



User Guide

Version 2021.8.0



This edition of the *User Guide* refers to version 2021.8.0 of Black Duck.

This document was created or updated on Thursday, August 26, 2021.

Please send your comments and suggestions to:

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

Copyright © 2021 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

Chapter 1: Getting started with Black Duck	1
Logging in to Black Duck	2
Scanning your code and mapping scans to projects	4
Mapping scans to projects	4
Administrative tasks	4
Importing a Protex BOM	4
Chapter 2: About Black Duck - Binary Analysis	5
Chapter 3: Understanding Component Scanning	6
Supported languages	7
Individual file matching	8
ISO files	8
Supported package managers	8
Scanning tools	8
Using Synopsys Detect (Desktop)	8
Downloading and installing Synopsys Detect (Desktop)	9
Configuring Synopsys Detect (Desktop)	10
Certificates	16
Scanning options	16
Creating a scan file	18
Managing scans	18
Uploading scan files to Black Duck	19
Viewing uploaded scans	20
About Rapid Scanning	21
Using the Signature Scanner	24
Signature Scanner client requirements	24
Downloading and installing the Signature Scanner CLI	25
Downloading the Signature Scanner CLI	25
Installing the Signature Scanner CLI	25
Defining your version of JRE for the Signature Scanner	25
Running a component scan using the Signature Scanner command line	26
Specifying the password	33
About package management files	33
Exit Statuses	33

Examples	34
Reducing the number of parameters entered on the command line for the Signature Scanner	36
Minimum Scan Interval	37
Changing the minimum scan interval	37
Accessing the Black Duck server via a proxy	37
Running an offline component scan using the Signature Scanner	39
Using certificate-based authentication with the Signature Scanner	40
Examples	40
Defining the scan name	42
Specifying names for BOM or JSON files	42
Approved signature lists	42
Resolving memory issues	43
About duplicate BOM detection	44
About component dependency duplication	45
Viewing component dependency duplication	45
Changing how duplicate dependencies are displayed	46
About snippet matching	46
Snippet scanning process	47
Uploading source files for snippet matching	49
Reviewing snippet matches	49
Retaining partial snippet identifications	49
Snippet matches and Vulnerabilities	50
Modifying the default maximum snippet file size	50
Snippet extensions	50
Resolving proxy errors	55
About custom scan signatures	55
Understanding the custom scan signature process	56
Defining default scanning levels	56
Creating custom scan signatures	56
Associating custom components to custom scan signatures	58
Disabling custom scan signatures	58
Hosting location for Synopsys Detect	58
Specifying the hosting location of Synopsys Detect	59
Chapter 4: Managing scans in the Black Duck UI	60
Filtering scans	60
Uploading a scan file using the Black Duck UI	61
Browsing scans	62
Mapping a scan to a project	64
Removing a scan from a project	65
Deleting a scan	66
Exporting a scan file	67

Viewing an audit log for a BOM file	68
Chapter 5: Understanding projects in Black Duck	71
Creating a project	73
Deleting a project	74
Watching projects	75
Viewing a list of your watched projects	75
Decreasing the number of watched projects	77
Watching projects	78
Cloning projects	79
Enabling cloning	79
Updating project information	80
Managing project team membership	83
Managing tags	88
About project versions	89
Creating a new version of a project	90
Updating project version information	92
Cloning project versions	93
Enabling cloning	94
Deleting a project version	94
About project version phases	95
About archived project versions	95
Chapter 6: Viewing a project version's BOM	97
Understanding the information in a project version's BOM	97
Header information	98
Risk graphs	100
Data Table	101
About the hierarchical BOM	107
Reviewing the contents of a BOM	108
Managing comments	110
Adding a comment	110
Viewing a comment	110
Editing a comment	111
Deleting a comment	111
Managing files associated with BOM components	111
Accessing the Source tab	111
About the Source tab	112
Modifying matches	114
Identifying unmatched files	114
Validating matched files	115
Resetting files	115
Deleting files from a BOM	115

Comparing BOMs	115
Printing a BOM	118
Viewing issues in a project	119
Viewing component versions with encryption	121
About Linux distributions in Black Duck	122
Viewing Linux distributions in Black Duck	122
Chapter 7: Editing a BOM	123
Applying edits to all versions of a project	123
Persistent edit examples	124
Enabling or disabling persistent edits for a project	126
Manually adding a component to a BOM	127
Excluding a component from a BOM	128
Deleting a component from a BOM	130
Removing components from a BOM	131
Ignoring a component in a BOM	133
Adjusting the component and/or component version in a BOM	134
Editing an origin or origin ID	135
Modifying licenses in a BOM	136
Selecting the license term fulfillment status	138
Editing license text in the BOM	139
Managing subprojects	141
Reviewing snippet matches	143
Snippets in the BOM	143
Viewing snippet matches in the Source tab	144
Retaining partial snippet identifications	148
Chapter 8: Managing components	149
About custom components	151
Managing custom components	152
Creating custom components	152
Viewing custom component information	153
Editing custom components	153
Deleting custom components	154
Managing custom component versions	154
Creating additional versions for a custom component	155
Viewing the projects where a version is used	155
Editing custom component versions	156
Deleting a custom component version	157
About Black Duck KnowledgeBase components	158
Understanding the component information available from the Black Duck KB	158
About the Overview tab	159
About the Settings tab	160

Understanding the component version information available from the Black Duck KB	161
Modifying KB components	164
Resetting a Black Duck KB component's values	166
About the KnowledgeBase Feedback Service	167
Setting or modify a component's status	168
Changing the status of components and/or versions	168
Chapter 9: Viewing risk in Black Duck	171
Dashboards	171
Project version pages	175
Viewing the health of your projects	176
Viewing your dashboards	179
Viewing dashboards	179
About the Watching and My Projects dashboards	180
About saved searches dashboards	182
Viewing overall risk for all projects	192
Understanding the types of project risk	193
Viewing overall risk at the project version level	194
Understanding the types of component risk	196
Chapter 10: About security risk	198
Security risk levels	198
Suggested work flow	199
Defining the default security risk calculation	199
Viewing the security vulnerabilities of your projects, project versions, and component versions	202
Related vulnerabilities	202
Viewing project version vulnerabilities	202
Security Risk graph	203
Components list	204
Filters	204
Vulnerabilities table	204
Analyzing the impact of a vulnerability	206
Vulnerability impact analysis process	206
Viewing reachable vulnerabilities	207
Viewing vulnerability details	209
Black Duck Security Advisories	209
Overview tab	211
Affected Projects tab	212
Technical tab	213
CVE References tab	214
Settings tab	215
CVE record	215
Overview tab	215

Affected Projects tab	216
References tab	217
Settings tab	217
Remediating security vulnerabilities	217
Remediating a vulnerability	218
Getting remediation guidance for components with security vulnerabilities	220
Managing global remediation for a vulnerability	222
Setting a global remediation for a vulnerability	222
Clearing a global default remediation status	223
Viewing all vulnerabilities with global remediation	224
Chapter 11: Managing policies	225
About the policy process	225
Viewing policy rules	226
Viewing policy rule violations	226
Overriding violations	227
Removing policy overrides	227
Default policy rules	227
Creating a policy rule	228
Policy conditions	228
Creating a policy	237
Creating policy rules for approved or barred items	238
Pre-approved policy rule examples	239
Barred policy rule example	240
Editing a policy rule	240
Copying a policy rule	240
Deleting a policy rule	241
Disabling or enabling a policy rule	241
Overriding policy violations	242
Removing policy overrides	243
Chapter 12: Managing open source licenses	245
Suggested work flow	246
About license families	247
Managing license families	249
About custom license families	251
Creating custom license families	251
Editing custom license families	252
Deleting custom license families	254
Viewing licenses	255
Viewing license text	258
Viewing license use	259
Determining license risk	261

Estimated licenses	262
Default license risk	262
License risk - by usage	262
License risk by license family	267
Managing deep license data	270
Enabling deep license data	271
Reviewing deep license data	273
Detecting embedded licenses	277
Supported file extensions/ file names	279
License detection process	280
Reviewing embedded licenses	280
Modifying licenses for a component	284
Reverting BOM-level license edits	287
About custom licenses	288
Creating custom licenses	288
Editing a custom license	290
Deleting custom licenses	292
About license terms	293
Suggested work flow	294
License terms process	294
Viewing license terms	296
About license term fulfillment	298
Defining fulfillment when viewing terms for a license	299
Creating license terms	301
Managing license term categories	305
Associating a license term to a license	308
Editing a custom license term	313
Deleting a license term	314
Deprecating or removing the deprecation status of a custom license term	315
Removing a license term	317
Deactivating a KnowledgeBase term	321
Restoring a KnowledgeBase license term	326
Editing a KnowledgeBase license	331
Restoring the original text and family of a KnowledgeBase license	333
Viewing detected copyright statements	334
Supported file extensions/ file names	336
Copyright detection process	337
Reviewing copyright data	337
Managing copyrights	341
Viewing and managing copyright statements	341
Creating custom copyright statements	343

Editing custom copyright statements	344
Deactivating copyright statements	344
Activating copyright statements	344
Editing KnowledgeBase copyright statements	344
Reverting KnowledgeBase copyright statements	345
Updating KnowledgeBase copyright statements	345
Managing attribution statements	346
About license conflicts	347
Defining conflicts for customer license terms	348
Defining an incompatible term	348
Viewing incompatible terms	350
Deleting incompatible license terms	351
Enabling management of license term conflicts	352
Enabling or disabling license conflicts for a specific project	353
Managing project license conflicts	355
Viewing project license conflicts	355
Chapter 13: Running a report	358
Notices File report	358
Excluding a component or subproject from the Notices File report	360
Project version reports	361
Detail reports	361
Vulnerability reports	363
Vulnerability Remediation report	364
Vulnerability Status report	365
Vulnerability Update report	366
Deleting reports	368
Chapter 14: Managing Black Duck user accounts	369
Configuring password requirements	369
Creating a user account	372
Disabling a user	375
Converting a user account	376
Viewing a user's groups	379
Viewing a user's projects	381
Changing your Black Duck password	382
Changing user account information	384
Changing a user's password	386
Understanding roles	387
Global roles	387
Project roles	389
Project Group roles	391
Managing user roles	391

Viewing your roles	393
Black Duck user role matrix	394
Global roles by task	395
Project roles	401
Project Group roles	406
Managing the Project Manager role	410
Authenticating users with LDAP	412
Configuring secure LDAP	415
Obtaining your LDAP information	416
Importing the server certificate	417
About locked out user accounts	419
Chapter 15: Managing groups in Black Duck	420
Viewing your groups	420
Creating groups	421
Managing group information	423
Managing group projects	425
Managing group roles	429
Adding a member to a group	431
Removing a member from a group	435
Deleting groups	439
About Project Groups	441
Project Group Hierarchy	441
Creating a Project Group	442
Editing a Project Group	443
Editing the name or description	444
Adding project sub-groups	444
Removing project sub-groups	444
Moving a project group to a different project group	445
Moving another project group into the selected group	445
Adding a member to a project group	445
Removing a member from a project group	446
Editing a member's roles in a project group	446
Adding a user group to a project group	446
Removing a user group from a project group	446
Chapter 16: About custom fields	448
Viewing custom field information in the Black Duck UI	449
Creating a custom field	452
Activating or deactivating a custom field	455
Determining the order of custom fields shown in the UI	456
Editing a custom field	456
Deleting a custom field	457

Chapter 17: Other administrative tasks	459
Creating system announcements	459
Markdown language	462
Viewing project and project version audit information	463
Viewing product registration information	465
Updating your product registration	466
Managing your code size limits	467
Managing user access tokens	468
Enabling license term fulfillment	471
Customizing the logo	472
Accessing log files	474
Viewing jobs	474
Appendix A: Understanding how to search in Black Duck	479
Searching for projects	479
About the search results	480
Using search filters	482
Sorting the search results	482
Searching for components	482
Using search filters	483
About the search results	485
Sorting the search results	489
Searching for vulnerabilities	489
Using search filters	490
About the search results	491
Saving and managing search results	492
Saving search results	493
Editing saved searches	493
Renaming saved searches	493
Deleting saved searches	494
Filtering the data shown in tables	494
Risk Graphs	494
Advanced Filters	495
Appendix B: Working with notifications	496
Viewing notifications	496
Viewing more information	496
Hiding notifications	497
Appendix C: About the Tools page	498
Downloads	498
Black Duck Open Source Integrations	498
Community and Education	499
Appendix D: Integrating Protex with Black Duck	500

Understanding the Protex BOM integration process	501
Requirements	502
Downloading the Protex BOM tool	502
Exporting a Protex BOM	503
Exit Statuses	505
Viewing multiple versions of a Protex BOM in Black Duck	505
Examples	506
Importing the Protex BOM file	507
Mapping or unmapping a Protex BOM	508

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 Swarm	install_swarm.pdf	Contains information about installing and upgrading Black Duck using Docker Swarm.
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.
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.

The installation methods for installing Black Duck software in a Kubernetes or OpenShift environment are Synopsysctl and Helm. Click the following links to view the documentation.

- [Helm](#) is a package manager for Kubernetes that you can use to install Black Duck.
- [Synopsysctl](#) is a cloud-native administration command-line tool for deploying Black Duck software in Kubernetes and Red Hat [OpenShift](#).

Black Duck integration documentation is available 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>
- Phone: See the Contact Us section at the bottom of our [support page](#) to find your local phone number.

To open a support case, please log in to the Synopsys Software Integrity Community site at <https://community.synopsys.com/s/contactsupport>.

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.

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> or for help with Black Duck, select **Black Duck Tutorials** from the Help menu () in the Black Duck UI.

Synopsys Statement on Inclusivity and Diversity

Synopsys is committed to creating an inclusive environment where every employee, customer, and partner feels welcomed. We are reviewing and removing exclusionary language from our products and supporting customer-facing collateral. Our effort also includes internal initiatives to remove biased language from our engineering and working environment, including terms that are embedded in our software and IPs. At the same time, we are working to ensure that our web content and software applications are usable to people of varying abilities. You may still find examples of non-inclusive language in our software or documentation as our IPs implement industry-standard specifications that are currently under review to remove exclusionary language.

Chapter 1: Getting started with Black Duck

The Synopsys Software Integrity Group (SIG) offers a comprehensive suite of services and tools that support customers on their security journey. From customers just starting with security, to customers strengthening an established program, SIG has the expertise, skills, and products necessary for success.



Black Duck, a Software Composition Analysis (SCA) tool, helps with managing the supply chain of software, understanding the third-party components in use and minimizing risks from known vulnerabilities and licensing. Black Duck is a comprehensive solution for supply chain management, based primarily on source analysis.

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

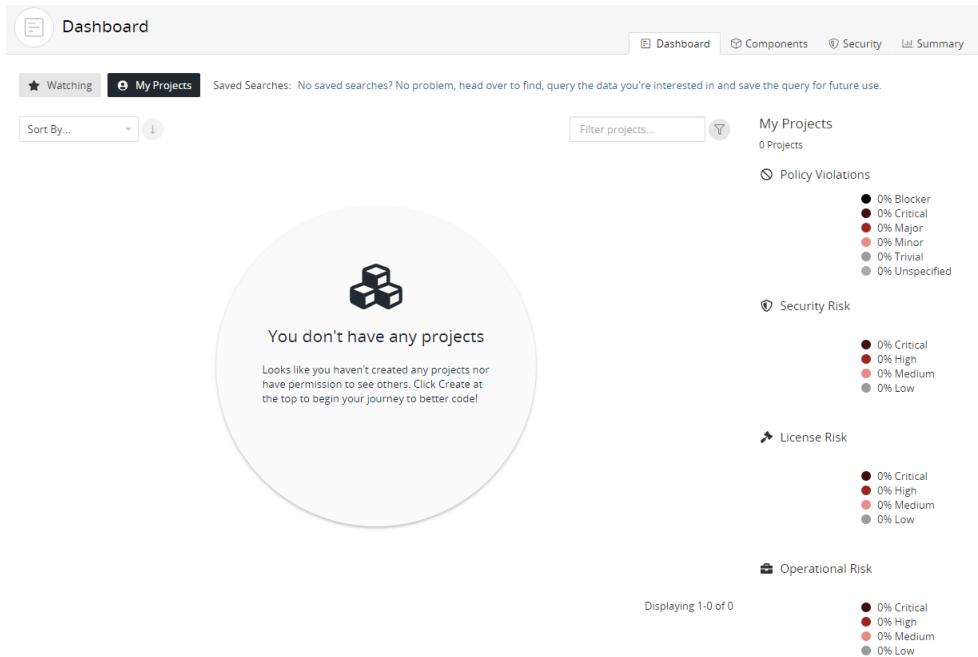
1. Using a browser, navigate to the Black Duck URL supplied by your system administrator. Typically, the URL is in the format `https://<server hostname>`.
2. Enter the username and password provided by your Black Duck administrator. Your password is case sensitive.

Note: If your administrator has enabled [password requirements](#) and your password does not meet the requirements, a dialog box appears notifying you that you must change your password. When updating your password, make sure that it meets the requirements, as listed in the dialog box. You will not be able to log in to Black Duck unless the password meets *all* requirements.

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 Dashboard 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 (specific [Dashboard](#) page or [Summary](#)) you viewed previously.

The Dashboard page has two default dashboards: the [Watching and My Projects dashboards](#). You can also create custom dashboards so that you can quickly view the project versions, component versions, or security vulnerabilities that are important to you: [search for projects](#), [components](#), and/or [security vulnerabilities](#) then [save the searches](#). Your saved searches appear on the Dashboard page.

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, [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.

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 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

For the current list of supported languages, refer to the list of supported languages shown in the [Synopsys Detect documentation](#).

Individual file matching

Individual file matching is the identification of a component based purely upon the checksum information of a single file. Prior to Black Duck 2020.2.0, for a small set of file extensions (.js, .apklib, .bin, .dll, .exe, .o, and .so), regular signature scanning matched files to components based upon a checksum match to the one file. Unfortunately, this matching was not always accurate and produced a fair amount of false positives that required you to spend additional effort reviewing and adjusting the BOM. Therefore, individual file matching is no longer the default behavior and instead is an optional capability as of the Black Duck 2020.2.0 release.

This may cause some components to drop off your BOM, which may or may not be desired. Therefore, in the Black Duck 2020.2.0 release, Synopsys provides parameters in the scanning tools so that you can re-enable individual file matching. Refer to the command line parameters for the [Signature Scanner CLI](#) and Synopsys Detect documentation for more information.

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

Refer to the [Synopsys Detect documentation](#) for a list of supported package managers.

Scanning tools

Download, install, and scan using one of the following tools:

- [Synopsys Detect](#). [Synopsys Detect](#) is the recommended scanning tool for Black Duck.
- [Synopsys Detect Desktop](#)
- [Command line](#) (CLI) version of the Signature Scanner.

Tip: Review the Scanning Best Practices Guide for information on the best practices for scanning.

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.

Be sure that your system meets the system requirements of Synopsys Detect.

- Click [here](#) for the system requirements for the latest version of Synopsys Detect.
- Click [here](#) for the documentation for previous versions of Synopsys Detect. Use this page to find the Synopsys Detect version and view the system requirements.

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.

If the Synopsys Detect (Desktop) does not open after installation and the following error message appears:

The SUID sandbox helper binary was found, but is not configured correctly.
Rather than run without sandboxing I'm aborting now. You need to make sure
that /opt/Synopsys Detect/chrome-sandbox is owned by root and has mode
4755.

your operating system does not support the Sandbox at the kernel layer. To run Synopsys Detect (Desktop) with the Sandbox disabled, enter the following at the command line:

```
synopsys-detect --no-sandbox
```

Command line options for Windows

- Unattended (silent) install for Synopsys Detect

```
./synopsys-detect-latest.exe /S
```

- Installing to a specific directory

```
./synopsys-detect-latest.exe /D=C:\directory
```

Installing the Linux version of Synopsys Detect (Desktop)

1. Download the executable from your Black Duck server, as described in the previous section.
2. Install Synopsys Detect (Desktop):

```
cd Downloads
```

To install on CentOS/RedHat:

```
sudo yum localinstall synopsys-detect-latest.rpm
```

To install on Ubuntu/Debian:

```
sudo apt install ./synopsys-detect-latest.deb
```

3. Change the permission of chrome-sandbox:

```
cd "/opt/Synopsys Detect"  
sudo chmod 4755 chrome-sandbox
```

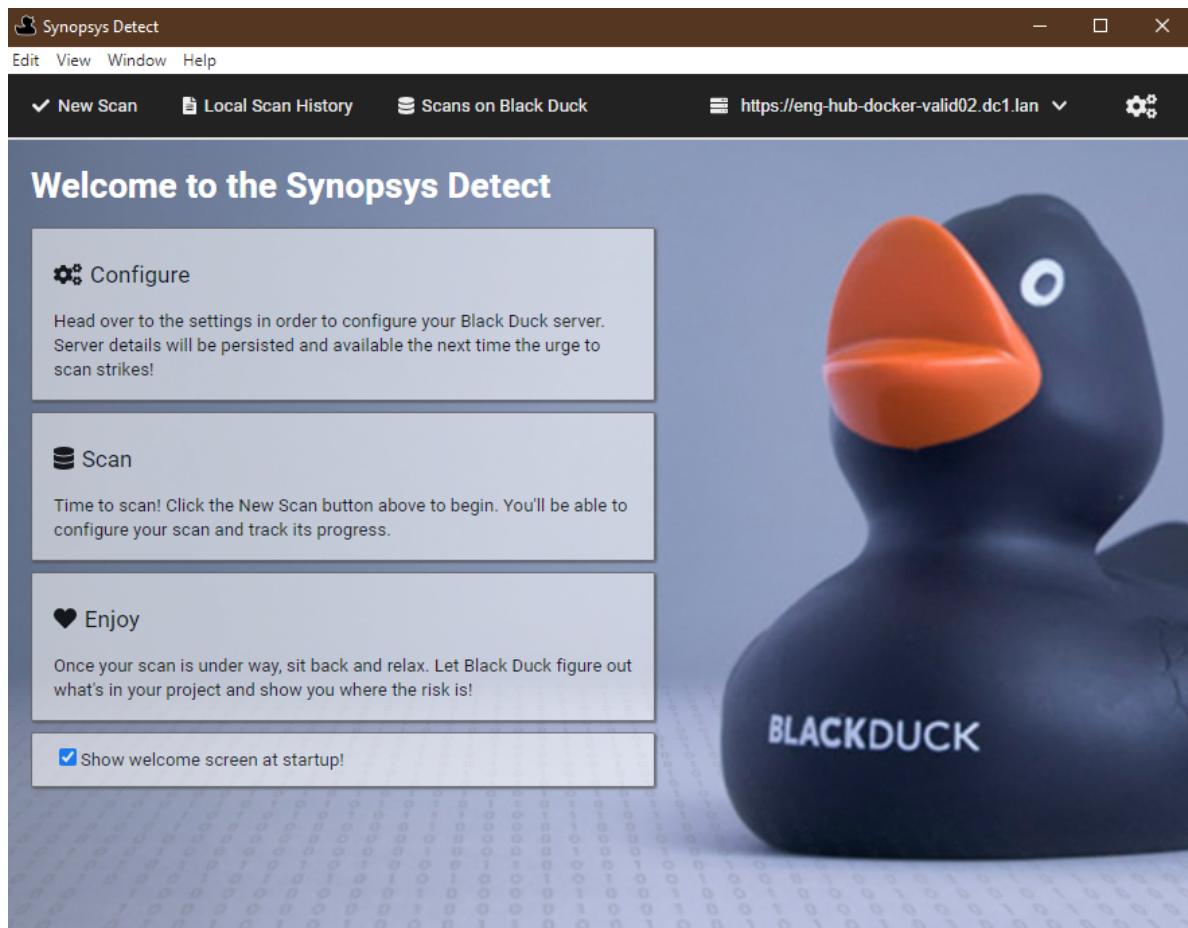
4. Run Synopsys Detect (Desktop):

```
./synopsys-detect --no-sandbox
```

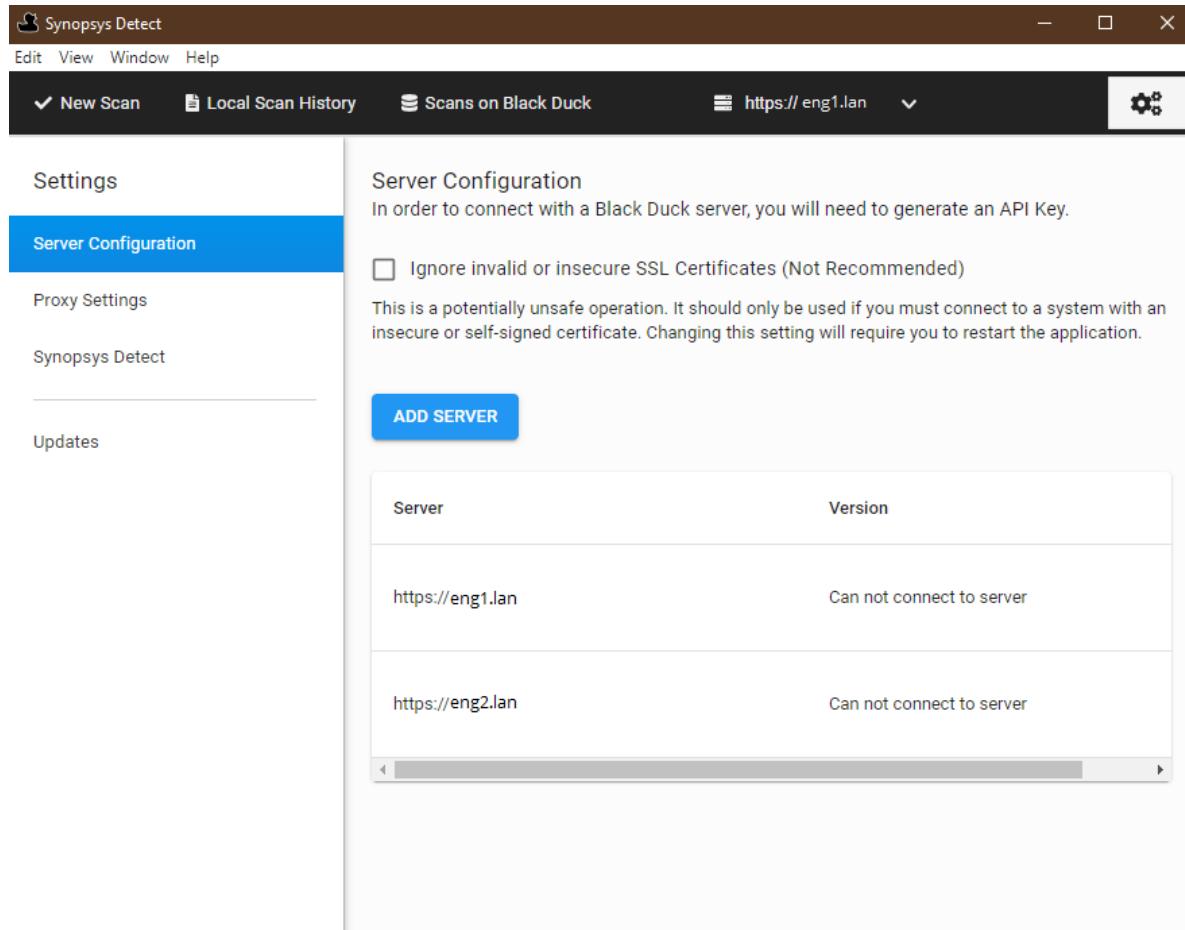
Configuring Synopsys Detect (Desktop)

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. Click the gear icon in the upper right corner to display the Settings page.



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

Black Duck server configuration

To add a server

1. Select the **Server Configuration** tab and click **Add Server**.

The Add Server dialog box appears.

Add Server

Black Duck Server URL

Generate New API Key [Already have a key?](#)

To generate a new API key, enter your username and password for your Black Duck server. The API key name is used to identify the key and must be unique.

API Key Name

Username *

Password *

2. 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.
3. Generate or enter an API key (user access token).
 - To generate a new API key:
 - a. Enter a key name, your username, and password.
 - b. Click **Create**.
 - To enter an API key:
 - a. Select **Already have a key?**.
 - b. Enter the API key in the field.
 - c. Click **Create**.
4. Click **Save**. Synopsys Detect (Desktop) connects to the Black Duck server and displays the version of Black Duck you are connected to.

To remove an API key

Removing the API key does not delete the key in Black Duck. It only removes it locally.

1. Select the **Server Configuration** tab.
2. Click **:** in the row of the server and select **Remove API Key**.
The Remove API Key dialog box appears.

3. Click **OK** to confirm.

To delete a configuration

1. Click **:** in the row of the server and select **Delete Configuration**.
The Delete Server Configuration dialog box appears.
2. Click **OK** to confirm.

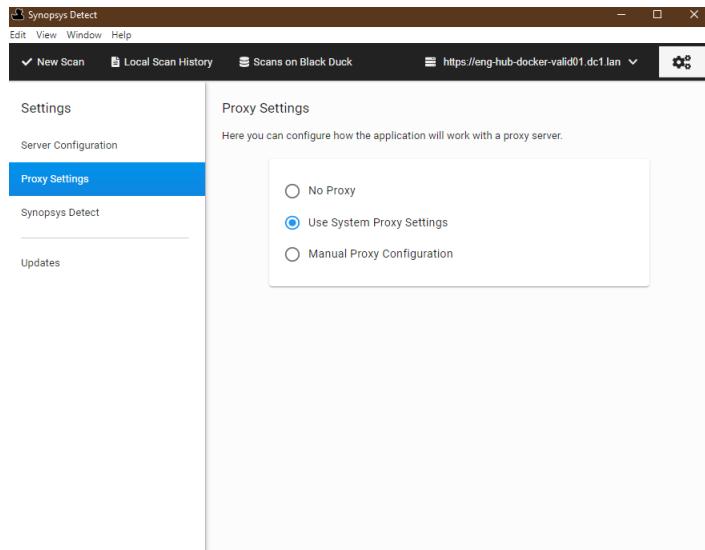
Proxy settings

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

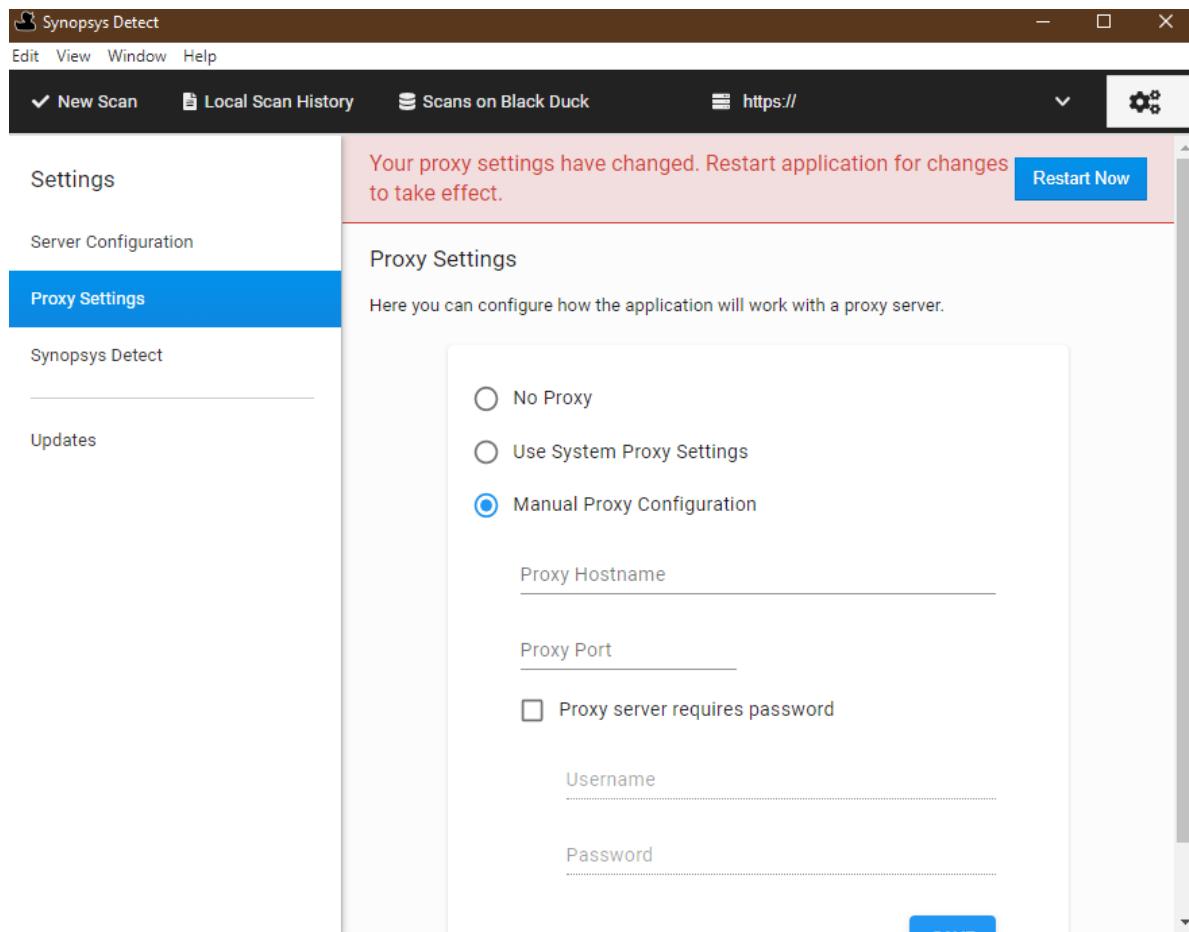
If you are required to manually enter your proxy settings or you do not require a proxy, you can modify these default settings.

To modify the default proxy settings

1. 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 **Synopsys Detect** 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, you can ignore invalid or insecure SSL certificates.

1. Click  located in the upper right corner display the Settings page.
2. Select the **Server Configuration** tab and select **Ignore invalid or insecure SSL Certificates**.
3. Restart the application.

Caution: This is a potentially unsafe operation. It should only be used if you must connect to a system with an insecure or self-signed certificate.

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 Matching** from the **Scan 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 **Local 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 **Local 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 archive (.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 **Local 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.

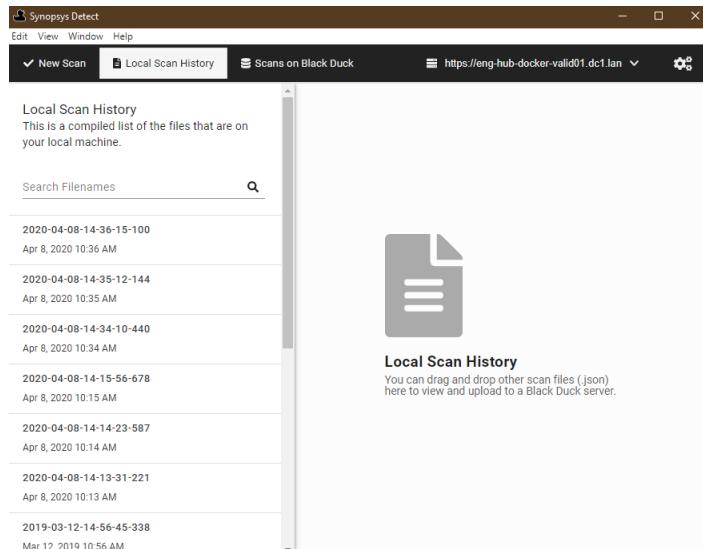
6. When the scan is complete, select the **Local Scan History** tab to view information on the completed scan.

Managing scans

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

1. Click **Local Scan History**.

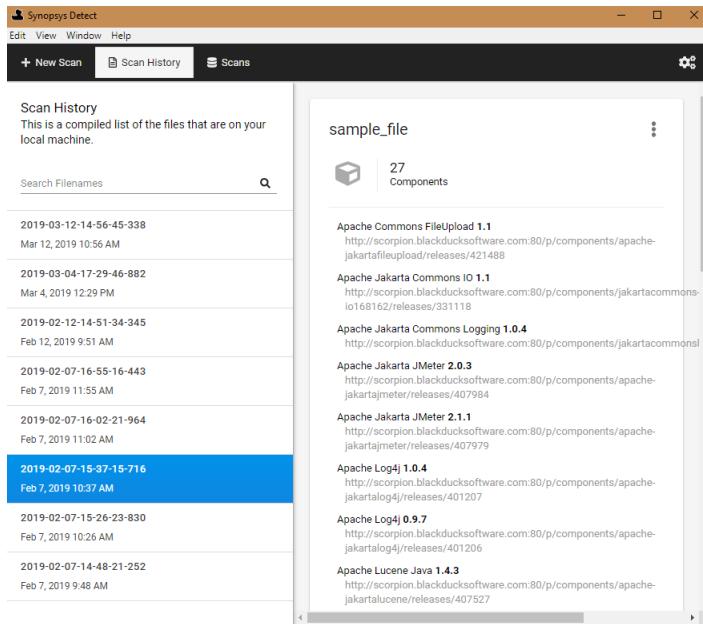
A list of scans on your local system appears in the left column of the tab.



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:

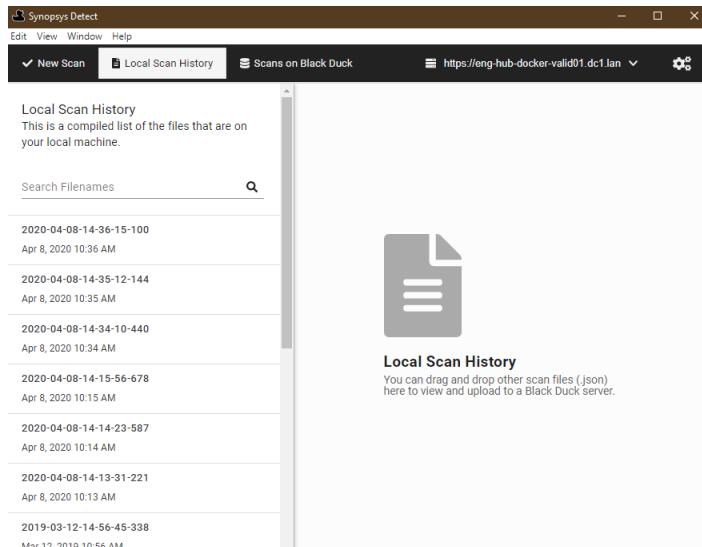


- View the location of the file on your system by clicking and selecting **Show File**.
- Upload the file, as described in the next section.
- Delete the scan by hovering over the scan name in the left column and clicking **Delete**. Click **Yes** to confirm.

Uploading scan files to Black Duck

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

1. Click Local Scan History.

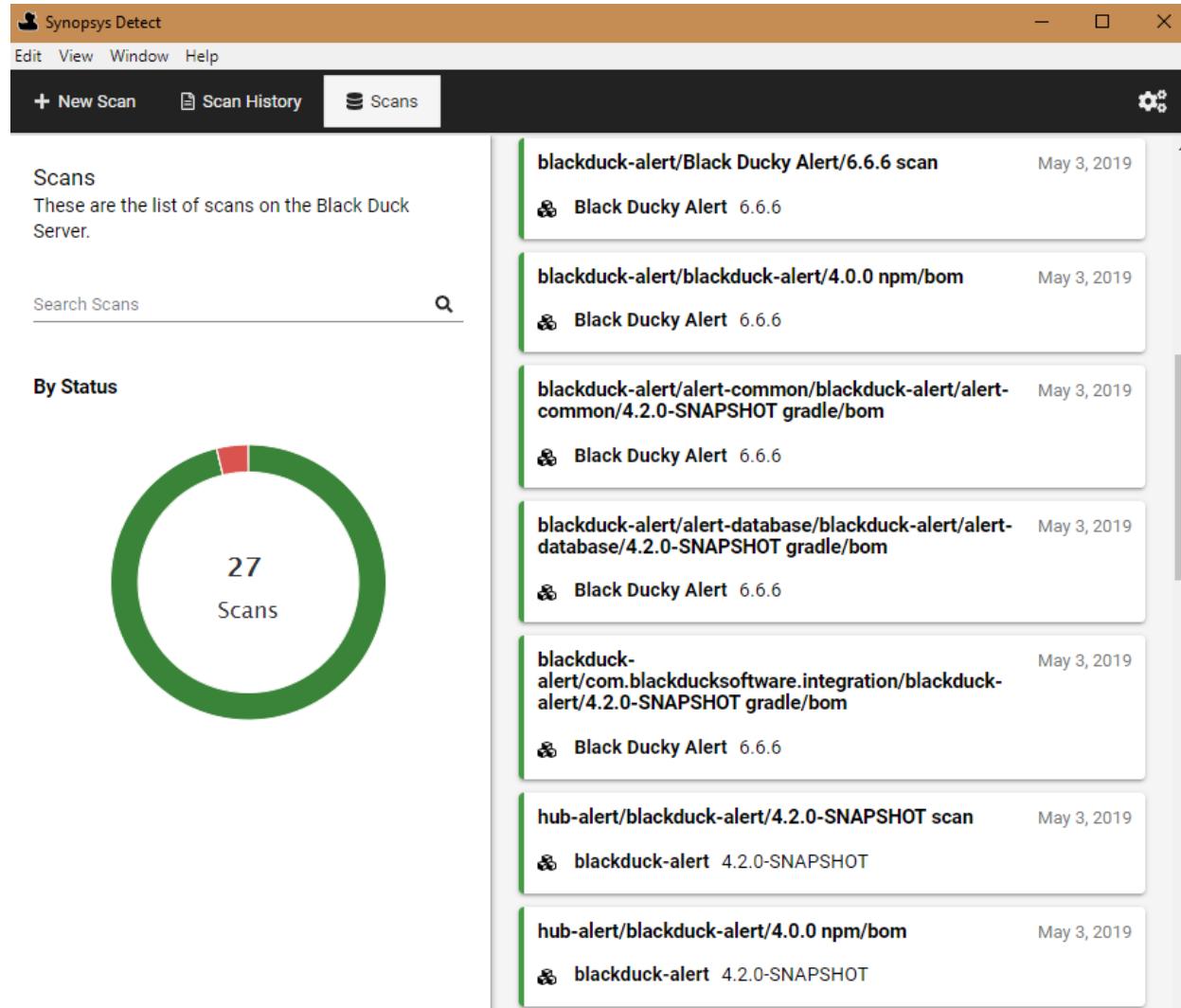


2. If the file is on your local system, you can drag and drop the scan file from your local machine to the **Scan History** tab.
3. Select the file to upload and click  in the upper right corner to display the file options.
4. Click **Upload Scan File to Black Duck**. The Upload Progress window appears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



Scans

These are the list of scans on the Black Duck Server.

Search Scans

By Status

27 Scans

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.

About Rapid Scanning

Black Duck's Rapid Scanning provides a way for developers to quickly determine if the versions of open source components included in a project violate corporate policies surrounding the use of open source. Using Synopsys Detect, Rapid Scanning quickly returns results as it only employs package manager scanning and does not interact with the Black Duck server database. Use Rapid Scanning when you need quick feedback and when persisting the data in Black Duck is not necessary.

Using Rapid Scanning enables you to run thousands of scans while eliminating the need to deploy additional instances of Black Duck. It provides you with actionable results (such as failing the build) that can be used without a project version or without access to Black Duck's user interface.

Results are printed as part of Synopsys Detect's normal log-style output. If there are policy violations, the output from Rapid Scanning lists the components with at least one policy violation, providing information such as the component name, version, component identifier, and policy name so that you can easily determine the component that is violating a policy.

Note that you can use a Synopsys Detect property to save the results as a .json file, as described below.

When a policy violation is found, the developer can:

- replace the triggering component with a component that does not violate policies.
- obtain an exception for the violation with the policy being modified to exclude the violating component.

For Rapid Scanning:

- Ensure that the user running Rapid Scanning has the ability to invoke Synopsys Detect.
- Ensure that your Black Duck system meets the system requirements needed to run Rapid Scanning.
- Create policy rules that will be triggered when a scan violates the rule.

Policy rules can apply to all projects or a subset of projects.

To use Rapid Scan to fetch all vulnerabilities regardless of policies, simply create a single policy,

setting the condition severity ≥ 0 .

Supported component conditions are:

Properties:

- Component (all versions or a specific version)
- Component Approval Status
- Component Modified
- Component Modification
- Component Purpose
- Component Usage
- Component Version Approval Status
- Match Type
- Newer Version Count
- Review Status
- Unknown Component Version

Licenses:

- Licenses (Declared)
- License Expiration Date (Declared)
- License Family (Declared)
- License Risk
- License Status (Declared)
- Unfulfilled License Terms

Operational:

- Component Release Date
- Commits in the past year
- Contributors in the past year

Vulnerabilities

- Critical Severity Vulnerability Count
- High Severity Vulnerability Count
- Medium Severity Vulnerability Count
- Low Severity Vulnerability Count
- Highest Vulnerability Score

Custom Fields:

- Component custom fields

- Component version custom fields
- Project custom fields
- Project version custom fields

Supported vulnerability conditions are:

- Overall Score
- CWE IDs
- Solution Available
- Workaround Available
- Exploit Available
- Reachable from Source
- Remediation Status

All other policy conditions are silently ignored.

