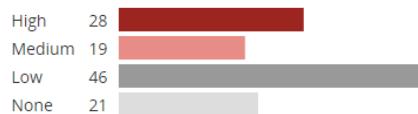
**Note:** You can print the vulnerability details using the native print functionality of your web browser.

## Filtering the data shown in tables

Risk graphs and/or advanced filters are available on some pages to help you filter the information shown in tables.

## Risk Graphs

Some pages display risk graphs which indicate the number of items in the table shown below the graphs that have that security, license, and/or operational risk at that severity level.



- **Critical/High or High** risk: 100% red
- **Medium** risk: 50% red
- **Low** risk: 100% gray
- **None**: 50% gray

Select a severity label/graph to filter the table to show only those items that have a specific type and severity of risk.

## Advanced Filters

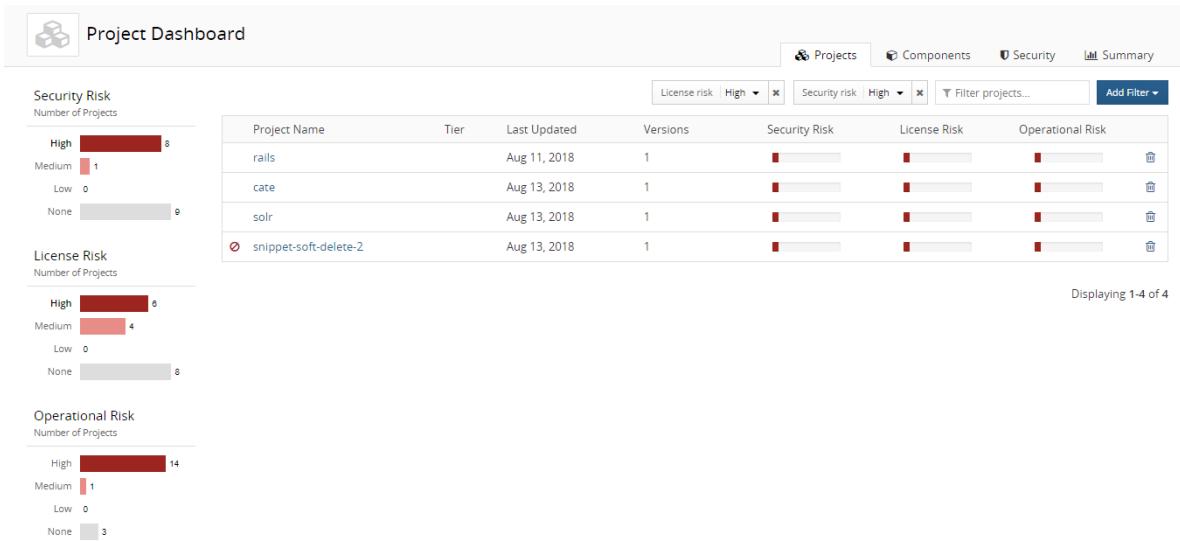
On some pages, tables have an advanced filter feature that provides an easy way to view the data. This feature provides you with a clear view of the filters that are being applied to the table.

### To use advanced filters

1. Click **Add filter ▾** to view the filters for this table.
2. Select a filter. The filter you selected appears at the top of the table.



3. Select values for this filter and click **OK**. If you select more than one value, Black Duck displays items that match *either* value.
4. Optionally, select additional filters. If you select more than one type of filter, Black Duck displays items that match *all* filters.
5. Black Duck displays items that match all selected filters. For example:



6. Click X to remove a filter.

**Tip:** For pages that have advanced filters and risk charts, advanced filters work with these charts so you can also select and/or clear multiple risk filters by using the graphs. Selecting a risk level displays filter (X) icon in the graph and a field appears above the table displaying the values you selected. Click X in the risk graphs or X in the filter fields to clear the filter.

# Appendix B: Working with notifications

Notifications alert you when:

- Security vulnerabilities are published or updated against components that are included in one or more of your projects.
- Actions you perform affect the vulnerabilities in BOM components, such as:
  - Editing, adding, or removing components which have vulnerabilities.
  - Unmapping a scan from a project.
  - Rescanning code or a Docker image.
  - Ignoring or no longer ignoring a component.
  - Modifying file(s) so that they are matched to a different component.
- Components have violated a policy.
- Policy violations have been overridden.
- Components no longer violate a policy.
- You are approaching or are exceeding your [code size limit](#).