Note that the severity of the policy rule determines how Synopsys Detect reacts to violations. Synopsys Detect treats violations of Blocker and Critical policies as errors; if any such violations are found, Synopsys Detect prints error messages and terminates with a non-zero exit code (so that builds can be failed if desired). For all other policy severities, Synopsys Detect treats violations as warnings; Synopsys Detect will print warning messages, but they alone will not cause Synopsys Detect to terminate with a non-zero exit code.

- Use Synopsys Detect 7.0 and later.
- Enable Rapid Scanning in Synopsys Detect by including these properties:
 - `--detect.blackduck.scan.mode="RAPID"`
 - `--detect.bom.aggregate.name=aggregated.bdio`

Refer to the [Synopsys Detect](#) documentation for more information about these properties.

- If `--detect.cleanup=false` is set, the raw JSON response from the Black Duck server will be saved as `<DETECT_OUPUT_PATH>/runs/<timestamp>/scan/<project_name>_<project_version>_BlackDuck_DeveloperMode_Result.json`.
- Any other Synopsys Detect properties that affect package manager scans are also applicable to Rapid Scanning.
- If a project and/or version is specified, but does not exist, policies will still be evaluated.
- Rapid Scanning will not create projects if they do not already exist, which is the opposite of what occurs when running other types of scans.
- For projects with complex builds, and especially when such projects use Gradle, the runtime of Rapid Scanning is frequently dominated by the runtime of the package manager when it is invoked to report the project's dependencies.
- A new job, `CollectScanStatsJob`, collects scan statistics which are shown on the **usage: rapid scan completion** section on the System Information page.
- Caching is used extensively. If any of the following values are changed, it may be up to 15 minutes for those changes to be reflected in Rapid Scanning results:

- Component Approval Status
- Component Custom Fields
- Component Modified
- Component Modification
- Component Purpose
- Component Usage
- Component Version Approval Status
- Component Version Custom Fields
- Match Type
- Project Custom Fields
- Project Name
- Project Version Name
- Project Tags
- Project Version Custom Fields
- Review Status
- Unfulfilled License Terms

You can override the Rapid Scanning caching interval by setting the value in the `blackduck.rapidscan.policy.cache.interval.mins` environment variable in the `blackduck-config.env` file.

- If a code location was previously scanned with a traditional scan where Synopsys Detect created a project for it, a subsequent rapid scan of the same code location will be associated with that project even if no project was explicitly provided.
- The following default values are used in evaluations if the project version is not known or the component is not found in the BOM.
 - Dynamically Linked for Component Usage
 - Not Reviewed for Review Status
 - File Dependency for Match Type
 - False for Component Purpose, Component Modified, Component Modification, and Component Terms Unfulfilled

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 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, from the Control Panel, click **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
-?, --help	Shows help for this tool.
<scan_path>	Path to the file directory location or archive that you want to scan.

Parameter	Description
--BinaryAllowedList <file extensions>	Use to create approved signature lists . <ul style="list-style-type: none"> • --BinaryAllowedList <i>x, y, z</i> where <i>x, y, z</i> are the approved file extensions for SHA-1 (binary) files.
--SourceAllowedList <file extensions>	<ul style="list-style-type: none"> • --SourceAllowedList <i>a, b, c</i> where <i>a, b, c</i>, are the approved file extensions for clean SHA-1 (source code) files.
--cloneFrom <version>	<p>Specifies the name of an existing project version to use as a clone.</p> <p>To clone a project version, use the:</p> <ul style="list-style-type: none"> • --project parameter to specify the project you wish to clone from. • --release parameter to specify the new project version. • --cloneFrom parameter to specify the project version to use as a clone. <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>
--context <context>	Additional URL context. Use this parameter, for example, if the X-Forwarded-Prefix header is being specified in a proxy server/load balancer configuration.
--copyright-search	Enables copyright text detection .
--dryRunReadFile <data directory>	Specifies the directory, including the file name, from a dryRun scan and posts the scan to the Black Duck server.
--dryRunWriteDir <data directory>	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.
--exclude <pattern>	Excludes a directory or several directories from scanning.
--exclude-from <Filename>	<p>The scanner automatically excludes these directories and the contents of these directories:</p> <ul style="list-style-type: none"> ■ CVS ■ .svn ■ .git ■ .hg ■ .bzr ■ __MACOSX <p>The scanner automatically excludes files named:</p> <ul style="list-style-type: none"> ■ .cvsignore ■ .git ■ .gitignore

Parameter	Description
	<ul style="list-style-type: none"> ■ .gitattributes ■ .gitmodules ■ .hgignore ■ .hgsub ■ .hgsubstate ■ .hgtags ■ .bzrignore ■ vssver.scc ■ .DS_Store <p>To exclude other directories, use --exclude to exclude a single directory; --exclude-from to specify a file that lists directories that should be excluded.</p> <p>Exclusion guidelines:</p> <ul style="list-style-type: none"> ■ Leading and trailing forward slashes are required. <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"> ■ Directory names cannot contain double asterisks (**). ■ 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. ■ You cannot exclude archives or contents within archives. <p>There are two additional methods you can use to exclude directories from scanning:</p> <ul style="list-style-type: none"> ■ Create an <code>ignore</code> file located in the <code>\$HOME/config/blackduck</code> directory. <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> <p>Lines in the <code>ignore</code> file must have the path from source root to the ignored directory, may have multiple subdirectories, and must have leading and trailing forward slashes (/).</p> <p>Create one of the following environment variables, as shown here, configured for Linux or Mac OS X:</p> <ul style="list-style-type: none"> • <code>export JAVA_TOOL_OPTIONS=" -Duser.home=<path> "</code> <p>The <code>.ignore</code> file must be located here: <code><user.home>/config/blackduck/ignore</code></p> <ul style="list-style-type: none"> • <code>export XDG_CONFIG_HOME=<path></code> <p>The <code>.ignore</code> file must be located here: <code>\$XDG_CONFIG_HOME/blackduck/ignore</code></p>

Parameter	Description
	<ul style="list-style-type: none"> • <code>export JAVA_TOOL_OPTIONS="-Dblackduck.scan.excludesFile=<path>/<file>"</code> <p>The <code>.ignore</code> file can be in any location and have any name.</p> <p>For Windows systems, use the Control Panel to access the Advanced System Settings dialog box to create the environment variables.</p> <ul style="list-style-type: none"> ■ Create individual <code>.bdignore</code> files which can be located in any directory. <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 directory that has subdirectories you want to exclude.</p> <p>You must also follow the exclusion guidelines as described above when using either of these methods.</p> <p>Tip: Use the <code>debug</code> parameter when excluding directories to ensure that the scanner visited and excluded the directory.</p>
<code>--host <host></code>	Server hosting the Black Duck installation.
<code>--individualFileMatching <option></code>	<p>Individual file matching is the identification of a component based purely upon the checksum information of a single file. By default, individual file matching is disabled. To enable individual file matching, select one of the following options:</p> <ul style="list-style-type: none"> • source. Performs individual file matching only on files with this extension: <code>.js</code>. • binary. Performs individual file matching on files with these extensions: <code>.apklib</code>, <code>.bin</code>, <code>.dll</code>, <code>.exe</code>, <code>.o</code>, and <code>.so</code>. • all. Performs individual file matching on all files with these extensions: <code>.js</code>, <code>.apklib</code>, <code>.bin</code>, <code>.dll</code>, <code>.exe</code>, <code>.o</code>, and <code>.so</code>. • both. Performs individual file matching on file extensions only using both approved signature lists. <p>Any other value will be ignored and individual file matching will remain disabled.</p> <p>Click here for more information on using this parameter with approved signature lists.</p>
<code>--insecure</code>	Ignores TLS validation errors, allowing the scanner to connect to the Black Duck server.
<code>--license-search</code>	Enables searching for embedded licenses .
<code>--logDir <log directory></code>	Location of the <code>log</code> directory which contains all scanner log files. You must specify the <code>--logDir</code> parameter for log files to be saved.
<code>--max-request-body-size <size></code>	Controls how scan data is streamed (buffered) from the Signature Scanner to Black Duck.
<code>--max-update-size <size></code>	In rare cases, you may need to modify these values to better suit your network, for example, decreasing the values if there are issues with your network or increasing the default values if your network is highly stable.

Parameter	Description
	<ul style="list-style-type: none"> --max-request-body-size. Size of the main request that uploads the scan data for scanned paths. <p>Specify a value, in bytes. The default is 20000000 bytes. The recommended minimum value is 2000000 bytes; the recommended maximum value is 2000000000 bytes.</p> <ul style="list-style-type: none"> --max-update-size Buffers an update request to inform Black Duck when the Signature Scanner has completed uploading the data of individual URIs (scanned paths). <p>Specify a value, in bytes. The default value of 10000 bytes. The recommended minimum value is 1000 bytes; the recommended maximum value is 1000000 bytes.</p>
--min-scan-interval=<time in hours>	Minimum scan interval setting (in hours), which may be used to limit daily rate of the signature scan for given code location. If set to greater than 0, signature scans will not be processed if they occur before the set scan interval.
--name <scan name>	<p>Unique name identifying this scan. This name is displayed on the Scans page. Click here for more information.</p> <p>Note: The --name parameter is not supported when specifying multiple scan paths in a single command line.</p>
--no-prompt	<p>Non-interactive mode.</p> <p>Instead of the --no-prompt parameter, you can set the BD_HUB_NO_PROMPT environment variable to enable non-interactive mode.</p>
--project <project>	<p>Name of the project 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"> If the project and project version exist, the scanner maps or remaps the scan results. If the project exists, but the version does not, the scanner creates the version and maps the scan results.

Parameter	Description
--password <password>	<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"> Specifying the --password parameter without the <i>password</i> value results in the scanner prompting you for the password. 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. <p>Set the BD_HUB_PASSWORD environment variable with the Black Duck server password instead of passing an argument to the --password parameter:</p> <ul style="list-style-type: none"> If you set this environment variable <i>and</i> specify the --password parameter, the scanner prompts you for the password; it does not check the password value against the value specified in the environment variable. If you set this environment variable <i>and do not</i> specify the --password parameter, the scanner does <i>not</i> prompt you for the password. <p>Important: Set the BD_HUB_PASSWORD environment variable with the Black Duck server password. If you supply the password parameter, an error message appears and the scan will not complete.</p> <p>If this environment variable is <i>not</i> set, the scanner prompts you for the password whether you include or omit the --password parameter.</p> <p>Note: If the password parameter is the parameter immediately before <i><scan_path></i> use -- to indicate you are finished passing parameters, for example --password -- <scan_path>. Otherwise, the scanner will try to use the <i><scan_path></i> value as the password.</p> <p>Instead of specifying a username and password, use the BD_HUB_TOKEN environment variable to specify a Black Duck API token.</p>
--port <port>	Port on which the Black Duck server instance is listening.
--release <release>	<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"> If the project and project version exist, the scanner maps or remaps the scan results. If the project exists, but the version does not, the scanner creates the version and maps the scan results.
--scheme <scheme>	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 --scheme https to specify the https protocol.
--statusWriteDir <directory>	Specifies the directory to which the scanner outputs a JSON file which contains the complete scan status information.

Parameter	Description
--selfTest	Performs a self-test; will not connect to or post the scan to the Black Duck server.
--snippet-matching --snippet-matching-only --full-snippet-scan	<p>Select one of the following for snippet matching:</p> <ul style="list-style-type: none"> --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. <p>Black Duck Software recommends using this parameter for snippet scanning.</p> <ul style="list-style-type: none"> --snippet-matching-only. Selecting this parameter runs a snippet scan only on files that have changed; a component scan is not performed. <p>You must have successfully completed a full file scan prior to selecting this parameter.</p> <ul style="list-style-type: none"> --full-snippet-scan. Selecting this parameter performs a snippet scan on all files. <p>This parameter must be used with the --snippet-matching or --snippet-matching-only parameter:</p> <ul style="list-style-type: none"> 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 <i>all</i> files. With the --snippet-matching-only parameter: A snippet scan is performed on all files; a component scan is <i>not</i> completed. <p>To upload source files, you must use the --upload-source parameter, as described below.</p>
--tlscertpass	<p>Forces the scanner to prompt you for the password for the client certificate.</p> <p>You can specify the --tlscertpass parameter and/or set the BD_HUB_CLIENTCERT_PASS environment variable which specifies the private key password for the client certificate, for example, when --tlscert points to an encrypted PKCS #12 key store.</p> <p>The result of specifying the --tlscertpass parameter depends on whether the key is encrypted.</p> <ul style="list-style-type: none"> If the key <i>is</i> encrypted, the scan will fail if you do not set the BD_HUB_CLIENTCERT_PASS environment variable <i>or</i> specify the --tlscertpass parameter. <ul style="list-style-type: none"> If you set the environment variable <i>and</i> specify the --tlscertpass parameter, the scanner prompts you for the password; it does not check the password value against the value specified in the environment variable. If the key <i>is not</i> encrypted, regardless of whether the BD_HUB_CLIENTCERT_PASS environment variable is set:

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

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.

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.

You can also find more information about these exit codes [here](#).

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-2021.8.0/scan.cli-2021.8.0/bin
```

Windows:

```
C:\scan.cli-2021.8.0\scan.cli-2021.8.0\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 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-2021.8.0/scan.cli-2021.8.0/bin
```

Windows:

```
C:\scan.cli-2021.8.0\scan.cli-2021.8.0\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
```

Minimum Scan Interval

This setting allows users to change the minimum hourly frequency of which signature scans can be performed for a given code location when using the intelligent persistence feature for signature scanning. This will allow customers to reduce the load on their servers, thus making scans running faster and with less errors which result from overloading the server.

The default setting is set to 0, or no minimum scan interval, meaning scans are not prevented from occurring regardless of frequency. If set to greater than 0, signature scans will not be processed if they occur before the set scan interval. For example, a setting of 4 will not allow signature rescans before 4 hours of time have elapsed.

Note: When this feature is enabled, signature scans with Detect will finish with a status of success even if the signature scan was not run due to the scan interval. A warning message will appear in the logs indicating the scan was not run, but there will be no other indication given to the user.

Changing the minimum scan interval

Users with the system administrator role can change this setting by:

1. Log in to Black Duck with the System Administrator role.



2. Click **Admin**.

3. Select **System Settings**.

4. Click **Scan**.

5. Under **Minimum Scan Interval**, enter an integer for the number of hours between subsequent signature scans.

6. Click **Save**. To indicate that the default value has changed, the button changes to **Saved**.

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, from the Control Panel, click **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. Click  **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-2021.8.0/scan.cli-2021.8.0/bin
```

Windows:

```
C:\scan.cli-2021.8.0\scan.cli-2021.8.0\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-2021.8.0/scan.cli-2021.8.0/bin
```

Windows:

```
C:\scan.cli-2021.8.0\scan.cli-2021.8.0\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 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"
```

Approved signature lists

As the Signature Scanner examines files, it generates “signatures” of the files and sends SHA-1 and clean SHA-1 signatures to Black Duck’s web application. Black Duck filters these signatures based on the individual file matching parameters (if selected) and/or allowed signature lists, which you can create. Black Duck then sends the signatures to the Black Duck KnowledgeBase (KB) web service to identify the open

source software contained in the your scanned code.

You can create an allowed signature list for SHA-1 and/or clean SHA-1 file extensions. Each list is optional and works independently of the other list.

To create a list of approved signatures:

- Use one or both of the following parameters in the Signature Scanner:
 - **--BinaryAllowedList** *x, y, z* where *x, y, z* are the approved file extensions for SHA-1 (binary) files.
 - **--SourceAllowedList** *a, b, c* where *a, b, c*, are the approved file extensions for clean SHA-1 (source code) files.
- Create an environment variable. The following example is for SHA-1 and clean SHA-1 signatures for Linux or Mac OS X.

```
export JAVA_TOOL_OPTIONS="-Dblackduck.scan.cli.BinaryAllowedList=x,y,z -  
Dblackduck.scan.cli.SourceAllowedList=a,b,c"
```

For Windows systems, use the Control Panel to access the Advanced System Settings dialog box to create the environment variable.

If you enable individual file matching (using the **--individualFileMatching** parameter) in the Signature Scanner *and* create list(s) of allowed signatures, the outcome depends on the option you select:

- **source** option
 - Replaces the existing file extension for the **source** option with the list of file extensions from your clean SHA-1 signature list (**SourceAllowedList**).
 - Does *not* use the list of file extensions from your SHA-1 signature list **BinaryAllowedList**.
- **binary** option
 - Replaces the existing list of file extensions used for the **binary** option with the list of file extensions from your SHA-1 signature list (**BinaryAllowedList**).
 - Does *not* use the list of file extensions from your clean SHA-1 signature list (**SourceAllowedList**).
- **all** option
 - Performs individual file matching on all files with these extensions: *.js, .apklib, .bin, .dll, .exe, .o, and .so*.
 - Uses the list of file extensions from your SHA-1 signature list (**BinaryAllowedList**).
 - Uses the list of file extensions from your clean SHA-1 signature list (**SourceAllowedList**).
- **both** option
 - *Only* uses the file extensions from your SHA-1 and clean SHA-1 signature list (**BinaryAllowedList** and **SourceAllowedList**).

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 duplicate BOM detection

Duplicate BOM detection determines if a new package manager scan duplicates the existing BOM, and if so, stops processing the scan and denotes it as complete. For high-frequency scans that generate redundant (identical) data, Black Duck's duplicate BOM detection can provide significant performance improvements.

The only indication in the Black Duck UI as to whether a scan is a duplicate is on the *Scan Name* page: for duplicate scans, the scan status is "Complete" and the number of matches is "Unchanged":

The screenshot shows the Black Duck interface for managing scans. At the top, there's a header with a user icon and the text "Scans" followed by "Duplicate BOM detection". Below this is a section titled "Scan Details - for the last completed scan" with the following data:

Path	/
Host	<unknown host>
Created on	Tue, Mar 23, 2021 11:26 AM
Scan Size	0 B
Match Count	6
Folders	0
Files	0

On the right, there's a "Mapped to Project Version" section showing "Duplicate BOM detection > 1.0.1" with a "Unmap from Project" button. Below this is a "Delete Scan" button.

Under "Scan History", there's a table listing three completed scans:

Status	Matches	Host	Path	Scan Size	Last Updated	Scan Initiated By	Action
Complete	Unchanged	<unknown host>	/	0 B	Tue, Mar 23, 2021 11:28 AM	sysadmin	View BOM Import Log
Complete	Unchanged	<unknown host>	/	0 B	Tue, Mar 23, 2021 11:27 AM	sysadmin	View BOM Import Log
Complete	6 Matches	<unknown host>	/	0 B	Tue, Mar 23, 2021 11:26 AM	sysadmin	View BOM Import Log

At the bottom right of the history table, it says "Displaying 1-3 of 3".

Note the following:

- Duplicate BOM Detection is currently for *package manager scans* only and works with any version of Synopsys Detect. No additional Synopsys Detect properties are required.
- This feature is automatically enabled, however, you can disable this feature. Refer to the *Installing Black Duck using Docker Swarm* guide for more information.
- Black Duck only compares a scan to recent BOMs: Black Duck will not compare a package manager scan to a BOM that is older than 7 days.
- If results were requested when configuring the scan, those results are still returned from the existing data.
- If Black Duck does not detect a duplicate BOM, scan processing proceeds as usual.
- Duplicate BOM information, such as the number of unique and total BOMs, is shown in the **usage: scan completion** section of the System Information page.

About component dependency duplication

When scanning a project, several different types of matching processes can happen. The signature match looks at the structure of directories and files and try to match the “signatures” to what’s stored in the KnowledgeBase. The snippet match looks at code snippets and looks for matches in what’s stored in the KnowledgeBase. The package manager match uses external tools to examine build configuration files to find declared dependencies and then find matching components in the KB.

After scan, match and BOM computation are completed, the Components tab will display all the components detected with the above matching processes. The Match Type column will display “Transitive Dependency” or “Direct Dependency” alone or together with other types. These components are detected by the package manager matching process. The Source column may show multiple matches representing the number of different paths in the dependency tree.

Viewing component dependency duplication

Clicking the matches link in the Sources column will direct you to the Source tab. This view has a left pane that shows the dependency tree (in addition to source code tree for signature match), and a right pane that

shows components (possibly filtered) under a tree node.

With default settings, all duplicate matches of a particular component will be agglomerated into a single entry. This means that if a project has multiple paths that lead to a specific component, only one entry will be displayed in the right-hand pane of the Source tab.

Changing how duplicate dependencies are displayed

Users with the system administrator role can define the depth of displayed component duplication where 1 is no duplication and 10 will display all components up to a maximum 10 levels of relation. Please note that setting this level too high will result in reduced product performance. The default level is 1.

1. Log in to Black Duck with the System Administrator role.



2. Click **Admin**.
3. Select **System Settings**.
4. Click **Scan**.
5. Under **Component Dependency Duplication Sensitivity**, enter an integer (1 to 10) for the number of levels to display more component dependency entries.
6. Click **Save**. To indicate that the default value has changed, the button changes to **Saved**.

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 to 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 open source files moved into your proprietary directories and matches that code with open source code found in one or more Black Duck KnowledgeBase files. Though many of the details are proprietary, the mechanism to find snippets is through creating “codeprints” over the contents of scanned source files with a sliding window algorithm. Then, a statistically relevant sampling of those codeprints is sent to the Black Duck KnowledgeBase for matching and the results are presented to the user for review in the Black Duck UI. Codeprints are a type of one-way cryptographic hash which cannot be decomposed back into the original source code. Codeprints are analogous to fingerprints used for crime scene investigation: a fingerprint can be used to identify a person, but a fingerprint cannot be turned into that person. Codeprints allow the Black Duck application to securely and accurately scan code for snippet reuse.

Typically, five to seven lines of average source code can generate a match depending upon the density of

non-ignored characters in the line of code. The scanner will ignore white space, tabs, and other non-relevant characters (for example, lines of *****). One line of code can generate a match if it has enough words/characters in it (see certain javascript libraries). Very short lines of code may require more than 20 lines for a match. So, the density of the information content over those lines plays a factor into generating matches. However, other factors can also come into effect and Black Duck has a variety of rules and exclusions to optimize the scanning process and reduce false positives, but which can also impact what gets scanned and what is detected.

Click [here](#) for the list of file extensions supported for snippet scanning.

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. **Run component analysis.**

The component scan is completed first. This identifies the open source components using directory/file-level signatures.

2. **Generate snippet codeprints.**

If enabled, a second-pass snippet scan is performed. This scan analyzes the unmatched files in the initial component scan. For example, individually matched files or files in directories which are matched to open source components do not get scanned for snippets, as they have already been identified and to further scan them for snippets is unnecessary for the typical scanning process. The unmatched files are those which, under the component scan, did not show indications of being open source and have a file extension which indicates they are a source file. These files are the candidates for further analysis.

Note that Black Duck only analyzes the first 2MB of data for codeprints.

3. **Perform snippet matching.**

Snippet codeprints for the file candidates are generated and sent to Black Duck which then sends them to the Black Duck KB Snippet Matching Service. Depending upon the scan parameters selected (see below), Black Duck will send the codeprints for all files or only changed files (delta scans) to the Black Duck KB for matching. The matching service first looks for an exact file match before looking for a snippet match. If any matches are found for the file, a list of matches is produced and a heuristic is run to select the best match as a likely source. Due to the nature of using codeprints over a sliding window, fuzzy matches (inexact or modified textual areas from the original) can be detected. All the matches are then consolidated and available for review in the Black Duck UI.

4. **The user reviews match details.**

Unlike components detected via signature or package management scans, components detected via snippet scans are not automatically added to the BOM. This is because the source of a snippet match can often be in many places. Black Duck attempts to choose the best match and show alternative options, but ultimately it is up to the individual users to review these matches and confirm them before they are added to their BOM. While reviewing, a user can look at the matched open source code and (optionally) compare their scanned code with the matched open source code. Please note however, that when viewing the matched area to an open source file, due to the nature of hashed-based scanning using a sliding window algorithm, the highlighted text is only an approximation of the matched area for references purposes. Parts of the match may exceed, and unmatched matched parts may be displayed, in this highlighted area.

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. If this scan is a first-time scan, then all snippet candidates will be analyzed for snippet matches and the performance benefit will happen on rescans of the same project code.

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. Its purpose is to add a snippet scan to an already existing component scan.

You must have successfully completed a full file scan prior to selecting this parameter, otherwise the scan will error.

- **--full-snippet-scan.** Selecting this parameter performs a snippet scan on *all* files, regardless of if they have changed or not. It effectively overrides the delta scanning capability at the cost of scan performance. On a first time scan, as all snippet candidates are analyzed for matching, this parameter will have no impact.

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* snippet candidate files.
- With the **--snippet-matching-only** parameter: A snippet scan is performed on *all* snippet candidate 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.matching** 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.

Retaining partial snippet identifications

By default, identifications you made to partial snippet matches are not retained in subsequent snippet rescans.

You can change this default setting so that you can minimize the number of snippet matches you need to

re-identify: in the project's **Settings** tab, in the **Snippet Adjustments** section, select **Apply IDs from partial snippet matches to new exact file matches**.

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.

Modifying the default maximum snippet file size

By default, Black Duck only analyzes the first 2MB of data for snippet codeprints.

You can modify this default value and select a value from 1MB to 16MB.

To modify the default maximum snippet file size

1. Log in to Black Duck with the System Administrator role.



2. Click **Admin**.

3. Select **System Settings**.

4. Click **Scan** in the left-hand menu.

5. In the **Snippet Max File Size** section, enter a value from 1 to 16 to set the maximum file size in MB for snippet scanning.

6. Click **Save**.

Snippet extensions

These are the file types supported for a snippet scan.

Extension	Language
.4th	Forth
.actionscript	ActionScript
.ada	Ada
.adb	Ada
.ads	Ada
.aidl	Interface Definition Language (IDL)
.as	ActionScript
.as8	Assembly

Extension	Language
.asm	Assembly
.asp	ASP.NET (C#)
.aspx	ASP.NET (VB)
.aug	Augeas
.awk	Awk
.bas	Classic BASIC
.bash	Shell
.bat	Windows batch
.bf	Brainf*ck
.bfpp	Brainf*ck++
.bi	Structured BASIC
.bms	Text
.bmx	BlitzMax
.boo	Boo
.c	C, C++
.c#	C#
.c++	C++
.cbl	COBOL
.cc	C++
.cfc	ColdFusion
.cfm	ColdFusion
.cgi	Text
.chai	ChaiScript
.clj	Clojure
.cljc	Clojure
.cljs	Clojure
.cls	Visual Basic
.cmd	Rexx
.com	DCL
.cpp	C++

Extension	Language
.cpy	Text
.cs	Text
.cu	CUDA
.cuh	CUDA
.cxx	C++
.d	D
.dpk	Delphi
.dylan	Dylan
.e	Eiffel
.ec	eC
.eh	eC
.el	Emacs Lisp
.erl	Erlang
.es	ECMAScript
.exec	Rexx
.exheres-0	Exheres
.exlib	Exheres
.f	Text
.f77	Text
.f90	Text
.factor	Factor
.for	Text
.fpp	Text
.fr	Forth
.frag	OpenGL Shading language (GLSL)
.frm	Visual Basic
.frx	Visual Basic
.fs	F#
.g77	Fortran (free-format)
.g90	Fortran (free-format)

Extension	Language
.glsl	Fortran (free-format)
.go	Go
.groovy	Groovy
.gs	Genie
.h	C, C++
.h++	C++
.haml	Ruby
.hh	C++
.hpp	C++
.hrl	Erlang
.hs	Haskell
.hx	Haxe
.hxx	C++
.i	Fortran (fixed-format)
.i3	Modula-3
.idl	Interface Definition Language (IDL)
.inc	Text
.java	Java
.js	JavaScript
.jsp	Java, JavaScript
.jws	Java
.l	C
.lhs	Haskell
.lisp	Lisp
.lsp	Lisp
.lua	Lua
.m	Text
.m2	Modula-2
.m3	Modula-3
.m4	Text

Extension	Language
.ml	OCaml
.mli	OCaml
.mm	Java
.mod	Modula-2
.nb	Mathematica
.nbs	Mathematica
.octave	Octave
.pas	Pascal
.php	PHP
.php3	PHP
.php4	PHP
.php5	PHP
.phps	PHP
.phtml	PHP
.pl	Prolog
.pm	Perl
.pp	Puppet
.py	Python
.r	R
.r3	Rebol
.rb	Ruby
.rc	Text
.reb	Rebol
.rebol	Rebol
.rexx	Rexx
.ru	Ruby
.s	Assembly
.sc	Scala
.scala	Scala
.scm	Scheme

Extension	Language
.sh	Shell
.sqb	SQL
.sql	SQL
.ss	Scheme
.st	Smalltalk
.swift	Swift
.tcl	Tcl
.tk	Text
.v	Coq
.vb	Visual Basic
.vba	Visual Basic
.vbe	VBScript
.vbs	VBScript
.vert	OpenGL Shading Language (GLSL)
.vhdl	VHDL
.vhdl	VHDL
.vim	Vimscript
.y	Text
.z80	Assembly

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

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,

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.

By default, custom scan signatures have been limited to the top five levels in the directory structure. System administrators can modify the global default value and Super Users or Project Managers can modify the setting for a specific project.

Defining default scanning levels

Users with the system administrator role can define the depth of the scan, as measured by number of levels in the directory structure, from root, to perform custom signature scanning. The default level is 5.

To configure the default custom signature scanning level

1. Log in to Black Duck with the System Administrator role.
2. Click  Admin.
3. Select **System Settings**.
4. In the **Custom Scan Signature Level** section, enter an integer for the number of levels to perform custom signature scanning. You cannot enter 0.
5. Click **Save**. To indicate that the default value has changed, the button changes to **Saved**.

Creating custom scan signatures

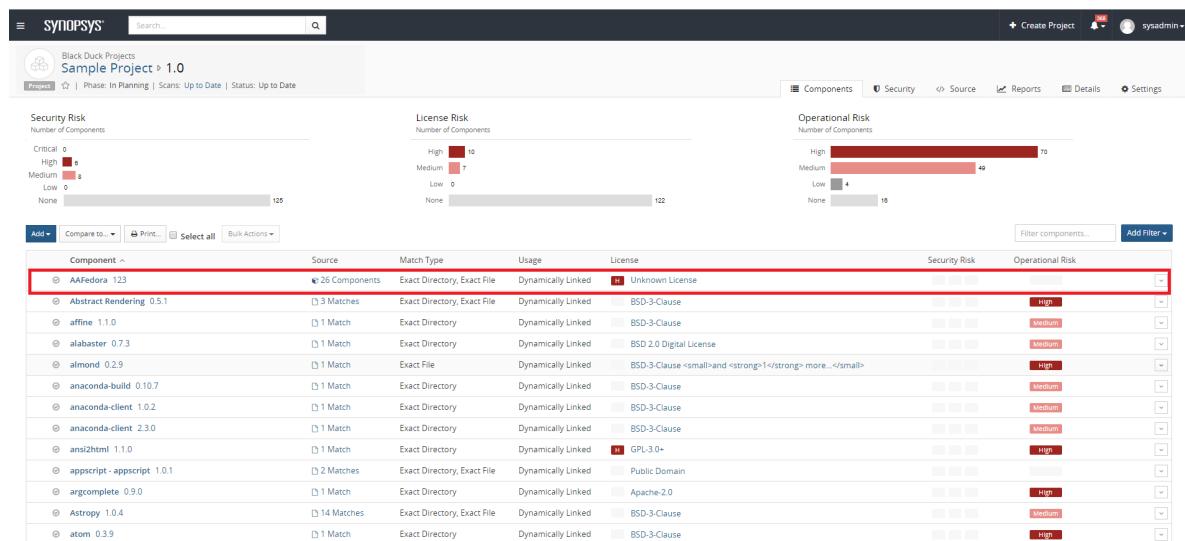
Custom code signatures are managed as projects and after identifying the code the custom code signatures are pulled into the BOM as a [subproject](#).

To create custom scan signatures

- Scan the third-party or proprietary code you wish to identify as a custom scan signature.

Use the `--blackduck.signature.scanner.individual.file.matching` property set to **ALL** in Synopsys Detect.

- Map the scan to a project version.
- Identify this project as a custom scan signature in the project's **Settings** tab:
 - Enable the feature
 - Optionally, select the depth, as measured by the number of levels in the directory structure, from root, to perform custom signature scanning. The value shown here is the default value, as defined by your system administrator.
 - Click **Save**.
- 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.

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 enable custom scan signatures.

- c. Define the number of levels to scan. The value shown here on the Settings tab

- d. Click **Save**.

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** 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.

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.

Hosting location for Synopsys Detect

Black Duck customers with limited external connectivity can define the internal hosting location of Synopsys Detect. Using this information, these users can leverage Code Sight for deployment across their developer base to run on-demand Software Composition Analysis (SCA) scans.

Specifying the hosting location of Synopsys Detect

To specify the hosting location of Synopsys Detect

1. Log in to Black Duck with the System Administrator role.



2. Click **Admin**.
3. Select **System Settings**.
4. Click **Synopsys Detect** in the left-hand menu.
5. In the **Hosting location for Synopsys Detect** section, enter the valid URI for your internal instance of Synopsys Detect.
6. Click **Save**.

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.](#)
- [Mapping a scan to a project.](#)
- [Removing a scan from a project.](#)
- [Deleting a scan.](#)
- [Viewing an audit log for a BOM file.](#)

Filtering scans

You can filter the scans on the *Scans* page by scan name, *Scan Status*, and/or by *Created Date*.

1. Log in to Black Duck.
2. Click  **Scans**.
3. Enter the desired text in the text field, and/or;
4. Click the **Add Filter** dropdown button and select an option in the dropdown menu.



Filtering by text

This filter allows you to view scans that contain specific text in its name.

Filtering by Scan Status

This filter allows you to view scans that match selected scan statuses. Selecting this filter opens a menu where you can select any number of statuses from the list below:

- Skipped: Scan that have been skipped.
- Complete: Scans and matching processes that are complete and a BOM is available for review.
- Not Started: Scans that have not been started.

- In Progress: Scans or the building of BOMs that are currently in progress.
- Error: Scans where an error has occurred.

Clicking the **OK** button in the dropdown menu will apply the filter to the Scans list.

Filtering by Created Date

This filter allows you to view scans that were created during the desired time frame. Selecting this filter opens a calendar selector allowing you to choose between two dates.

 to  

Removing filters

You can remove an active filter by clicking the  button to the right of the filter.

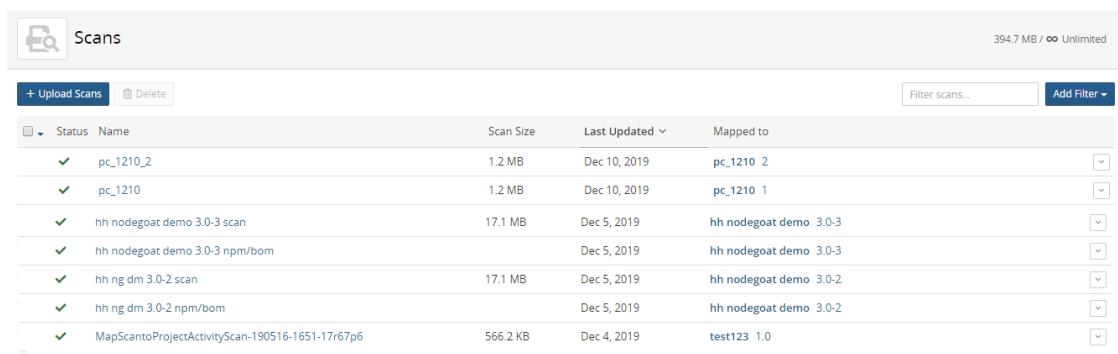
Uploading a scan file using the Black Duck UI

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.
2. Do one of the following:

- Click .



Status	Name	Scan Size	Last Updated	Mapped to
✓	pc_1210_2	1.2 MB	Dec 10, 2019	pc_1210_2
✓	pc_1210	1.2 MB	Dec 10, 2019	pc_1210_1
✓	hh nodegoat demo 3.0-3 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh nodegoat demo 3.0-3 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh ng dm 3.0-2 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-2
✓	hh ng dm 3.0-2 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-2
✓	MapScantoProjectActivityScan-190516-1651-17r67p6	566.2 KB	Dec 4, 2019	test123_1.0

- From the **Settings** tab for a project version, select **Scans**.

Scans

Status	Name	Scan Size	Last Updated
✓	bds00992#C:/Scan36/scan.cli-3.6.0-SNAPSHOT	195.24 MB	Dec 7, 2020

Displaying 1-1 of 1

3. Click **Upload Scans**.
4. Use the Upload Scans dialog box to locate the file and upload it.
5. Click **Close** in the Uploads Scan dialog box after uploading the file.

Note: The scan will not appear on the project version's **Settings** tab unless you mapped the scan to this project version during the scan; view the scan on the Scans page.

After uploading the file(s), if the scan is unmapped, 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.
2. Click  **Scans**.
3. Select the path of the scan that you want to view the results to open the *Scan Name* page.

Scans

ComplexBomMainProject_2015-12-04 10:28:23

Scan Details - for the last completed scan

Path	/
Host	scorpion.blackducksoftware.com
Created on	Mon, Aug 15, 2016 6:06 PM
Scan Size	1.19 MB
Match Count	74
Folders	22
Files	73

Delete Scan

Scan History

Status	Matches	Host	Path	Scan Size	Last Updated	Scan Initiated By
complete	74 Matches	scorpion.blackducksoftware.com	/	1.19 MB	Tue, Sep 29, 2020 1:17 PM	sysadmin

Displaying 1-1 of 1

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 **Mapped to Project Version** section displays the project and project versions to which the scan is currently mapped. If this scan is unmapped, use the **Map Scan to Project Version** 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.
 - **Cloned.** Black Duck is cloning the project version.
 - **Skipped.** The scan has been skipped.
 - **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. Note that this status also appears if Black Duck has determined that the scan was a [duplicate](#).
 - **Cancelled.** A user cancelled the scan before it was completed.

The following scan statuses indicate the scan has completed with or without error: COMPLETE,

CANCELLED, CLONED, SKIPPED, ERROR_SCANNING, ERROR_SAVING_SCAN_DATA, ERROR_MATCHING, ERROR_BUILDING_BOM, and ERROR.

The following scan statuses indicate the scan is in progress: UNSTARTED, SCANNING, SAVING_SCAN_DATA, SCAN_DATA_SAVE_COMPLETE, REQUESTED_MATCH_JOB, MATCHING, BOM_VERSION_CHECK, and BUILDING_BOM.

The [header on the BOM page](#) displays the status of scans being processed for this BOM.

- 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.

Mapping a scan to a project

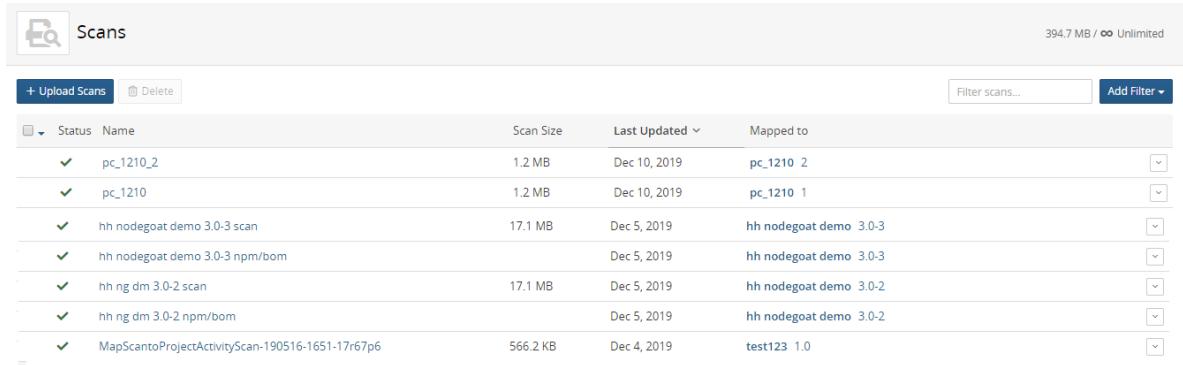
Mapping a scan adds the scan data to the BOM of a project version.

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.

2. Click  **Scans**.



Status	Name	Scan Size	Last Updated	Mapped to
✓	pc_1210_2	1.2 MB	Dec 10, 2019	pc_1210_2
✓	pc_1210	1.2 MB	Dec 10, 2019	pc_1210_1
✓	hh nodegoat demo 3.0-3 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh nodegoat demo 3.0-3 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh ng dm 3.0-2 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-2
✓	hh ng dm 3.0-2 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-2
✓	MapScantoProjectActivity/Scan-190516-1651-1767p6	566.2 KB	Dec 4, 2019	test123 1.0

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.

The screenshot shows the Black Duck UI interface. At the top, there's a header with a search icon and the word "Scans". Below it, a title bar says "ComplexBomMainProject_2015-12-04 10:28:23".

Scan Details - for the last completed scan:

Path	/	Match Count	74
Host	scorpion.blackducksoftware.com	Folders	22
Created on	Mon, Aug 15, 2016 6:06 PM	Files	73
Scan Size	1.19 MB		

Delete Scan

Map Scan to Project Version:

This scan is not mapped to any versions.

+ Create Project

Project:
start typing to select project...

Version:
Select a project to list its versions

Save

Scan History:

Status	Matches	Host	Path	Scan Size	Last Updated	Scan Initiated By	View BOM Import Log
Complete	74 Matches	scorpion.blackducksoftware.com	/	1.19 MB	Tue, Sep 29, 2020 1:17 PM	sysadmin	View BOM Import Log

Displaying 1-1 of 1

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.

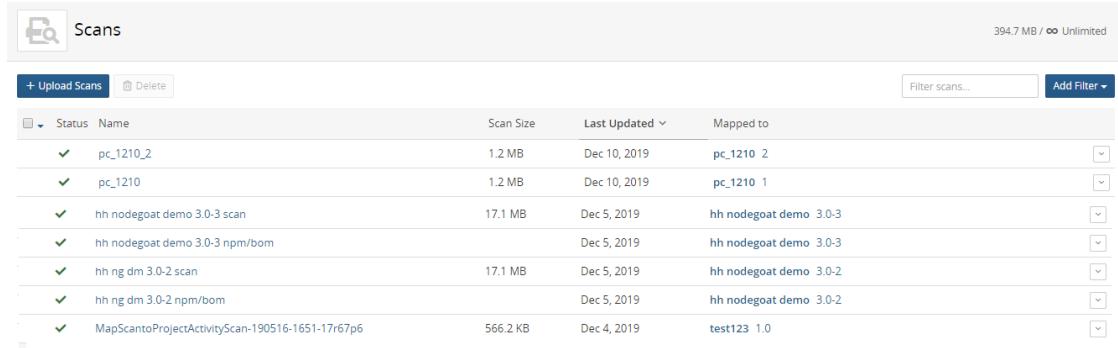
Removing a scan from a project

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

To remove a mapping

1. Log in to Black Duck.
2. Do one of the following:

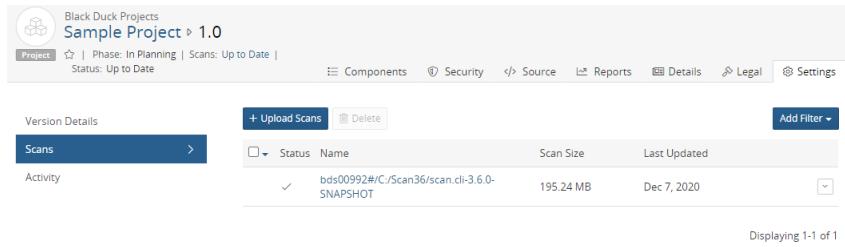
- Click



The screenshot shows the 'Scans' page in the Black Duck UI. At the top, there are buttons for '+ Upload Scans' and 'Delete'. A search bar says 'Filter scans...' and a 'Add Filter' button. The main area is a table with columns: Status, Name, Scan Size, Last Updated, and Mapped to. The table lists several scans, each with a dropdown menu icon. The total storage used is 394.7 MB / ∞ Unlimited.

Status	Name	Scan Size	Last Updated	Mapped to
✓	pc_1210_2	1.2 MB	Dec 10, 2019	pc_1210 2
✓	pc_1210	1.2 MB	Dec 10, 2019	pc_1210 1
✓	hh nodegoat demo 3.0-3 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh nodegoat demo 3.0-3 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh ng dm 3.0-2 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-2
✓	hh ng dm 3.0-2 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-2
✓	MapScantoProjectActivityScan-190516-1651-17r67p6	566.2 KB	Dec 4, 2019	test123 1.0

- From the **Settings** tab for a project version, select **Scans**.



The screenshot shows the 'Sample Project > 1.0' settings page. The 'Scans' tab is selected. The table shows one scan entry: 'bds00992#/C/Scan36/scan.cli-3.6.0-SNAPSHOT' with a size of 195.24 MB and a last update date of Dec 7, 2020. Below the table, it says 'Displaying 1-1 of 1'.

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

Deleting a scan

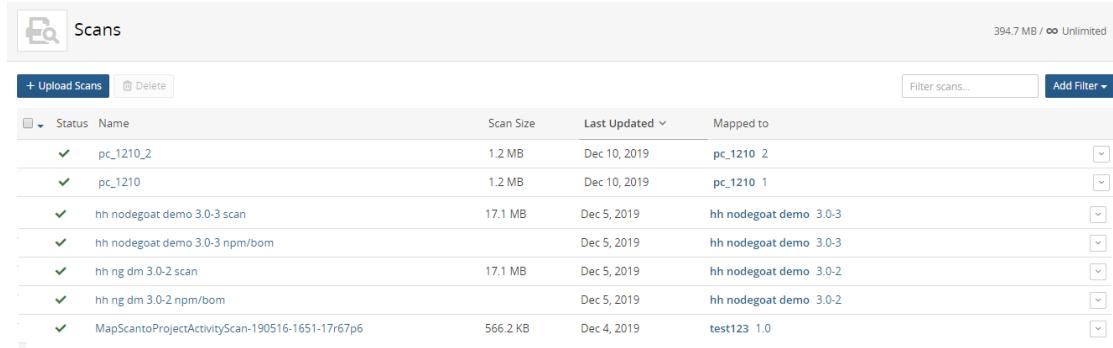
If you have scanned an incorrect path or Docker image, no longer require the scan, or want to free up space, you can delete the scan(s).

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

To delete a scan

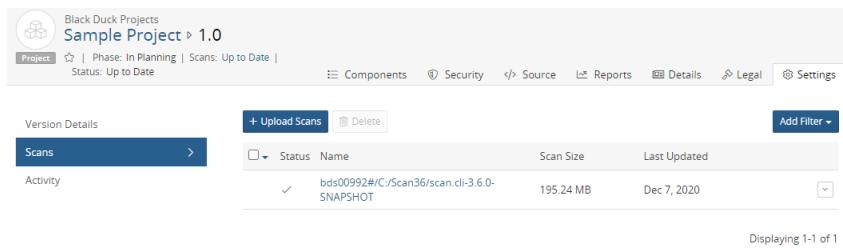
- Log in to Black Duck.
- Do one of the following:

- Click .



	Status	Name	Scan Size	Last Updated	Mapped to	
✓		pc_1210_2	1.2 MB	Dec 10, 2019	pc_1210 2	<input type="checkbox"/>
✓		pc_1210	1.2 MB	Dec 10, 2019	pc_1210 1	<input type="checkbox"/>
✓		hh nodegoat demo 3.0-3 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-3	<input type="checkbox"/>
✓		hh nodegoat demo 3.0-3 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-3	<input type="checkbox"/>
✓		hh ng dm 3.0-2 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-2	<input type="checkbox"/>
✓		hh ng dm 3.0-2 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-2	<input type="checkbox"/>
✓		MapScantoProjectActivityScan-190516-1651-17r67p6	566.2 KB	Dec 4, 2019	test123 1.0	<input type="checkbox"/>

- If the scan is mapped to a project version, from the **Settings** tab for a project version, select **Scans**.



Black Duck Projects
Sample Project > 1.0
Project | Phase: In Planning | Scans: Up to Date | Status: Up to Date
Components Security Source Reports Details Legal Settings

	Status	Name	Scan Size	Last Updated
✓		bds00992#/C:/Scan36/scan.cli-3.6.0-SNAPSHOT	195.24 MB	Dec 7, 2020

Displaying 1-1 of 1

- 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.

- 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

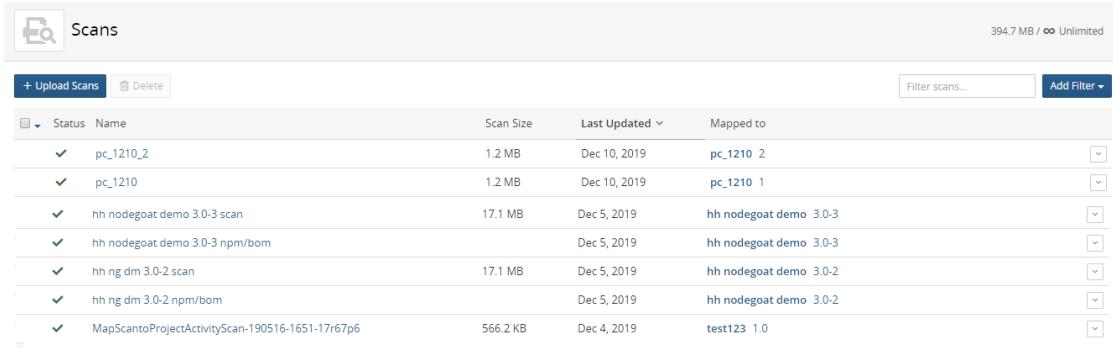
You may need a scan file, which is a file of a scan that has been imported to Black Duck, similar to a dry run file. For example, you may need to provide Customer Support with the scan file if you are experiencing scanning issues, as this file may help them investigate the issue.

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

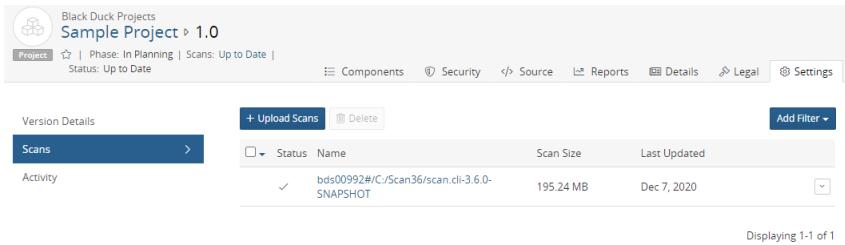
- Log in to Black Duck.
- Do one of the following:

- For unmapped scans, click .



	Status	Name	Scan Size	Last Updated	Mapped to
✓		pc_1210_2	1.2 MB	Dec 10, 2019	pc_1210 2
✓		pc_1210	1.2 MB	Dec 10, 2019	pc_1210 1
✓		hh nodegoat demo 3.0-3 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-3
✓		hh nodegoat demo 3.0-3 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-3
✓		hh ng dm 3.0-2 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-2
✓		hh ng dm 3.0-2 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-2
✓		MapScantoProjectActivityScan-190516-1651-17r67p6	566.2 KB	Dec 4, 2019	test123 1.0

- For scans mapped to a project version, from the **Settings** tab for a project version, select **Scans**.



Black Duck Projects
Sample Project > 1.0
Project | Phase: In Planning | Scans: Up to Date | Status: Up to Date
Components Security Source Reports Details Legal Settings

Version Details

Scans >

	Status	Name	Scan Size	Last Updated
✓		bds00992#C:/Scan36/scan.cli-3.6.0-SNAPSHOT	195.24 MB	Dec 7, 2020

Displaying 1-1 of 1

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

The file is downloaded with a `.bdio` extension, and is a zip file.

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

- Log in to Black Duck.



The Scans page appears.

Scans				
		Scan Size	Last Updated	
Status	Name	Scan Size	Last Updated	Mapped to
✓	pc_1210_2	1.2 MB	Dec 10, 2019	pc_1210_2
✓	pc_1210	1.2 MB	Dec 10, 2019	pc_1210_1
✓	hh nodegoat demo 3.0-3 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh nodegoat demo 3.0-3 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh ng dm 3.0-2 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-2
✓	hh ng dm 3.0-2 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-2
✓	MapScantoProjectActivityScan-190516-1651-17r67p6	566.2 KB	Dec 4, 2019	test123 1.0

3. Select the scan you wish to view.

The *Scan Name* page appears.

Scan Details - for the last completed scan		Mapped to Project Version	
Path	/	Sample Project	3.0
Host	scorpion.blackducksoftware.com	Match Count	74
Created on	Mon, Aug 15, 2016 6:06 PM	Folders	22
Scan Size	1.19 MB	Files	73
Delete Scan		Unmap from Project	
Scan History			
Status	Matches	Host	Path
Complete	74 Matches	scorpion.blackducksoftware.com	/
			1.19 MB
			Tue, Sep 29, 2020 1:17 PM
			sysadmin
			View BOM Import Log
Displaying 1-1 of 1			

4. Select **View BOM Import Log** 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.		
23 Components Mapped	3 Components Not Found	17 Licenses Mapped
0 License Not Found		
		Add Filter ▾
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 [My Projects dashboard](#) 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 'Overview' tab of the Black Duck Project page for 'Sample Project'. At the top, there's a navigation bar with 'Project' (highlighted), 'Watching Project | Versions: 1', 'Overview' (selected), and 'Settings'. Below the navigation, there are sections for 'Description' (a project used for demonstration purposes only) and 'Additional Fields' (select the team that originally created this project). An 'Architecture' section is also present. A 'Create Version' button is located at the bottom left. The main area displays a table with one row for 'Version 1.0' (Phase: In Planning, Last Updated: 8:33 AM, Last Scanned: Mar 19, 2021, License: GPL 3.0 with Classpath Exception). To the right of the table are three risk bars: Security Risk (red), License Risk (red), and Operational Risk (dark red). Filter and add filter buttons are at the top right of the table. The bottom right corner shows 'Displaying 1-1 of 1'.

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

Column	Description
N/A	<p>Icons shown to the left of the version name:</p> <ul style="list-style-type: none"> ■ Policy violation. Select the icon to view information on the policy violation. ■ Policy violation has been overridden. <p>Select the icon to view information on the policy violation.</p>
Version	Name of the project version.
Phase	<p>The development phase of this version. The possible values are:</p> <ul style="list-style-type: none"> • In Planning • In Development • Pre-release • Released • Deprecated • Archived <p>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.</p>
Last Updated	<p>When this project version was last updated. Hover over the value to see:</p> <ul style="list-style-type: none"> • 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. • 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.
Last Scanned	Date of the last scan for this project version. Hover over the value to see the date and time.
License	Name of the license for this project version.
Security Risk	<p>Bars show the critical, high, medium, and low security risk levels for the OSS components in this version of the project.</p> <p>Select the 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 the 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 the bar to view the number of affected components.</p>

Above 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.
- **Additional Fields.** Project custom field information.

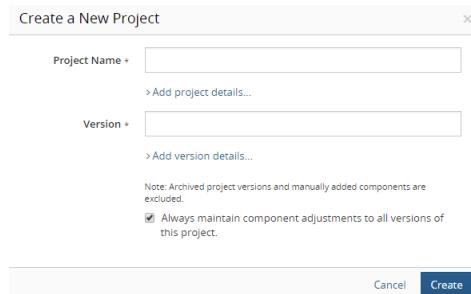
Creating a project

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.¹
- Select the attributes you wish to [clone for new versions of this project](#).
- Select whether to enable [custom scan signatures](#).

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	Low	Medium	High

Displaying 1-1 of 1

Description
No description.

Created
Sep 4, 2018 by sysadmin

Updated
Sep 4, 2018 by sysadmin

Tags
No Tags

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. Locate the project by using the **Watching** or **My Projects** dashboard.

¹A tier lets you categorize projects in terms of importance to your company. Tier 0 projects are defined as those that are most critical to the company, where Tier 5 projects are defined as least critical.

3. Click  in the project you want to delete and select **Delete**.
4. In the Delete Project dialog box, confirm that you have selected the correct project, and click **Delete**.
The project is deleted.

Watching projects

If you created a project, became a project member, or became a group member of a project, you are automatically "watching" that project.

- You will receive notifications for all projects (and the components in the projects) you are watching.
- The **Watching** dashboard, one of the default dashboards, displays all your watched projects.
- Your watched projects is also a filter available on the Find page for project searches.

You can remove projects you are watching and add projects you previously stopped watching.

Note: The **My Projects** dashboard lists all your projects, including those you are no longer watching.

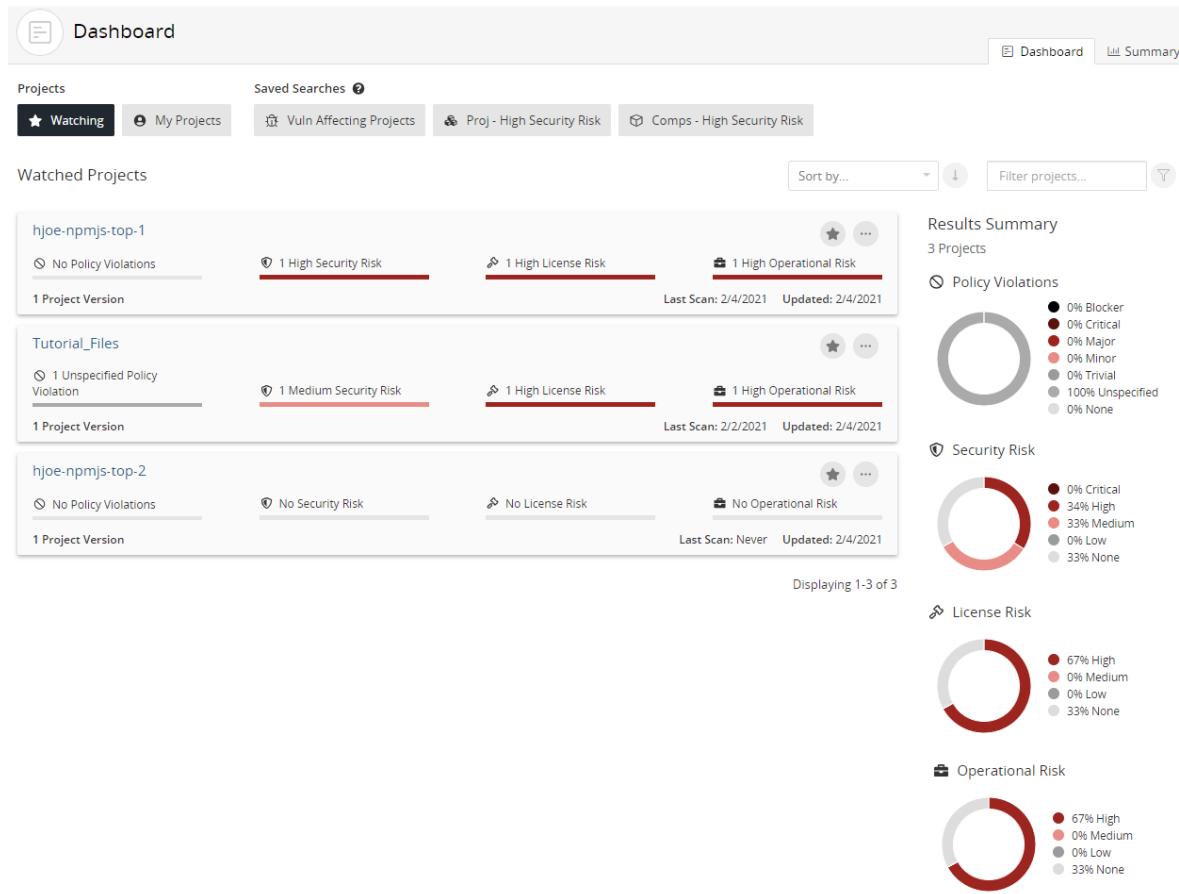
Viewing a list of your watched projects

The **Watching** dashboard lists your watched projects. The list of your watched projects also appears on the My Profile page.

To view a list of your projects using the Watching dashboard

1. Click .
2. If not selected, click **Watching**.

The dashboard of your watched projects appears.



To view a list of your watched projects from the My Profile page

1. From the user menu located on the top navigation bar, select **My Profile**.

The My Profile page appears.

My Profile

Profile >

Overall Roles

User Groups

Watched Projects

Username: SampleUser

Email: Email

First Name: FirstName

Last Name: LastName

Change Password

2. Select Watched Projects.

The table with your watched projects appears.

My Profile

Profile >

Overall Roles

User Groups

Watched Projects >

Watched Projects
Projects you are watching will generate notifications. The following is a list of all the projects you are currently watching.

Stop Watching Projects

Name	Created
Test Project	Apr 7, 2020

Displaying 1-1 of 1

The table lists each project name; select the project name to view the *Project Name Overview* tab. The **Created** column displays the date or time (if today) you become a watcher for this project. This could be date or time the project was created, you became a project member, or when you selected to watch the project.

Decreasing the number of watched projects

When you stop watching a project, you will no longer receive notifications for the project and its versions and the project is removed from the **Watching** dashboard.

To stop watching a project

Do one of the following:

- Click  in the **Watching** or **My Projects** dashboards.

The project no longer appears on the **Watching** dashboard.

 The Not Watching icon () appears for the project on the **My Projects** dashboard.

- In the **Watched Projects** tab of the My Profile page:
 - Click  in the row of the project you no longer wish to watch. The project is removed from the table.
 - To easily unwatch one or more projects, click  to the left of the project name and click **Stop Watching Projects**.
- Click **Confirm** in the Stop Watching Project dialog box. The project is removed from the table.
- From the *Project Name Overview* or **Settings** tab, click **Watching Project** in the project banner.



The heading now indicates that you are no longer watching this project:



Watching projects

If you selected to stop watching a project, you can select to watch it again. You will receive notifications for the project again and the project appears on the **Watching** dashboard.

To watch a project

Do one of the following:

- Click  in the **My Projects** dashboard of the project you wish to watch.

The icon  now indicates that you are watching the project.

The project now appears on your **Watching** dashboard.

- From the *Project Name Overview* or **Settings** tab, click **Not Watching Project** in the project banner to watch the project.



The heading now indicates you are watching the project:



Cloning projects

Use project cloning to fork an existing project to a new project. Cloning helps reduce your workload by using the data, analysis, and resolutions you defined in an existing project as a baseline for a new project.

Users who can create projects can clone projects. For each project, select the versions you wish to clone and the project's attributes, such as the project's settings or project members and groups. Note that the attributes cloned for each project version depend on the [project version cloning](#) settings you selected, as shown in the **Cloning** section in the **Project Details** section of the **Settings** tab:

Cloning Select the attributes you'd like to clone for any new versions of this project.

- Additional Fields
- Component Edits
- License Fulfillment Status
- Remediation Details

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

To successfully use cloning:

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

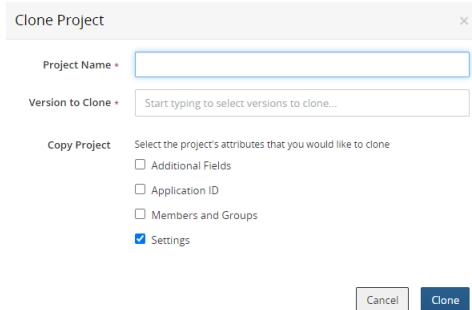
Cloned information appears in the cloned project versions for components that are the *same* as in the original project version for this project. A scan will need to be performed to replicate the components of the base project to the cloned project. If a component is not included in the newly scanned 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.

Enabling cloning

To clone a project

1. Open the **Project Name Settings** tab for the project you wish to clone.
2. Click **Clone Project** in the **Clone Project** section.

The Clone Project dialog box appears.



3. Do the following:

- Enter a name for this clone.
- Select the versions you wish cloned.
- Select what you would like to clone:
 - Additional Fields.
 - Application ID.
 - Members and Groups.
 - Settings. This option is selected by default. This includes the values of all attributes, excluding the project name, shown in the **Settings** section in the **Project Details** tab, for this project.

4. Click **Clone**.

Updating project information

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

- Project name.
- Project description.
- Project owner.
- Tier.¹
- [Ability to apply edits to all versions of a project.](#)
- [Snippet adjustments.](#)
- [Cloning project version settings.](#)
- [Custom scan signatures.](#)
- [Custom fields.](#)
- [Deep license data.](#)

¹A tier lets you categorize projects in terms of importance to your company. Tier 0 projects are defined as those that are most critical to the company, where Tier 5 projects are defined as least critical.

- Application ID.¹
- [Cloning projects](#).

To configure project settings for a project

1. Log in to Black Duck.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the **Settings** tab.

¹A 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 'Project Details' tab selected in the navigation bar. The main area is titled 'Settings'. It includes fields for 'Project Name' (Sample Project 1), 'Description' (empty), 'Owner' (dropdown placeholder 'start typing to select owner...'), 'Tier' (dropdown placeholder 'select tier...'), 'Component Adjustments' (checkbox 'Always maintain component adjustments to all versions of this project' checked), 'Cloning' (checkboxes for 'Additional Fields', 'Component Edits', 'License Fulfillment Status', 'Remediation Details', and 'Version Settings' all checked), 'Custom Scan Signature' (checkbox 'Enable Custom Scan Signature' unchecked), 'Depth' (input field set to 5), 'Deep License Data' (checkbox 'Apply Deep License Data to bill of materials' unchecked), 'License Conflicts' (checkbox 'Apply License Conflicts Data to bill of materials' unchecked), and a 'Save' button. Below this, there's an 'Additional Fields' section with a 'Data Sensitivity' dropdown set to 'Does the application contain or access important confidential information or personally identifiable data.' and a 'Save' button. Further down are sections for 'Application ID' (input field empty) and 'Clone Project' (button with icon). At the bottom is a 'Delete Project' section with a note about不可恢复的删除 and a red 'Delete Project' button.

4. 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.

5. Click **Save**.

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

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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. 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

Displaying 1-2 of 2

4. Click **+ Add Member**.
5. 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.
6. Select the username to add this user to the project team.
7. Optionally, to add multiple users, type and select the name of additional users.
8. Select the roles for this user for this project.
9. 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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. 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

Displaying 1-2 of 2

4. Click in the row of the user you want to remove from the project team and select **Remove**.
5. 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.

	Group Name	Status
	Htest	Active

Displaying 1-1 of 1

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.

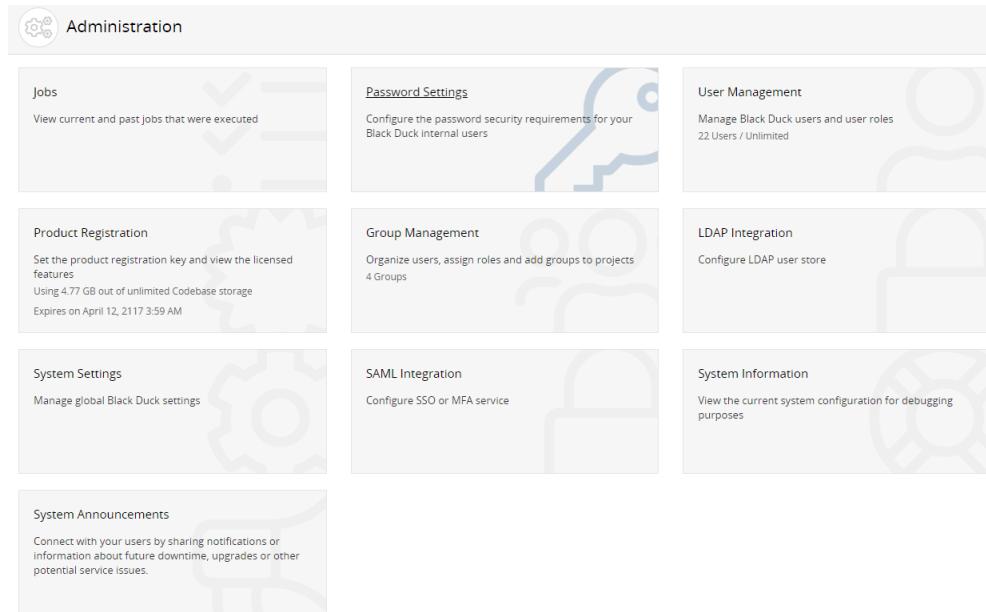
From the *Group Name* page:

1. Log in to Black Duck.

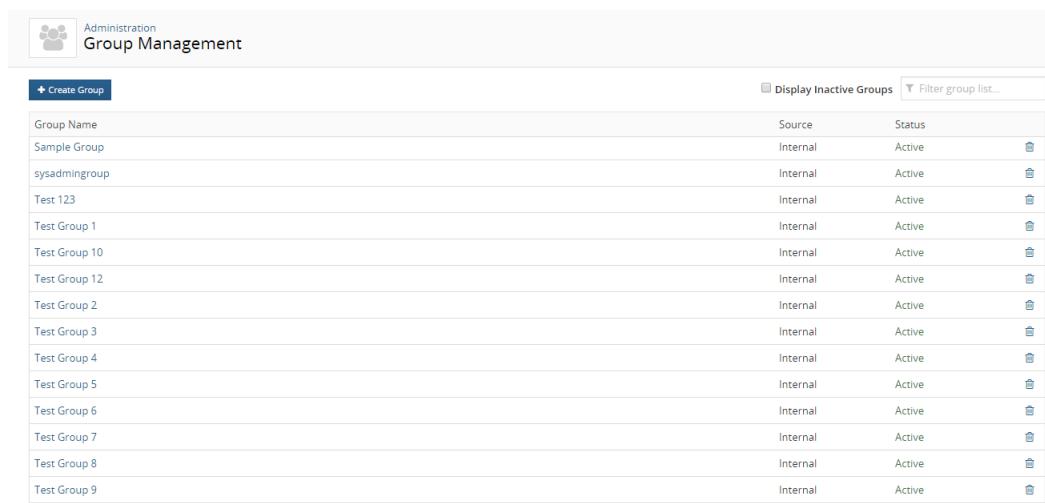


2. Click **Admin**.

The Administration page appears.

A screenshot of the Black Duck Administration page. The page has a header "Administration" with a gear icon. Below the header are nine cards arranged in a grid. The cards are: "Jobs" (View current and past jobs that were executed), "Password Settings" (Configure the password security requirements for your Black Duck internal users), "User Management" (Manage Black Duck users and user roles, 22 Users / Unlimited), "Product Registration" (Set the product registration key and view the licensed features, Using 4.77 GB out of unlimited Codebase storage, Expires on April 12, 2017 3:59 AM), "Group Management" (Organize users, assign roles and add groups to projects, 4 Groups), "LDAP Integration" (Configure LDAP user store), "System Settings" (Manage global Black Duck settings), "SAML Integration" (Configure SSO or MFA service), and "System Information" (View the current system configuration for debugging purposes). There is also a "System Announcements" card at the bottom left.

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

A screenshot of the Black Duck Group Management page. The page has a header "Group Management" with a group icon. Below the header is a search bar with "Display Inactive Groups" and "Filter group list..." options. A "Create Group" button is located at the top left. A table lists 18 groups: Sample Group, sysadminingroup, 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 table columns are "Group Name", "Source", and "Status". All groups listed are Internal and Active.

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

The screenshot shows the 'Administration / Group Management' section. A 'Sample Group' is selected. The 'Group Details' section contains a 'Group Name' field set to 'Sample Group' and an 'Active Group' checkbox which is checked. Below this are four role checkboxes: 'BOM Manager' (BOM manager is granted the privilege of editing BOM), 'Code Scanner' (Manages code-related scans), 'Policy Manager' (Policy Manager is granted the privilege of editing policy rules), and 'System Administrator' (System Administrator are granted the privilege of editing users and other system settings). The 'Group Members' section has a table with columns 'Username', 'First Name', 'Last Name', 'Email', and 'Status'. A button '+ Add Member' is visible above the table, and a message 'No Results Found.' is displayed below it.

- Click **Add Project** in the **Group Projects** section to display the Add Project dialog box.
- Enter the name of the project.
- 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:

- Log in to Black Duck.
- Locate the project using the **Projects** tab on the Dashboard.
- 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 Projects' section for 'Sample Project 1'. It includes tabs for 'Project' (selected) and 'Overview'. The 'Groups' tab is active, showing a table with columns 'Group Name' and 'Status'. One row is present with 'Htest' and 'Active'. A 'Remove' link is located to the right of the 'Htest' row. Other tabs include 'Members' and 'Project Details'.

- Click in the row of the group that you want to remove from the project team and select **Remove**.

6. In the Remove Group dialog box, click **Remove** to confirm.

The group is removed from the project team.

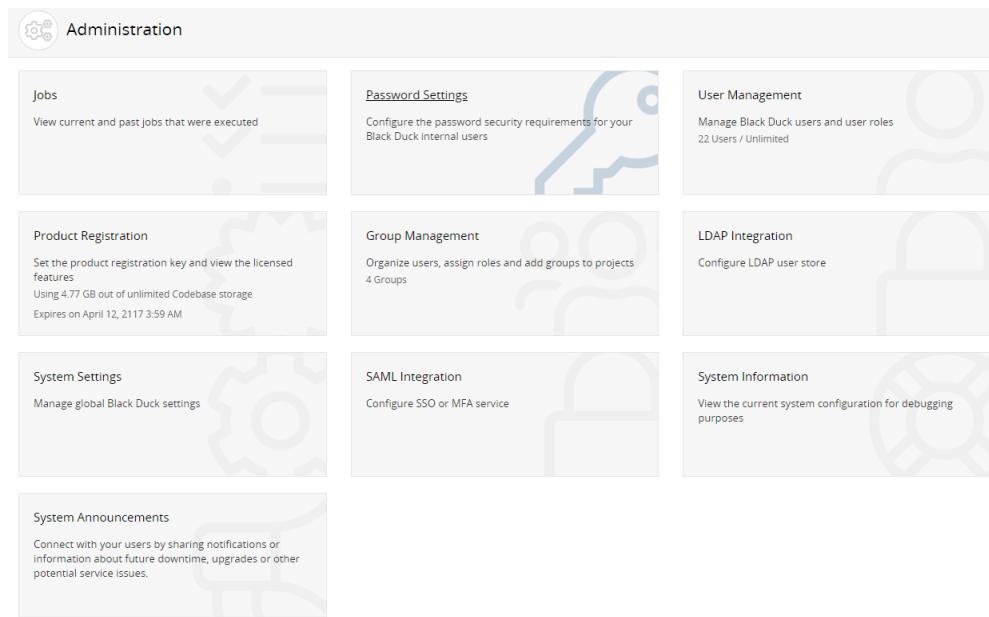
From the *Group Name* page:

1. Log in to Black Duck.

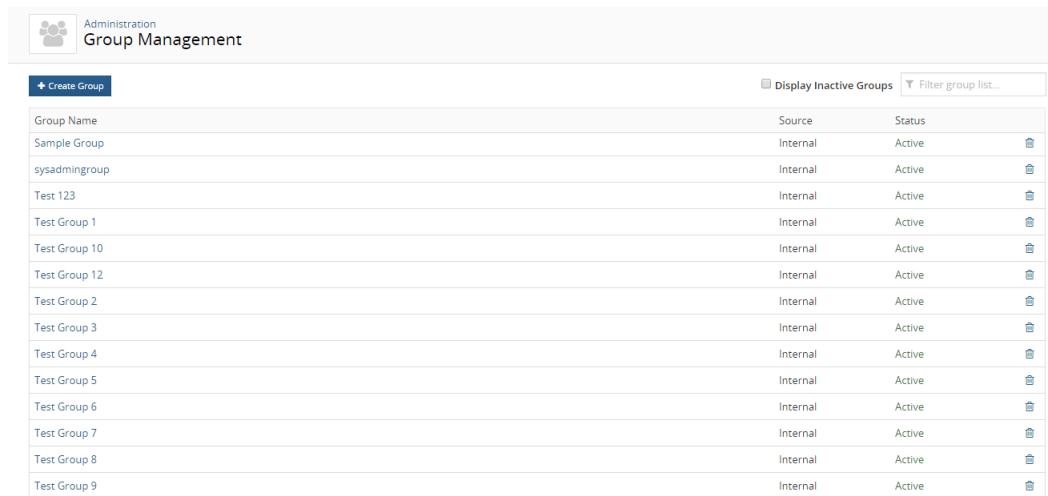


2. Click **Admin**.

The Administration page appears.

A screenshot of the Black Duck Administration page. The top navigation bar has a gear icon and the text "Administration". Below the navigation are several cards: "Jobs" (View current and past jobs that were executed), "Password Settings" (Configure the password security requirements for your Black Duck internal users), "User Management" (Manage Black Duck users and user roles, 22 Users / Unlimited), "Product Registration" (Set the product registration key and view the licensed features, Using 4.77 GB out of unlimited Codebase storage, Expires on April 12, 2017 3:59 AM), "Group Management" (Organize users, assign roles and add groups to projects, 4 Groups), "LDAP Integration" (Configure LDAP user store), "System Settings" (Manage global Black Duck settings), "SAML Integration" (Configure SSO or MFA service), "System Information" (View the current system configuration for debugging purposes), and "System Announcements" (Connect with your users by sharing notifications or information about future downtime, upgrades or other potential service issues).

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

A screenshot of the Black Duck Group Management page. The top navigation bar has a group icon and the text "Administration" followed by "Group Management". Below the navigation is a search bar with "Create Group" and filters for "Display Inactive Groups" and "Filter group list...". A table lists groups with columns for "Group Name", "Source", and "Status". The table includes rows for "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", all marked as "Internal" and "Active".

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

The screenshot shows the 'Administration / Group Management' section. A group named 'Sample Group' is selected. The 'Group Details' section shows the group name as 'Sample Group' and the 'Active Group' checkbox is checked. Below it are 'Delete Group' and 'Save' buttons. The 'Roles' section lists several roles with descriptions: BOM Manager, Code Scanner, Policy Manager, and System Administrator. The 'Group Members' section shows a table with columns: Username, First Name, Last Name, Email, and Status. A '+ Add Member' button is at the top of the table, and a message 'No Results Found.' is displayed below it.

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.

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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Hover over the **Tags** area of the page and click ↗ to display the tags field.
4. 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 **Manage** > **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.

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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.

Tip: 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 Clone Version dialog box appears with the information in the **Version to Clone** field completed.

3. Click **+ Create Version**.

The Create a New Version dialog box appears.

The screenshot shows a 'Create a New Version' dialog box. It contains the following fields:

- Version: A text input field.
- License: A dropdown menu.
- Notes: A text input field.
- Nickname: A text input field.
- Release Date: A text input field with a calendar icon.
- Phase: A dropdown menu with 'In Planning' selected.
- Distribution: A dropdown menu with 'External' selected.

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

4. 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.
 5. 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.
 6. 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.
 7. 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.
 8. 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.
 9. 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.
10. 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.

11. Click **Save**.

Black Duck saves the project version.

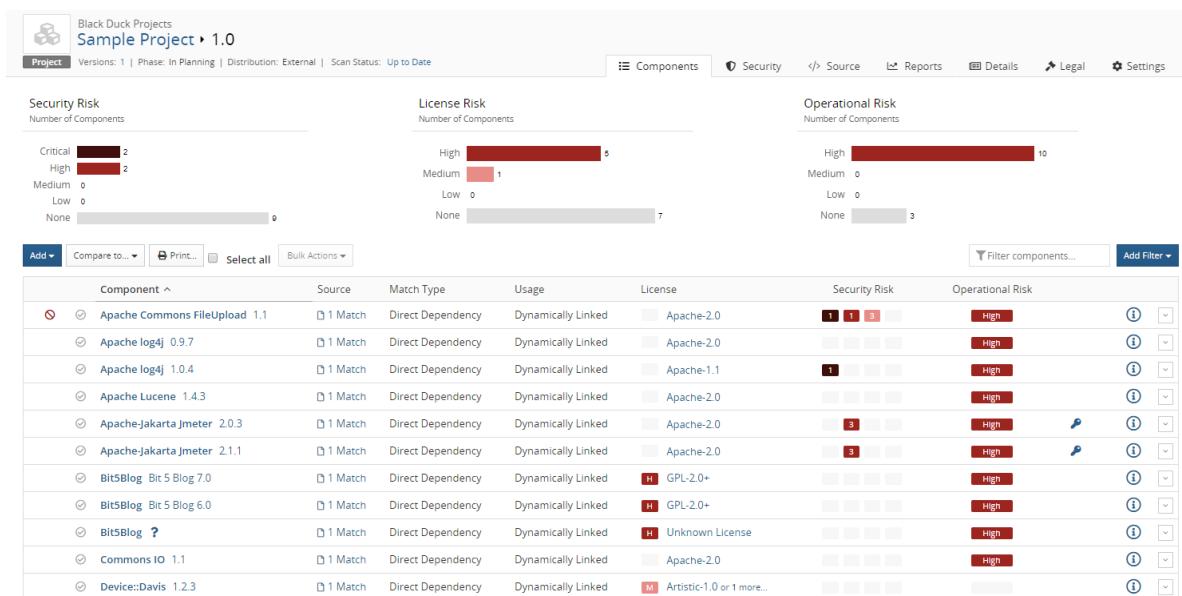
Updating project version information

You can rename a project version and update its information.

To update project version information

1. Log in to Black Duck.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name of the project that you want to manage.

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



4. 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.

5. 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 synchronizes 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. Note that if you have enabled persistent edits, then edits made to the baseline version *will be* propagated to the cloned version.

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, all options are cloned:

- Additional Fields: Custom field information.
- 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
 - Version Settings:
 - License
 - Notes
 - Nickname
 - Release Date
 - Phase
 - Distribution

You can modify these settings by using the **Cloning** section 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:

- Select **Clone** for the version you wish to clone from the *Project Name* page. The version of the Clone Version dialog box that appears depends on whether **Version Settings** was selected as a cloning attribute for the project:
 - If **Version Settings** was selected, specify the version name in the Clone Version dialog box and click **Clone**.
 - If **Version Settings** was not selected, specify the version name in the Clone Version dialog box, enter any of the other project version settings, and click **Clone**.
- Use 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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page

appears.

3. Click  in the row of the version of the project you want to delete and select **Delete**.
4. 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.

Black Duck treats In Planning, In Development, Pre-release, Released, and Deprecated project versions the same. Black Duck does not differentiate between these phases: these phases are to help you manage your projects. Project versions with these phases are included in project risk calculations.

Archived project versions are treated differently than the other project version phases.

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.

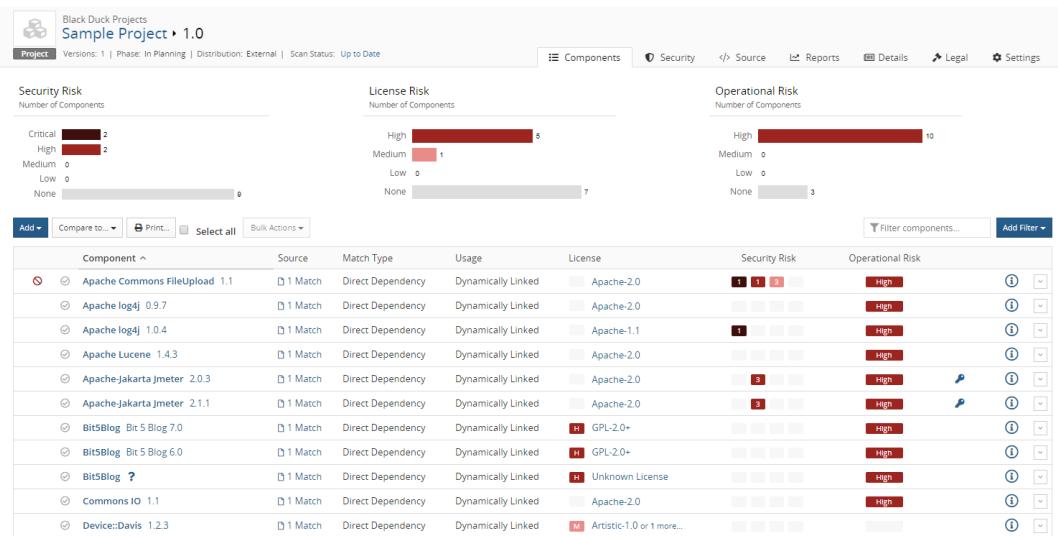
Chapter 6: Viewing a project version's BOM

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 name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the version that you want to view.

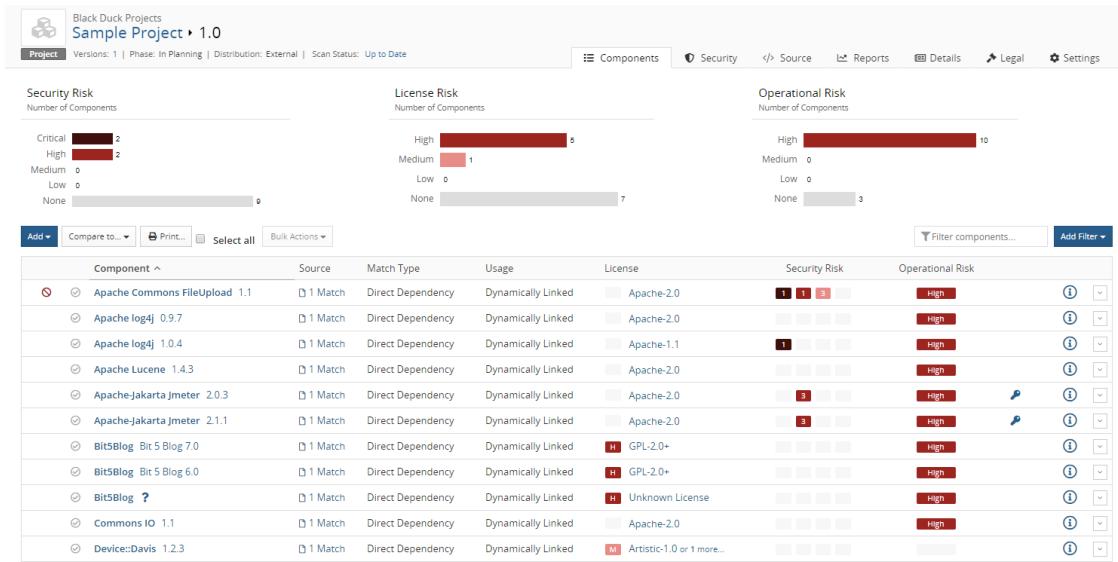
The **Components** tab displays the BOM. The example below is what appears for a user with the BOM Manager [role](#) using the List view:



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

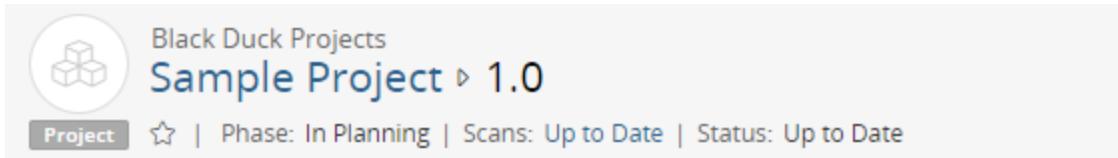
On a project version page (from the Dashboard, select **Project** tab > **Project Name** > **Project Version**), the **Components** tab displays the BOM. The page displays a header, risk graphs and a data table.



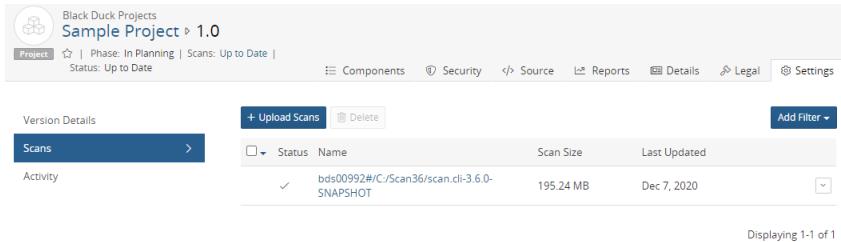
The above example is the list (or "flat") view of the BOM. You can also view a [hierarchical version](#) of the BOM.

Header information

Black Duck displays information in the header about the project version (such as the phase) along with the status of scans and the BOM.



Scans provides the status of the scans being processed for this BOM. Once the scan completes successfully, an [Up to Date](#) status appears. Select the link to view the **Scans** tab of the *Project Name Version Name Settings* tab. Use this page to manage the scans for this project version.



Status provides the current status of the BOM. It has these possible values:

- [Processing](#). The Black Duck system is processing events to create or update the BOM.
- [Up to Date](#). The BOM is up-to-date; there are no errors.

-  **Error**. An error has occurred while processing an event or the Black Duck system is currently not processing any events and is up-to-date, however an error has occurred.

For  and  statuses, select the link to open the BOM Processing Status dialog box.

BOM Processing Status

Check your BOM events processing status and details as they are submitted by a user or executed by the system. You have visibility of errors that may occur and the ability to dismiss them.

Event	Submitted	Start Time	Elapsed Time	Event Status
Project	 System Administrator Oct 14, 2020 11:13 PM	-	00:00:00	 Queued
Project	 System Administrator Oct 14, 2020 11:13 PM	Oct 14, 2020 11:18 PM	10:45:55	 Processing

Displaying 1-2 of 2



This dialog box lists each event, who submitted it, including the date and time, the time the event started, elapsed time, and current status.

Use this dialog box to see which events are pending or taking a long time to complete. If errors occurred during processing, the BOM Processing Status dialog box notifies you as to which event failed.

BOM Processing Status				
Check your BOM events processing status and details as they are submitted by a user or executed by the system. You have visibility of errors that may occur and the ability to dismiss them.				
X Dismiss All Errors				
Event	Submitted	Start Time	Elapsed Time	Event Status
Project	System Administrator Oct 14, 2020 11:13 PM	-	00:00:00	Queued
Project	System Administrator Oct 14, 2020 11:13 PM	Oct 14, 2020 11:18 PM	10:42:50	Processing
> File Adjustment	System Administrator Oct 14, 2020 11:13 PM	Oct 14, 2020 11:13 PM	-	Error
> File Adjustment	System Administrator Oct 14, 2020 11:13 PM	Oct 14, 2020 11:13 PM	-	Error

Displaying 1-4 of 4

Close

Click > located next to failed events to view the error message. Click to dismiss individual errors or dismiss all errors.