## Viewing all notifications



1. Open the notifications list by selecting .
2. Select **See All Notifications** located at the bottom of the list.  
The All notifications page appears.
3. By default, the page is filtered. Select **Add Filter** to [change these settings](#).

## Viewing more information

Viewing a notification removes it from the notifications list. Click **View all notifications** to view notifications that you have either viewed or deleted from the list.

### To view more information on security vulnerabilities and BOM component adjustments

1. Log in to Black Duck.
2. Open the notifications list by selecting and select **See All Notifications**.
  - Select a component version to open the **Security** tab of the Black Duck KB [component version page](#).

- Select a vulnerability record (such as CVE-2017-1234) to view the vulnerability details page for that security vulnerability.

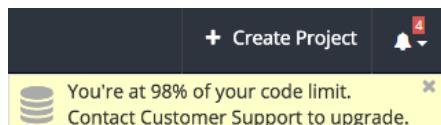
⚙ To view more information on policy violations and overrides

1. Open the notifications list by selecting  and select **See All Notifications**.
2. Select a policy violation or a policy violation override to open the BOM page.

Users with the appropriate role can [override a policy violation](#) or [remove a policy violation that was overridden](#).

⚙ To view more information on code limits

The notification automatically appears at the top of the page when you are close to exceeding your [code size limits](#):



1. Open the notifications list by selecting .
2. Select **See All Notifications** located at the bottom of the list.  
The All notifications page appears.
3. To upgrade your code limit, contact Customer Support.

## Deleting notifications

1. Log in to Black Duck.
2. Open the notifications list by selecting .
3. Click **X** located in the upper right corner of a notification.

# Appendix C: About the Tools page

The Tools page is divided into these sections: **Downloads**, **Documentation** and **Developer Tools**.

## Downloads

This section of the Tools page provides links for Synopsys Detect (Desktop), Synopsys Detect CLI, and Legacy Downloads (the Signature Scanner).

- **Synopsys Detect (Desktop)**. Select the link to download the Mac OS X, Linux, or Windows version of Synopsys Detect (Desktop) from Google Cloud Storage. This tool scans your file system and generates a Bill of Materials (BOM).

Synopsys Detect (Desktop) client systems must meet the following requirements:

- Mac OS X. Version 10.9 or later. A minimum of 8 GB of RAM.
- Windows. Windows 7 or later. A minimum of 8 GB of RAM.

- **Synopsys Detect (CLI)**. Select the link to go to the Integrations Documentation page. From here, select **Synopsys Detect** to view download instructions and documentation for Synopsys Detect, a command line interface (CLI) that integrates with your build jobs to identify package manager dependencies as well as file system matches.

- **Legacy Downloads**. Select **Toggle All** to view the download links for the Linux, Mac OS X, or Windows CLI of the Signature Scanner, a tool to scan your file system and generate a BOM.

The Signature Scanner client systems must meet the following requirements:

- Linux. A minimum of 8 GB of RAM on the supported operating systems.
- Mac OS X. Version 10.9 or later. A minimum of 8 GB of RAM.
- Windows. Windows 7 or later. A minimum of 8 GB of RAM.

**Note:** The Signature Scanner is included with Synopsys Detect. We recommend you use Synopsys Detect to create a more complete Bill of Materials.

## Documentation

This section of the Tools page provides the following links:

- **Synopsys Software Integrity Community**. Clicking this link on the Tools page displays an [online resource for customer support, solutions, and information](#).
- **Synopsys Software Integrity, Customer Education**. Clicking this link on the Tools page opens a web

page that lists the catalog of classes.

**Note:** Access to the classes requires a training subscription.

- **Help.** Clicking this link on the Tools page displays the home page of the online help.

## Developer's Tools

This section of the Tools page provides the following links:

- **REST API Developers Guide.** Clicking this link on the Tools page displays the documentation available for the REST APIs as well as test the request and response of each API.
- **Black Duck Open Source Integrations.** Clicking this link on the Tools page opens the Black Duck Integrations page on [GitHub](#).