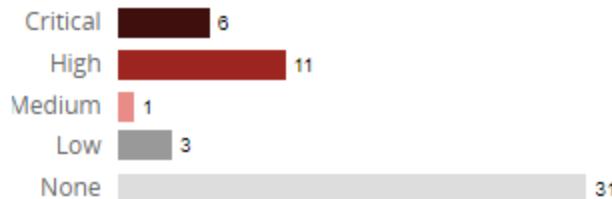
If left open, the dialog box updates the information shown in the table every 30 seconds, otherwise close and reopen the BOM Processing Status dialog box for a fresh update on the status of events.

Refer to the installation guide for information on configuring the frequency of the BOM event cleanup job (VersionBomEventCleanupJob) which clears BOM events that might be stuck because of processing errors or topology changes.

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:



- **Critical** risk: dark red - 50% black and 50% red (security risk only)
- **High** risk: 100% red
- **Medium** risk: light red - 50% red
- **Low** risk: 100% gray
- **None**: light gray - 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](#)), an information icon appears in the table row to indicate that a manual adjustment was made to this component:

		Add ▾	Bulk Actions ▾	Compare to... ▾	Print...	Match Status	Confirmed ▾	Ignore	Not Ignored ▾	Filter components...	Add Filter ▾
□ ▾	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						

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"> •  Policy violation. •  Policy violation in a child component. This icon appears next to the parent component when the child components are not displayed. •  Policy violation has been overridden. •  Policy violation has been overridden at the child component level. •  Component or subproject has not been reviewed. •  Component or subproject has been reviewed.
Component	<p>For subprojects: name and version of the project.</p> <p>Select a subproject version to open the Details 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"> • Select the version number to open the Black Duck KB component version page which displays a list of the projects and project versions in which this version of the component is used. • Select ?, which indicates an unknown version, to open the Black Duck KB component page which provides general information about the component. • Mouse over the version to view the origin and origin ID. <p>Note: 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: </p> <p>For automatic matches, the number of files that were identified in the component scan and matched to this version of the component appears. Select the text to open the Source tab.</p> <p>For parent components, this value does not include child component values.</p> <p>For subprojects: Number of components in the subproject. For example: </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>

Column	Description
Match Type	<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"> • Binary. Scanning identified the binaries in use within your codebase. This match type only appears if Black Duck - Binary Analysis is enabled. • Direct Dependency Binary. Scanning identified that the binaries in use are a direct dependency. • Transitive Dependency Binary. Scanning identified that the binaries in use are a transitive dependency. • Exact Directory. Scanning identified the archive as an exact match to a directory or archive in the Black Duck KB. • Exact File. Scanning identified an exact match to a single file in the Black Duck KB. <p>This match type may appear when scanning Docker images or if you enabled individual file matching.</p> <ul style="list-style-type: none"> • File Dependency. 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. • Files Added/Deleted. 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. • Files Modified. 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. • Direct Dependency. Scanning identified a match to a component in the Black Duck KB via a direct (declared) dependency. • Transitive Dependency. Scanning identified a match to a component in the Black Duck KB via a transitive dependency. • Manually Added. The component was manually added to the BOM. • Manually Identified. 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. <p>The following are automatic matches from an imported Protex BOM:</p> <ul style="list-style-type: none"> • Exact • Partial • File Dependency <p>Click here for more information.</p> <p>The match type for subprojects is Manually Added.</p>
Usage	For components: Indicates how this component is intended to be included in the project when 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

Column	Description
	<p>released version of the project.</p> <p>Tip: To remove components from the project version's risk calculations because they will not be released with the project, exclude them from the BOM.</p> <p>The possible usage statuses are:</p> <ul style="list-style-type: none">• Dynamically linked. A moderately-integrated component that is dynamically linked in, such as with DLLs or .jar files. This is the default value.• Statically linked. A tightly-integrated component that is statically linked in and distributed with your project.• Source Code. Source code such as .java or .cpp files. Could be used when packaging a component's sources with the build, a binary, or distribution; usually due to open source requirements.• Separate Work. 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.• Merely Aggregated. 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.• Implementation of Standard. Intended for cases where you implemented according to a standard. For example, a Java spec request that ships with your project.• Prerequisite. Intended for components that are required but not provided by your distribution.• Dev. Tool / Excluded. 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.• Unspecified. The usage for this component has not yet been determined. You can use Unspecified to indicate that you need to investigate the usage of this component. <p>For subprojects, usage defaults to Dynamically Linked, as described above.</p>

Column	Description
License	<p>Declared license of the component or subproject in use in this version of your project.</p> <ul style="list-style-type: none"> •  indicates that the component/subproject has a high license risk. •  indicates that the component/subproject has a medium license risk. •  indicates that the component/subproject has a low license risk. • (white box) indicates that there is no license risk. <p>For known licenses, select the license name to view license details and license text.</p> <p>For parent components, 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 here for more information on how license risk for a component is determined.</p>
Security Risk	<p>Number of critical/high or 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>Select a value to open the project version page Security tab 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>Note: If you do not have permission to view the project, you will not be able to access this page.</p> <p>For parent components, this column shows the security risk of the parent and all of its children.</p>

Column	Description
	Indicates that this component version has encryption algorithms .
Operational Risk	<p>Operational risk level for the component or subproject in use in this version of your project:</p> <ul style="list-style-type: none"> • High risk • Medium risk • Low risk <p>The operational risk level in this version of your project is calculated using a combination of:</p> <ul style="list-style-type: none"> • 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. • 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. <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> <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>Note: 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 parent and child components, use the component list view to hover over the value to obtain more</p>

Column	Description
	information.

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	[H] GNU General Public License v2.0 only and 5 more...	2 1 1	Low
ceph 13.2.1	12 Matches	Files Modified	Dynamically Linked	[M] GNU Lesser General Public License v2.1 or later	2 1	Low
+ ceph v14.0.0	104 Matches	Exact Directory, Files Added/Deleted, Files Modified	Dynamically Linked	[H] Creative Commons Attribution Share Alike 3.0 and 7 more...	2 1 1	Low
Chart.js ?	1 Match	Direct Dependency	Dynamically Linked	[W] Unknown License	2 1 1	Low
core.js ?	1 Match	Direct Dependency	Dynamically Linked	MIT License	2 1 1	Low
Flot 0.8.3	2 Matches	Exact File	Dynamically Linked	MIT License	2 1 1	High

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	2 1 1	High
	Apache XML Commons 1.0.b2	1 Match	Files Added/Deleted	Statically Linked	Apache License 1.1	2 1 1	High
	Commons IO 1.1	1 Match	Exact Directory	Statically Linked	Apache License 2.0	2 1 1	High
	db-charmer 1.6.10	1 Match	Exact Directory	Statically Linked	MIT License	2 1 1	High
-	Mercurial Toolbar Initial Release	1 Match	Exact Directory	Statically Linked	[H] GNU General Public License v2.0 or later	2 1 1	High
	Mercurial Toolbar Initial Release	1 Match	Exact File	Statically Linked	[H] GNU General Public License v2.0 or later	2 1 1	High

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.



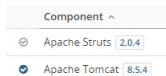
- As this hierarchical view displays components based on file relationships, components that were manually added in the component list view do not appear in this view of the BOM.
- The number of archives or files that match, as identified in the **Source** column, applies to the parent component only. 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:



- - 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 `components_date_time.csv` and the `bom_component_custom_fields_date_time.csv` files in the [Project Version report](#) include the review status, the username of reviewers, and the review date.

- Changing the review status does not cause the [Information icon](#) (ⓘ) 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.

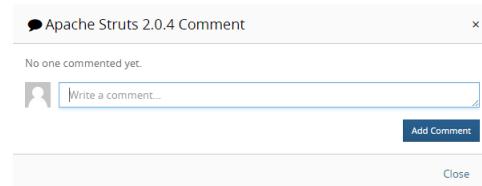
Note:

- 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	Match Count	Match Type	Usage	License	Security Risk	Operational Risk
Apache Struts 2.0.4	0	Manually Added	Dynamically Linked	Apache-2.0		

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](#).
- [Reviewing detected embedded licenses](#).

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 name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the version name to open the **Components** tab and view the BOM.
3. Do one of the following:
 - Select the **Source** tab to view all files in this BOM.

No Folders or Filters Selected
Select a node from the file tree or one of the filters above to begin navigating the content of your project.

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	Discovery Types
org.apache.struts.xwork:xwork-core:2.3.7	Apache Struts 2.3.7	Transitive Dependency	Apache-2.0		
org.apache.struts.struts2-core:2.3.7	Apache Struts 2.3.7	Direct Dependency	Apache-2.0		

Displaying 1-2 of 2

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.

Selecting to view an archive:

Name	Component	Match Type	License	Usage	Discovery Types
com.sun.grizzly.grizzly-framework:1.9.14	grizzly-framework 1.9.14	Transitive Dependency	CDDL-1.0	Dynamically Linked	

Displaying 1-1 of 1

Selecting to view a folder:

Name	Component	Match Type	License	Usage	Discovery Types
:travis.yml					
AcmeAuthor.java					
AcmeConfig.java					
AuthorsConfig.java					
README.md					
UserService.java					
WEB-INF					
acme					
azure-pipelines.yml					
bootstrap.js	Bootstrap (Twitter) 3.3.2	Exact File	MIT	Dynamically Linked	

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 appear above the table.

Click and select **Copy path** to copy the path to your clipboard.

- A table which provides the following information on the item selected in the pane.
 - **Name.**
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.
 - **Discovery Types.** Indicates the type of discovery. Possible values of License and License Reference are for [embedded licenses detected during the scan](#).
- Filters located above the table, to filter the information shown on the tab.
- Check box located above the table, to view subfolder information. Select **All Subfolders** to include information on all subfolders and files.

- **Files/Discoveries** tab to view files or discoveries. Select **Discoveries** to view [embedded license information](#) detected in the scan.

The tab uses the following icons:

- Package manager scan/archive
- or Signature scan/directory
- Empty dependency tree - which contains no OSS matches
- or File
- Snippet information. Click [here](#) for more information.
- Source file. Used when [reviewing snippet matches](#) and [detected embedded licenses](#).

Modifying matches

To modify a match

1. Open the **Source** tab as described above.
2. Select one or more items in the table and click 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 . 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 **Add filter ▾** and select **Match Type > Type of match(es)** and click **OK**.
3. Select one or more entries and click . 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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the version name to open the **Components** tab and view the BOM.
This is the "current" version of the BOM.
3. 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 components. Below this is a table with 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 (Added, Removed, New), usage (Dynamically Linked), license (e.g., The Open SSL License and 1 more..., MIT License, PCRE License, curl License), and security risk levels (represented by red, orange, and green bars).

Component	Version	Changes	Usage	License	Security Risk
OpenSSL	1.0.1e	Removed	Dynamically Linked	The Open SSL License and 1 more...	[Red: 13, Orange: 64, Green: 17]
GNU C Library	2.17	Removed	Dynamically Linked	GNU Lesser General Public License v2.1 or later	[Red: 12, Orange: 28, Green: 8]
BIND (Berkeley Internet Name Domain)	9.10.1	New	Dynamically Linked	Bind License	[Red: 10, Orange: 15, Green: 3]
libxml2-python	2.9.1	Removed	Dynamically Linked	libxml2 License	[Red: 8, Orange: 35, Green: 2]
libxml2	2.9.1	Removed	Dynamically Linked	MIT License	[Red: 6, Orange: 36, Green: 11]
PCRE	8.32	Removed	Dynamically Linked	PCRE License	[Red: 6, Orange: 9, Green: 4]
cURL	7.40.0	Added	Dynamically Linked	curl License	[Red: 5, Orange: 21, Green: 14]
curl	7.15.3	Added	Dynamically Linked	curl License	[Red: 4, Orange: 25, Green: 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	Possible values are: <ul style="list-style-type: none">• Added. 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.• Modified. The usage or license for this component/subproject version has changed.• New. The component or subproject is new - it was not in the "compared to" version of the BOM.• Removed. The component/subproject was in the "compared to" version of the BOM, however, it is not in the "current" version of the BOM.• Replaced. 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.

Column	Description										
	<p>For modifications to ignored components:</p> <ul style="list-style-type: none"> Components ignored in both versions are not compared. Components ignored in the "compared to" version but not ignored in the "current" version have a value of Added. Components ignored in the "current" version but not ignored in the "compared to" version have a value of Removed. <p>Note that for a modification to the version:</p> <ul style="list-style-type: none"> The component/subproject and original version are shown with Replaced as the value in the Changes column. The component/subproject and new version are shown with Added as the value in the Changes column. <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							
Usage	<p>Usage 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>										
License	<p>Declared license 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>										
Security Risk	<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 Security Risk column indicates an increase or decrease in security risk depending on the value in the Changes column. If the value in the Changes column is:</p> <ul style="list-style-type: none"> Removed or Replaced. The value indicates a decrease in security risk from the "compared to" version of the BOM. New, Modified, or Added. The value indicates an increase in security risk from the "compared to" version of the BOM. 										

Printing a BOM

You can print a BOM.

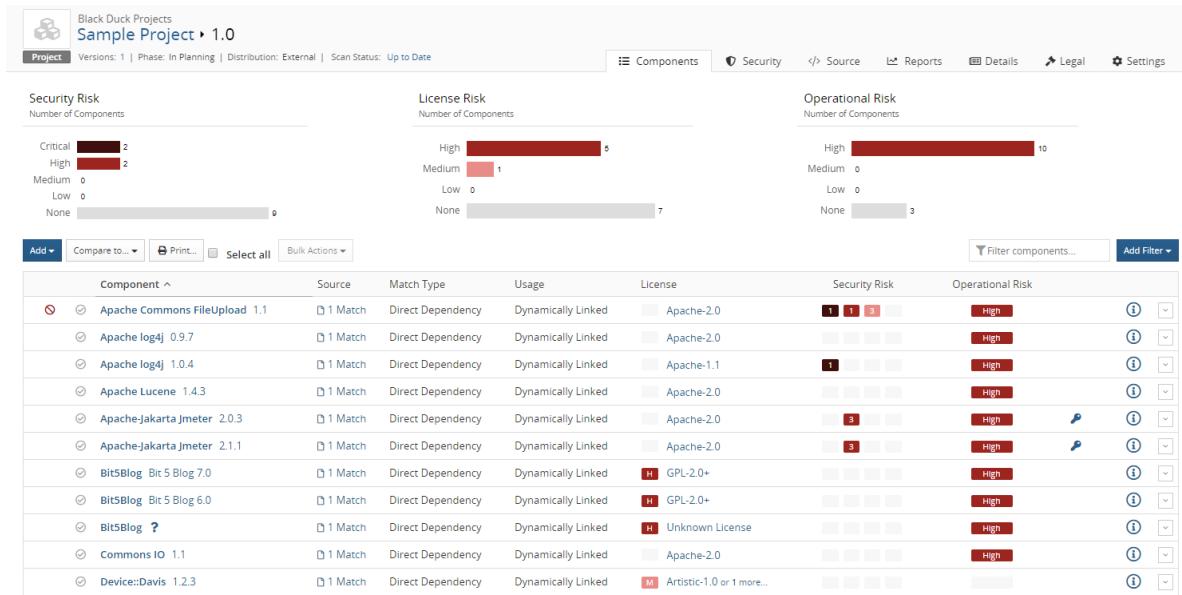
The printout displays the BOM similar to what is shown in the UI: security, license, and operational risk graphs appear at the top of the page; component and subproject information is listed in a table.

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 name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the version that you want to view.

The **Components** tab displays the BOM.



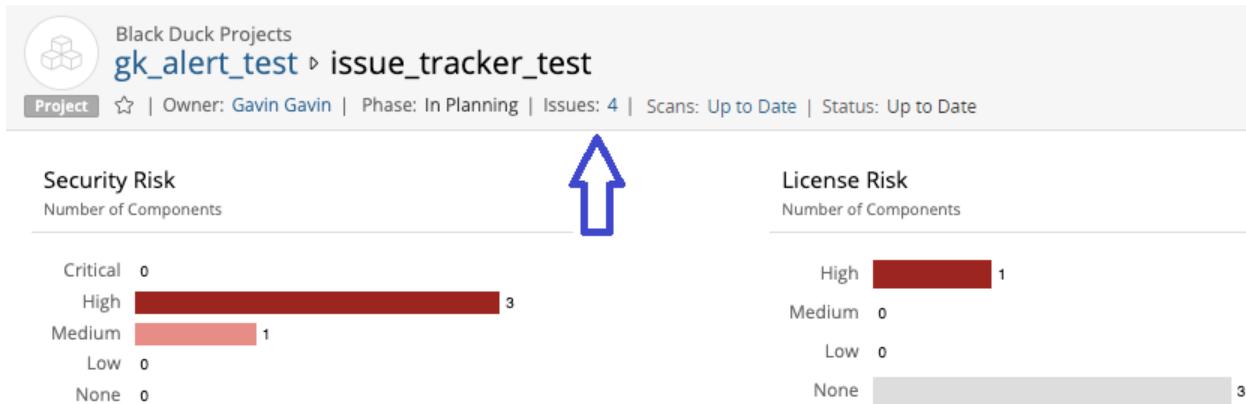
3. Optionally, filter the BOM so that the printout only shows the information you want to see.

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

Viewing issues in a project

Black Duck provides information on the issues associated with a project version as monitored by an issue tracking system. Currently, this feature is supported using Synopsys Alert 6.2.0 and later.

Black Duck displays an **Issues** link in the project version header for a project version if an issue tracking system was configured to the Black Duck project version using Synopsys Alert. Once Synopsys Alert creates issues for this project version, the link appears. No additional configuration is needed.



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

Users with the Super User [role](#) and all project members (users assigned to the project) can select the **Issues** value to display the Issue Management table.

Issue Management					
Component	ID	Summary	Assignee	Status	Updated
Apache Struts 1.3.5	ALERT-3	Alert - Provider: Black Duck, Project: gk_alert_test, Project Version: issue_tracker_test, Component: Apache Struts, Component Version: 1.3.5, Vulnerability	Alert User	Created by Alert	Never
Apache Struts 1.3.8	ALERT-4	Alert - Provider: Black Duck, Project: gk_alert_test, Project Version: issue_tracker_test, Component: Apache Struts, Component Version: 1.3.8, Vulnerability	Alert User	Created by Alert	Jul 16, 2020
Apache Struts 1.3.5	GALERT-689	Alert - Provider: Black Duck, Project: gk_alert_test, Project Version: issue_tracker_test, Component: Apache Struts, Component Version: 1.3.5, Vulnerability	Alert User	Created by Alert	Never
Apache Struts 1.3.8	GALERT-690	Alert - Provider: Black Duck, Project: gk_alert_test, Project Version: issue_tracker_test, Component: Apache Struts, Component Version: 1.3.8, Vulnerability	Alert User	Created by Alert	Jul 16, 2020

Displaying 1-4 of 4

[Close](#)

This table lists the issues created in the external issue tracking systems. You can then use this table to see the status of the issue in your workflow.

The table provides the following information for each issue:

Column	Description
Component	Component name and version affected by this ticket.
ID	Issue identifier.
Summary	Summary of the external issue.
Assignee	User assigned to this ticket.
Status	Status of the ticket.
Updated	Time when this ticket was last updated.

Note that this table does not display all changes from the external issue tracker system. Changes to the issue, such as manual changes to the description, will not be reflected in the Black Duck issue table other than those changes made automatically to the issue by Synopsys Alert.

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 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.

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

The screenshot shows the Black Duck KnowledgeBase interface. At the top, it displays "KnowledgeBase" and "Apache-Jakarta Jmeter • 2.0.3". Below that, it says "Versions: 128". The navigation bar includes tabs for "Security", "Cryptography" (which is highlighted in purple), "Copyrights", "Details", and "Settings". The main content area is titled "Cryptography" and contains the following text: "This is a list of all the potential algorithms Apache-Jakarta Jmeter - 2.0.3 can use, but it doesn't necessarily mean they are in use by this project." A table is shown with one row for "NTLM". The table has columns for "Algorithm" (NTLM), "Description" (Also known as Unicode Hash and NT Password Hash. Used by Windows NT and XP to create password hashes. Simply the MD4 hash of the password.), "Key Lengths" (Key Lengths: User Definable Key Length Unconstrained), and "Originator" (Microsoft). A note at the bottom of the table says "Displaying 1-1 of 1".

- 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.

Chapter 7: Editing a BOM

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:

		Add	Bulk Actions	Compare to...	Print...	Match Status	Confirmed	Ignore	Not Ignored	Filter components...	Add Filter
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	High	Info					

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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the **Settings** tab.

The screenshot shows the 'Settings' tab of the 'Sample Project 1' configuration page. The page is divided into several sections:

- Project Details:** Includes fields for 'Project Name' (Sample Project 1), 'Description', 'Owner' (dropdown), and 'Tier' (dropdown).
- Component Adjustments:** A note about archived versions and manually added components, with a checked checkbox for 'Always maintain component adjustments to all versions of this project'.
- Cloning:** A section for selecting attributes to clone for new versions, with all checkboxes checked: 'Additional Fields', 'Component Edits', 'License Fulfillment Status', 'Remediation Details', and 'Version Settings'.
- Custom Scan Signature:** A note about identifying third-party software, with an unchecked checkbox for 'Enable Custom Scan Signature'.
- Depth:** A dropdown menu set to '5' for the number of levels in the directory structure for custom signature scanning.
- Deep License Data:** A note about applying deep license data to components, with an unchecked checkbox for 'Apply Deep License Data to bill of materials'.
- License Conflicts:** A note about applying license conflicts data to components, with an unchecked checkbox for 'Apply License Conflicts Data to bill of materials'.
- Save Buttons:** Two blue 'Save' buttons located at the bottom of each main section.
- Additional Fields:** A section for managing data sensitivity, with a dropdown menu for 'Data Sensitivity' and a note about containing important confidential information.
- Application ID:** A section for storing an external mapping ID, with a text input field and a note about its purpose.
- Clone Project:** A section for cloning project settings, members, groups, additional fields, component edits, and application ID, with a 'Clone Project' button.
- Delete Project:** A section for deleting the project, with a note about losing all information and a red 'Delete Project' button.

4. Do one of the following in the **Component Adjustments** area of 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.
5. Click **Save**.

Manually adding a component to a BOM

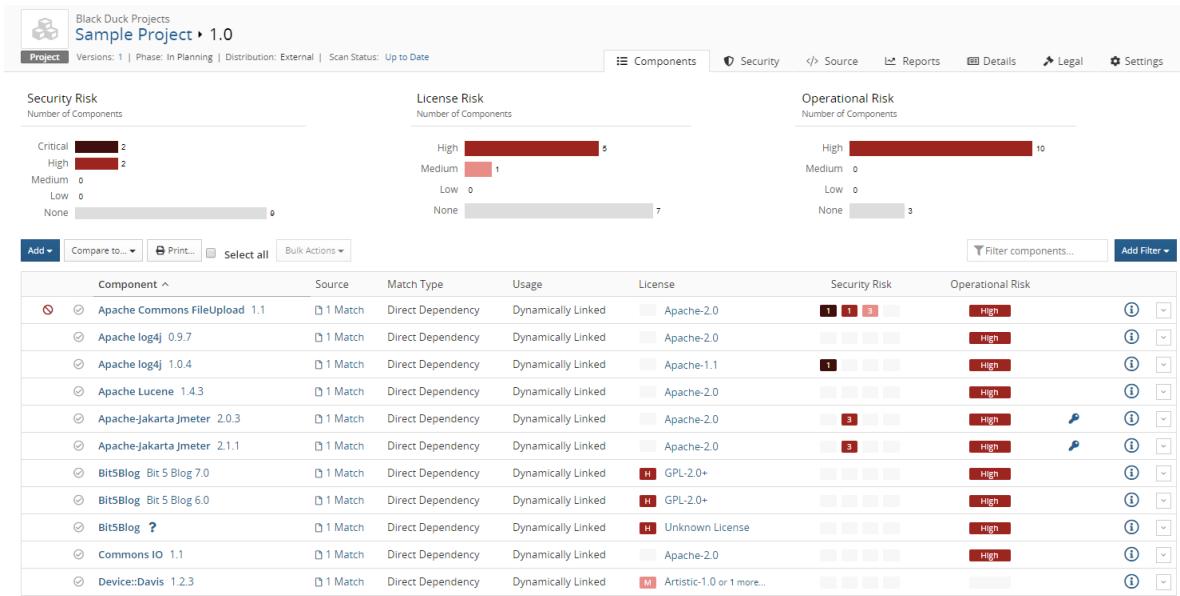
Once you have mapped a component scan to a project version, the scan results automatically populate the project version's BOM with the discovered components. Although the BOM contains all the components discovered in the mapped scan, there may be other components that you are using in that version of 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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab.



- Click **Add** and select **Component** to open the Add Component dialog box.
- Enter the name of the component that you want to add.
- Optionally, enter or select a version and an origin ID.
- 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.
- 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
- Unspecified

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

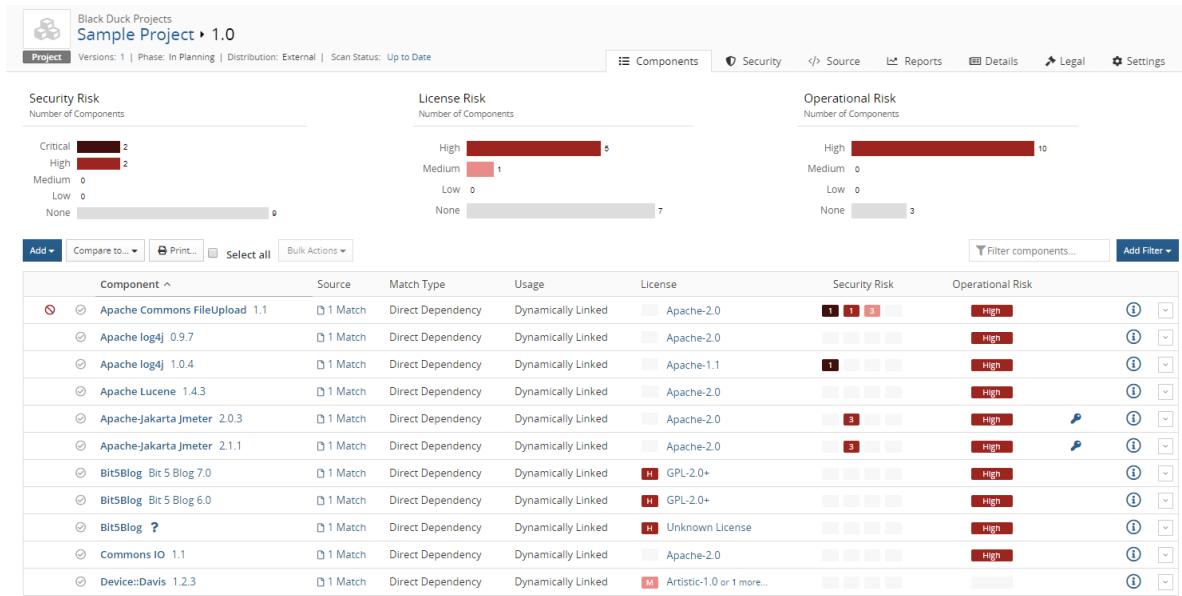
You can change a component's usage to indicate that it is not included in the project version's BOM because it is not actually being distributed with the released project version. For example, if scanning identified development tools in scanned code or a Docker image mapped to the project version, but 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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab and view the BOM.



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

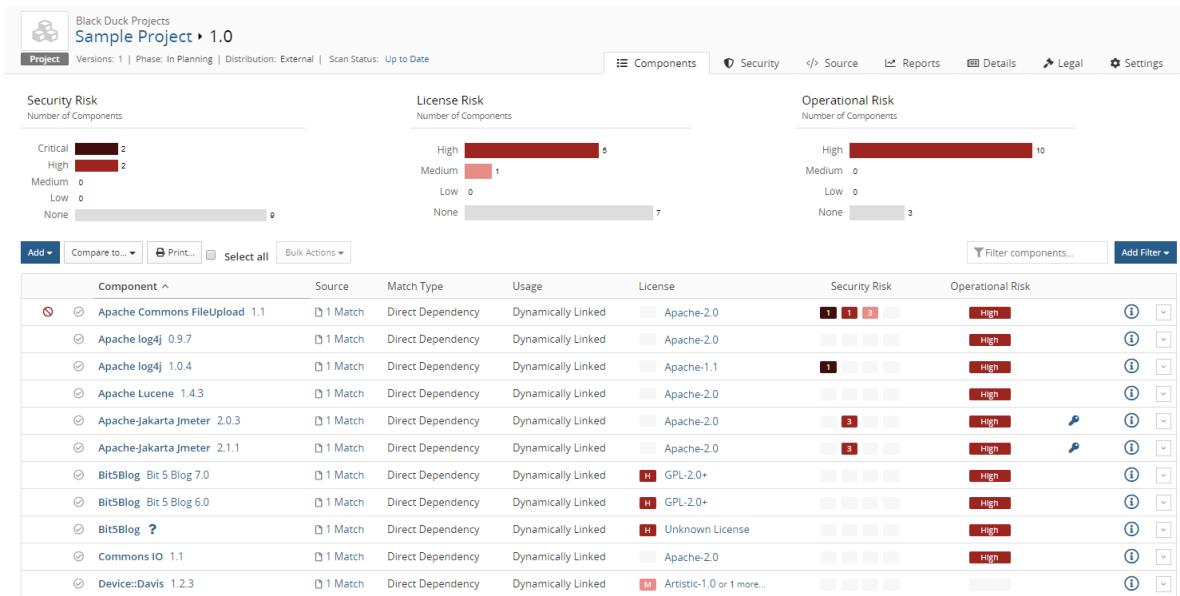
Common reasons to delete a component that was added manually include:

- 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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab.



4. In the List view of the BOM, click and select **Delete** to open the Delete Component dialog box.
5. 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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab and view the BOM.

The screenshot shows the 'Components' tab of a project named 'Sample Project'. It displays three risk distribution charts: Security Risk (with categories Critical, High, Medium, Low, None), License Risk (with categories High, Medium, Low, None), and Operational Risk (with categories High, Medium, Low, None). Below these charts is a table listing various dependencies with columns for Component, Source, Match Type, Usage, License, Security Risk, and Operational Risk. Each row includes a checkbox and a 'Details' link.

Component	Source	Match Type	Usage	License	Security Risk	Operational Risk
Apache Commons FileUpload 1.1	1 Match	Direct Dependency	Dynamically Linked	Apache-2.0	1 (High)	High
Apache log4j 0.9.7	1 Match	Direct Dependency	Dynamically Linked	Apache-2.0	0 (Medium)	High
Apache log4j 1.0.4	1 Match	Direct Dependency	Dynamically Linked	Apache-1.1	1 (High)	High
Apache Lucene 1.4.3	1 Match	Direct Dependency	Dynamically Linked	Apache-2.0	0 (Medium)	High
Apache-Jakarta Jmeter 2.0.3	1 Match	Direct Dependency	Dynamically Linked	Apache-2.0	3 (Medium)	High
Apache-Jakarta Jmeter 2.1.1	1 Match	Direct Dependency	Dynamically Linked	Apache-2.0	3 (Medium)	High
Bit5Blog Bit 5 Blog 7.0	1 Match	Direct Dependency	Dynamically Linked	GPL-2.0+	0 (Medium)	High
Bit5Blog Bit 5 Blog 6.0	1 Match	Direct Dependency	Dynamically Linked	GPL-2.0+	0 (Medium)	High
Bit5Blog ?	1 Match	Direct Dependency	Dynamically Linked	Unknown License	0 (Medium)	High
Commons IO 1.1	1 Match	Direct Dependency	Dynamically Linked	Apache-2.0	0 (Medium)	High
Device::Davis 1.2.3	1 Match	Direct Dependency	Dynamically Linked	Artistic-1.0 or more...	0 (Medium)	High

4. 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.

The screenshot shows the 'Scans' tab of the project. It displays a summary message: 'Your project version includes 1 scan with 1.19 MB of code scanned.' Below this is a table with columns for Status, Name, Scan Size, and Last Updated. A single row is listed: 'ComplexBomMainProject_2015-12-04 10:28:23' with a status of 'Completed', a scan size of '1.19 MB', and a last updated date of 'May 29, 2019'.

Status	Name	Scan Size	Last Updated
Completed	ComplexBomMainProject_2015-12-04 10:28:23	1.19 MB	May 29, 2019

5. 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.

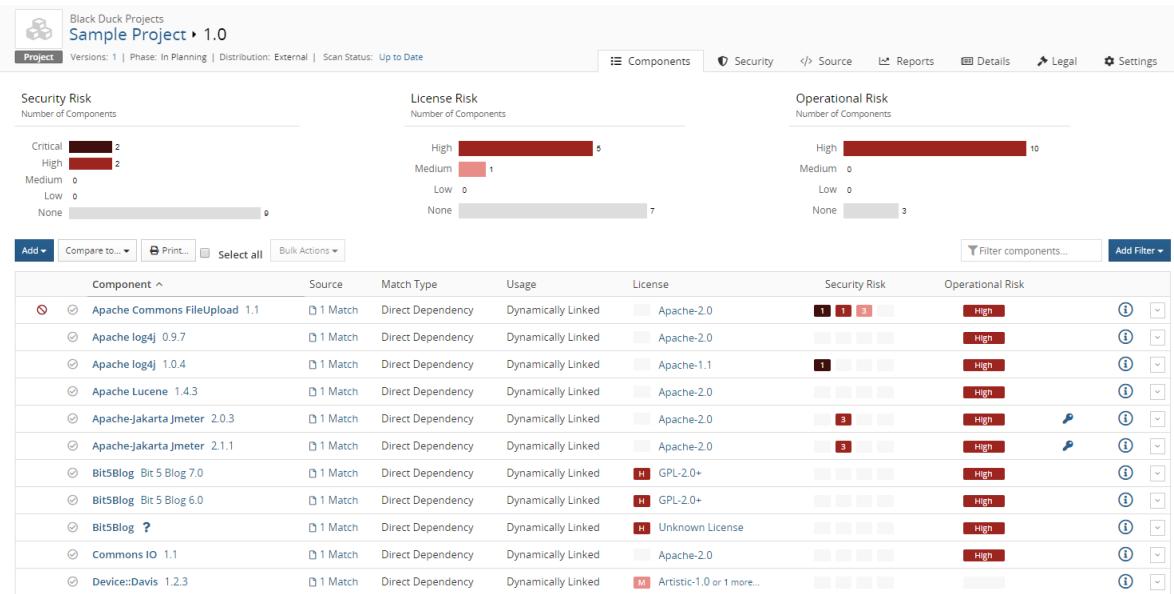
Ignoring a component is considered a component adjustment. Therefore, if you selected to apply [persistent edits](#), ignoring a component applies to all versions of the project.

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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab and view the BOM.



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

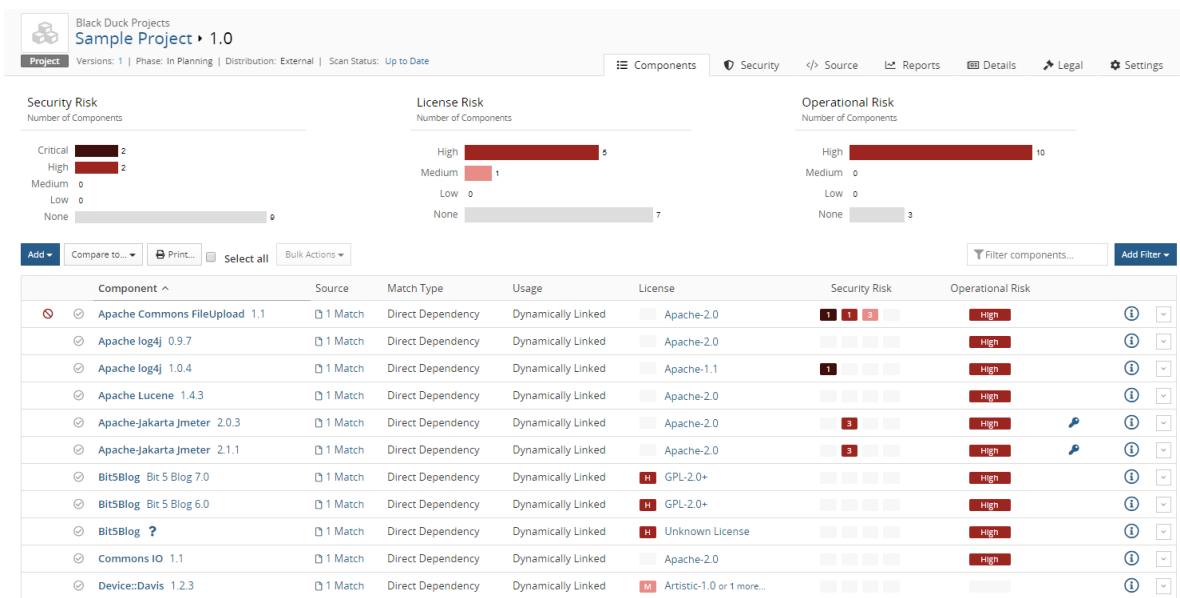
Adjusting the component and/or component version in a BOM

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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab and view the BOM.



4. In the component list view of the BOM, click and select **Edit** to open the Edit component dialog box.
5. Type the name of the OSS component in the **Component** field and select the alternate match.
6. 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.
7. Optionally, enter a purpose for this adjustment and/or select the **Modification** checkbox and optionally, enter information regarding this modification in the field.
8. Click **Save**.

The component and version for the BOM entry are updated. The Information 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:

		Match Status: Confirmed					Ignore Not Ignored		Filter components...		Add Filter
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	1	High					

Editing an origin or origin ID

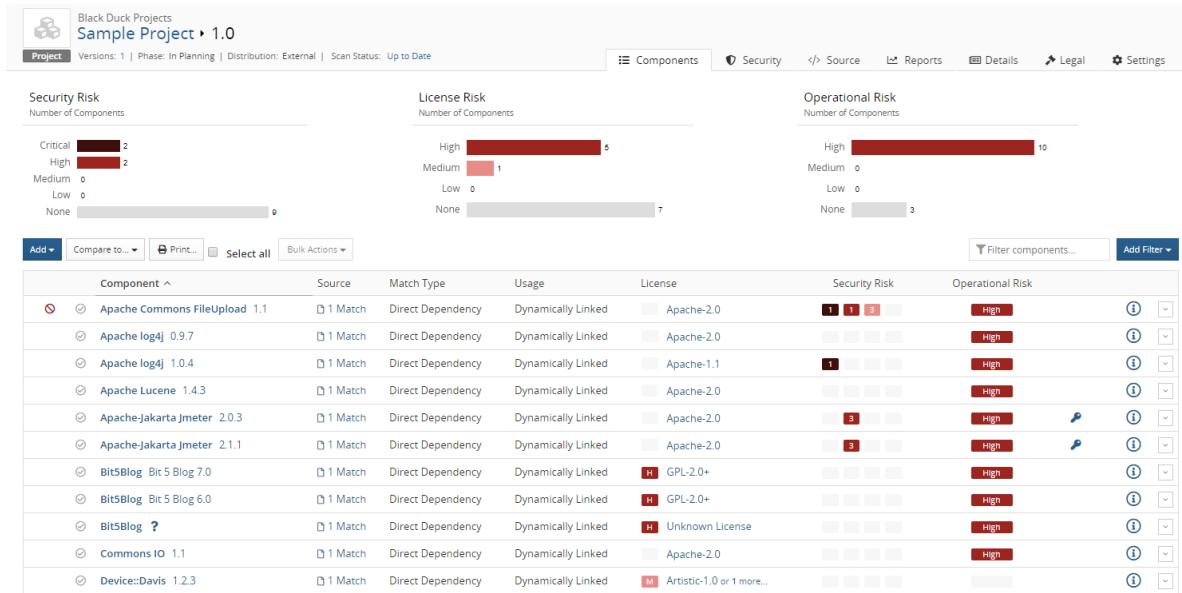
You can select a different origin or origin ID shown for a Linux distribution and used in a project version's BOM.

To select a different origin or origin ID

1. Log in to Black Duck.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page

appears.

- Select the version name to display the **Components** tab and view the BOM.



- In the component list view of the BOM, click and select **Edit** to open the Edit component dialog box.
- 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.
- 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.

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

Modifying licenses in a BOM

So that you can successfully manage license risk, you may need to edit the license(s) for a component version used in a BOM so that it is different from the component's declared license identified in the Black Duck KB.

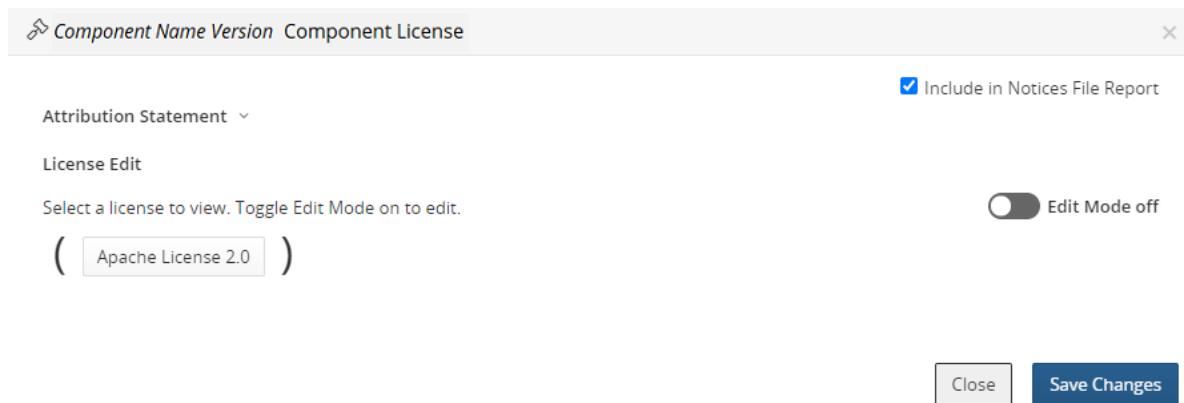
You can modify a single license or include multi-license scenarios, such as "License A AND License B" or "License A OR License B". This lets you accurately represent the licenses in Black Duck for the components in your projects

If you have modified a license, you can select to revert it back to the license as defined by the Black Duck KnowledgeBase.

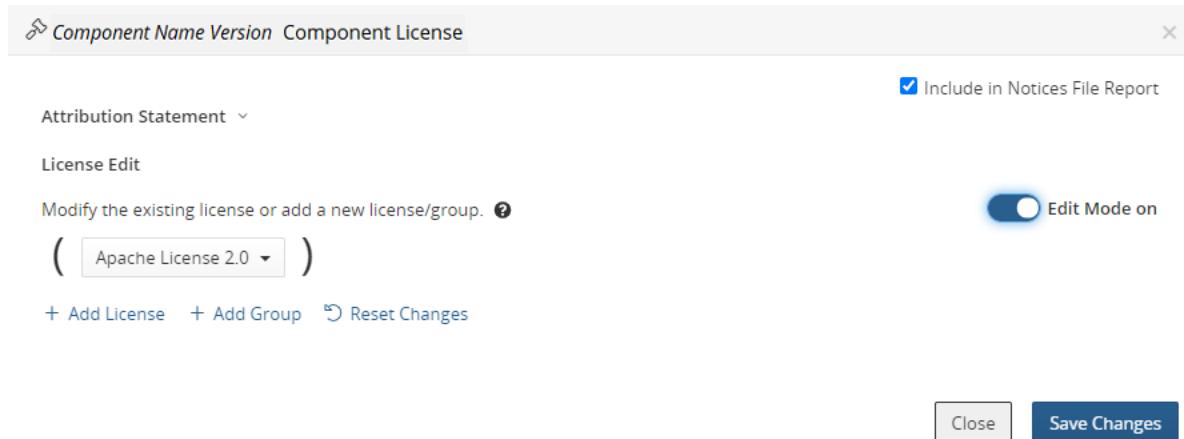
Note: Edits made to a license in the BOM are *local* edits. These edits apply to this version of the component in this BOM only.

To modify licenses

1. Log in to Black Duck.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab and view the BOM.
4. Select the single license or multi-license to open the *Component Name Version Component License* dialog box.



5. Select the Edit Mode option to enable editing.



6. Edit the license as described [here](#).

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 License Fulfillment tab on 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, and if necessary, the **Term Fulfilment** tab, to view a list of license terms that require fulfillment.