## Appendix D: Integrating Protex with Black Duck

Black Duck provides the ability to import Protex BOMs into Black Duck.

This feature gives Protex users the ability to use Black Duck to view and manage security vulnerabilities in their existing BOMs. It also provides Black Duck customers the ability to use the greater language support that is available in Protex.

There are three basic methods for importing Protex data into Black Duck:

- Components Only

This option is akin to the mapping that is currently done between Protex and Code Center – the BOM in Code Center only has the list of component/versions and not any of the associated file mappings. Similarly, using this technique to import a Protex BoM into Black Duck only preserves the components/versions. As only the component and version information is being mapped, there is less of a performance impact compared to the other methods.

This is the default output of the [Protex BOM Tool](#).

- Components and Files

This method maps the existing Protex BOM into a comparable BOM within Black Duck, preserving the identified components and the associated file mappings. Note that the resultant BOM in Black Duck is only as good as the identifications that were made manually in Protex, therefore, it is important that the people doing the identification work in Protex pay attention to the versions they are selecting for each component. Historically, for license compliance, having the correct version for a component was less important as licenses rarely changed between versions of the same component. However, for security risk, having the correct version for a component is very important as vulnerabilities are mapped to specific versions of components. Therefore, if you will be using Protex with Black Duck, it is important for you to be aware of this as you are doing your identification work.

The [Protex BOM Tool](#) can export a BOM from Protex and import it directly into Black Duck, mapping it to a specific project and release. Or, the tool can be used to export the BOM into a JSON file which can be later imported into Black Duck using the Black Duck UI.

**Note:** The component and version identifiers are different between the Protex KB and Black Duck KB. During the import process, Black Duck application will remap each BOM component/version from its Protex KB identifier to the corresponding Black Duck KB identifier. Not all components will have a KB identifier and will therefore not be reflected in Black Duck BOM, for example, custom or local components, or components that do not have a corresponding ID in Black Duck KB.

An [audit log](#) lists the Protex components and licenses that were mapped to Black Duck and provides details around any items that were unable to be mapped between the Protex KB and the Black Duck KB.

To use this method, include the **--include-files** parameter when running the [Protex BOM Tool](#).

**Note:** Due to the amount of file information contained in many Protex BOMs, there may be some performance impact both during the import process and when navigating to UI pages involving these projects.

- File Metadata > Black Duck Signatures

This method takes the original file metadata that was captured during the Protex scan and imports it into Black Duck such that Black Duck treats it as if the scanner was scanning the files and directories directly. A new Black Duck BOM is created which will likely be different from the original Protex BOM. As the scanner takes advantage of the full context of file and directory information, it can identify the correct version information for a component. Thus, in many cases you will see more accurate version information using this method and get better results for security use cases.

To use this method, use the **--dryRunWriteDir** and **--include-files** parameters when running the [Protex BOM Tool](#).

**Note:** For the best results using this approach, archives need to be expanded when running the Protex scan. This may produce longer scan times for some projects depending on the number of archives in the project.

## Understanding the Protex BOM integration process

The process for integrating a Protex BOM into Black Duck is:

1. Log in to Black Duck.
2. [Download](#) and install the Protex BOM tool. The Protex BOM tool provides several different ways by which you can import a Protex BOM into Black Duck.
3. [Export](#) the Protex BOM file.

**Note:** Only projects assigned to the user whose credentials are supplied in the tool will be available for export.
4. If you do not use the Protex BOM tool to import the BOM into Black Duck or map the BOM to a project, then use Black Duck UI to:
  - [Import](#) the Protex BOM file into Black Duck.

- [Map](#) the Protex BOM to a Black Duck project.

Once the Protex BOM is imported and mapped, you can [view and manage](#) its contents as you manage any other BOM in Black Duck.