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

Displaying 1-3 of 3

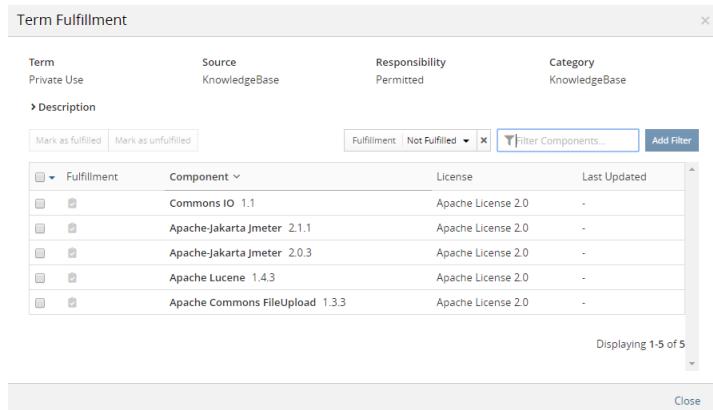
By default, the **Legal** tab is filtered to show all license terms that are not fulfilled.

The tab displays the following information:

Column	Description
Fulfillment	Indicates fulfillment status: <ul style="list-style-type: none"> indicates this license term is not fulfilled. indicates this license term is fulfilled.
Term Name	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.
Responsibility	Indicates the responsibility for this term. Possible values are Required, Forbidden, or Permitted.
Category	Category 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.



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.

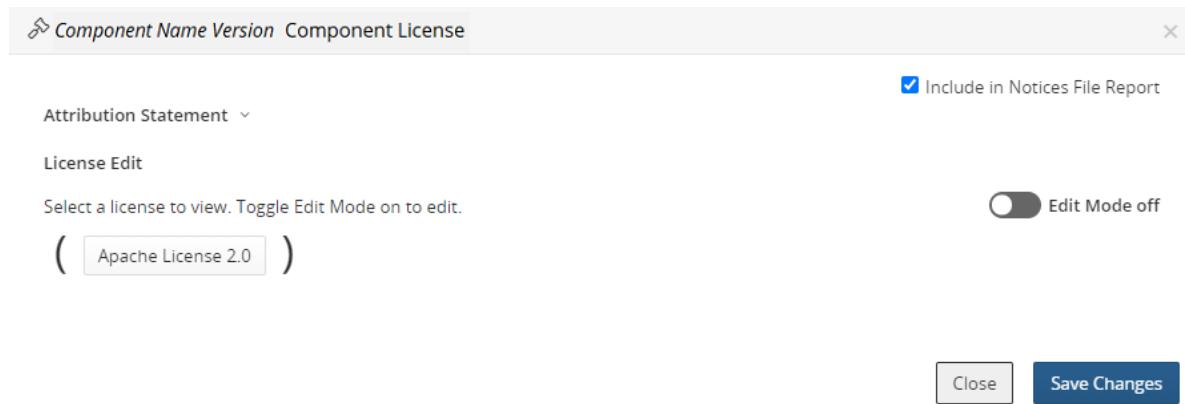
Updated by System Administrator - 11:16 AM 

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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
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.



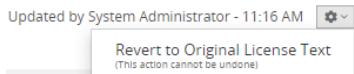
5. Select the license you wish to edit.

The dialog box expands to show the obligations and license text for the selected license.

6. Edit the text directly in the field.
7. 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:

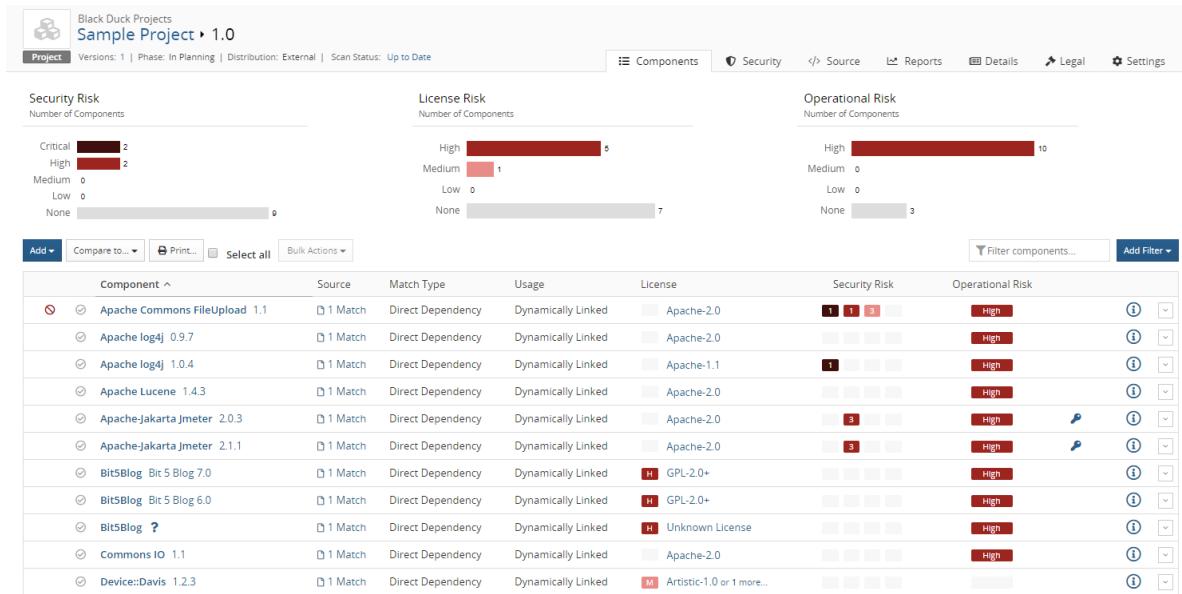
- If you modify the project version license outside of the BOM and have *not* edited the subproject license from within the BOM, the edited license will appear in the BOM for the subproject.
- If you modify the project version license outside of the BOM and have edited the subproject license from within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

To add a project

1. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the version name to open the **Components** tab.



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

4. Enter the name and version of the project.

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

5. 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.

6. 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 interface for a project named 'KBHUB-262 • 032117'. The 'Components' tab is selected. In the top right, there are tabs for 'Components', 'Security', 'Source', 'Reports', 'Details' (which is selected), and 'Settings'. Below the tabs, the 'Where Used' table is displayed. It has columns for Project, Version, Tier, Released, Distribution, and Phase. One row is shown: 'Sample Project 1' with Version '1.0', Tier 'Never', Released 'External', and Phase 'In Planning'. To the right of the table, there are sections for Description ('No description.'), Released on ('Unknown'), Licenses ('Unknown License'), and Tags ('No Tags'). At the bottom right of the table area, it says 'Displaying 1-1 of 1'.

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

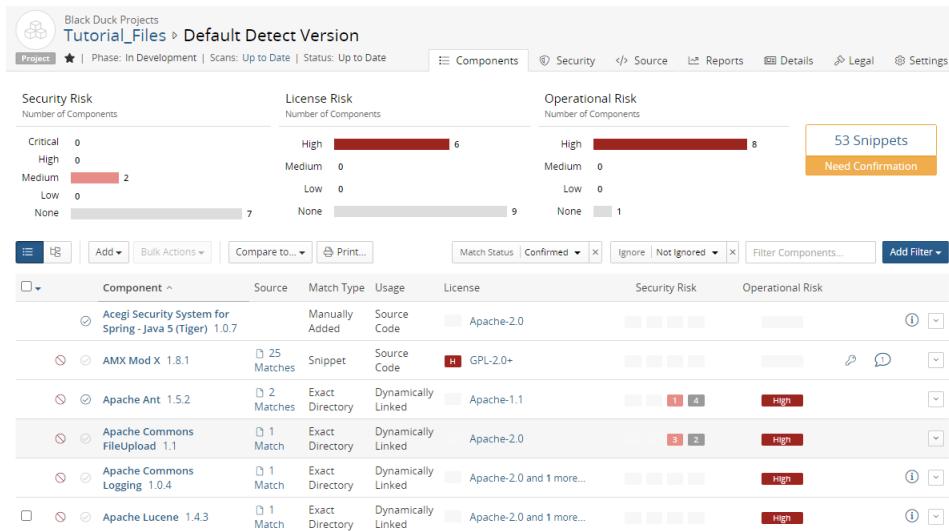
If a snippet scan has been run and snippet matches were found, a snippet badge appears next to the risk charts in the BOM indicating the number of snippets that need confirmation.



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.

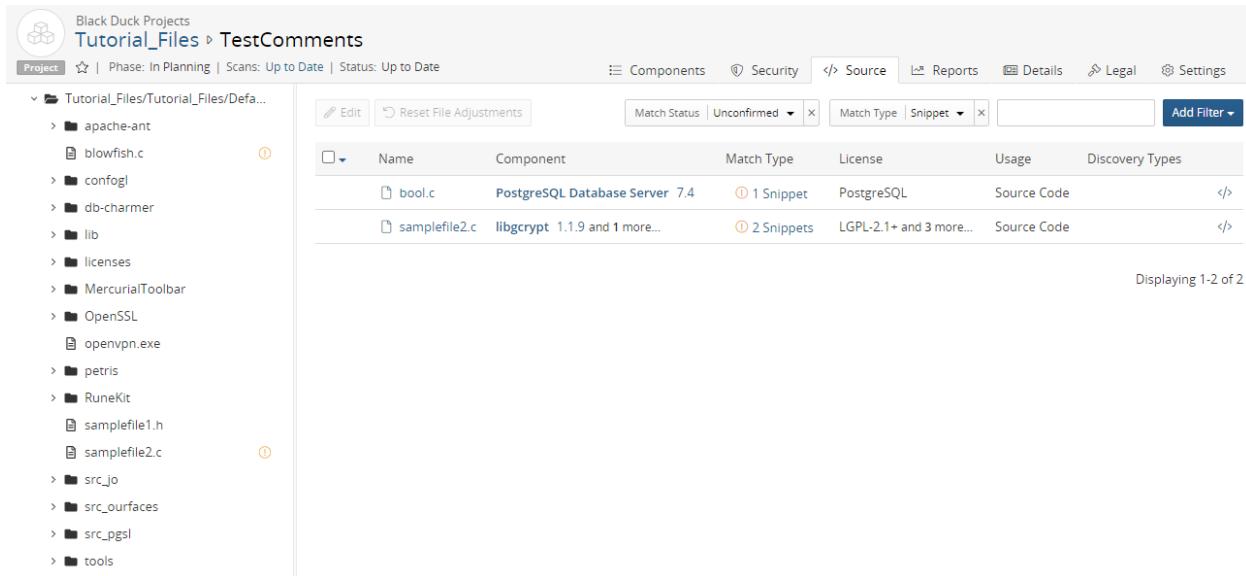
You can filter the BOM to view unconfirmed snippet matches by selecting the **Unconfirmed** option for the **Match Status** filter and the not ignored snippet matches by selecting the **Not Ignored** option for the **Ignored**

filter.



Viewing snippet matches in the Source tab

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



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.
 - indicates the location of an unconfirmed snippet.
- The table provides information, such as the name, component, match type, license, and usage.
 - ○ indicates an unconfirmed snippet match.

- ⓘ indicates an ignored snippet match.
- ⓘ indicates a confirmed 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 a window titled "Source Code View". The left pane displays the Java code for Cache.java, and the right pane shows the corresponding component definition from the PostgreSQL Database Server 7.4 database. The component definition includes copyright information and two sections of code, each with some lines highlighted in yellow to indicate matches.

```

Source Code View
File
Cache.java
file:///Users/eford/Downloads/Tutorial_Files/src_jo's%20files/util/Cache.java
Size: 3.46 KB

1 ---dxiINQHHS2kBShdjEWdCrgNW15exrfz0480mB
2 Content-Disposition: form-data; name="file"; filename="Cache.java"
3 Content-Type: text/x-java-source
4 Content-Length: 3545
5
6 /*
7 * $Id: Cache.java,v 1.7 2003/11/07 20:16:25 dfe 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

```

- Clicking #snippet 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 a window titled "Snippet View". It compares a scanned file named "ascii.c" with a PostgreSQL component. The left pane shows the source code for "ascii.c", and the right pane shows the corresponding PostgreSQL component. Both panes have their own scroll bars. Lines of code are highlighted in yellow to indicate matches between the two components.

Scanned File	Matched Component
ascii.c Scanned File Path <code>file:///Users/florac/Downloads/Tutorial_Files_60/Tutorial_Files/src_pgsql/ascii.c</code> File Size: 3.22 KB	PostgreSQL Database Server 7.4 License: PostgreSQL License Release Date: Nov 16, 2003 Matched File Path <code>/postgresql-7.4/src/backend/utils/adt/ascii.c</code> Snippet Match: 100%
<pre> 1 /* 2 * ascii.c 3 * The PostgreSQL routine for string to ascii conversion. 4 */ 5 * Portions Copyright (c) 1999-2003, PostgreSQL Global Development Group 6 * 7 * IDENTIFICATION 8 * \$Header: /cvsroot/pgsql-server/src/backend/utils/adt/ascii.c,v 1. 9 * 10 */ 11 #include "postgres.h" 12 13 #include "utils/builtins.h" 14 #include "mb/pg_wchar.h" 15 #include "utils/ascii.h" 16 17 static void pg_to_ascii(unsigned char *src, unsigned char *src_end, 18 unsigned char *dest, int enc); 19 static text *encode_to_ascii(text *data, int enc); 20 </pre>	<pre> 1 /* 2 * ascii.c 3 * The PostgreSQL routine for string to ascii conversion. 4 */ 5 * Portions Copyright (c) 1999-2003, PostgreSQL Global Development Group 6 * 7 * IDENTIFICATION 8 * \$Header: /cvsroot/pgsql-server/src/backend/utils/adt/ascii.c,v 1. 9 * 10 */ 11 #include "postgres.h" 12 13 #include "utils/builtins.h" 14 #include "mb/pg_wchar.h" 15 #include "utils/ascii.h" 16 17 static void pg_to_ascii(unsigned char *src, unsigned char *src_end, 18 unsigned char *dest, int enc); 19 static text *encode_to_ascii(text *data, int enc); 20 </pre>

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:

The screenshot shows the 'Snippet View' window. On the left, under 'Scanned File', it lists 'config.com' with its file path: 'file:///Users/calvin1/Desktop/snippet-scanning-example/config.com' and file size: '2.45 KB'. On the right, under 'Matched Component', it shows 'OpenSSL 1.1.0-pre4' with its release date: 'Mar 16, 2016'. Below this, there's a status message: 'Needs confirmation'. A large text area labeled 'Matched Lines: 1 - 77' displays a snippet of C code from OpenSSL. The code is mostly black, with some lines highlighted in yellow, indicating they are matches. At the bottom of the code area, there are navigation buttons: 'Previous Match' (disabled), 'Next Match', and 'Close'.

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.
 - Match file path.
 - Percentage of the scanned file that matches the component file.



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 Match**
 - **Ignore Match**
 - **Unignore Match**

To review a snippet match

1. In the **Source** tab, select **#snippet** 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. Select one of these options:

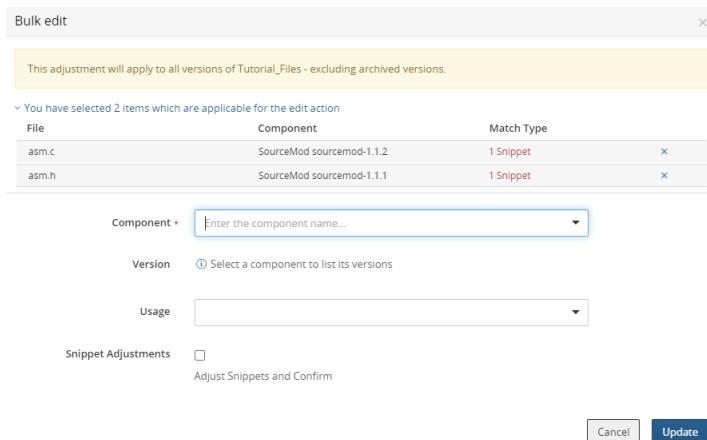
- **Confirm**
- **Ignore Match**
- **Unignore Match**

To bulk edit snippet matches

1. Select more than one snippet match.

2. Click .

The Bulk edit dialog box appears.



This adjustment will apply to all versions of Tutorial_Files - excluding archived versions.

File	Component	Match Type
asm.c	SourceMod sourcemod-1.1.2	1 Snippet
asm.h	SourceMod sourcemod-1.1.1	1 Snippet

Component +

Version

Usage

Snippet Adjustments Adjust Snippets and Confirm

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**.

Retaining partial snippet identifications

By default, identifications you made to partial snippet matches are not retained in subsequent snippet rescans.

You can change this default setting so that you can minimize the number of snippet matches you need to re-identify: in the project's **Settings** tab, in the **Snippet Adjustments** section, select **Apply IDs from partial snippet matches to new exact file matches**.

Chapter 8: Managing components

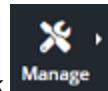
Users with the [Component Manager role](#) can:

- 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 **Manage** > **Component Management**.

The **Components** tab appears.

A screenshot of 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 is a toolbar with 'Add', 'Filter components...', and 'Add Filter' buttons. The main area 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 [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'.

The table in **Components** tab contains the following information:

Column	Description
Component	<p>Name of the component.</p> <p>Select the component name to open the Overview tab of the <i>Component Name</i> page.</p> <p>If there are multiple versions for this component, select > to display the versions.</p> <p>Select a version to open the Details tab of the <i>Component Name > Version</i> page.</p>  indicates that there is a note for this component or component version. Hover over the icon to view the information.
License	License for this component.
Source	<p>Source for this component. Possible values are:</p> <ul style="list-style-type: none"> • Custom. A custom component. • KnowledgeBase. An unmodified Black Duck KnowledgeBase component. • Modified KnowledgeBase. A modified Black Duck KnowledgeBase component.
Approval Status	<p>Approval status of this component. Possible values are:</p> <ul style="list-style-type: none"> • Unreviewed • In Review • Reviewed • Approved • Limited Approval • Rejected • Deprecated
Last Modified	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
Component Version	<p>Name of the component version.</p> <p>Select the component name to open the Overview tab of the <i>Component Name</i> page.</p> <p>Select the version to open the Details tab of the <i>Component Name > Version</i> page.</p>
License	License for this component version.
Source	<p>Source for this component version. Possible values are:</p> <ul style="list-style-type: none"> • Custom. A custom component version. • KnowledgeBase. An unmodified Black Duck KnowledgeBase component version. • Modified KnowledgeBase. A modified Black Duck KnowledgeBase component version.

Column	Description
Approval Status	Approval status of this component version. Possible values are: <ul style="list-style-type: none"> • Unreviewed • In Review • Reviewed • Approved • Limited Approval • Rejected • Deprecated
Last Modified	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. 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 `components_date_time.csv` and `bom_component_custom_fields_date_time.csv` files 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 **Manage** > **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 **Manage** > **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

Status: Unreviewed
Description: No description.
Tags: No Tags
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
Version	Version(s) for this component. Select a version to open the Details tab of the <i>Component Name</i> > <i>Version</i> which lists the projects that use this component version.
Used Count	Number of projects that use this component version.
License	License for this component version.
Release	Release date for this component version. Never is listed if a value was not entered.

Editing custom components

1. Log in to Black Duck with the Component Manager [role](#).



2. Click **Manage** > **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 a component management interface. At the top, there's a header with a cube icon and the text "Custom My Custom Component". Below the header, the "Component Details" tab is selected. The main area contains several input fields: "Component Name" (set to "My Custom Component"), "Description", "Url", "Notes", and "Status" (set to "Unreviewed"). To the right of the status field is a dropdown arrow. At the bottom right of the main area is a blue "Save" button. Below the main form is a horizontal line and a "Delete component" section. It asks "Are you sure you want to delete this component?" and contains a red "Delete Component" button with a trash icon.

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 **Manage** > **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 **Manage > 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**.

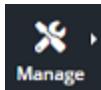
The Create a New Version dialog box appears.

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 **Manage > Component Management**.

The **Components** tab appears.

3. Select the **Component Versions** tab.

A screenshot of the Black Duck Component Management interface. The title bar says "Component Management". Below it, there are two tabs: "Components" and "Component Versions", with "Component Versions" being the active tab. A sub-header "Component Version" is visible. On the left, there's a toolbar with an "Add" button. The main area is a table with columns: Component Version, License, Source, Status, and Last Modified. There are three rows of 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, it says "Displaying 1-3 of 3".

4. Select the version to open the **Details** tab of the *Component Name > Version* page.

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 **Manage > Component Management**.

The **Components** tab appears.

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

3. Select the **Component Versions** tab.

The screenshot shows the 'Component Management' page with the 'Components' tab selected. At the top, there are buttons for 'Add', 'Local Only', 'Filter components...', and 'Add Filter'. Below is a table with columns: Component Version, License, Source, Status, and Last Modified. The data includes:

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, 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.

The screenshot shows the 'My custom component > 1.0' settings page. It has tabs for 'Cryptography', 'Details' (selected), and 'Settings'. Under 'Component Details', fields include Version (1.0), Release Date, Notes, and Approval Status (Unreviewed). A 'Save' button is at the bottom right. Below the form, a 'Delete Version' section contains a note about不可恢复删除 and a red 'Delete Version' button.

- Select **Component Details** to edit the version, release date, notes or approval status.
- Select **License** to [modify the existing license or add a new license or group](#).

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.

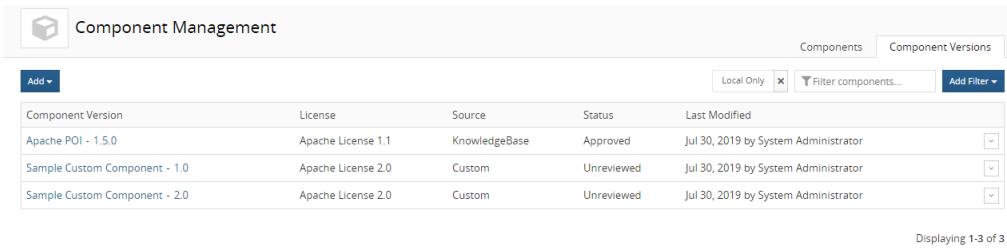
To delete a custom component version

1. Log in to Black Duck with the Component Manager [role](#).

2. Click **Manage** > **Component Management**.

The **Components** tab appears.

3. Select the **Component Versions** tab.



The screenshot shows a software application window titled "Component Management". At the top, there are tabs for "Components" and "Component Versions", with "Component Versions" being the active tab. Below the tabs is a search bar with the placeholder "Filter components..." and a "Local Only" checkbox. To the right of the search bar is a "Add Filter" button. The main area displays a table with three rows of data. The columns are labeled "Component Version", "License", "Source", "Status", and "Last Modified". The data is as follows:

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

At the bottom right of the table area, it says "Displaying 1-3 of 3".

4. Click  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.

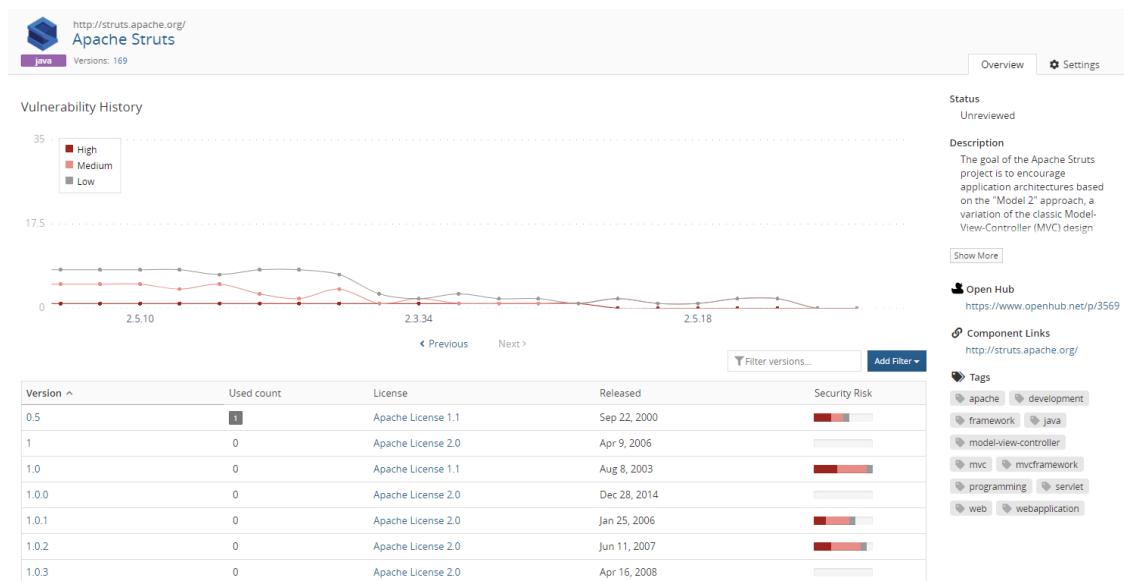
The Black Duck KB provides information about OSS components at the [component level](#) and at the [component version level](#).

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 an approval 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.



This page is comprised of two tabs: an **Overview** tab and a **Settings** tab.

About the Overview tab

The **Overview** tab displays information such as a status, 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:



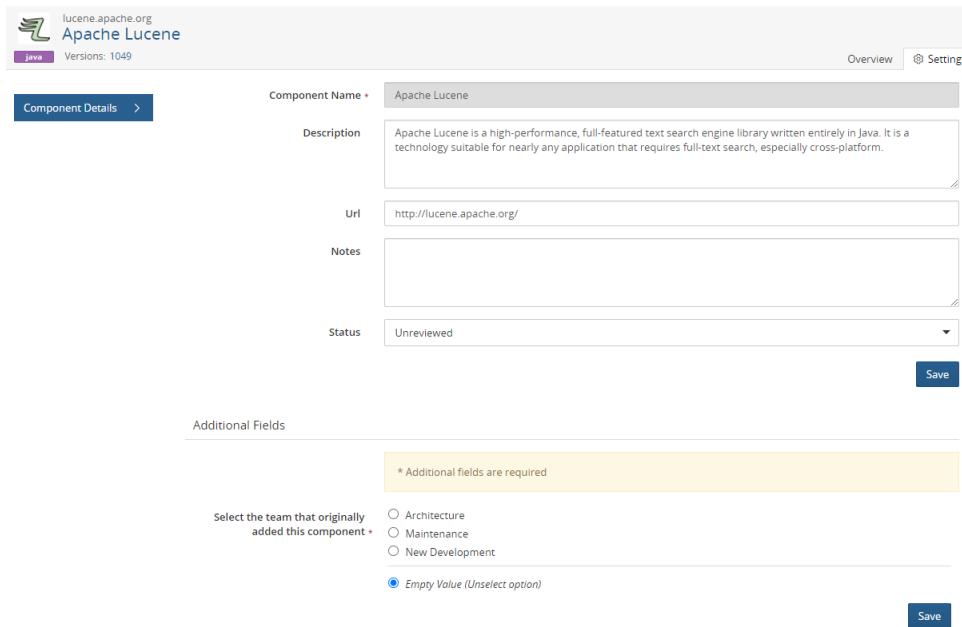
To view information on versions that interest you, use the filter, located above the table, to filter the versions shown in the vulnerability graph and in the table below.

The following information is available for each version:

Column	Description
Version	<p>Release number of this version of the component.</p> <p>Select the version number to display the Component Name > Version page.</p>
Used Count	<p>Number of project version BOMs in which this version of the OSS project is used.</p> <p>Tip: Select the number to go to the Details 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.</p>
License	<p>Declared license of this version of the OSS component. Other license types include:</p> <ul style="list-style-type: none"> "Unknown" indicates that the OSS component version's license is not known. "License Not Found" indicates that although researched by Synopsys, no declared license was found for the component. "No License" indicates that Synopsys has found a declaration of 'No License' for the component. <p>For known licenses, select the license name to view license details and license text.</p>
Released	The date this version of the OSS component was released.
Security Risk	<p>A graph which shows the number of high risk, medium risk, and low risk vulnerabilities associated with this version of the OSS component</p> <p>Select a value in the security risk graph to display the Component Name > Version page.</p>

About the Settings tab

The **Settings** tab shows details on this component. Information shown here appears on the **Overview** tab.



The screenshot shows the 'Apache Lucene' component settings page. At the top, there are tabs for 'Overview' and 'Settings'. Below the tabs, the component name 'Apache Lucene' is displayed. The main area contains several input fields: 'Description' (Apache Lucene is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.), 'Url' (http://lucene.apache.org/), 'Notes' (empty), and 'Status' (Unreviewed). A 'Save' button is located at the bottom right of this section. Below this, there is a section titled 'Additional Fields' with a note: '* Additional fields are required'. It includes a dropdown menu for selecting the team that originally added the component, with options: 'Architecture', 'Maintenance', 'New Development', and 'Empty Value (Unselect option)'. The 'Empty Value (Unselect option)' option is selected. Another 'Save' button is located at the bottom right of this section.

Users with the Component Manager [role](#) can use the **Settings** tab to edit the description, URL, notes, and

status for this KB component. Click [here](#) for information on editing component information and [here](#) for information on modifying a component's status.

Users with the System Administrator role can use the **Settings** tab to edit the component [custom field information](#), as shown in the **Additional Fields** section.

Understanding the component version information available from the Black Duck KB

On the *Component Name Version* page, the **Details** tab provides the following information:

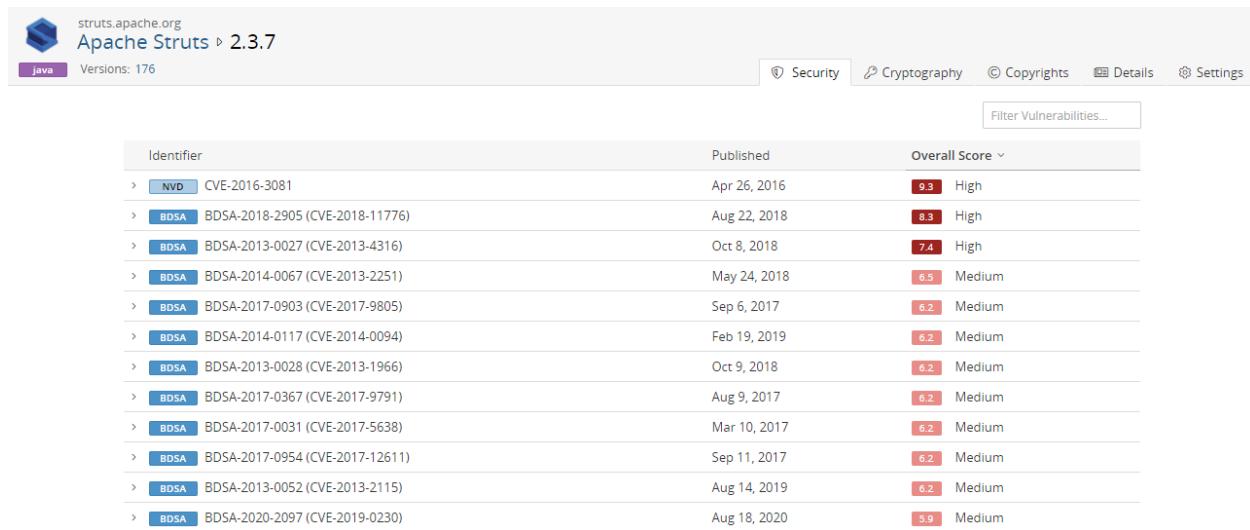
- Description.
- Count of known security vulnerabilities.
- Associated licenses.
- Component links, if available.
- Tags, if available.
- Date this version was released.
- Number of newer versions.
- [Approval Status](#) of this version.
- Date this component was last updated.
- Commit activity and the 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 screenshot shows the component details for Apache Struts 2.3.7. At the top, there is a navigation bar with tabs for Security, Cryptography, Copyrights, Details (which is selected), and Settings. Below the navigation bar, the component name is displayed as "Apache Struts 2.3.7" with a Java icon and a link to "http://struts.apache.org/". The page title is "Apache Struts 2.3.7". The "Versions" count is 173. The main content area includes sections for Description, Activity, and Where Used. The Description section states: "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." The Activity section shows commit and contributor trends over the last 12 months. The Where Used section displays a table with one row, showing the project "bds_jenkins_test" with version "2020_04_03-15_49_49", released "Never", and phase "In Development". To the right of the main content, there are sections for Licenses (Apache License 2.0), Open Hub (https://www.openhub.net/p/3569), Component Links (http://struts.apache.org/), and Tags (apache, development, framework, java, model-view-controller, mvc, mvframework, programming, servlet, web, webapplication). A red banner at the top right indicates 40 vulnerabilities.

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 Overview tab of the Project Name 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.



The screenshot shows the Black Duck KB interface for the Apache Struts 2.3.7 component. The top navigation bar includes links for struts.apache.org, Apache Struts > 2.3.7, Versions: 176, Security (selected), Cryptography, Copyrights, Details, and Settings. A search bar at the top right is labeled "Filter Vulnerabilities...". Below the navigation is a table with columns: Identifier, Published, and Overall Score. The table lists 15 vulnerabilities, each with a link to either BDSA or NVD. The overall scores range from 5.9 to 9.3, with most being High risk.

Identifier	Published	Overall Score
> NVD CVE-2016-3081	Apr 26, 2016	9.3 High
> BDSA BDSA-2018-2905 (CVE-2018-11776)	Aug 22, 2018	8.3 High
> BDSA BDSA-2013-0027 (CVE-2013-4316)	Oct 8, 2018	7.4 High
> BDSA BDSA-2014-0067 (CVE-2013-2251)	May 24, 2018	6.5 Medium
> BDSA BDSA-2017-0903 (CVE-2017-9805)	Sep 6, 2017	6.2 Medium
> BDSA BDSA-2014-0117 (CVE-2014-0094)	Feb 19, 2019	6.2 Medium
> BDSA BDSA-2013-0028 (CVE-2013-1966)	Oct 9, 2018	6.2 Medium
> BDSA BDSA-2017-0367 (CVE-2017-9791)	Aug 9, 2017	6.2 Medium
> BDSA BDSA-2017-0031 (CVE-2017-5638)	Mar 10, 2017	6.2 Medium
> BDSA BDSA-2017-0954 (CVE-2017-12611)	Sep 11, 2017	6.2 Medium
> BDSA BDSA-2013-0052 (CVE-2013-2115)	Aug 14, 2019	6.2 Medium
> BDSA BDSA-2020-2097 (CVE-2019-0230)	Aug 18, 2020	5.9 Medium

This tab contains the following information:

Column	Description
Identifier	The identifier and value associated with this vulnerability. Select > in the table next to the vulnerability to view a brief description. Depending on the identifier, select to view the BDSA record or the CVE record .
Published	Date on which the vulnerability was published.
Overall Score	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. <ul style="list-style-type: none"> • For BDSA, the Temporal, Base, Exploitability, and Impact scores are shown. • For NVD, the Base, Exploitability, and Impact scores are shown. The Temporal score represents time-dependent qualities of a vulnerability, taking into account the

Column	Description
	<p>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"> • Access Vector (AV) - CVSS v2 / Attack Vector (AV) - CVSS v3.x • Access Complexity (AC) - CVSS v2 / Attack Complexity (AC) - CVSS v3.x • Authentication (Au) • Integrity (I) • Availability (A) • Confidentiality (C) <p>Note: The Authentication value is not available for CVSS v3.x 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. Click [here](#) for more information.

The **Copyrights** tab shows the copyright statements for this component version. Click [here](#) for more information.

The **Settings** tab shows details on this component version. Information shown here also appears on the **Details** tab.

OpenSSL http://www.openssl.org/
OpenSSL • 0.8.1b
Versions: 360

Component Details >

Version	0.8.1b
License	SSLeay License
Release Date	12/21/1998
Notes	(empty)
Status	Unreviewed

Save

Additional Fields

Select the team who originally added this component version

Architecture
 Maintenance
 New Development

Save

Users with the Component Manager [role](#) can use the **Settings** tab to edit information for this KB component

version.

- Select **Component Details** to edit the release date, notes, and status for this KB component version.
- Select **License** to [modify the existing license or add a new license or group](#).

Click [here](#) for information on editing component information and [here](#) for information on modifying a component version's status.

Users with the System Administrator role can use the **Settings** tab to edit the component version [custom field information](#), as shown in the **Additional Fields** section.

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.

Note: Setting the [status](#) of a KB component and all versions listed in the Component Management table to **Unreviewed** removes the KB component and its versions from the Component Management table. Note that this does not apply to those KB components and versions shown with a source of **Modified KnowledgeBase**.

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 **Manage** > **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 a "Add" button, a search bar labeled "Filter components...", and a "Add Filter" button. The main area has a table 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.

The unreviewed status is not available when adding a KnowledgeBase 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 **Manage** > **Component Management**.

The **Components** tab appears.

3. Select the KB component you wish to modify.

The **Overview** tab for the *Component Name* page appears.

Note: You can also display the **Overview** tab by searching for the component and selecting to view it from the search results.

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 **Manage** > **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.
 - Select **Component Details** to edit the release date, notes, and status for this KB component version.
 - Select **License** to [modify the existing license or add a new license or group](#).
5. Modify the information and click **Save**.

If you modified the license or release date, 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 **Manage** > **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** to 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 **Manage** > **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** to 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.

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.

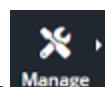
The unreviewed status is not available for KB components.

- 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

1. Log in to Black Duck with the Component Manager role.



2. Click **Manage** > **Component Management**.

The **Components** tab appears.

A screenshot of the Black Duck Component Management interface. The top navigation bar includes 'Component Management' and tabs for 'Components' and 'Component Versions'. Below is a table with columns: Component, License, Source, Status, and Last Modified. The table shows 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

Displaying 1-3 of 3

3. 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 **Approval Status** list and click **Save**.

To modify the status for a component version

1. Log in to Black Duck with the Component Manager role.



2. Click **Manage** > **Component Management**.

The **Components** tab appears.

Component Management				
		Components Component Versions		
Add	Component	License	Source	Status Last Modified
>	Apache POI [1 Version]	Apache License 2.0	KnowledgeBase	Unreviewed Jul 30, 2019 by sysadmin
	Bash	Apache License 2.0	Modified KnowledgeBase	Approved Jul 30, 2019 by sysadmin
>	Sample Custom Component [2 Versions]	Apache License 2.0	Custom	Unreviewed Jul 30, 2019 by sysadmin

Displaying 1-3 of 3

3. Select the **Component Versions** tab.

Component Management				
		Components Component Versions		
Add	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. 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:
 - 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 **Approval 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 page/**Security** tab

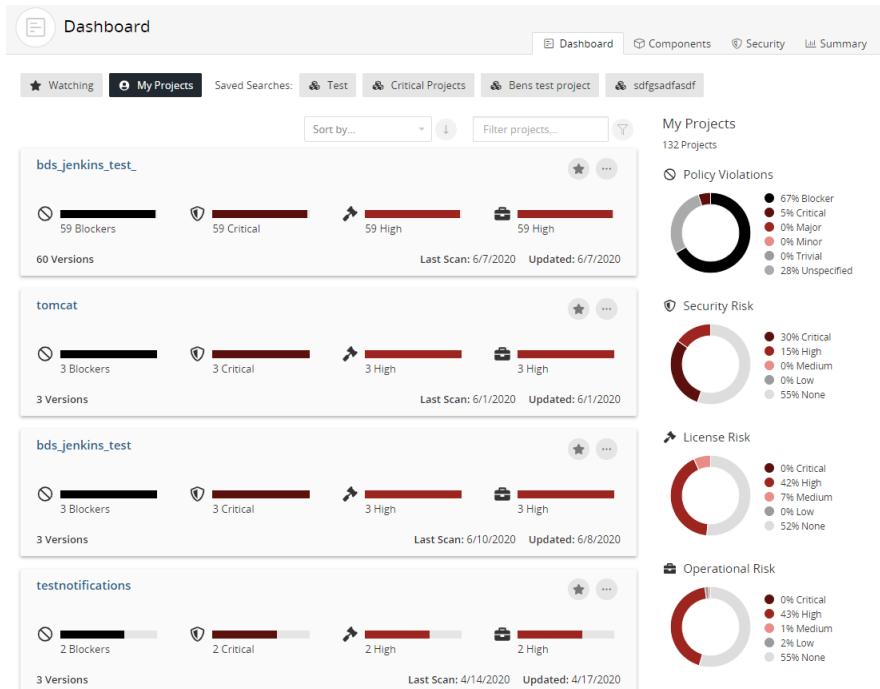
Note that the security risk values shown use CVSS v2 or CVSS v3.x scores, depending on which [security risk calculation you selected](#); by default, CVSS v2 scores are shown. Note that the security risk graph displays a Critical risk category with a value of 0, if you selected CVSS v2.

Dashboards

Dashboards provide a high-level overview of risk from different perspectives.

Note: Dashboards 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.

- You can view the projects that interest you by using the **Watching** or **My Projects** dashboard or create a custom dashboard by [saving your project search results](#).



Create a saved [component search](#) to view the components that interest you that are used in one or more projects.

Dashboard

Projects Saved Searches [?](#)

★ Watching My Projects [Vuln Affecting Projects](#) [Proj - High Security Risk](#) [Comps - High Security Risk](#)

Comps - High Security Risk

Sort by... [↓](#) Filter Components... [T](#)

Apache Struts ^{2.3.7}	Used By 4 Project Versions	No Policy Violations	No License Risk	High		Last Vuln: 2/4/2021
Approval Status: Unreviewed First Detected: 2/1/2021 Release Date: 11/6/2012 Newer Versions: 81						
Growl ^{1.9.2}	Used By 1 Project Version	No Policy Violations	No License Risk	High		Last Vuln: 2/4/2021
Approval Status: Unreviewed First Detected: 2/4/2021 Release Date: 2/21/2016 Newer Versions: 7						
Handlebars.js ^{4.0.5}	Used By 1 Project Version	No Policy Violations	No License Risk	High		Last Vuln: 2/4/2021
Approval Status: Unreviewed First Detected: 2/4/2021 Release Date: 11/20/2015 Newer Versions: 52						
Request - Simple HTTP Client ^{2.9.203}	Used By 1 Project Version	No Policy Violations	No License Risk	High		Last Vuln: 2/4/2021
Approval Status: Unreviewed First Detected: 2/4/2021 Release Date: 6/28/2012 Newer Versions: 163						
lodash.merge ^{4.5.1}	Used By 1 Project Version	No Policy Violations	No License Risk	High		Last Vuln: 2/4/2021
Approval Status: Unreviewed First Detected: 2/4/2021 Release Date: 2/22/2016 Newer Versions: 7						
lodash.template ^{3.6.2}	Used By 1 Project Version	No Policy Violations	No License Risk	High		Last Vuln: 2/4/2021
Approval Status: Unreviewed First Detected: 2/4/2021 Release Date: 6/30/2015 Newer Versions: 14						
lodash.template ^{4.3.0}	Used By 1 Project Version	No Policy Violations	No License Risk	High		Last Vuln: 2/4/2021
Approval Status: Unreviewed First Detected: 2/4/2021 Release Date: 2/8/2016 Newer Versions: 6						
node-ini ^{1.3.4}	Used By 1 Project Version	No Policy Violations	No License Risk	High		Last Vuln: 2/4/2021
Approval Status: Unreviewed First Detected: 2/4/2021 Release Date: 6/6/2015 Newer Versions: 5						
sequelize ^{3.23.6}	Used By 1 Project Version	No Policy Violations	No License Risk	High		Last Vuln: 2/4/2021
Approval Status: Unreviewed First Detected: 2/4/2021 Release Date: 7/19/2016 Newer Versions: 445						

Results Summary
9 Components
Results updated at Feb 5, 2021 7:56 AM
[Saved Search Settings](#)

Policy Violations

Security Risk

License Risk

Operational Risk

Displaying 1-9 of 9

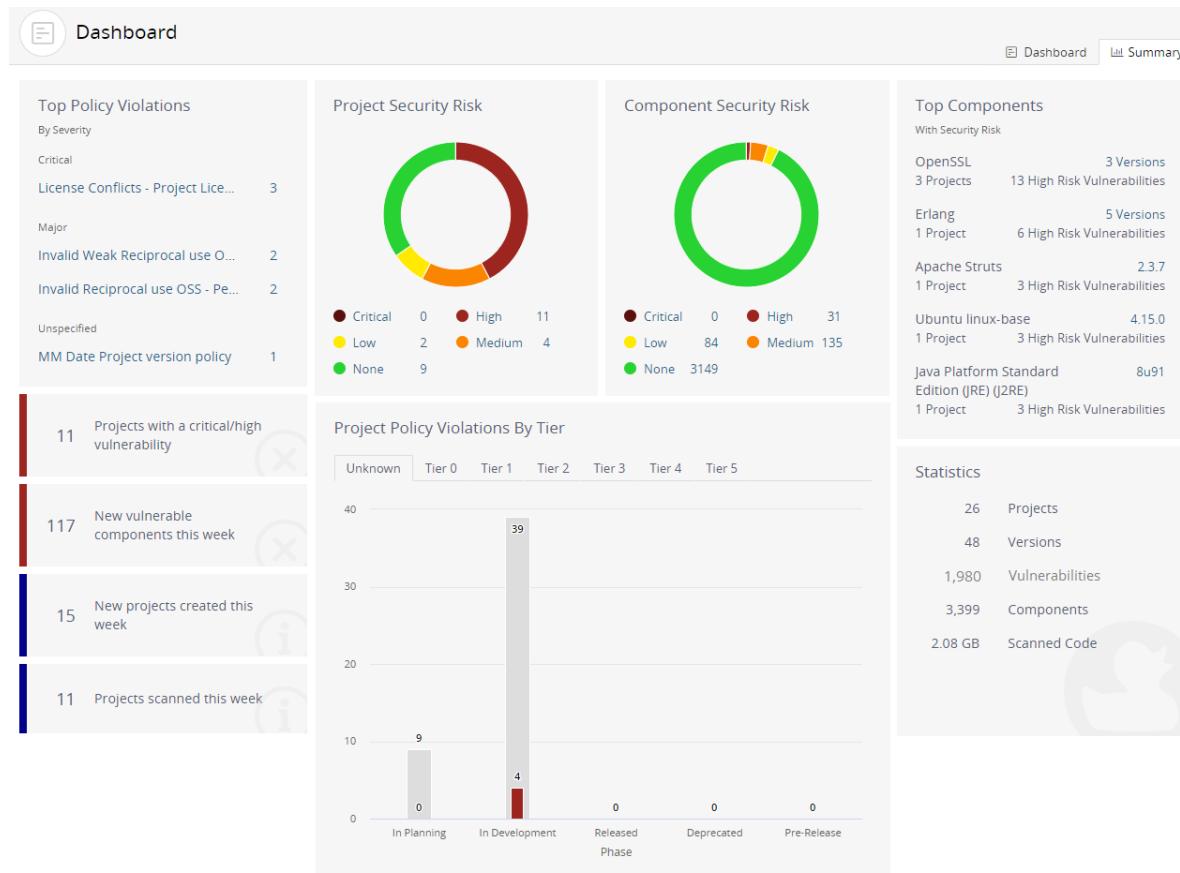
- Create a saved [vulnerability search](#) to view the vulnerabilities that interest you.