## Requirements

To import a Protex BOM into Black Duck, you must be running:

- Protex version 7.1.2 or higher
- Black Duck version 2.3 or higher

Note the following:

- Imported Protex data is processed in Black Duck and the Black Duck KB through a new KnowledgeBase matching service. This service converts all Protex Suite IDs to Black Duck KnowledgeBase IDs.
- Matched and unmatched file information is available in Black Duck. The following table lists the Protex discovery type, usage, and the corresponding Black Duck match type:

Protex Discovery Type	Protex Usage	Black Duck
*	Component	Exact
Code Match	File	Exact
Code Match	Snippet	Partial
String Search	Snippet	Partial
Dependency	Snippet	Dependency

- The following Protex BOM components are not available in Black Duck:
  - Custom Components
  - Custom Licenses

These components are dropped during the import process.

- If you use Protex to make any changes to the Protex BOM, the changes persist when the Protex BOM is reimported to Black Duck: only the changes made in the imported Protex BOM are updated in the Black Duck project.

## Downloading the Protex BOM tool

The Protex BOM tool command line interface (CLI) client is packaged as a .zip file.

Enter the following URL to download the zip file: <https://<Black Duck hostname>/download/scan.protex.cli.zip>

After you unzip the Protex BOM tool client file, use it to [import a Protex BOM](#) into Black Duck.

## Exporting a Protex BOM

The Protex BOM tool provides several different ways by which you can import a Protex BOM into Black Duck.

For example, you can use the tool to:

- [Export the Protex BOM from the Protex server](#) and import it into Black Duck using the tool.
- [Export the Protex BOM from the Protex server to a file](#) and manually import it through Black Duck UI.
- [Import a Protex BOM file](#) into Black Duck using the tool.

The tool does not require any specific role in Black Duck or in Protex to use the tool.

By default, the tool outputs component/version data only; use the **--include-files** parameter to include file data.

The Protex BOM tool has these parameters:

Parameter	Description
<b>-?, --help</b>	Shows help for this tool.
<b>-A, --dest &lt;host: port&gt;</b>	Specifies Black Duck host name and port.
<b>-P, --hub-project &lt;name&gt;</b>	Specifies the name of the Black Duck project to which you want to map this Protex BOM. If the project does not exist, the tool creates the project and maps the BOM to the project.
<b>-R, --hub-release &lt;name&gt;</b>	Specifies the name of the Black Duck project version to which you want to map this BOM. If the version does not exist, the tool creates the version and maps the BOM to this version of the project.  If you specify <b>hub-project</b> , <b>hub-release</b> is optional. If you do not specify <b>hub-release</b> , the version defaults to the value of the <b>release</b> parameter.
<b>-S, --secure-dest</b>	Uses HTTPS to connect to the server hosting Black Duck. If you do not include this parameter, HTTP is used.
<b>-U, --dest-user &lt;user&gt;</b>	Specifies the username to log in to the Black Duck server.
<b>-W, --dest-password</b>	Forces the tool to prompt you for a password for the Black Duck server. When the tool runs, a prompt appears requesting the password for the specified user.  For non-interactive use, set the BD_HUB_PASSWORD environment variable with the password for the Black Duck server. If you set this variable, the <b>dest-password</b> parameter is optional: the tool prompts the user for the password; it does not check the password against the variable.
<b>-a, --address &lt;host:port&gt;</b>	Specifies the Protex host name and port.
<b>-r --release&lt;name&gt;</b>	Specifies a value to use to identify the current state

Parameter	Description
	<p>of the Protex BOM. You can use any value for &lt;name&gt;.</p> <p>Use this parameter to enable viewing multiple "versions" of a Protex BOM in Black Duck. Click <a href="#">here</a> or more information.</p>
<b>--list-projects</b> <SearchQuery>	<p>Lists all Protex project identifiers for all projects to which you have access, one per line, on the console.</p> <p>&lt;SearchQuery&gt; is optional.</p> <p>To export multiple Protex projects, use the output from this parameter to write a script which iterates over multiple project identifiers.</p>
<b>--data</b> <path>	Specifies the path to the Protex BOM file.
<b>--output</b> <path>	Writes the BOM out to a file or directory with the project name.
<b>-p, --project</b> <id or name>	Specifies the Protex project identifier or project name.
<b>-s, --secure</b>	Uses HTTPS to connect to the server hosting Protex. If you do not specify this parameter, HTTP is used.
<b>-u, --user</b> <user>	Specifies the username to log in to the Protex server.
<b>-w, --password</b>	<p>Forces the tool to prompt you for a password. When the tool runs, a prompt appears requesting the Protex server password for the specified user.</p> <p>For non-interactive use, set the BD_PROTEX_PASSWORD environment variable with the password for the Protex server. If you set this variable, the <b>password</b> parameter is optional.</p>
<b>-V, --version</b>	Shows the version information of this tool.
<b>-v, --verbose</b>	Sets the logging level to verbose.
<b>--dryRunWriteDir</b> <dryRunWriteDir>	Specifies the directory to which the Protex BOM Tool outputs a JSON file with the original file metadata used for scanning.
<b>--debug</b>	Shows debug output.
<b>--include-files</b>	Includes the Protex code tree and match details.

By default, the tool generates the Protex BOM to standard out, if you don't specify an output (file) or use the tool to import the BOM to Black Duck.

## Exit Statuses

The possible exit statuses are:

- **0:** SUCCESS. The export completed successfully.
- **1:** FAILURE. Generic failure.
- **64:** USAGE. The command to run the tool was used incorrectly, for example, with the wrong number of arguments or a bad syntax.
- **65:** DATA\_ERROR. The input data was incorrect.
- **66:** NO\_INPUT. An input file (not a system file) did not exist or was not readable.
- **67:** NO\_USER. The specified user does not exist.
- **68:** NO\_HOST. The specified host does not exist.
- **69:** UNAVAILABLE. A service is unavailable.
- **70:** SOFTWARE. An internal software error has been detected.
- **71:** OS\_ERROR. An operating system error has been detected.
- **72:** OS\_FILE. A system file does not exist, cannot be opened, or has some sort of error, for example a syntax error.
- **73:** CANNOT\_CREATE. An output file cannot be created.
- **74:** IO\_ERROR. An error occurred while doing input/output on a file.
- **75:** TEMPORARY\_FAILURE. Temporary failure,
- **76:** PROTOCOL. The remote system returned something that was "not possible" during a protocol exchange.
- **77:** NO\_PERMISSION. You did not have sufficient permission to perform the operation.
- **78:** CONFIGURATION. Something was found in an unconfigured or misconfigured state.
- **79:** NO\_REGISTRATION. Registration to Black Duck or Protex was not valid.

## Viewing multiple versions of a Protex BOM in Black Duck

When you import a Protex BOM, Black Duck creates a file (labeled a BOM File in Black Duck UI) that is associated with that BOM. In Black Duck, a BOM File can only be mapped to a single project and version – if you import the Protex BOM again, the new file is added to the existing BOM File.

You may want to view multiple versions, or snapshots, of a Protex BOM in Black Duck. Although Protex does not have project versions, you can use the **release** parameter in the Protex BOM tool to denote a snapshot of your Protex BOM. When you use the **release** parameter, Black Duck creates a new BOM file for that snapshot. You can then map that BOM file to a different project or to a different version of a project. This gives you the flexibility to create multiple snapshots of a single Protex BOM and view them at the same time in Black Duck.

Note that if you specify a value for **release** that has already been used for that Protex BOM, a new BOM File is not created. Instead, the new file will be added to the existing BOM File.

## Examples

The following are examples of using the Protex BOM tool:

- [Exporting the Protex BOM and importing it into Black Duck using the export tool](#)
- [Exporting the Protex BOM to a file](#)
- [Importing a Protex BOM from a file](#)

Note that the examples show the required parameters.

## Using the Protex BOM tool to map the Protex BOM

In these examples, you have the option of using these parameters to specify the Black Duck project and version that this BOM should be mapped to:

- **hub-project <name>**
- **hub-release <name>**

If you specify a value for the **release** parameter and wish to use the tool to map the Protex BOM, the **hub-release** parameter is optional: if you do not specify a value for **hub-release**, Black Duck project version defaults to the value of **release**.

If you do not specify **hub-project** and **release** or **hub-release**, you must [map the Protex BOM](#) using the Black Duck UI.

## Exporting the Protex BOM and importing it into Black Duck using the export tool

This example exports the Protex BOM from the Protex server and imports it into Black Duck using the tool.