The screenshot shows the Black Duck Summary Dashboard. At the top, there's a navigation bar with links for 'Dashboard' and 'Summary'. Below the navigation, there are sections for 'Projects' (Watching, My Projects), 'Saved Searches' (Vuln Affecting Projects, Proj - High Security Risk, Comps - High Security Risk), and a 'Results Summary' section indicating 224 Vulnerabilities last updated at Feb 5, 2021 7:56 AM with a link to 'Saved Search Settings'.

The main area is titled 'Vuln Affecting Projects' and lists five entries:

- BDSA-2019-1853 (CVE-2019-11272): Overall Risk Medium (5.5). Solution available, no workaround or exploit. First Detected: 2/1/2021, Published: 6/21/2019, Last Modified: 6/21/2019.
- BDSA-2013-0030 (CVE-2013-1965): Overall Risk Medium (5.9). Solution available, no workaround or exploit. First Detected: 2/1/2021, Published: 10/10/2018, Last Modified: 4/3/2020.
- BDSA-2018-1901 (CVE-2018-11040): Overall Risk Low (3.2). Solution available, workaround available, no exploit. First Detected: 2/1/2021, Published: 6/20/2018, Last Modified: 6/20/2018.
- BDSA-2019-4008 (CVE-2019-17571): Overall Risk Medium (6.4). No solution, workaround available, no exploit. First Detected: 2/1/2021, Published: 12/20/2019, Last Modified: 7/16/2020.
- BDSA-2019-1179 (CVE-2019-10246): Overall Risk Low (3.9). Solution available, no workaround or exploit. First Detected: 2/1/2021, Published: 4/23/2019, Last Modified: 4/23/2019.

- Use the [Summary Dashboard](#) to view the overall health of the projects you have permission to view and identify areas of concern.

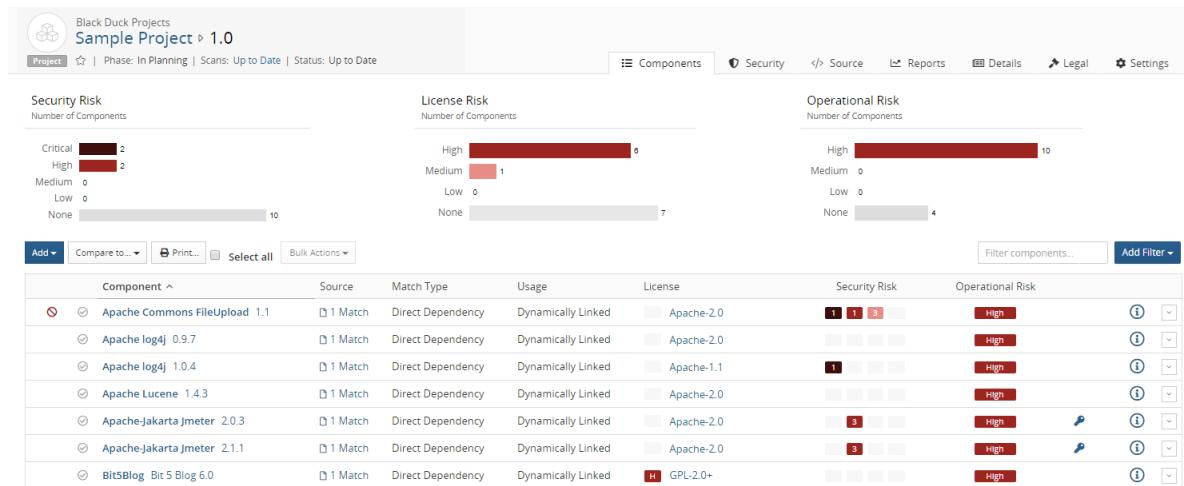


Note:

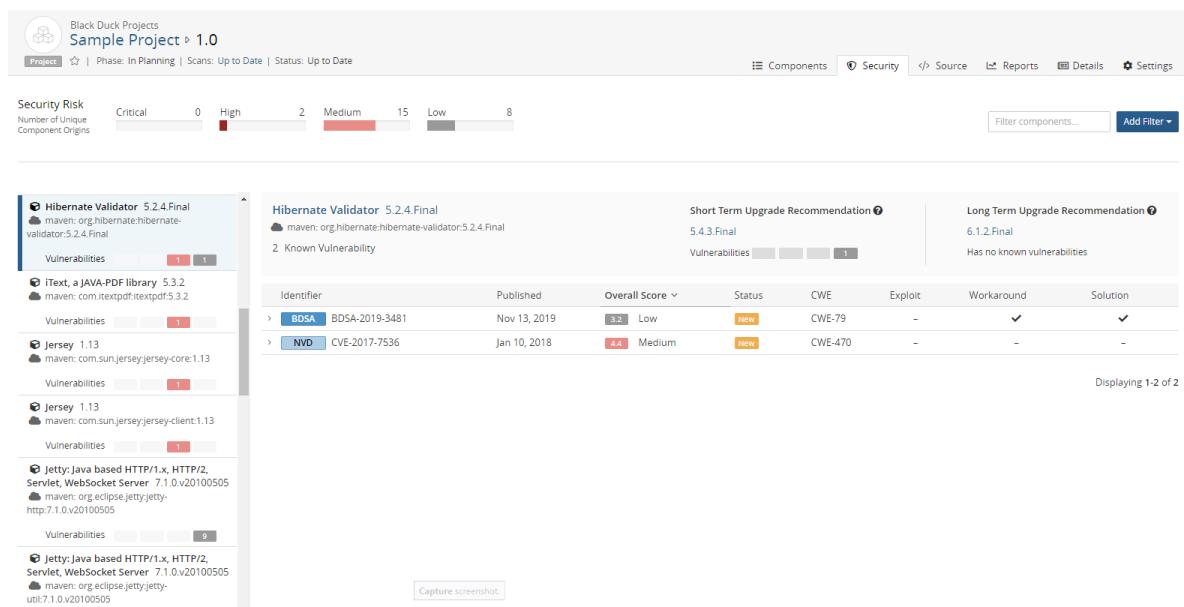
- The Dashboard page that appears when you log in depends on the last main dashboard (Dashboard or Summary) you viewed prior to previously logging out.
- Click or the logo in the upper left corner of the navigation bar to view the last dashboard (Dashboard or Summary) you viewed.

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.

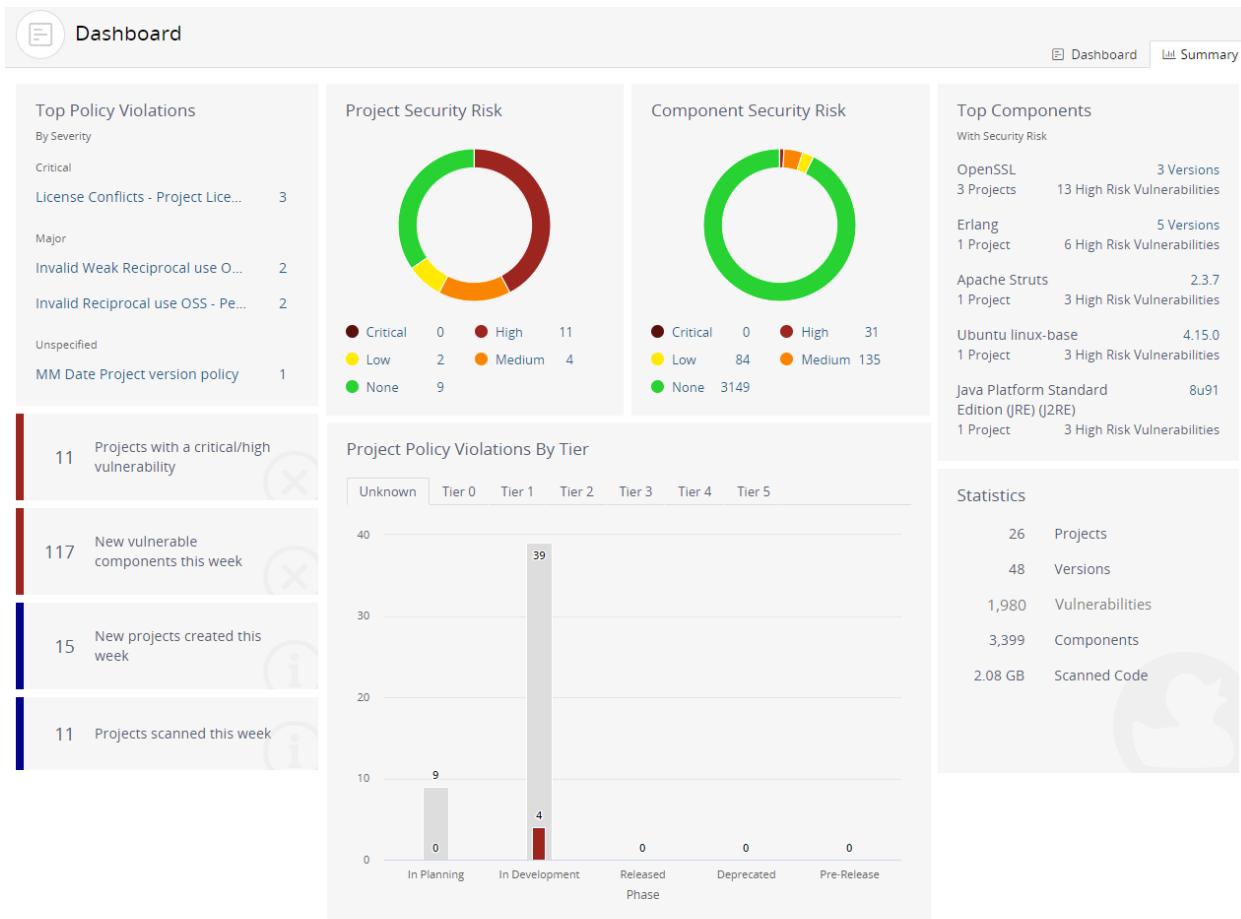


- Use the [project version page/ Security tab](#) to view the security vulnerabilities of each severity associated with the components used in a project version.



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.



Description	More Information
<p>The Top Policy Violations widget displays up to the top five policy violations across all projects that you have permission to view.</p> <p>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"> If you do not have the Policy Management module, this widget will not appear on the page. A message appears if you have the Policy Management module but do not have any policy rules configured or have any policy violations. 	Select a policy rule to view the My Projects tab filtered to display the projects with a version that violates that policy rule.
<p>The Project Security Risk widget displays the number of projects you have permission to view for each level of security risk.</p> <p>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 Component Security Risk widget displays the number of components in projects you have permission to view for each security risk level.</p> <p>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 security risk.</p>	Hover over the graph to view the number of components with that level of security risk.
<p>The Top Components with Security Risk 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"> Component name and number of versions used in your projects. If only one version is used, the specific version is listed here. Number of your projects that have this component. Number of security risks in this component, with the highest security risk listed here. <p>Components are organized by security risk, with those components with the highest risk listed first.</p>	Select the specific version or number of versions to view the Component Version Details page .
<p>The Projects have a critical/high vulnerability widget displays the number of projects with versions that contain components with a critical and/or high security risk.</p>	N/A.
<p>The New vulnerable components this week 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 New projects created this week widget displays the number of projects that you have permission to view that have been created in the past seven days, including today.</p>	N/A.

Description	More Information
The Projects scanned this week widget displays the number of projects with scans from the past seven days, including today.	N/A.
<p>The Project Policy Violations by Tier widget displays the total number of projects by phase that have a policy violation, grouped by tiers.</p> <ul style="list-style-type: none"> • If you do not use tiers for your projects, projects are grouped in a single category called Unknown. • If you do not have the Policy Management module, this widget displays Projects by Tier. 	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 Statistics widget displays the following information:</p> <ul style="list-style-type: none"> • Projects lists the number of your projects. • Versions lists the number of project versions for your projects. • Vulnerabilities lists the number of vulnerabilities in your projects. • Components lists the number of components used in your projects, <i>including</i> ignored components. • Scanned Code lists the number of GBs scanned for all scans. 	N/A.

Viewing your dashboards

Use dashboards to view the types and severity of risk and policy violations that are associated with the components that are in one or more versions of your projects. Dashboards provide an overall view across your projects, components, and vulnerabilities.

So that you can view the projects and project versions that are important to you, Black Duck's provides two default dashboards and the ability for you to create an unlimited number of custom dashboards.

Black Duck displays these two default dashboards:

- **Watching.** Your [watched projects](#).
- **My Projects.** All of your projects, including projects that you are not watching.

These dashboards display information on the Dashboard page at the project level.

In addition, you can create custom dashboards so that you can quickly view the project versions, component versions, and vulnerabilities that are important to you: [search for projects](#), [components](#), and/or [vulnerabilities](#) and then [save the searches](#); use the Dashboard page to view the information from those saved searches.

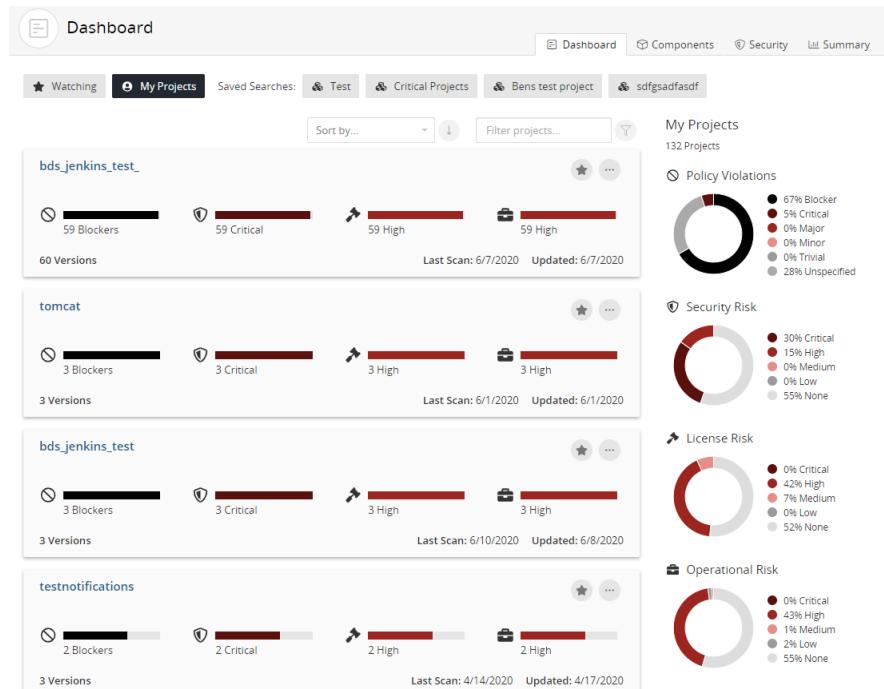
Viewing dashboards

To view the dashboards

1. Click  to display the dashboards.

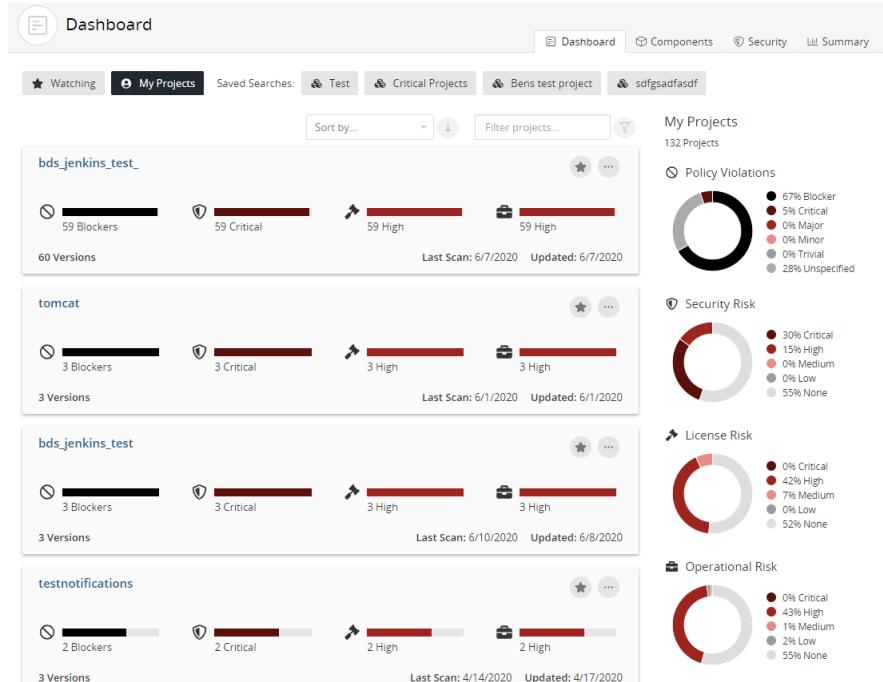
The dashboard page that appears depends on the last dashboard (a specific Dashboard page or

[Summary](#) Dashboard) you viewed previously. If not displayed, select **Dashboard** to display your dashboards.



About the Watching and My Projects dashboards

Use the **Watching** or **My Projects** dashboards to view risk and policy violation information at the *project level*.



The following information is shown for each project:

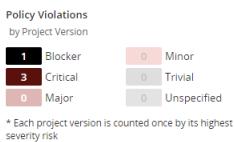


- To view policy violation information for a specific project:
 - Use the bar to view the number of project versions with the highest policy severity level.



Note: The text states the number of project versions with this highest policy severity level, not all policy severity levels affecting this project.

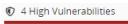
- Hover over the bar to see the number of project versions with their highest severity level of policy violations:



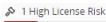
In the above example, there are four project versions which have policy violations; one version has a policy violation which has Blocker as the highest severity level, the other three versions have Critical as the highest severity level. Note that this does not indicate the number of policy violations in these versions, just the highest severity level for each version.

- To view risk information:
 - Use the risk bar to view the number of project versions with the highest risk level:

Security risk:



License risk:



Operational risk:



Note: The text states the number of project versions with this highest risk level, not all risk levels affecting the versions.

- Hover over a risk bar to see the number of versions of this project with their highest level of risk.



If a project version has risk, the version is only counted once and only its highest risk level is shown.

- Use the graphs to see overview information for all projects in this dashboard.
 - The risk graph shows the percentage of projects in this dashboard that have policy violations by severity level. You can also hover over an area in the graph to view this information:



- The risk graphs show the percentage of projects in this dashboard that have this level of security, license, or operational risk. You can also hover over an area in the graph to view this information:



- Hover over a value in the legend to highlight the value in the graph.
- View additional information for each project, including:
 - Number of versions.
 - Last scan date.
 - Date when this project was last updated, such as when a scan that was mapped to any project version was last run or when the BOM for any project version was last updated, either manually or by a new scan.
- Select a project name to view the *Project Name* page which lists all versions of this project.
- Manage how the projects are shown in these dashboards:
 - Use the **Sort by** field to select an attribute to sort by and click an arrow to select the sort order (ascending) or (descending).
 - Use the **Filter projects** field to filter the projects shown in either dashboard.
- Use the icons to [manage your watched projects](#) or [delete a project](#).

About saved searches dashboards

Use a saved search to view the project versions, component versions, and vulnerabilities that are important to you.

For each saved search, Black Duck lists the date and time this search was last updated.

Results Summary

9 Components

Results updated at Feb 8, 2021 10:03 AM

[Saved Search Settings](#)

Select **Saved Search Settings** to view the filters for this saved search.

Saved Search Settings
 • Security Risk: High
[Edit Saved Search >](#)

Select **Edit Saved Search** to open the Find page displaying your saved search. Use the page to edit and save this revised saved search.

Project version saved searches

The screenshot shows the Black Duck interface for viewing risk in project versions. The main area displays five project versions with their respective risk counts:

- bds_jenkins_test > 2021_02_03--23_21_31: 1 High Security Risk, 2 High License Risks, 53 High Operational Risks
- bds_jenkins_test > 2021_02_01--08_51_58: 1 High Security Risk, 2 High License Risks, 53 High Operational Risks
- bds_jenkins_test > 2021_02_01--23_18_05: 1 High Security Risk, 2 High License Risks, 53 High Operational Risks
- bds_jenkins_test > 2021_02_02--23_18_51: 1 High Security Risk, 2 High License Risks, 53 High Operational Risks
- hjoe-npmjs-top-1 > 1.0: 8 High Security Risks, 2 High License Risks, 847 High Operational Risks

To the right, there are four donut charts showing the distribution of risks:

- Results Summary**: Policy Violations (0% Blocker, 0% Critical, 0% Major, 0% Minor, 0% Trivial, 0% Unspecified, 100% None)
- Security Risk**: 100% Critical, 0% High, 0% Medium, 0% Low, 0% None
- License Risk**: 100% High, 0% Medium, 0% Low, 0% None
- Operational Risk**: 100% High, 0% Medium, 0% Low, 0% None

The following information is shown for each project version:



- ⓘ located in front of the saved search name indicates that this is a project saved search.
- To view policy violation information for a specific project version:
 - Use the bar to see the number of components with the highest policy severity level for this project version.

For example, the following shows that while there are components with lower severity levels, the highest policy severity level for this project version is Blocker and there are five components that have Blocker as their highest policy severity level.

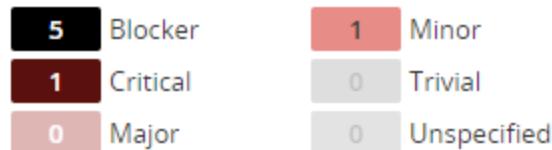
ⓘ 5 Blocker Policy Violations

Note: The text states the number of components with the highest policy severity level for this project version, not all policy severity levels affecting this project version.

- Hover over the bar to see the number of components with policy violations by the highest policy severity level:

Policy Violations

by Component



* Each component is counted once by its highest severity risk

If a component has a policy violation, the component is only counted once and only its highest policy severity level is shown.

- To view risk information:
 - Use the risk bars to quickly view the number of components with the highest level of security, license, or operational risk.

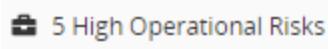
Security risk:

ⓘ 1 High Vulnerability

License risk:

ⓘ 2 High License Risks

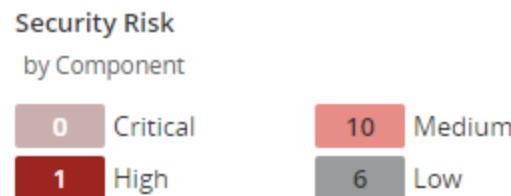
Operational risk:



For example, the following shows that while there are components with lower risk, the highest security risk for this project version is High and that one component in this project version has a high level of security risk as their highest risk level:



- Hover over the bar to see the number of components for each risk category.



* Each component is counted once by its highest severity risk

In this example, there is one component that has a high risk level as its highest risk, 10 components that have medium risk as their highest risk level, and six components that have low risk as their highest risk level.

Note: Each component is only counted once and is shown with its highest risk level.

- Use the graphs to view overview information for all project versions in this dashboard categorized by policy severity and risk levels. The graphs lists the percentages for each level. You can also:

- Hover over the graph to view the percentage of project versions with policy violations for each policy severity level.



- Hover over the graph to view the percentage of project versions in this dashboard for each risk level.



- Hover over a value in the legend to highlight the value in the graph.

- For each project version, the dashboard also shows:

- Number of components in this project version.
 - Last scan date.
 - Date when this project version was last updated, such as when a scan that was mapped to this project version was last run or when the BOM for this project version was last updated, either manually or by a new scan.
 - License of this project version.
 - Phase for this project version.
 - Distribution of this project version.
- Select the project or version name to view the BOM.
 - Manage how the projects are shown in these dashboards:
 - Use the **Sort by** field to select an attribute to sort by and click an arrow to select the sort order  (ascending) or  (descending).
 - Use the **Filter projects** field to filter the projects shown in the dashboard.

Component saved searches

Dashboard

Saved Searches [Comps - High Security Risk](#)

Comps - High Security Risk

Sort by... [Filter Components...](#)

Component	Version	Used By	Policy Violations	License Risk	Risk Level	Last Vuln
Apache Struts	2.3.7	4 Project Versions	No Policy Violations	No License Risk	High	2/4/2021
Growl	1.9.2	1 Project Version	No Policy Violations	No License Risk	High	2/4/2021
Handlebars.js	4.0.5	1 Project Version	No Policy Violations	No License Risk	High	2/4/2021
Request - Simple HTTP Client	2.9.203	1 Project Version	No Policy Violations	No License Risk	High	2/4/2021
lodash.merge	4.5.1	1 Project Version	No Policy Violations	No License Risk	High	2/4/2021
lodash.template	3.6.2	1 Project Version	No Policy Violations	No License Risk	High	2/4/2021
lodash.template	4.3.0	1 Project Version	No Policy Violations	No License Risk	High	2/4/2021
node-ini	1.3.4	1 Project Version	No Policy Violations	No License Risk	High	2/4/2021
sequelize	3.23.6	1 Project Version	No Policy Violations	No License Risk	High	2/4/2021

Results Summary
9 Components
Results updated at Feb 5, 2021 7:56 AM
[Saved Search Settings](#)

Policy Violations

0% Blocker
0% Critical
0% Major
0% Minor
0% Trivial
0% Unspecified
100% None

Security Risk

0% Critical
100% High
0% Medium
0% Low
0% None

License Risk

0% High
0% Medium
0% Low
100% None

Operational Risk

100% High
0% Medium
0% Low
0% None

Displaying 1-9 of 9

The following information is shown for each component.

Apache Struts	2.3.7	9 Project Versions	4 Critical Policy Violations	No License Risk	High	3 28 11
Approval Status:	Unreviewed	First Detected:	Never	Released Date:	11/6/2012	Newer Versions: 80
Last Vuln: 10/9/2020						

- A cube icon located in front of the saved search name indicates that this is a component saved search.
- Select the component name/version to display the [Component Name Version page](#).
- View the number of project versions that use this component version as shown by the value next to **Used By**.

Used By | 2 Project Versions

Select **Project Versions** to open the Where Used dialog box.

The dialog box has a header 'Used in' and a close button 'X'. It displays a message: 'Apache Struts - 1.2.2 is being used in 1 Project Version'. Below this is a table with columns: Project Name, Phase, License, Review Status, and Security Risk. One row is shown: Sample Project - 4.0, In Planning, Apache License 2.0, Not Reviewed, and a bar with values 0, 3, 6, 0. A 'Close' button is at the bottom right.

Project Name	Phase	License	Review Status	Security Risk
Sample Project - 4.0	In Planning	Apache License 2.0	Not Reviewed	0 3 6 0

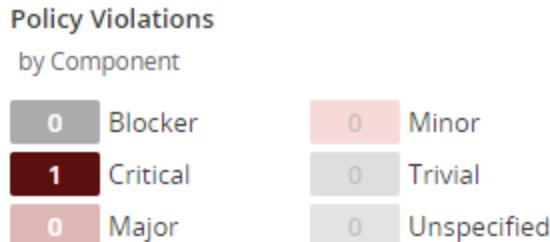
This dialog box shows the project versions that use this version of the component.

Column	Description
Project Name	Name of project and version that uses this component version. Select the project name to display the project version's Components tab.
Phase	Project Phase .
License	License for this component version.
Review Status	Whether this component has been reviewed in this project version.
Security Risk	<p>Lists the vulnerabilities for each severity level, from left to right: Critical, High, Medium, and Low.</p> <p>0 3 28 11</p> <p>Select a value to display the Security tab of the the Black Duck KB Component Name Version page, which lists the vulnerabilities associated with this version of this component.</p>

- Use the bar to quickly see the number of components with the highest policy severity level.

0 1 Critical Policy Violation

Select the bar to see the number of components with policy violations by severity level:



* Each component is counted once by its highest severity risk

Note: A component is only counted once with the highest policy severity level, not all policy severity levels affecting this component.

- Use the bar to quickly view the number of components with the highest level of license risk.

1 High License Risk

Select the bar to view the number of components in each risk category.



* Each component is counted once by its highest severity risk

- View the operational risk for this component version:

High

- View the number of vulnerabilities by severity associated with this component version for each severity level, from left to right: Critical, High, Medium, and Low.

The **Last Vuln** date is the date when a vulnerability for this component was last updated in Black Duck (by the Black Duck KnowledgeBase or a user).

0 3 28 11
Last Vuln: 10/7/2020

Select a value to display the **Security** tab of the the Black Duck KBComponent Name Version page, which lists the vulnerabilities associated with this version of this component.

The screenshot shows a table of vulnerabilities for the Apache Commons Collections 3.2.1 component. The columns are Identifier, Published, and Overall Score. The table contains four rows of data:

Identifier	Published	Overall Score
BDSA-2015-0001 (RCE)	Apr 3, 2017	8.3 High
BDSA-2015-0753 (CVE-2015-6420) (RCE)	May 3, 2019	8.3 High
BDSA-2017-2285 (CVE-2017-15708) (RCE)	Dec 14, 2017	5.5 Medium
BDSA-2015-0766 (RCE)	Aug 6, 2019	5.5 Medium

Displaying 1-4 of 4

- For each component version, the search results also show:
 - Approval status. Status indicates whether this component version has been reviewed.
 - First detected date.
 - Date this component version was released.
 - Number of newer versions.
 - Date when a vulnerability for the component was last updated in Black Duck (by updates from the Black Duck KnowledgeBase or a user manually changing the associated vulnerability and so on).
- Manage how the components are shown in these dashboards:
 - Use the **Sort by** field to select an attribute to sort by and click an arrow to select the sort order (ascending) or (descending).
 - Use the filter field to filter the components shown in the dashboard.

Vulnerability saved searches

The screenshot shows the Black Duck interface with the 'Dashboard' tab selected. In the top navigation bar, there are links for 'Dashboard' and 'Summary'. Below the navigation, there are tabs for 'Projects' (Watching, My Projects), 'Saved Searches' (Vuln Affecting Projects, Proj - High Security Risk, Comps - High Security Risk), and 'Vuln Affecting Projects' (which is currently active).

Vuln Affecting Projects

The main content area displays a list of vulnerabilities:

- BDSA** BDSA-2019-1853 (CVE-2019-11272)
 - Used By: 4 Project Versions
 - Overall Risk: 5.5 Medium
 - First Detected: 2/1/2021 Published: 6/21/2019 Last Modified: 6/21/2019
 - Solution: ✓
 - Workaround: No Workaround
 - Exploit: No Exploit
 - CWE: CWE-287
- BDSA** BDSA-2013-0030 (CVE-2013-1965)
 - Used By: 4 Project Versions
 - Overall Risk: 5.9 Medium
 - First Detected: 2/1/2021 Published: 10/10/2018 Last Modified: 4/3/2020
 - Solution: ✓
 - Workaround: No Workaround
 - Exploit: Exploit
 - CWE: CWE-95, CWE-94
- BDSA** BDSA-2018-1901 (CVE-2018-11040)
 - Used By: 4 Project Versions
 - Overall Risk: 3.2 Low
 - First Detected: 2/1/2021 Published: 6/20/2018 Last Modified: 6/20/2018
 - Solution: ✓
 - Workaround: ✓
 - Exploit: No Exploit
 - CWE: CWE-200
- BDSA** BDSA-2019-4008 (CVE-2019-17571)
 - Used By: 4 Project Versions
 - Overall Risk: 6.4 Medium
 - First Detected: 2/1/2021 Published: 12/20/2019 Last Modified: 7/16/2020
 - Solution: No Solution
 - Workaround: ✓
 - Exploit: Exploit
 - CWE: CWE-502
- BDSA** BDSA-2019-1179 (CVE-2019-10246)
 - Used By: 4 Project Versions
 - Overall Risk: 3.9 Low
 - First Detected: 2/1/2021 Published: 4/23/2019 Last Modified: 4/23/2019
 - Solution: ✓
 - Workaround: No Workaround
 - Exploit: Exploit
 - CWE: CWE-200

Results Summary
224 Vulnerabilities
Results updated at Feb 5, 2021 7:56 AM
[Saved Search Settings](#)

The following information is shown for each vulnerability:

A single vulnerability card is shown:

BDSA BDSA-2020-1234 (CVE-2020-13430)

Used By: 0 Project Versions Overall Risk: 8.1 High

First Detected: Never Published: 5/27/2020 Last Modified: 7/27/2020

Solution: ✓ Workaround: ✓ Exploit: No Exploit

CWE: CWE-79

- Select the vulnerability ID to view more information about the vulnerability, such as additional score values. 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](#).
- View the number of project versions that affected by this vulnerability next to **Used By**.

Used By: 2 Project Versions

Select **Project Versions** to open the **Affected Projects** tab for the vulnerability which lists the project versions affected by this vulnerability.

The screenshot shows a Black Duck Security Advisory page for a specific vulnerability. At the top, there's a header with the title "Apache HttpClient Vulnerable to Man-In-The-Middle (MITM) Attack via SSL Hostname Verification Bypass", the date "BDSA-2014-0126 | CVE-2014-3577 | Published May 30, 2019 | Updated Feb 7, 2020", and navigation tabs for "Overview", "Affected Projects", "Technical", "CVE References", and "Settings". Below the header is a search bar with the placeholder "Filter projects...". A "Remediate" button is also present. The main content is a table with the following data:

Project	Component	Component Origin	Status	Target date	Actual date
cloudfoundry-identity-parent 3.6.3	Apache HttpClient 3.1	maven/commons-httpclient:commons-httpclient:3.1	New	Never	Never
cloudfoundry-identity-parent 3.6.3	Apache HttpClient 4.3.3	maven/org.apache.httpcomponents:httpclient:4.3.3	New	Never	Never
cloudfoundry-identity-parent 3.6.3	Apache HttpComponents Core 4.3.2	maven/org.apache.httpcomponents:httpcore:4.3.2	New	Never	Never
cloudfoundry-identity-parent 3.6.3	Apache HttpComponents Core 4.3.3	maven/org.apache.httpcomponents:httpcore:4.3.3	New	Never	Never

At the bottom right of the table area, it says "Displaying 1-4 of 4".

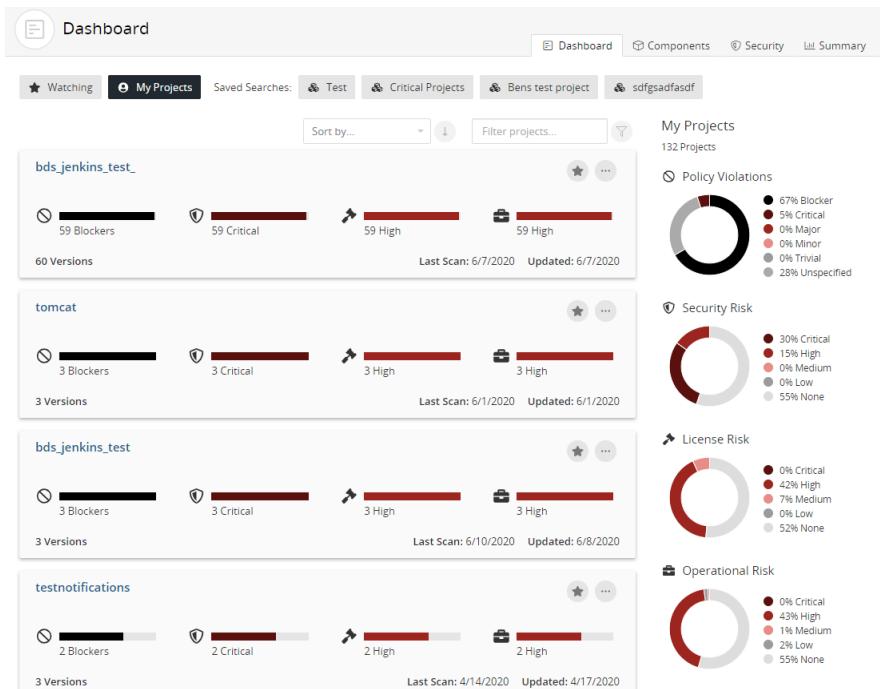
- View the overall risk score. The search results show the Temporal Score for BDSA vulnerabilities, or the Base Score for NVD vulnerabilities and the associated risk level. Note that the score shown and risk level depends on the [selected security rankings](#).

Select the score to view individual scores: temporal, base, exploitability, and impact for BDSA; base, exploitability, and impact for NVD.

- View whether a solution, workaround, or exploit is available:
 - ✓ indicates that there is a solution or workaround available for this vulnerability.
 - ✗ indicates there is an exploit for this vulnerability.
- For each vulnerability, the search results also show:
 - First Detected.
 - Published date.
 - Last modified date.
 - Common Weakness Enumeration (CWE) number for this security vulnerability.

Viewing overall risk for all projects

The Watching, My Projects, and saved search dashboards show the overall risk across all projects where you are a project team member.



Click [here](#) for more information about using this page to understand security vulnerabilities associated with your projects.

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 v2 or CVSS v3.x scores, depending on which [security risk calculation you selected](#); by default, CVSS v2 scores are shown. Note that the graph displays a Critical risk category with a value of 0, if you selected CVSS v2.

Possible risk categories are:

- Critical. The project has critical severity vulnerabilities.
- High. The project has 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.

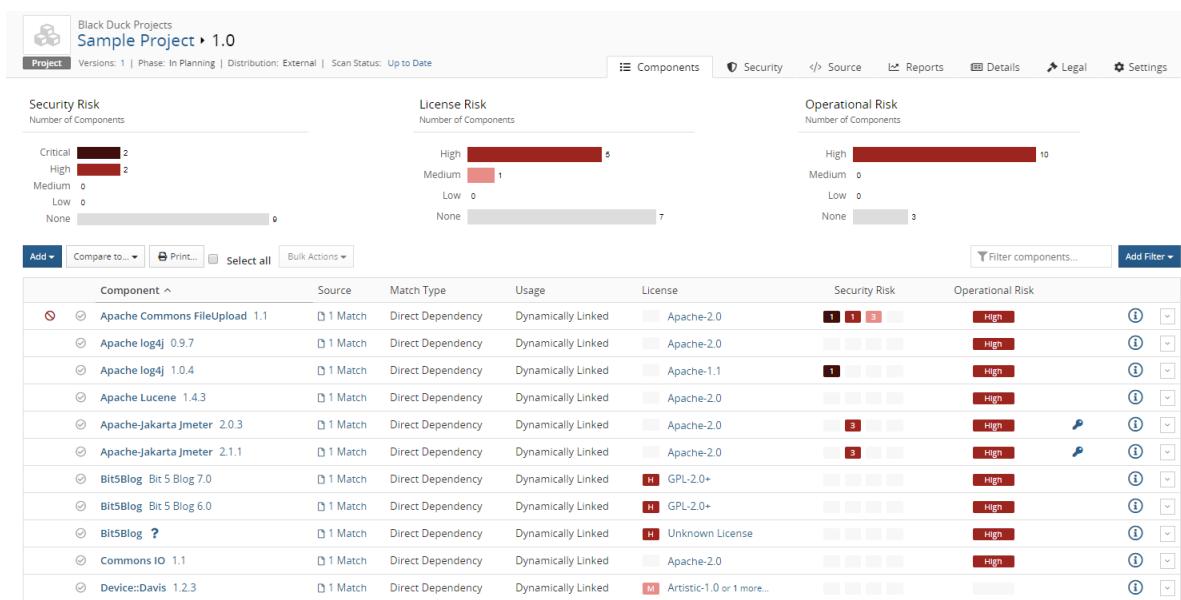
- **Operational Risk.** Operational risk is based on a combination of factors: (1) the strength of the component community, including the number of contributors and the level of commit activity; and (2) the number of newer versions of the component that are available than the one that is currently in 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.

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 v2 or CVSS v3.x scores, depending on which [security risk calculation you selected](#); by default, CVSS v2 scores are shown. Note that the graph displays a Critical risk category with a value of 0, if you selected CVSS v2.

Possible risk categories are:

- Critical. The number of components in this project version with critical severity vulnerabilities.
- High. The number of components in this project version with 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.

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 v2 or CVSS v3.x scores, depending on which [security risk calculation you selected](#); by default, CVSS v2 scores are shown. Note that the graph displays a Critical risk category with a value of 0, if you selected CVSS v2.

Possible risk categories are:

- Critical. The component has critical high severity vulnerabilities.
- High. The component has 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.

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. Note that Black Duck displays the numbers together in reports and in the UI because they represent the same vulnerability from different sources.

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 v2 or CVSS v3.x scores. By default, Black Duck displays CVSS v2 scores.

- CVSS v2 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.

Although CVSS v2 does not have a Critical risk category, the security graphs In the Black Duck UI display a Critical risk category. This category will display a value of 0 for CVSS v2.

- CVSS v3.x 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 the scores shown for CVSS v3.x can be v3.0 or v3.1 scores.

Suggested work flow

To manage security risk using Black Duck:

1. With the assistance of your security team, determine your security risk policies.
2. If necessary, users with the system administrator role can [define the default security ranking](#).

Note that the security ranking also defines how vulnerabilities appear in reports. Depending on the data available, the vulnerability will be presented as either: BDSA (NVD) or NVD (BDSA). For example, if the security ranking is NVD2, BDSA2, BDSA3, NVD3 then:

- Vulnerability A has data for just NVD3. The vulnerability is listed as NVD-1234-5678 in the report.
 - Vulnerability B has data for NVD3 and BDSA3. The report lists it as BDSA (NVD).
 - Vulnerability C has data for everything. The report lists it as NVD (BDSA).
3. [Create policies](#) that trigger violations when components do not comply with your security policies.
 4. Depending on your interests:
 - Use the [Summary Dashboard](#) to view the overall health of your projects and identify areas of concern. Use this page to quickly assess areas where you need to focus your attention.
 - Use these Dashboard pages for a high-level overview of risk:
 - [Use the Watched or My Project dashboards](#) to view the security risk across all your projects.
 - [Create saved searches](#) to customize the information shown on the Dashboard page to view the projects, components, and vulnerabilities that interests you.
 - Use these pages for project version-level information:
 - [project version page/Components tab](#), also known as the project version BOM, to view the components specific to that project version, that have security risk.
 - [project version page/ Security tab](#) to view the security vulnerabilities of each severity associated with the components used in a project version.
 5. Investigate vulnerabilities and policy violations. For detailed information on security vulnerabilities, view the:
 - [CVE page](#)
 - [BDSA page](#) if you have licensed Black Duck Security Advisories (BDSA)
 6. After reviewing the severity of the vulnerability, users with the appropriate [role](#) can [change the remediation status](#) of the security vulnerability.
 7. [Monitor notifications](#) for any new security vulnerabilities.

You will receive notification alerts if security vulnerabilities are published or updated against components that are included in one or more of your projects.

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 v2
 2. NVD v3.x
- If you have BDSA licensed, the default order is:
 1. BDSA v2
 2. NVD v2
 3. BDSA v3.x
 4. NVD v3.x

As shown above, by default Black Duck defines security risk initially using CVSS v2 scores. You can modify the order by which Black Duck determines security risk so that CVSS v3.x scores are used.

Note the following:

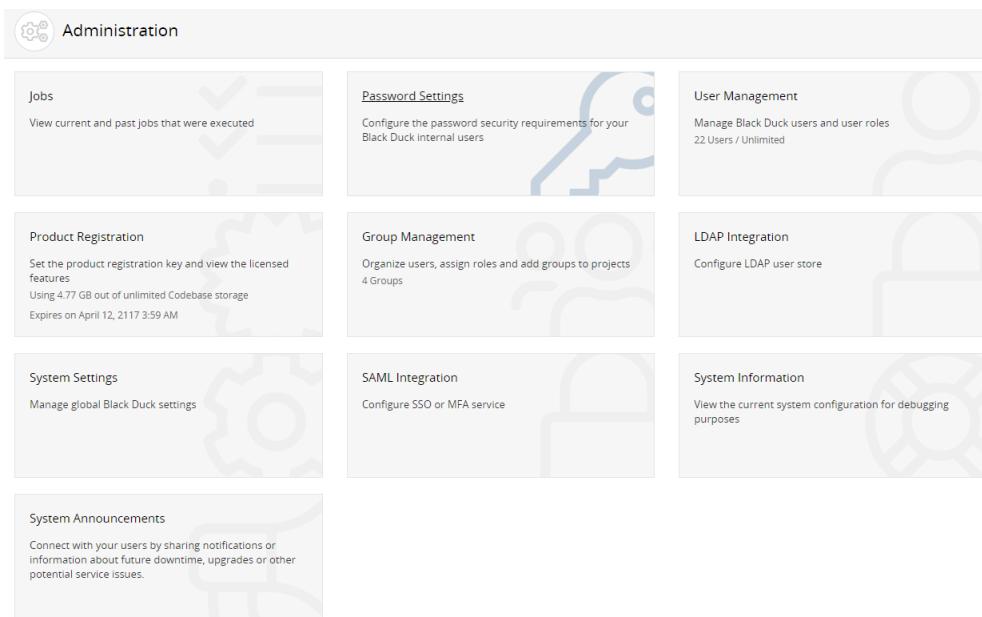
- Changing the order of the security risk configuration will result in revised security risk calculations for all project version BOMs and may result in new policy violations. These calculations may take a *considerable amount of time* to complete.
- The ability to change the security risk ranking is 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  Admin.

The Administration page appears.



3. Select System Settings.

The System Settings page appears.

The screenshot shows the System Settings page. At the top, there is a "Logo" section where a logo for "SYNOPSYS" is displayed. Below it is a "System Logs" section with a "Download Logs (.zip)" button. The "Legal Tab Visibility" section has a "Disable" button. The "Security Risk Configuration Ranking" section contains a list of tiles for "BDSA (CVSS v2)", "NVD (CVSS v2)", "BDSA (CVSS v3.x)", and "NVD (CVSS v3.x)". A "Save" button is located to the right of this section. The "Custom Scan Signature Level" section includes a "Levels" input field set to "5" and a "Save" button. The "Snippet Max File Size" and "Maximum Snippet File Size" sections both have "Save" buttons.

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 the security vulnerabilities of your projects, project versions, and component versions

Use your [dashboards](#) to view the types and severity of risk that are associated with the components that are in one or more versions of your projects. Dashboards provide an overall view of risk across the components in your projects and project versions. Use the project version **Security** tab to view a list of vulnerabilities for each component version origin.

Related vulnerabilities

Note that BDSA-1234-6789 or CVE-1234-5678 is the ID for a single vulnerability from BDSA or NVD: there is one vulnerability, but there are two databases and each has its own set of IDs to distinguish the same vulnerability.

There can be instances when the Black Duck UI shows vulnerabilities as related and in other instances (for example a different component version origin) when the same vulnerabilities are not shown as related. This may occur as sometimes NVD or BDSA does not evaluate certain origins, components, or component versions.

For example, suppose vulnerability X is found; NVD identifies it as CVE1 and BDSA identifies it as BDSA1. NVD has also found vulnerability X in component version origin A and component version origin B but BDSA has only found it in component version origin B (BDSA has either decided NVD is incorrect or not evaluated it). If your BOM has component version origin A, the project version's **Security** tab displays just the NVD identifier (CVE1) for that component version origin. BDSA does not apply in this context because it is not linked to this component version origin. If your BOM has component version origin B, both NVD and BDSA have found that Vulnerability X applies. You will see either BDSA 1 (CVE 1) or CVE 1 (BDSA 1) depending on your security priority system settings.

If NVD does not find the exploit at all then Black Duck only lists the BDSA ID, whether it be for specific component version origins or just generally looking up the BDSA identifier in the Black Duck application.

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 v2 or CVSS v3.x scores, depending on which [security risk calculation you selected](#); by default CVSS v2 scores are shown. Note that the graph displays a Critical risk category

with a value of 0, if you selected CVSS v2.

The screenshot shows the Black Duck Project interface for the 'Sample Project' dated 2021_06_15-23_27_39. The top navigation bar includes tabs for Components, Security, Source, Reports, Details, Legal, and Settings. A search bar and filter buttons are also present. On the left, a 'Security Risk' graph shows a horizontal scale from Critical (0) to Low (10), with 19 Medium vulnerabilities highlighted in red. Below the graph, a sidebar lists components with their versions and vulnerability counts: Apache Commons Codec 1.10 (0), Apache Commons FileUpload 1.2.2 (2), Apache HttpClient 4.4.1 (3), and Apache Commons FileUpload 1.2.2 (2). The main content area displays the 'Apache Commons FileUpload 1.2.2' component details. It shows a 'Transitive dependency' of commons-fileupload-1.4 with 1 match. A 'Short Term Upgrade Recommendation' for commons-fileupload-1.4 indicates no known vulnerabilities. A 'Long Term Upgrade Recommendation' for commons-fileupload-1.4 also indicates no known vulnerabilities. A table lists 5 known vulnerabilities, each with an identifier (BDSA), date (e.g., BDSA-2013-0013), overall score (e.g., 6.5), status (Medium), CWE (e.g., CWE-626), exploit (✓), workaround (✓), and solution (✓). The table is paginated at the bottom with 'Displaying 1-5 of 5'.

This page has these sections:

- Security Risk graph.
- Components list.
- Filters.
- Remediation guidance section, shown above the vulnerabilities table. Click [here](#) for more information about this feature.
- Vulnerabilities table.

Security Risk graph

The Security Risk graph shows how many vulnerabilities of each severity for each component version and subproject used in this version of the project.

The Security Risk graph shows the number of components with vulnerabilities for each severity 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.

Note: The number of components with vulnerabilities shown here may not be the same value as shown in your project version BOM (**Components** tab). In the BOM, the security graph aggregates similar components with different origins. On this page, the graph displays security risk by unique component origins, as a vulnerability may be origin-specific.

Select a severity level in the Security Risks graph to view all components that share the same level of risk.

Components list

This section lists each component with vulnerabilities. For each component, the component name, component version, and origin are shown along with risk bars that list how many vulnerabilities of each severity exist in this component version or subproject.

Select the component to display its vulnerabilities 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 list, then select the link shown on the page which displays the vulnerabilities for the subproject.

Filters

Use the **Filter components** field to view specific components. Click  to view other available filters.

- Some filter options apply to the values shown in the vulnerabilities table. If you select those filter options, components that have at least one vulnerability with the specified filter value will appear on the page.
- Filters filter the list of components shown on the left side of the page. However, the data shown in the vulnerability table for those components is not filtered.

For example, if you select to view those components that have vulnerabilities with an overall score greater than 9.0, the page displays the list of components that have at least one vulnerability with an overall score greater than 9.0. The information shown in the vulnerability table for those components is not filtered: it still shows all vulnerabilities for the filtered components, including those vulnerabilities with an overall score less than 9.0.

Vulnerabilities table

Initially, the vulnerabilities table shows the vulnerabilities of the first component in the Components list. Select a component to display its vulnerabilities.

The vulnerabilities table lists the following information for each vulnerability:

Column	Description
Identifier	<p>The identifier, value associated with this vulnerability, and any vulnerability tags (if applicable).</p> <p>Currently, the only vulnerability tag displayed is the Remote Code Execution tag. Hovering your cursor over the tag will provide a description of the vulnerability.</p> <p>Select > in the table next to the vulnerability to view a brief description. Depending on the identifier, select to view the BDSA record and/or the CVE record.</p> <p>Users with the appropriate role can also use this section to remediate the vulnerability.</p>
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"> For BDSA, the Temporal, Base, Exploitability, and Impact scores are shown. For NVD, the Base, Exploitability, and Impact scores are shown. <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"> Access Vector (AV) - CVSS v2 / Attack Vector (AV) - CVSS v3.x Access Complexity (AC) - CVSS v2 / Attack Complexity (AC) - CVSS v3.x Authentication (Au) Integrity (I) Availability (A) Confidentiality (C) <p>Note: The Authentication value is not available for CVSS v3.x 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	Remediation status of this vulnerability. Possible values are: Duplicate, Ignored, Needs Review, New, Mitigated, Patched, Remediation Complete, or Remediation Required.
CWE	Common Weakness Enumeration (CWE) number for this security vulnerability. - indicates a CWE number is not available.
Exploit	<p>Indicates whether an exploit for this vulnerability is available:</p> <ul style="list-style-type: none"> - No exploit available ✓ Exploit available

Column	Description
Workaround	Indicates whether a workaround for this vulnerability is available: <ul style="list-style-type: none"> • - No workaround available • ✓ Workaround available
Solution	Indicates whether a solution for this vulnerability is available: <ul style="list-style-type: none"> • - No solution available • ✓ Solution available

Analyzing the impact of a vulnerability

If a project version has several vulnerabilities, how can you decide which vulnerability you should focus on first?

To help you to prioritize which vulnerabilities you should address first, Black Duck can determine if any external public methods called by your Java applications are potentially involved in a known vulnerability. Black Duck can identify the called fully qualified public functional names in your source code and match them to the known function names being exploited by a vulnerability. By knowing whether any external public methods called by your Java applications are potentially involved in a known vulnerability, you can prioritize what vulnerabilities you need to concentrate on.

Vulnerability impact analysis works with Black Duck Security Advisories (BDSAs), a Black Duck-exclusive vulnerability data feed. Vulnerability metadata has been added to BDSAs that includes the fully qualified public function names that expose the vulnerability. Using this data, Black Duck can determine if vulnerable code is more likely to be invoked, and flags those vulnerabilities as reachable, indicating to you that these vulnerabilities are a higher priority for remediation.

Important: Vulnerability impact analysis can help you triage vulnerabilities in open source components. It is *not* a definitive list of the vulnerabilities that do or do not affect your code. Although Black Duck may indicate that a vulnerability is reachable, it does not indicate that the vulnerability definitely affects your code, as your code may not trigger the vulnerability. Likewise, Black Duck may not denote that a vulnerability is reachable. However, that does not indicate that your code is safe from the vulnerability as Synopsys may not have data for a specific vulnerability.

Vulnerability impact analysis process

The process to display the possible impact of a vulnerability in Black Duck is:

1. Synopsys Detect analyzes the code

When the `--detect.impact.analysis.enabled` property in Synopsys Detect to set to `true`, Synopsys Detect creates a call graph (a list of calls made by your code) to understand the public methods your code is using in your application. The call graph shows the fully qualified public method names as well as the line number where the function was called.

Along with creating BOM files, Synopsys Detect creates a file which will be used by the Black Duck KnowledgeBase for matching call graph signatures against BDSA-provided function signatures. This file is

encrypted with SHA1 hashes. Hashing of the call graph signatures is completed at the client system.

The data is packaged into a single file and Synopsys Detect sends the file over HTTPS to the Black Duck server.

2. Black Duck sends data to the KnowledgeBase

Black Duck sends the hashed call graph function signatures to the Black Duck KnowledgeBase via a Black Duck KnowledgeBase API.

3. Black Duck KnowledgeBase identifies vulnerable methods

The Black Duck KnowledgeBase uses the Function Signature Match Service to compare the hashed call graph signatures to KnowledgeBase hashed data for BDSA vulnerabilities with associated fully qualified public method names metadata. Fully qualified public method name matching is similar to signature matching: a discovered set of fully qualified public method names is compared against those known to the Black Duck KnowledgeBase.

The Black Duck KnowledgeBase sends the vulnerability metadata (vulnerable methods) to the Black Duck server via HTTPS.

4. Black Duck identifies vulnerabilities

Black Duck creates the data that needs to persist in the Black Duck PostgreSQL database. The data is stored as a new scan type (Vulnerability Impact) in Black Duck.

Black Duck cross references the identified methods with the vulnerable methods from BDSA to identify which vulnerabilities the user is calling in their code.

Black Duck displays the vulnerabilities for a component that have a method that is being called in the project version's **Security** tab.

Viewing reachable vulnerabilities

To view reachable vulnerabilities:

1. Set the **--detect.impact.analysis.enabled** property in Synopsys Detect to **true** to enable vulnerability impact analysis.
2. Once scanning completes and Black Duck has built the BOM, open the project version's **Security** tab which lists the security vulnerabilities for this project version.
3. Select a component from the **Component** list on the left side of the page to view a table which lists the vulnerabilities for this component.

The screenshot shows the Black Duck Project vulnImpactProject dashboard. At the top, there's a navigation bar with tabs for Components, Security, Source, Reports, Details, and Settings. Below the navigation is a 'Security Risk' summary with a horizontal bar chart showing the number of unique component origins across Critical, High, Medium, and Low risk levels. A 'Filter Components...' button and an 'Add Filter' button are also present.

The main content area displays a list of components on the left, including Apache Ant 1.8.2, Apache Commons BeanUtils 1.9.3, and Apache Commons Email 1.2. On the right, detailed information is shown for Apache Commons Email 1.2, including a 'Known Vulnerabilities' section (2 known vulnerabilities), a 'Short Term Upgrade Recommendation' (version 1.18, status 1.18, no known vulnerabilities), and a 'Long Term Upgrade Recommendation' (version 6.2, status 6.2, no known vulnerabilities). A table below lists two BDSA records: BDSA-2017-0721 (CVE-2017-9801) and BDSA-2018-0558 (CVE-2018-1294), both marked as 'Reachable'. The table includes columns for Identifier, Overall Score, Status, CWE, Exploit, Workaround, and Solution. A note at the bottom right says 'Displaying 1-2 of 2'.

located next to the BDSA record number indicates that there is a function call in your code that makes this vulnerability reachable. Use the **Reachable** filter to view all such BDSA records.

Click to view more information for an individual BDSA record.

The dialog box is titled 'Reachable Vulnerability'. It contains a message: 'Vulnerable function calls in your code that make this vulnerability reachable.' Below this is a list: 'Apache Commons Email 1.0' with 'BDSA-2017-0721 (CVE-2017-9801)'. Under 'Vulnerable Functions', there's a dropdown menu showing 'org.apache.commons.mail.Email.setSubject'. In the 'Calls' section, it says 'We found 1 function call' and shows 'Called by: com.nickavv.struts2test.App.main Line: 17'. A note at the bottom right says 'Displaying 1-1 of 1'. A 'Close' button is at the bottom right.

The Reachable Vulnerability dialog box lists:

- Component name and version
- BDSA record
- A list of all vulnerable function calls in your code.

Select a function name to view the method name and line number in your code.

Note the following:

- This feature is available in Synopsys Detect version 6.5 or later (and Synopsys Detect (Desktop) that uses Synopsys Detect 6.5 and later). Set the **--detect.impact.analysis.enabled** property in Synopsys Detect to **true** to enable vulnerability impact analysis.
- This feature is for Java applications only. Synopsys Detect looks at Java .class files only.

- Synopsys Detect only discovers vulnerabilities in public methods that call potentially vulnerable functions.
- This feature displays reachable functions for BDSAs only.
- There is a [project version report](#), `vulnerability_matches_date_time.csv`, that lists the component, vulnerability data, and vulnerability impact analysis data for each component potentially reached by a vulnerability.
- A vulnerability condition, **Reachable from Source**, is available enabling you to create policy rules for vulnerabilities which has been identified as reachable. Use this condition to create policy rules that prioritize those vulnerabilities.

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 Commons FileUpload Vulnerable to NULL Byte Poisoning
BDSA BDSA-2013-0013 | CVE-2013-2186 | Published Mar 22, 2018 | Updated Mar 5, 2020

MEDIUM 6.5 BDSA Fix Available Nov 23, 2016 Exploit Available Apr 14, 2016 2823 Days Vulnerability Age

Apache Commons FileUpload was discovered to be vulnerable to NULL byte poisoning due to the incorrect handling of filenames containing NULL bytes. The failure to check for NULL byte allows a remote unauthenticated attacker to upload or overwrite files on the system leading to unauthorized file modification, denial-of-service or remote code execution (RCE).

Remote Code Execution
This vulnerability allows an attacker to remotely execute arbitrary code.

How to fix it

 **Solution - Fix Available**
Fixed in version 1.3.3 with this commit.

Workaround
Applications utilising FileUpload prior to version 1.3.3 in the classpath can remediate by limiting the classes, including DiskFileUpload, that are deserialized and through validation and verification of filenames in FileUpload control.

Common Vulnerability Scoring System (CVSS)

BDSA (CVSS v2)	BDSA (CVSS v3.x)	NVD (CVSS v2)
6.5	6.5	7.5
Overall (Temporal)	Temporal	Base
		Exploitability
		Impact
6.5 Overall		
(AV:N/AC:L/Au:N/C:P/I:P/A:P/E:H/RL:OF/RC:C)		

6.5 Temporal

TEMPORAL METRICS

Exploitability	Not Defined	Unproven	Proof of Concept	Functional	High
Remediation Level	Not Defined	Official Fix	Temporary Fix	Workaround	Unavailable
Report Confidence	Not Defined	Unconfirmed	Uncorroborated	Confirmed	

10 Exploitability

EXPLOITABILITY METRICS

Access Vector	Local	Adjacent Network	Network
Access Complexity	High	Medium	Low
Authentication	Multiple Instances	Single Instance	None

6.4 Impact

IMPACT METRICS

Confidentiality Impact	None	Partial	Complete
Integrity Impact	None	Partial	Complete
Availability Impact	None	Partial	Complete

Common Weakness Enumeration (CWE)
CWE-626 - Null Byte Interaction Error (Poison Null Byte)
The product does not properly handle null bytes or NUL characters when passing data between different representations or components.

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**.

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.
 - Vulnerability Age. Today's date - Disclosure date.
 - A brief description of the vulnerability.
- A Remote Code Execution banner is displayed if this vulnerability is found for this particular record.



Remote Code Execution

This vulnerability allows an attacker to remotely execute arbitrary code.

- 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 v2, CVSS v3.x, and if available, NVD v2 and NVD v3.x scores.

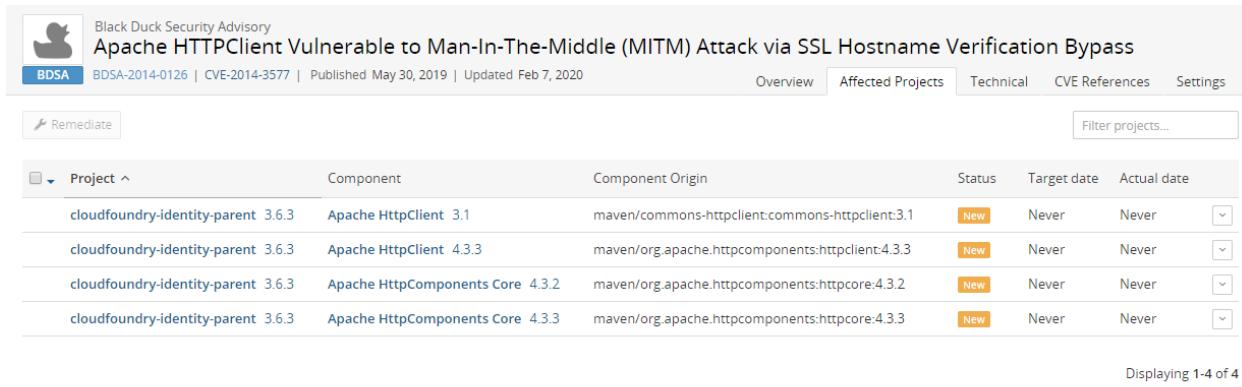


Select a value above the graph to view the information in the graph and details below.

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 web-based security advisory tool. At the top, it displays the title "Black Duck Security Advisory" and the specific advisory "Apache HttpClient Vulnerable to Man-In-The-Middle (MITM) Attack via SSL Hostname Verification Bypass". Below this, it shows the identifier "BDSA-2014-0126 | CVE-2014-3577 | Published May 30, 2019 | Updated Feb 7, 2020". A navigation bar at the top right includes links for "Overview", "Affected Projects", "Technical", "CVE References", and "Settings". Below the navigation, there is a button labeled "Remediate" and a search bar labeled "Filter projects...". The main content area is a table titled "Affected Projects" with the following columns: "Project", "Component", "Component Origin", "Status", "Target date", and "Actual date". The table lists four entries, all of which are marked as "New" status and have a target date of "Never". The rows are for "cloudfoundry-identity-parent" version 3.6.3, which contains "Apache HttpClient 3.1" and "Apache HttpClient 4.3.3" from "maven/commons-httpclient:commons-httpclient:3.1" and "maven/org.apache.httpcomponents:httpclient:4.3.3" respectively, and also contains "Apache HttpComponents Core 4.3.2" and "Apache HttpComponents Core 4.3.3" from "maven/org.apache.httpcomponents:httpcore:4.3.2" and "maven/org.apache.httpcomponents:httpcore:4.3.3". The bottom right corner of the screenshot indicates "Displaying 1-4 of 4".

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.
- Component origin 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.

Select  in the row of a project 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.

Use this tab to remediate the vulnerability for one or more projects by origin:

- In the row of the single project you want to remediate, do one of the following:
 - Select , select **Update Remediation Plan**, enter the remediation details, and click **Update**.
 - Select  and click **Remediate**. Enter the remediation details, and click **Update**.
- For multiple projects that need the same remediation status, select  in each row and click **Remediate**. In the Bulk Remediation dialog box, enter the remediation details, and click **Update**

Technical tab

Select the **Technical** tab to view a technical description and a list of references and related links.

The screenshot shows a detailed security advisory for Apache Tomcat. At the top, there's a logo for Black Duck Security Advisory and the title "Apache Tomcat Vulnerable to Reflected Cross-Site Scripting (XSS) via SSI 'printenv' Debugging Command". Below the title, it says "BDSA-2019-1661 | CVE-2019-0221 | Published May 30, 2019 | Updated May 30, 2019". There are tabs for "Overview", "Affected Projects", "Technical" (which is selected), "CVE References", and "Settings". The "Technical Description" section explains that 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. The "References and Related Links" section lists "Advisories" and "Vendor Upgrade" links, each with a small icon and a list of URLs.

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.
- Vulnerability Age. Today's date - Disclosure date.
- 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 list of affected projects for this vulnerability. The 'All' tab is selected, showing 27 entries. Other tabs include BID (1), BUGTRAQ (1), CONFIRM (3), DEBIAN (1), FEDORA (2), FULLDISC (1), GENTOO (1), MISC (2), MLIST (8), N/A (1), REDHAT (2), SUSE (2), and UBUNTU (2). Each entry includes a link to the original source.

Project	Count
All	27 >
BID	1
BUGTRAQ	1
CONFIRM	3
DEBIAN	1
FEDORA	2
FULLDISC	1
GENTOO	1
MISC	2
MLIST	8
N/A	1
REDHAT	2
SUSE	2
UBUNTU	2

Settings tab

Use this tab to manage the global remediation for this vulnerability. Click [here](#) for more information.

CVE record

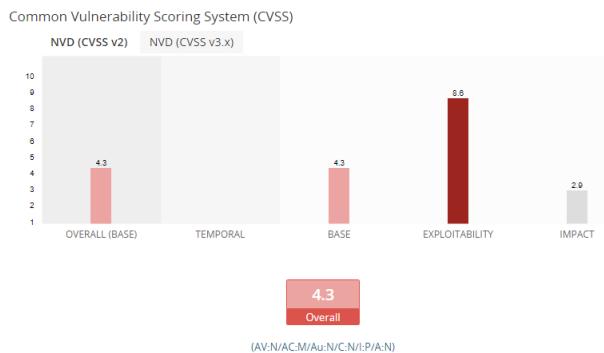
Vulnerabilities are linked to components by the Common Vulnerabilities and Exposures numbers (CVEs), as reported in the National Vulnerabilities Database (NVD) maintained by the National Institutes of Standards and Technology (NIST).

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 v2 and CVSS v3.x scores.



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 the "Affected Projects" tab for CVE-2014-3577. The page header includes the NVD logo, the title "National Vulnerability Database", the CVE number "CVE-2014-3577", and the publication date "Published Aug 21, 2014 | Updated Jul 18, 2018 | https://nvd.nist.gov/vuln/detail/CVE-2014-3577". The tabs at the top are Overview, Affected Projects (which is selected), References, and Settings. A "Filter projects..." input field is present. The main content table lists two projects affected by the vulnerability:

Project	Component	Component Origin	Status	Target date	Actual date
cloudfoundry-identity-parent 3.6.3	Apache HttpClient 3.1	maven/commons-httpclient:commons-httpclient:3.1	New	Never	Never
cloudfoundry-identity-parent 3.6.3	Apache HttpClient 4.3.3	maven/org.apache.httpcomponents:httpclient:4.3.3	New	Never	Never

At the bottom right, it says "Displaying 1-2 of 2".

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.

Select in the row of a project 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.

Use this tab to remediate the vulnerability for one or more projects by origin:

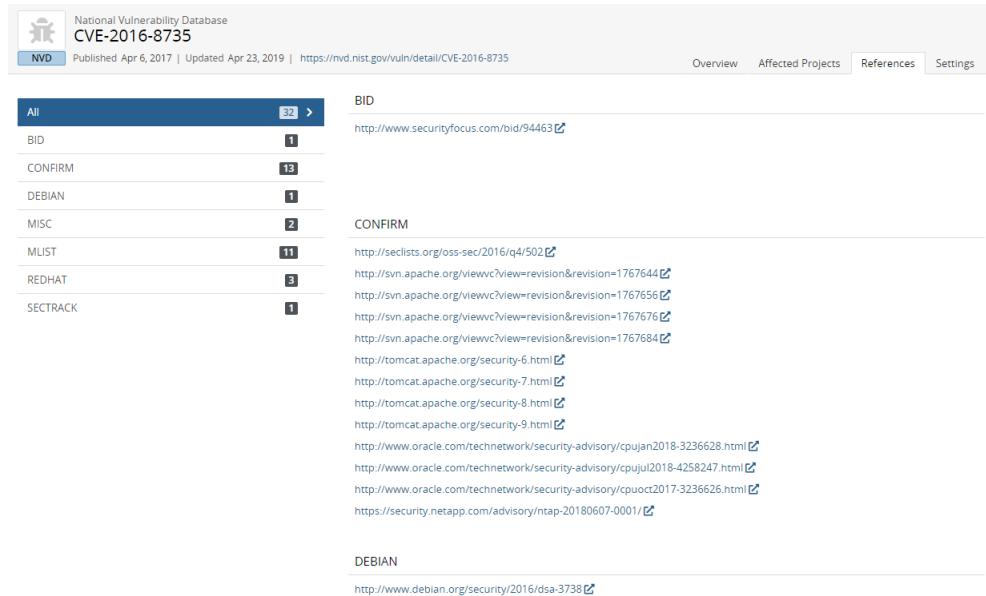
- In the row of the single project you want to remediate, do one of the following:
 - Select

 , select **Update Remediation Plan**, enter the remediation details, and click **Update**.

- Select  and click **Remediate**. Enter the remediation details, and click **Update**.
- For multiple projects that need the same remediation status, select  in each row and click **Remediate**. In the Bulk Remediation dialog box, enter the remediation details, and click **Update**

References tab

Select the **References** tab to view links for additional information.



The screenshot shows the National Vulnerability Database (NVD) interface for CVE-2016-8735. The top navigation bar includes the NVD logo, the title 'National Vulnerability Database CVE-2016-8735', and tabs for 'Overview', 'Affected Projects', 'References' (which is selected), and 'Settings'. Below the navigation is a search bar with the placeholder 'BID'. A sidebar on the left lists categories: All (32), BID (1), CONFIRM (13), DEBIAN (1), MISC (2), MLIST (11), REDHAT (3), and SECTRACK (1). The main content area displays a list of references under the 'CONFIRM' category, each with a link icon. The list includes URLs from securityfocus.com, seclists.org, snyk.io, apache.org, and oracle.com, along with a link to a NetApp advisory.

Settings tab

Use this tab to manage the global remediation for this vulnerability. Click [here](#) for more information.

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 can help 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.

You can remediate a vulnerability for current projects or set a [global remediation status](#) that applies to new instances of that vulnerability when that component appears on new BOMs.

Only users with the appropriate [role](#) can remediate vulnerabilities.

To remediate vulnerabilities for current projects

Use this method to identify and remediate the vulnerabilities affecting a specific project version.

1. Log in to Black Duck.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
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	CWE	Exploit	Workaround	Solution
BDSA	Nov 13, 2019	3.2	Low	New	CWE-79	-	✓
NVD	Jan 10, 2018	4.4	Medium	New	CWE-470	-	-

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 expand the row. A brief description, additional information, and fields to remediate the vulnerability appear. 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 and 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 project versions by origin:

- In the row of the single project you want to remediate, do one of the following:
 - Select



- , select **Update Remediation Plan**, enter the remediation details, and click **Update**.
- Select and click **Remediate**. Enter the remediation details, and click **Update**.
 - For multiple projects that need the same remediation status, select in each row and click **Remediate**. In the Bulk Remediation dialog box, enter the remediation details, and click **Update**.

Getting remediation guidance for components with security vulnerabilities

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 the possible versions of the component that are available to you:

- The version used in your BOM with the number of vulnerabilities.
- Dependency information. When direct or transitive dependencies are found in a Synopsys Detect scan, Black Duck lists the number of matches for each type of dependency.

Apache Tomcat 8.5.32

Cloud maven: org.apache.tomcat:tomcat-juli:8.5.32

Direct dependency 39 matches

Transitive dependency 2 matches

16 Known Vulnerabilities

Select **Transitive dependency** to view the dependency tree for that component.

Dependency Tree for Apache Tomcat 8.5.32

Cloud maven: org.apache.tomcat:tomcat-juli:8.5.32

Transitive dependency brought in by the following components:

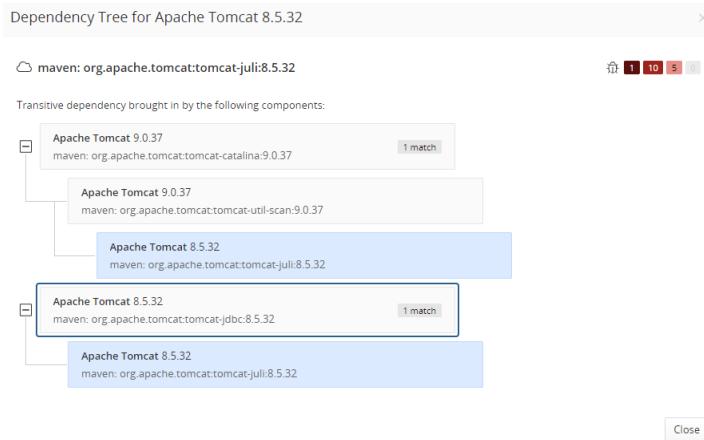
- + Apache Tomcat 9.0.37
maven: org.apache.tomcat:tomcat-catalina:9.0.37 1 match
- + Apache Tomcat 8.5.32
maven: org.apache.tomcat:tomcat-jdbc:8.5.32 1 match

Close

The dependency tree shows the components that brought in this dependency. In the upper right

corner is a list of the vulnerabilities by severity level, from left to right: Critical, High, Medium, and Low for this origin of this component version. The match count is the number of times the component was brought in with that dependency path.

Click + to open the tree.



- **Recommendations.** If available, Black Duck provides a short term and long term upgrade recommendation. In both instances, the recommended version has fewer reported vulnerabilities than the version you are currently using in your BOM. The recommended version is also from the same origin as the version you are currently using in your BOM.

- **Short Term Upgrade Recommendation.** This recommendation provides a short-term upgrade path as it is typically the same major version as the version currently used in your BOM.

Components using non-semantic versioning will not have a short-term recommendation.

- **Long Term Upgrade Recommendation.** Unlike the short term upgrade recommendation, this recommendation usually requires a major version upgrade. This may require more planning and/or engineering work to implement.

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 project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
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.

Identifier	Published	Overall Score	Status	CWE	Exploit	Workaround	Solution
BDSA-2019-3481	Nov 13, 2019	3.2	Low	New	CWE-79	-	✓
CVE-2017-7536	Jan 10, 2018	4.4	Medium	New	CWE-470	-	-

Managing global remediation for a vulnerability

There may be a vulnerability that appears frequently in your BOMs. Instead of repeatedly reviewing and remediating that vulnerability, set a global default remediation status for it.

After you set a global remediation status, when that vulnerability appears in new BOMs, it will automatically get the global remediation status you defined.

Note: Any component in your existing BOMs that has that vulnerability retains its current remediation status. The global remediation status only applies to *new* instances of the vulnerability.

You must have the Global Security Manager [role](#) to set or remove a global remediation status for a vulnerability.

Setting a global remediation for a vulnerability

Use this process to set a global remediation status or edit an existing status.

- Find the vulnerability you want to remediate globally. For example, you can:
 - Use the [Security tab for a specific project version](#).
 - Use the [Search feature](#).
- View the security record.
 - When viewing a list of vulnerabilities in a table, select > in the table next to the vulnerability to view a brief description. Then select either **View BDSA record** or **View CVE record**.

- When using the Search feature, select the BDSA or CVE record in the search results.
3. Select the **Settings** tab.

The screenshot shows a web-based application for managing software vulnerabilities. At the top, there's a header with the Black Duck logo, the title "Black Duck Security Advisory", and the specific advisory details: "Mozilla Thunderbird Vulnerable to Information Disclosure via Out-of-Bounds Read in 'nsParseMailbox.cpp' File". Below the header, a navigation bar includes links for "Overview", "Affected Projects", "Technical", "CVE References", and the currently selected "Settings" tab. The main content area is titled "Default Remediation Status" and contains a sub-instruction: "Select the remediation status that will be applied to this vulnerability in all future Project Versions.". Underneath this, there's a dropdown menu labeled "Status" with the placeholder "Select Default Remediation Status...". To the right of the dropdown is a large text input field for "Comments". At the bottom of the form are two buttons: "Clear" and "Save", with "Save" being highlighted in blue.

4. Select a default status and optionally, enter a comment. This comment appears when viewing the description, as described below.
5. Click **Save**.

The Default Remediation Status Confirmation dialog box appears.

6. Click **Confirm**.

Clearing a global default remediation status

You can remove a global default remediation status. Clearing a status only affects future vulnerabilities: components with the existing global vulnerability status will retain that status. To modify the status of the existing vulnerabilities, modify the remediation status manually either individually or by using bulk remediation.

1. Find and display the vulnerability record as described in the previous section.
2. Select the **Settings** tab.

The screenshot shows a web-based configuration interface for a security advisory. At the top left is a small icon of a duck. To its right, the text "Black Duck Security Advisory" is displayed. Below this, the title "Mozilla Thunderbird Vulnerable to Information Disclosure via Out-of-Bounds Read in 'nsParseMailbox.cpp' File" is shown. Underneath the title, there is a blue button labeled "BDSA" and some smaller text indicating the record number "BDSA-2020-0294" and the date "Published Feb 18, 2020 | Updated Feb 18, 2020". To the right of this information are several tabs: "Overview", "Affected Projects", "Technical", "CVE References", and "Settings". The "Settings" tab is currently selected, as indicated by a blue border around it. The main content area is titled "Default Remediation Status" and contains a sub-instruction: "Select the remediation status that will be applied to this vulnerability in all future Project Versions." Below this, there is a dropdown menu labeled "Status *" with the option "Remediation Required" selected. To the right of the dropdown is a text input field labeled "Comments" containing the text "Although the security risk is low, this vulnerability must be remediated." At the bottom right of the form are two buttons: "Clear" and "Save", with "Save" being the larger and darker blue button.

3. Click **Clear**.

The Default Remediation Status confirmation dialog box appears.

4. Click **Confirm**.

Viewing all vulnerabilities with global remediation

You can view all vulnerabilities with global remediation by selecting the **Default Remediation** filter when [searching for vulnerabilities](#). Select the BDSA or CVE record number in the search results and then select the **Settings** tab, as described previously to view the remediation status.

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.
 - Select a category for your rule. Black Duck provides these categories for a policy rule: component, security, license, operational, and uncategorized (default).

By using categories and filters, you can easily find policies (on the Policy Management page) or policy violations (on the BOM page) by category.

- [View the Project Version report](#). This report includes policy violation information:
 - The `components_date_time.csv`, `bom_component_custom_fields_date_time.csv`, and `source_date_time.csv` files list the policy status and override information.
 - The `version_date_time.csv` file indicates whether this version of the project has a

policy violation.

To assist you, Black Duck provides five [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 [Manage](#) and selecting **Policy Management**:

A screenshot of the Black Duck Policy Management interface. The title bar says 'Policy Management'. Below it is a table with three rows of policy rules. The columns are 'Policy Rule' (with a dropdown arrow), 'Description', 'Severity', and 'Category'. The first rule is 'No External Tier 1 or Tier 2 Projects With More Than 1 High Vulnerability', described as 'Disallow External Projects With More Than 1 High Vulnerability at Tier 1 or 2', with Severity 'Critical' and Category 'Security'. The second rule is 'No External Tier 1 or Tier 2 Projects With More Than 3 Medium Vulnerabilities', described as 'Disallow External Projects With More Than 3 Medium Vulnerabilities at Tier 1 or 2', with Severity 'Major' and Category 'Security'. The third rule is 'No Modified Components Without Description', described as 'Disallow components that have the modification flag on but do not have a description.', with Severity 'Unspecified' and Category 'Uncategorized'. At the bottom right of the table, it says 'Displaying 1-3 of 3'.

Policy Rule	Description	Severity	Category
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	Critical	Security
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	Major	Security
No Modified Components Without Description	Disallow components that have the modification flag on but do not have a description.	Unspecified	Uncategorized

- The page is filtered to display enabled rules. Modify or clear the filter to view disabled rules.
- All rules can be overridden unless noted.
- Click > to view the conditions of this rule and who created and last updated it.

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 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.

- Custom dashboards. 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:



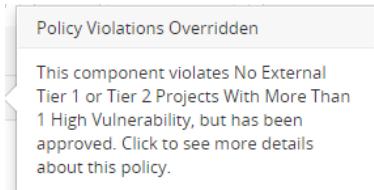
Clicking the icon (when viewing the BOM using the List view) displays the Policy Violations dialog box from which you can override the policy violation.

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. Click the icon to view more information.

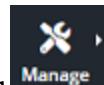


Removing policy overrides

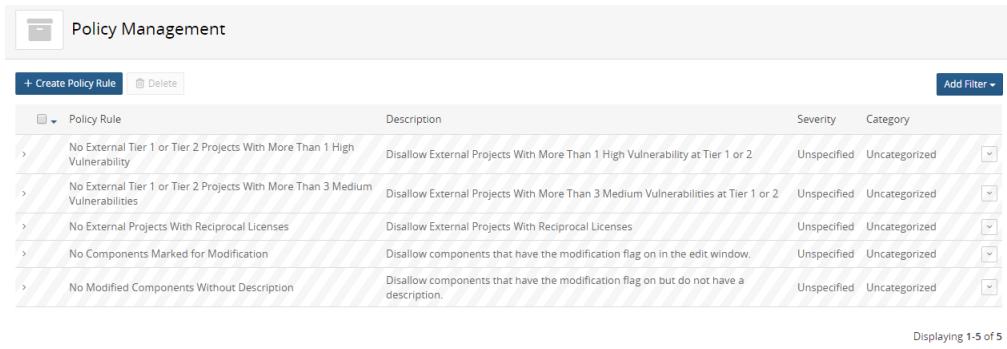
If a violation of a policy should not have been overridden, you can [remove](#) the override.

Default policy rules

Black Duck provides five 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 **Manage > Policy Management** and selecting to view disabled rules:



The screenshot shows a table titled "Policy Management" with the following data:

Policy Rule	Description	Severity	Category
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	Unspecified	Uncategorized
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	Unspecified	Uncategorized
No External Projects With Reciprocal Licenses	Disallow External Projects With Reciprocal Licenses	Unspecified	Uncategorized
No Components Marked for Modification	Disallow components that have the modification flag on in the edit window.	Unspecified	Uncategorized
No Modified Components Without Description	Disallow components that have the modification flag on but do not have a description.	Unspecified	Uncategorized

Displaying 1-5 of 5

Click > to view a description and the conditions for these rules.

These policy rules are in the **Uncategorized** category.

The default rules are:

- No External Projects With Reciprocal Licenses
- No External Tier 0, Tier 1 or Tier 2 Projects With More Than 1 High Vulnerability
- No External Tier 0, 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, component version, or vulnerability 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 to (all or specific project attributes) and then selecting:

1. A component and/or vulnerability attribute

You can create a policy rule for a component, a vulnerability, or for a component and vulnerability combination.

2. An operator (such as equals or greater than)
3. A value (depending on the option you selected)

Components that meet the conditions will violate the policy rule. For vulnerability conditions, components that have vulnerabilities that meet the vulnerability conditions violate the policy rule.

You can create multiple conditions for a policy rule: *all* conditions must be true for a component to be in

violation.

When evaluating components with multiple licenses for policy rules created using one or more of these license conditions: license, license status, license family and/or license expiration date, each license is evaluated and *all* license conditions must be true for a policy violation. If license risk is included as a policy condition, license risk is evaluated independently: all licenses for the component are evaluated, not just the license that met the other license policy conditions. Therefore, a policy violation can be triggered if one license meets the policy rule for multiple conditions while another license for that component meets the license risk condition.

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 are divided into these categories:

- Properties
- Project Custom Fields
- Project Version Custom Fields

Properties

Project Filters	Value
Project Name	Begin typing to view possible values.
Project Tags	Enter the tag name.
Project Tier	Enter one of the following values: 0 - 5.
Project Phase	Select one of the following values: <ul style="list-style-type: none">• Archived• Deprecated• In Development• In Planning• Pre-Release• Released
Project Distribution Type	Select one of the following values: <ul style="list-style-type: none">• External• Internal• Open Source• SaaS

Project Custom Fields

Project Filters	Value
Project Custom Field Name	Available for Boolean, Date, Drop Down, Multiple Selections, Single Selection, and Text field types. Select a value.

Project Version Custom Fields

Project Filters	Value
Project Version Custom Field Name	Available for Boolean, Date, Drop Down, Multiple Selections, Single Selection, and Text field types. Select a value.

Component conditions

Component conditions are divided into these categories:

- Properties
- Operational
- Vulnerabilities
- Licenses
- Custom Fields: BOM Component, Component, and Component Version

Properties

Component Condition	Value
Component	Begin typing to view possible component values. After selecting a component, the version field appears whereby you can enter a version number. Any Version is the default value if you do not enter a specific version.
Component Usage	Select one of the following values: <ul style="list-style-type: none"> • Dev. Tool / Excluded • Dynamically Linked • Source Code • Statically Linked • Separate Work • Implementation of Standard • Prerequisite • Merely Aggregated • Unspecified

Component Condition	Value
Review Status	Select one of the following values: <ul style="list-style-type: none"> • Not Reviewed • Reviewed
Newer Versions Count	Enter a number.
Match Type	Select one of the following values: <ul style="list-style-type: none"> • Files Added/Deleted • File Dependency • Direct Dependency • Transitive Dependency • Exact Directory • Exact File • Files Modified • Manually Added • Manually Identified • Partial • Snippet • Binary
Component Purpose	Select either Yes or No. Indicates whether information was added in the Purpose field when manually adding or editing a component.
Component Modified	Select either Yes or No. Indicates whether the Modification option was selected when manually adding or editing a component.
Component Modification	Select either Yes or No. Indicates whether information was added to the Modification field when manually adding or editing a component.
Component Approval Status	Select one of the following values: <ul style="list-style-type: none"> • Unreviewed • In Review • Reviewed • Approved • Limited Approval • Rejected • Deprecated

Component Condition	Value
Component Version Approval Status	Select one of the following values: <ul style="list-style-type: none"> • Unreviewed • In Review • Reviewed • Approved • Limited Approval • Rejected • Deprecated
Unknown Component Version	Select True or False . If you select True , any component that has a ? as the version will trigger a policy violation.

Operational

Component Condition	Value
Component Release Date	Select a date.
Commits in the past year	Enter a number.
Contributors in the past year	Enter a number.

Vulnerabilities

Component Condition	Value
Critical Severity Vulnerability Count	Enter a number.
High Severity Vulnerability Count	Enter a number.
Medium Severity Vulnerability Count	Enter a number.
Low Severity Vulnerability Count	Enter a number.
Highest Vulnerability Score	Enter a number between 0 and 10, including decimal numbers.

Licenses

Component Condition	Value
Unfulfilled License Terms	<p>Select True or False.</p> <p>If you select True, any component that has unfulfilled license terms will trigger a policy violation.</p> <p>Note: The Legal tab must be enabled for a user to indicate that a term is fulfilled. If the Legal tab is disabled, a user will be unable to indicate that a term is fulfilled, and policy violations cannot be cleared.</p>
License Conflict with Project Version	<p>Select True or False.</p> <p>If you select True, a policy violation is triggered when a component's license conflicts with the license for a project version.</p>
License Risk	<p>Select one of the following values:</p> <ul style="list-style-type: none"> • None • Low • Medium • High
License (Declared)	Begin typing to view possible declared license values.
License Family (Declared)	Select one of the following KB license families (Permissive, Reciprocal, Weak Reciprocal, AGPL, or Unknown) or a custom license family for the declared license.
License Status (Declared)	<p>Select one of the following values:</p> <ul style="list-style-type: none"> • Unreviewed • In Review • Reviewed • Approved • Limited Approval • Rejected • Deprecated
License Expiration Date (Declared)	Select an expiration date for the declared license.