1. Open a command prompt.
2. Go to the directory where the tool is installed and run the following command:

### Linux example

```
./scan.protex.cli.sh --address <host:port> --user <user> --password --  
project <id> --output <path> --dest-address <host:port> --dest-user <user>  
--dest-password
```

### Windows example

```
scan.protex.cli.bat --address <host:port> --user <user> --password --  
project <id> --output <path> --dest-address <host:port> --dest-user <user>  
--dest-password
```

## Exporting the Protex BOM to a file

This example exports the Protex BOM from the Protex server to a JSON file. You then need to use the Black Duck UI to [manually import the file](#).

1. Open a command prompt.
2. Go to the directory where the tool is installed and run the following command:

### Linux example

```
./scan.protex.cli.sh --address <host:port> --user <user> --password --  
project <id> --output <path>
```

### Windows example

```
scan.protex.cli.bat --address <host:port> --user <user> --password --  
project <id> --output <path>
```

### Importing a Protex BOM from a file

This example imports a Protex BOM file into Black Duck using the tool.

1. Open a command prompt.
2. Go to the directory where the tool is installed and run the following command:

### Linux example

```
./scan.protex.cli.sh --data <path> --dest-address <host:port> --dest-user  
<user> --dest-password
```

### Windows example

```
scan.protex.cli.bat --data <path> --dest-address <host:port> --dest-user  
<user> --dest-password
```

## Importing the Protex BOM file

If you output the Protex BOM to a file, you need to import the file into Black Duck.

### To import a Protex BOM file

Status	Name	Scan Size	Last Updated	Mapped to
✓	cowboy-mac#/Users/cowboy/node_modules/get-stdin	3.58 KB	Aug 13, 2018	testScan2_2
✓	cowboy-mac#/Users/cowboy/node_modules/lodash	823.26 KB	Aug 13, 2018	testScan1_2
✓	FitNesseScanCodeLocation_2	184.28 KB	Aug 13, 2018	testScan1_2
✓	hubui_10518	148.89 MB	Aug 13, 2018	testScan1_2

1. Log in to Black Duck and click the expanding menu (≡) icon.
2. Select **Scans**.
3. In the Scans page, click **Upload Scans**.
4. Use the Upload Files dialog box to locate the Protex BOM file
5. Click **Close**.

If you did not use the Protex BOM tool to automatically map the BOM to a project, use Black Duck to [map the file to a project](#).

## Mapping or unmapping a Protex BOM

You must use Black Duck to map the Protex BOM to a project if you did not use the Protex BOM tool to do so.

### ⚙️ To map a Protex BOM to a project



1. Log in to Black Duck and click the expanding menu (☰) icon.
2. Select **Scans**.

The screenshot shows the Black Duck 'Scans' page. At the top, there's a header with a 'Scans' icon and the word 'Scans'. On the right, it shows '479.03 MB / ∞ Unlimited'. Below the header is a toolbar with '+ Upload Scans' and 'Delete' buttons, and filters for 'Filter scans...' and 'Add Filter'. The main area is a table with columns: Status, Name, Scan Size, Last Updated, and Mapped to. There are four rows of data:

Status	Name	Scan Size	Last Updated	Mapped to
✓	cowboy-mac#/Users/cowboy/node_modules/get-stdin	3.58 KB	Aug 13, 2018	testScan2_2
✓	cowboy-mac#/Users/cowboy/node_modules/lodash	823.26 KB	Aug 13, 2018	testScan1_2
✓	FitNesseScanCodeLocation_2	184.28 KB	Aug 13, 2018	
✓	hubui_10518	148.89 MB	Aug 13, 2018	

3. If you did not use the Protex BOM tool to import the BOM, [use Black Duck's UI to import it](#).
4. Click and select **Map to Project** in the row of the Protex BOM you want to map.
5. In the Map Scan dialog box, start typing the name of a project to progressively display matches.
6. Select the project version to which you want to map the Protex BOM.
7. Click **Save**.

Black Duck displays the name and version of the project to which you mapped the Protex BOM. Select the link to open the [BOM page](#).

### ⚙️ To unmap a Protex

You can remove the mapping of a Protex BOM.



1. Log in to Black Duck and click the expanding menu (☰) icon.
2. Select **Scans**.
3. Click and select **Unmap from Project** in the row of the Protex BOM that you want to remove the mapping.
4. Click **Remove** to confirm.