Component Condition	Value
License Expiration Date Comparison (Declared)	<p>Use this condition to compare the declared license expiration date of a component to the project version release date. Specify a number which equals the number of days to <i>add to</i> the project version release date for the comparison. Black Duck triggers a policy violation if the date is less than or greater than that date.</p> <ul style="list-style-type: none"> • Use 'before' to trigger a policy violation when the license expiration date is more than X number of days before (or less than), the project version release date. • Use 'after' to trigger a policy violation when the license expiration date is more than X number of day after (or greater than), the project version release date. <p>The following are examples using 'before.' The project version release date is the 10th.</p> <ul style="list-style-type: none"> • Number of days = 0: triggers a policy violation when the license expiration date is the day before the project version release date (the 9th) or earlier. • Number of days = 1: triggers a policy violation when the license expiration date is the same day as the project version release date (the 10th) or earlier. • Number of days = 2: triggers a policy violation when the license expiration date is the 11th or earlier. • Number of days = -2: triggers a policy violation when the license expiration date is the 7th and earlier. <p>The following are examples using 'after.' The project version release date is the 10th.</p> <ul style="list-style-type: none"> • Number of days = 0: triggers a policy violation when the license expiration date is the day after the project version release date (the 11th) or later. • Number of days = -1: triggers a policy violation when the license expiration date is the same as the project version release date (the 10th) and later. • Number of days = 2: triggers a policy violation when the license expiration date is the 13th and later. • Number of days is -2: triggers a policy violation when the license expiration date is the 9th and later.
License (Deep License)	Begin typing to view possible deep (embedded) license values.
License Family (Deep License)	Select one of the following KB license families (Permissive, Reciprocal, Weak Reciprocal, AGPL, or Unknown) or a custom license family for the deep (embedded) license .
License Status (Deep License)	<p>Select one of the following values for the deep (embedded) license:</p> <ul style="list-style-type: none"> • Unreviewed • In Review • Reviewed

Component Condition	Value
	<ul style="list-style-type: none"> Approved Limited Approval Rejected Deprecated
License Expiration Date (Deep License)	Select an expiration date for the deep (embedded) license .
License Expiration Date Comparison (Deep License)	<p>Use this condition to compare the deep license expiration date of a component to the project version release date. Specify a number which equals the number of days to <i>add to</i> the project version release date for the comparison. Black Duck triggers a policy violation if the date is less than or greater than that date.</p> <ul style="list-style-type: none"> Use 'before' to trigger a policy violation when the license expiration date is more than X number of days before, or less than, the project version release date. Use 'after' to trigger a policy violation when the license expiration date is more than X number of day after, or greater than, the project version release date. <p>The following are examples using 'before.' The project version release date is the 10th.</p> <ul style="list-style-type: none"> Number of days = 0: triggers a policy violation when the license expiration date is the day before the project version release date (the 9th) or earlier. Number of days = 1: triggers a policy violation when the license expiration date is the same day as the project version release date (the 10th) or earlier. Number of days = 2: triggers a policy violation when the license expiration date is the 11th or earlier. Number of days = -2: triggers a policy violation when the license expiration date is the 7th and earlier. <p>The following are examples using 'after.' The project version release date is the 10th.</p> <ul style="list-style-type: none"> Number of days = 0: triggers a policy violation when the license expiration date is the day after the project version release date (the 11th) or later. Number of days = -1: triggers a policy violation when the license expiration date is the same as the project version release date (the 10th) and later. Number of days = 2: triggers a policy violation when the license expiration date is the 13th and later. Number of days is -2: triggers a policy violation when the license expiration date is the 9th and later.

BOM Component Custom Fields

Component Condition	Value
BOM Component Custom Field Name	Available for Boolean, Date, Drop Down, Multiple Selections, Single Selection, and Text field types. Select a value.

Component Custom Fields

Component Condition	Value
Component Custom Field Name	Available for Boolean, Date, Drop Down, Multiple Selections, Single Selection, and Text field types. Select a value.

Component Version Custom Fields

Component Condition	Value
Component Version Custom Field Name	Available for Boolean, Date, Drop Down, Multiple Selections, Single Selection, and Text field types. Select a value.

The table below shows the vulnerability attributes that you can select and the values you can specify.

Vulnerability conditions

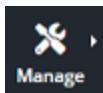
Vulnerability condition	Value
Overall Score	Enter a number from 0 to 10.
CWE IDs	Enter a Common Weakness Enumeration (CWE) number.
Solution Available	Select either Yes or No. Indicates whether there is a solution for the vulnerability.
Workaround Available	Select either Yes or No. Indicates whether there is a workaround available for the vulnerability.
Exploit Available	Select either Yes or No. Indicates whether there is an exploit for the vulnerability.

Vulnerability condition	Value
Reachable from Source	Select either Yes or No. Indicates whether the vulnerability is reachable from the source code .
Remediation Status	Select one or more of the following values: <ul style="list-style-type: none"> • Duplicate • Ignored • Mitigated • Needs Review • New • Patched • Remediation Complete • Remediation Required

Creating a policy

To create a policy

1. Log in to Black Duck as a user with the Policy Manager role.



2. Click **Manage** and select **Policy Management**.
3. Select **Create Policy Rule** to display the Create Policy Rule dialog box.
4. Complete the following:

- **Name**. Required. Name of this policy.
- **Category**. Optional. Assign one of the following categories for this policy:
 - Uncategorized. This is the default value.
 - Component.
 - License.
 - Operational.
 - Security.

Black Duck provides a filter in the project version BOM (to view policy violations by category) and the Policy Management pages (to view policies by category).

- **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**.

- **Scan Modes.** Select whether this policy rule applies to Full Scans (default value), Rapid Scans, or both.
- **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.

5. For a component condition: select an attribute from the **Component Conditions** list, select an operator, and specify a value.

Click **+ Component Condition** to specify additional component conditions.

6. For a vulnerability condition: select an attribute from the **Vulnerability Conditions** list, select an operator, and specify a value.

Click **+ Vulnerability Condition** to specify additional vulnerability conditions.

7. To remove a condition, click  in the row of the condition you wish to remove.

8. 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 of component or vulnerability conditions, the Policy Violation icon (ⓘ) appears next to component name.

Creating policy rules for approved or barred items

You can create policy rules that enforce your company's policy of approved or barred items. For example, you can create a policy rule to:

- pre-approve a component version in your BOM: any component version that does not match your approval list triggers a policy violation.
- bar a component version from your BOM: a policy violation is automatically triggered for any component version that matches your list of barred components.

Pre-approved policy rule examples

Suppose you want to create a policy rule 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 two sets of filter conditions. The top section, labeled 'Project Distribution Type', has fields for 'equals' and 'External'. Below it, under 'Policy Rules', is another set of fields for 'License Family' (set to 'not equal to' 'Permissive'). Both sections include '+ Add Filter' and '+ Add Rule' buttons.

Suppose you want to create a policy rule 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 two conditions for 'Component'. The first condition uses '>equals' with 'Apache Tomcat' and 'Any Version'. The second condition uses 'not in' with 'Apache Tomcat' and '8.0.1'. Both sections include '+ Add Rule' buttons.

In this example, a policy violation is triggered when the Apache Tomcat version is not 8.0.1.

Suppose you want to create a policy rule 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 two conditions for 'Component'. The first condition uses '>equals' with 'Apache Tomcat' and 'Any Version'. The second condition uses 'not in' with 'Apache Tomcat' and '8.0.3'. Below the second condition, a list shows 'Apache Tomcat 8.0.1' and 'Apache Tomcat 8.0.3' with a delete icon. A '+ Add Rule' button is also present.

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.

Barred policy rule example

Suppose you want to create a 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:

Projects All Filtered

Project Distribution Type	equals	SaaS	
Project Phase	equals	In Development In Planning	

[+ Add Filter](#)

Policy Rules

License Family	equals	AGPL	
----------------	--------	------	--

[+ Add Rule](#)

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  > **Policy Management**.
2. Click  in the row of the policy you want to edit and select **Edit** to display the Edit Policy Rule dialog box.
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 **Manage** > **Policy Management**.

The Policy Management page appears.

2. Click in the row of the policy you want to copy and select **Copy**.

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 **Manage** > **Policy Management**.

The Policy Management page appears.

3. Click in the row of the policy you want to delete and select **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 **Manage** > **Policy Management**.

The Policy Management page appears.

2. Click in the row of the policy rule that you want to enable or disable and select **Edit**.
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**.

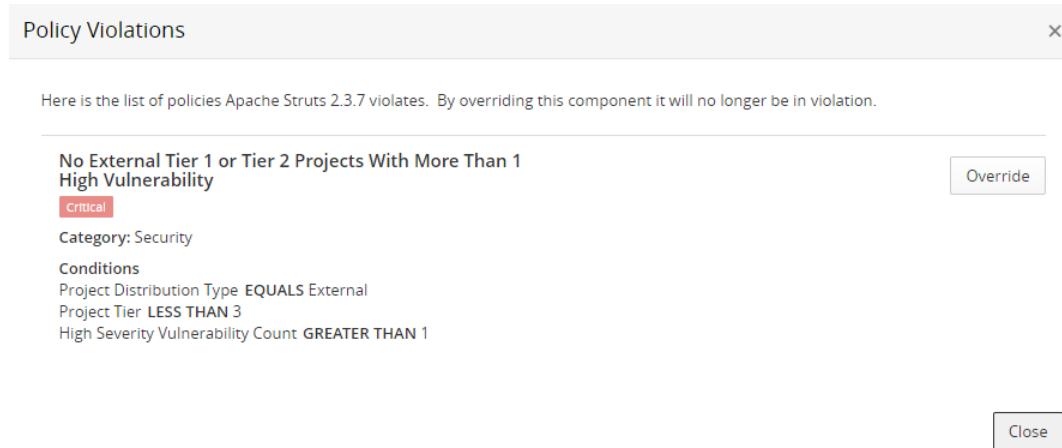
Overriding policy violations

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, click the Policy Violation icon (ⓘ) of the component 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 Policy Violation Override icon (ⓘ) appears next to the component 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 (ⓘ) will still appear.

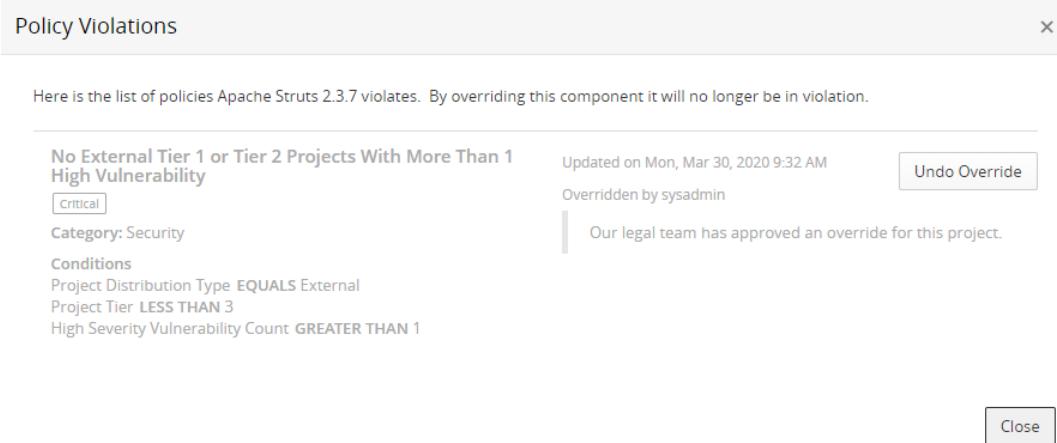
Note: Overrides can be [removed](#).

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, click the Policy Violation Override icon (ⓘ) located next to the component. 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 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 log4j 1.2.15	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.

Copyright Data

Apache log4j 1.2.15

- Copyright 1999-2005 The Apache Software Foundation
- Copyright 2007 The Apache Software Foundation

Licenses

Apache License 2.0

Apache log4j 1.2.15

```
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.
```

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](#) 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. Determine whether you want to enable [deep license data](#).
 6. 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.
 7. [Review copyright statements](#) and/or review [detected copyright statements](#). Optionally edit the Black Duck KnowledgeBase copyright statements and/or create custom copyright statements.
 8. Create the [Notices File report](#). Optionally, 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 source 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:

- **Affero 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.

- **Restrictive Third Party Proprietary**

Licenses in the Restrictive Third Party Proprietary family are for the licenses which cover other company's commercial proprietary code. Typically Restrictive Third Party Proprietary licenses have restrictions on the use of the code and can be risky.

- **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.

- **Internal Proprietary**

Licenses in the Internal Proprietary family are typically for your licenses which are used to cover your company-owned proprietary software. Licenses in this family tend to not place restrictions on your use of the code and are generally not very risky when you use code with licenses in this family.

- **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
Affero General Public License (AGPL)	<ul style="list-style-type: none"> GNU Affero General Public License v3 or later
Reciprocal	<ul style="list-style-type: none"> GNU General Public License (GPL) 2.0 or 3.0 Sun GPL with Classpath Exception v2.0
Weak Reciprocal	<ul style="list-style-type: none"> Code Project Open License 1.02 Common Development and Distribution License (CDDL) 1.0 or 1.1 Eclipse Public License GNU Lesser General Public License (LGPL) 2.1 or 3.0 Microsoft Reciprocal License Mozilla Public License
Permissive	<ul style="list-style-type: none"> Apache 2.0 Artistic License BSD License 2.0 (2-clause Simplified, 3-clause, New, or Revised) Do What The F*ck You Want To Public License ISC License Microsoft Public License MIT License Zlib-Libpng License
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](#).

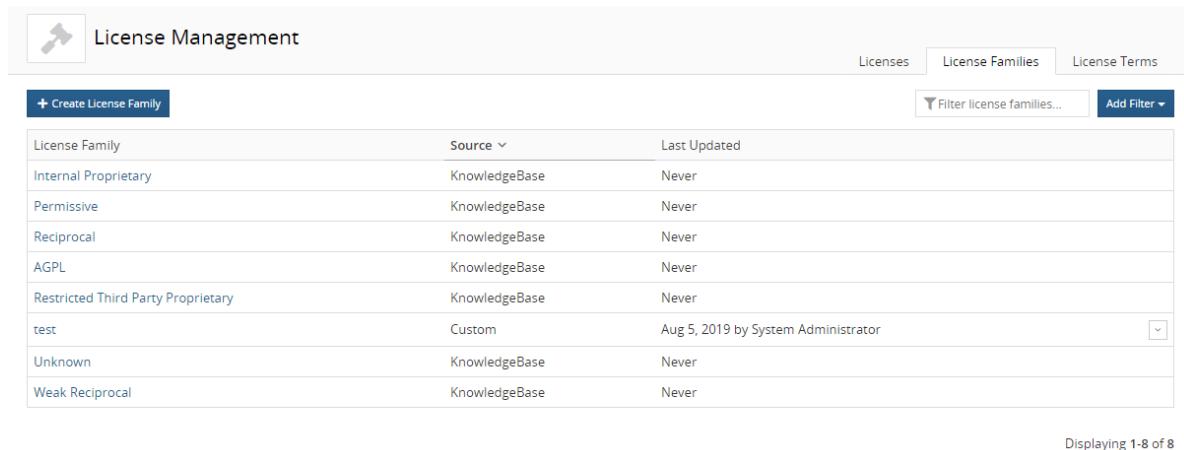
To view the License Management page

1. Log in to Black Duck with the License Manager [role](#).

- 
2. Click **Manage** > **License Management**.

The License Management page appears.

3. Select the **License Families** tab to view a table which lists all license families.



The screenshot shows the 'License Management' interface with the 'License Families' tab selected. The table displays the following data:

License Family	Source	Last Updated
Internal Proprietary	KnowledgeBase	Never
Permissive	KnowledgeBase	Never
Reciprocal	KnowledgeBase	Never
AGPL	KnowledgeBase	Never
Restricted Third Party Proprietary	KnowledgeBase	Never
test	Custom	Aug 5, 2019 by System Administrator
Unknown	KnowledgeBase	Never
Weak Reciprocal	KnowledgeBase	Never

Displaying 1-8 of 8

The table provides the following information:

Column	Description																																																			
License Family	<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; margin-top: 10px;"> <p>Reciprocal</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">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> <tr> <td style="text-align: center;">Component Usage</td> <td style="text-align: center;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></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>Statically Linked</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>Separate Work</td> <td>None</td> <td>None</td> <td>None</td> <td>None</td> </tr> <tr> <td>Merely Aggregated</td> <td>None</td> <td>None</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>Prerequisite</td> <td>Medium</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> </tbody> </table> </td> </tr> </table> <p style="text-align: right;">Close</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	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></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>Statically Linked</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>Separate Work</td> <td>None</td> <td>None</td> <td>None</td> <td>None</td> </tr> <tr> <td>Merely Aggregated</td> <td>None</td> <td>None</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>Prerequisite</td> <td>Medium</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> </tbody> </table>		External	SaaS	Internal	Open Source	Source Code	High	Low	None	None	Statically Linked	High	Low	None	None	Dynamically Linked	High	Low	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
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	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th></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>Statically Linked</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>Separate Work</td> <td>None</td> <td>None</td> <td>None</td> <td>None</td> </tr> <tr> <td>Merely Aggregated</td> <td>None</td> <td>None</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>Prerequisite</td> <td>Medium</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> </tbody> </table>		External	SaaS	Internal	Open Source	Source Code	High	Low	None	None	Statically Linked	High	Low	None	None	Dynamically Linked	High	Low	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						
	External	SaaS	Internal	Open Source																																																
Source Code	High	Low	None	None																																																
Statically Linked	High	Low	None	None																																																
Dynamically Linked	High	Low	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																																																
Source	<p>Source for this license. Possible values are:</p> <ul style="list-style-type: none"> KnowledgeBase. From the Black Duck KnowledgeBase. Custom. Custom license family. 																																																			
Last Updated	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.

Note: If your License Manager created a custom license family labeled "Restrictive Third Party Proprietary" or "Internal Proprietary" before the 2019.10.0 release, the number "(1)" is appended to those custom license family names.

Creating custom license families

Only users with the License Manager role can create [custom license families](#).

To create a custom license family

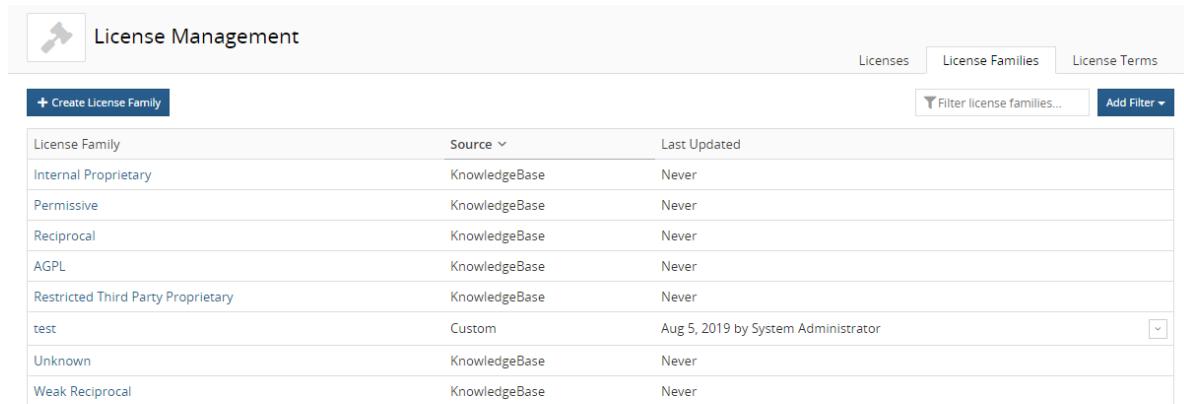
1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **License Management**.

The License Management page appears.

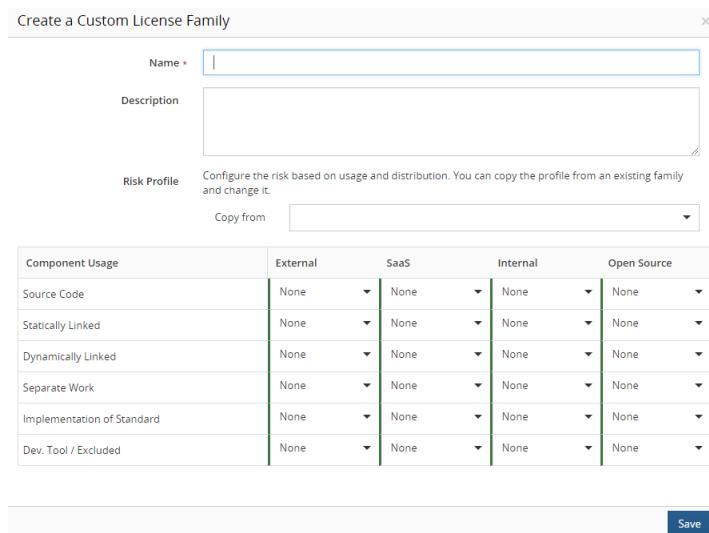
Select the **License Families** tab to display all license families.



The screenshot shows the 'License Management' interface. At the top, there are tabs for 'Licenses', 'License Families' (which is selected), and 'License Terms'. Below the tabs is a search bar labeled 'Filter license families...' and a 'Add Filter' button. A 'Create License Family' button is located at the top left of the main content area. The main content is a table with columns: 'License Family', 'Source', and 'Last Updated'. The table lists several license families: Internal Proprietary, Permissive, Reciprocal, AGPL, Restricted Third Party Proprietary, test, Unknown, and Weak Reciprocal. Each row shows the source as 'KnowledgeBase' and the last update as 'Never' except for the 'test' entry which was updated on Aug 5, 2019 by System Administrator.

Displaying 1-8 of 8

- Click **Create License Family** to open the Create a Custom License Family dialog box.



The dialog box has a title 'Create a Custom License Family'. It contains fields for 'Name' (with a red asterisk) and 'Description'. Below these is a 'Risk Profile' section with a note: 'Configure the risk based on usage and distribution. You can copy the profile from an existing family and change it.' A 'Copy from' dropdown menu is shown. The main part of the dialog is a table titled 'Component Usage' with columns: External, SaaS, Internal, and Open Source. The table rows include: Source Code, Statically Linked, Dynamically Linked, Separate Work, Implementation of Standard, and Dev. Tool / Excluded. Each row has dropdown menus for selecting risk levels for each usage type. At the bottom right is a 'Save' button.

- Enter a name for this license family.
- Optionally, enter a description.
- 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.
- Click **Save**.

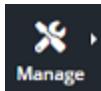
Editing custom license families

Custom license families can be edited by users with the License Manager [role](#).

Note: Adjusting the risk profile for a license family will not change the calculated license risk for components on existing BOMs. Changes will only be reflected on project versions when the BOM is recalculated, such as during rescans or when assigning a scan to a project version.

To edit a custom license family

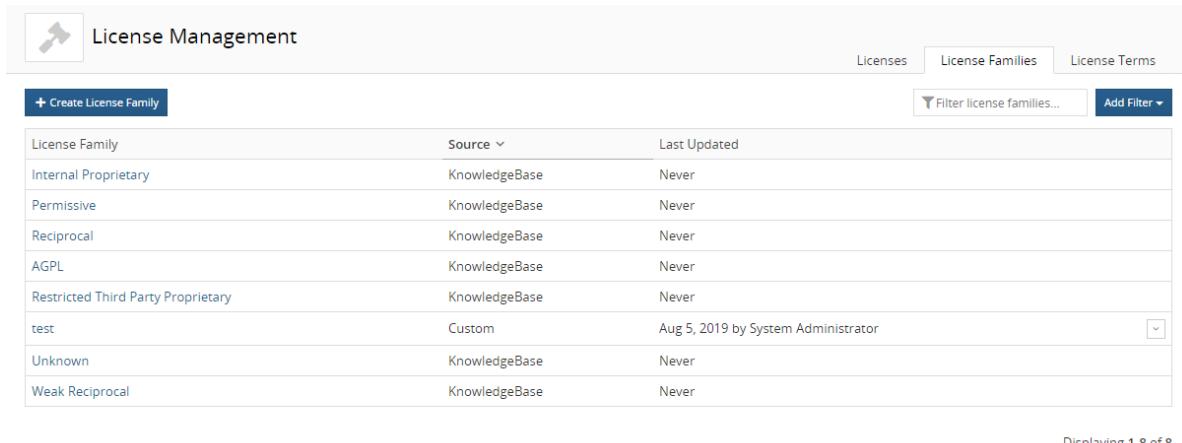
1. Log in to Black Duck with the License Manager role.



2. Click **Manage** > **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 title bar says "License Management". Below it is a toolbar with a wrench icon, a "Create License Family" button, 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 lists various license families: Internal Proprietary, Permissive, Reciprocal, AGPL, Restricted Third Party Proprietary, test, Unknown, and Weak Reciprocal. Each row shows "KnowledgeBase" as the source and "Never" as the last updated date. At the bottom right of the table, it says "Displaying 1-8 of 8".

License Family	Source	Last Updated
Internal Proprietary	KnowledgeBase	Never
Permissive	KnowledgeBase	Never
Reciprocal	KnowledgeBase	Never
AGPL	KnowledgeBase	Never
Restricted Third Party Proprietary	KnowledgeBase	Never
test	Custom	Aug 5, 2019 by System Administrator
Unknown	KnowledgeBase	Never
Weak Reciprocal	KnowledgeBase	Never

4. 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.

Edit Custom License Family

Name *	Reciprocal - Modified for SaaS			
Description	Modified SaaS risk levels to Medium			
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

5. Modify the information shown for this custom license family.
6. 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.

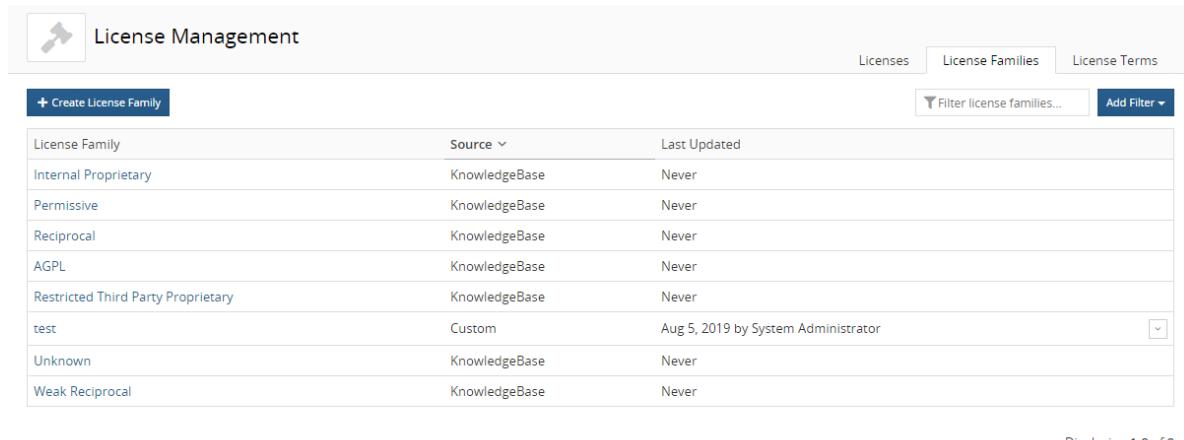
1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **License Management**.

The License Management page appears.

3. Select the **License Family** tab to display all license families.



The screenshot shows the 'License Management' page with a table of license families. The columns are 'License Family', 'Source', and 'Last Updated'. The table includes rows for Internal Proprietary, Permissive, Reciprocal, AGPL, Restricted Third Party Proprietary, test (custom), Unknown, and Weak Reciprocal. The 'test' row is highlighted with a yellow background. A dropdown menu icon is visible next to the 'test' entry.

License Family	Source	Last Updated
Internal Proprietary	KnowledgeBase	Never
Permissive	KnowledgeBase	Never
Reciprocal	KnowledgeBase	Never
AGPL	KnowledgeBase	Never
Restricted Third Party Proprietary	KnowledgeBase	Never
test	Custom	Aug 5, 2019 by System Administrator
Unknown	KnowledgeBase	Never
Weak Reciprocal	KnowledgeBase	Never

Displaying 1-8 of 8

- 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.

- 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 **Manage** > **License Management**.

The License Management page appears.

License Management						
		Components	License Family	Last Updated	User	Source
License						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.

Note: Newly added Black Duck KnowledgeBase licenses or modifications made to Black Duck KnowledgeBase licenses may not be visible here for up to 30 minutes.

Column	Description
License	<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"> • Settings tab to view information for this license, such as the license family and license text. • License Terms tab to view the terms for this license. • Where Used tab to view the component and subproject versions where this license is used.
Components	<p>Number of components or subprojects in all projects that have this license.</p> <p>Note: 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 where this license is used.</p>
License Family	<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>

Column	Description																																													
	<p>Reciprocal</p> <div style="border: 1px solid #ccc; padding: 10px;"> <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.</p> <p>Risk Profile License risk is determined by the license usage of the OSS components in the project version's bill of materials.</p> <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>Statically Linked</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>Separate Work</td> <td>None</td> <td>None</td> <td>None</td> <td>None</td> </tr> <tr> <td>Merely Aggregated</td> <td>None</td> <td>None</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>Prerequisite</td> <td>Medium</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> </tbody> </table> </div> <p style="text-align: right;">Close</p>	Component Usage	External	SaaS	Internal	Open Source	Source Code	High	Low	None	None	Statically Linked	High	Low	None	None	Dynamically Linked	High	Low	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
Component Usage	External	SaaS	Internal	Open Source																																										
Source Code	High	Low	None	None																																										
Statically Linked	High	Low	None	None																																										
Dynamically Linked	High	Low	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																																										
Last Updated	Time, if updated today, or date that the license was last updated.																																													
User	<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>																																													
Source	<p>Source for this license. Possible values are:</p> <ul style="list-style-type: none"> KnowledgeBase. From the Black Duck KnowledgeBase. Modified KnowledgeBase. An edited the Black Duck KnowledgeBase license. Custom. Custom license. 																																													
Status	<p>The review status for the license. Possible values are:</p> <ul style="list-style-type: none"> Unreviewed In Review Reviewed Approved Limited Approval Rejected Deprecated 																																													

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 or a custom license family.
- License Status.

- **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.

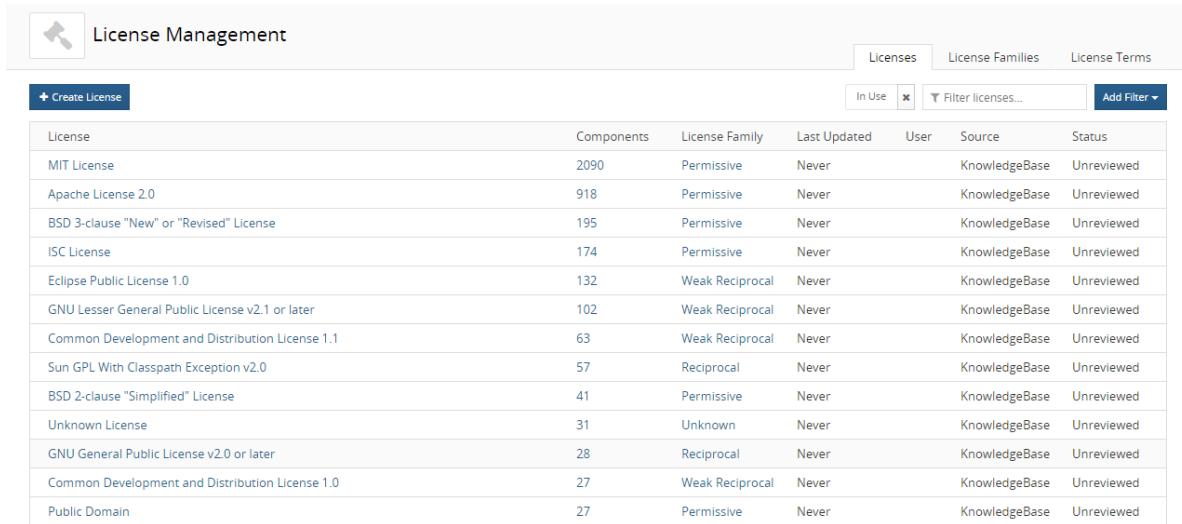
To view license text

1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **License Management**.

The License Management page appears.



A screenshot of the Black Duck License Management page. The page has a header with a wrench icon and the title "License Management". Below the header is a navigation bar with tabs: "Licenses" (selected), "License Families", and "License Terms". There are also buttons for "Create License", "In Use" (with a delete icon), "Filter licenses...", and "Add Filter". The main area is a table with the following columns: License, Components, License Family, Last Updated, User, Source, and Status. The table lists various open source licenses:

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 'Apache License 2.0' settings page. At the top, there is a header with the license name and its family. Below the header, there are several input fields and dropdown menus:

- License Terms**: A dropdown menu showing 'Permissive'.
- Where Used**: A dropdown menu showing 'Unreviewed'.
- Name**: A text input field containing 'Apache License 2.0'.
- License Family**: A dropdown menu showing 'Permissive'.
- Status**: A dropdown menu showing 'Unreviewed'.
- Notes**: An empty text area.
- Expiration Date**: An empty text input field.
- License Text**: A scrollable text area containing the Apache License text, starting with 'Apache License Version 2.0, January 2004'.

On the right side of the form, there are two status indicators:
Created: never
Updated: never

A blue 'Save' button is located at the bottom right of the form.

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.

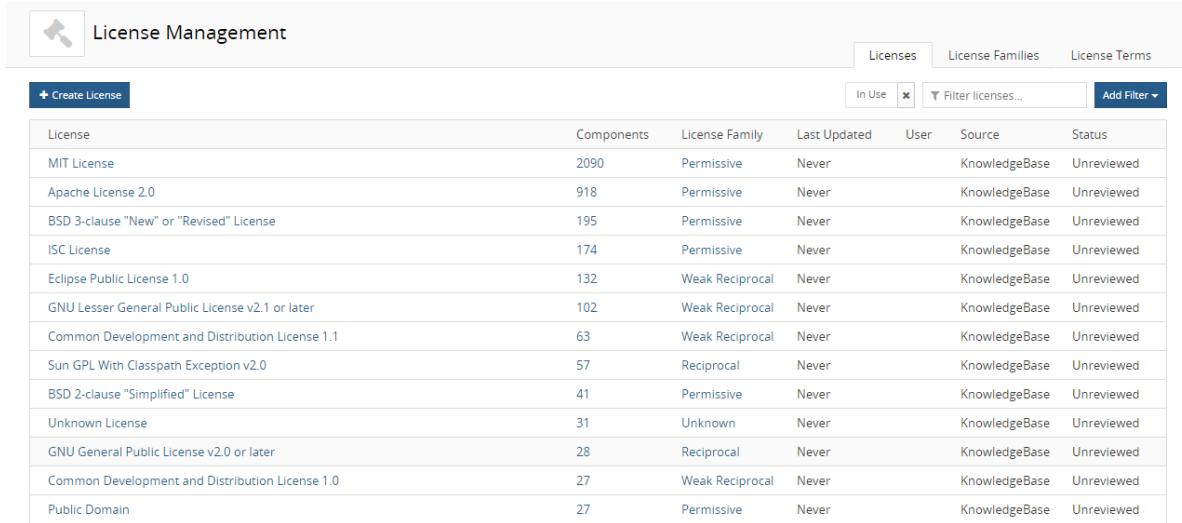
Note: The information shown here lists the components and subprojects that use a license. It does not include licenses assigned to project versions.

To view where a license is used

1. Log in to Black Duck with the License Manager role.

2. Click **Manage** > **License Management**.

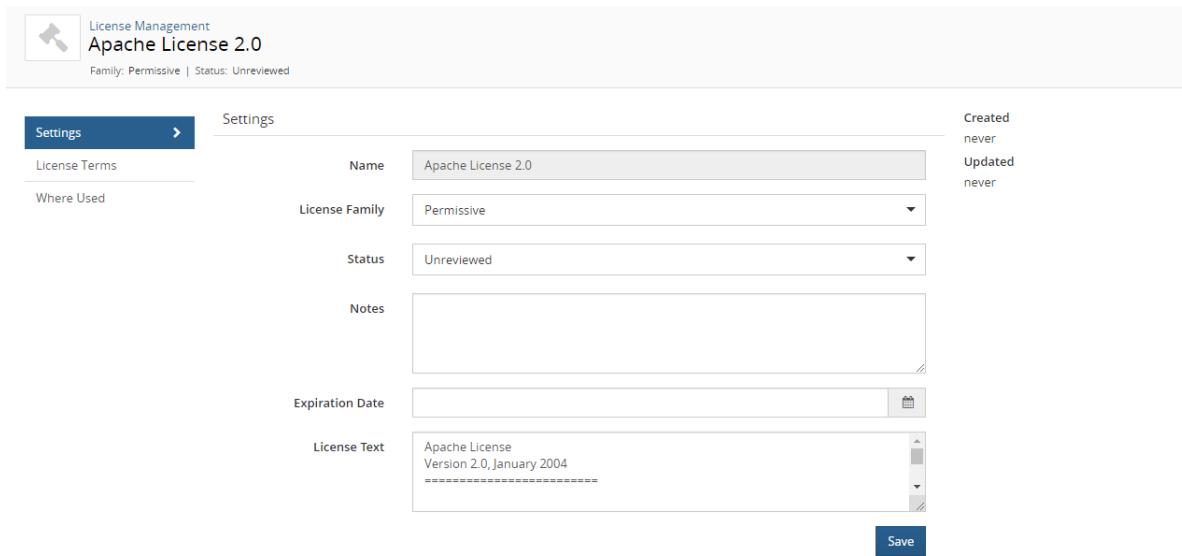
The License Management page appears.



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	

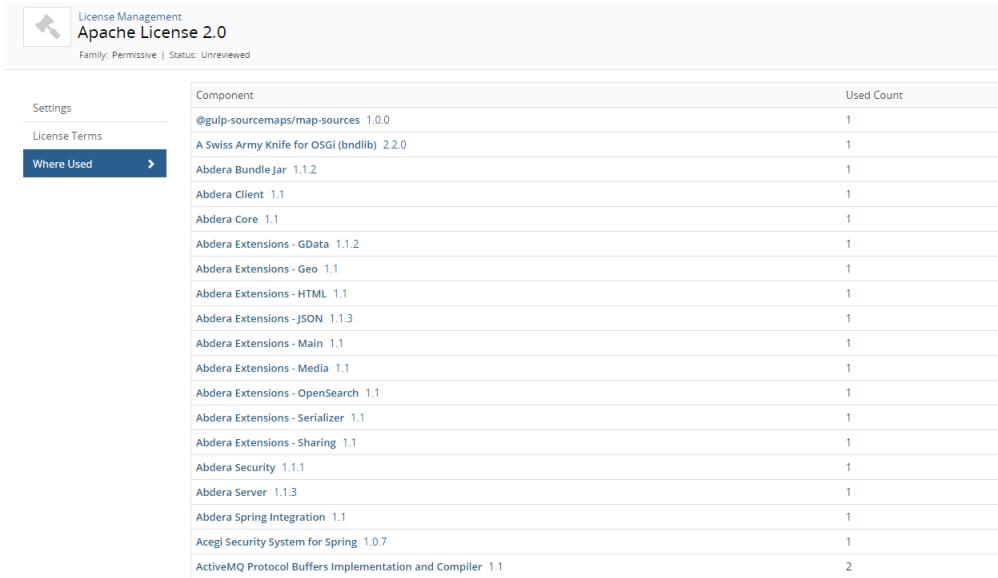
3. Select the license name to display the *License Name Settings* tab.



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 indicates the license was created and updated never.

Settings	
License Terms	Name: Apache License 2.0
Where Used	Created never Updated never
	License Family: Permissive
	Status: Unreviewed
Notes	(empty)
Expiration Date	(empty)
License Text	Apache License Version 2.0, January 2004 =====
Save	

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
Acegi 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, separate work, and unspecified).

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:

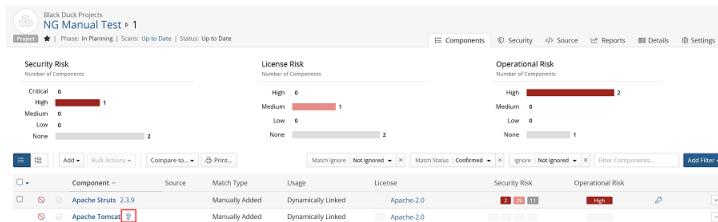
- "AND" licenses: license risk is determined by the license with the highest risk.
- "OR" licenses: license risk is determined by the license with the lowest risk.

Risk calculations assume that your project is being distributed under a proprietary license.

Estimated licenses

A default license may be assigned to components with an unknown version found during a scan. This is an estimated license based on greatest number of times it shows up across the top 1,000 versions of the component.

When viewing the BOM for a project, components with unknown versions will have a question mark next to the component name.



Clicking the license in the License column will open the [Modify License](#) window which will display the following warning banner:

Note: Black Duck wasn't able to identify the component version, therefore this is an estimated license. It was defined based on popularity across multiple versions. For a more accurate result, manually specify a version for this component.

[Edit Component](#) [Learn More](#)

It is recommended that you review these components and manually specify a version for more accurate results.

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.

Note: If your License Manager created a custom license family labeled "Restrictive Third Party Proprietary" or "Internal Proprietary" before the 2019.10.0 release, the number "(1)" is appended to those custom license family names.

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	High	None	None
Reciprocal	High	Low	None	None
Weak Reciprocal	High	Low	None	None
Permissive	None	None	None	None
Restrictive Third Party Proprietary	Medium	Medium	Medium	High
Internal Proprietary	None	None	None	Medium
Unknown	High	High	High	High

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	High	None	None
Reciprocal	High	Low	None	None
Weak Reciprocal	Medium	Low	None	None
Permissive	None	None	None	None
Restrictive Third Party Proprietary	Medium	Medium	Medium	High
Internal Proprietary	None	None	None	Medium
Unknown	High	High	High	High

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	High	None	None
Reciprocal	High	Low	None	None

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Weak Reciprocal	High	Low	None	None
Permissive	None	None	None	None
Restrictive Third Party Proprietary	Medium	Medium	Medium	High
Internal Proprietary	None	None	None	Medium
Unknown	High	High	High	High

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
Restrictive Third Party Proprietary	Low	Low	Low	Low
Internal Proprietary	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
Permissive	None	None	None	None

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Restrictive Third Party Proprietary	Low	Low	Low	Low
Internal Proprietary	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
Restrictive Third Party Proprietary	Medium	Medium	Medium	High
Internal Proprietary	None	None	None	Medium
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
Restrictive Third Party Proprietary	Medium	Medium	Medium	High

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Internal Proprietary	None	None	None	Medium
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
Restrictive Third Party Proprietary	Medium	Medium	Medium	High
Internal Proprietary	None	None	None	Medium
Unknown	Medium	Medium	Low	Low

Unspecified

The following table lists the license risk when the component's usage is **Unspecified**.

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Affero General Public License (AGPL)	High	High	None	None
Reciprocal	High	Low	None	None
Weak Reciprocal	High	Low	None	None
Permissive	None	None	None	None
Restrictive Third Party Proprietary	Medium	Medium	Medium	High
Internal Proprietary	None	None	None	Medium
Unknown	High	High	High	High

License risk by license family

Affero General Public License (AGPL)

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Source Code	High	High	None	None
Statically Linked	High	High	None	None
Dynamically Linked	High	High	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
Unspecified	High	High	None	None

Reciprocal

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Source Code	High	Low	None	None
Statically Linked	High	Low	None	None
Dynamically Linked	High	Low	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
Unspecified	High	Low	None	None

Weak Reciprocal

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Source Code	High	Low	None	None
Statically Linked	High	Low	None	None
Dynamically Linked	Medium	Low	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
Unspecified	High	Low	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
Unspecified	None	None	None	None

Restrictive Third Party Proprietary

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Source Code	Medium	Medium	Medium	High
Statically Linked	Medium	Medium	Medium	High
Dynamically Linked	Medium	Medium	Medium	High
Separate Work	Medium	Medium	Medium	High
Merely Aggregated	Medium	Medium	Medium	High
Implementation of Standard	Low	Low	Low	Low
Prerequisite	Medium	Medium	Medium	High
Dev. Tool/Excluded	Low	Low	Low	Low
Unspecified	Medium	Medium	Medium	High

Internal Proprietary

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Source Code	None	None	None	Medium
Statically Linked	None	None	None	Medium
Dynamically Linked	None	None	None	Medium
Separate Work	None	None	None	Medium
Merely Aggregated	None	None	None	Medium
Implementation of Standard	None	None	None	None
Prerequisite	None	None	None	Medium
Dev. Tool/Excluded	None	None	None	None
Unspecified	None	None	None	Medium

Unknown

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
Source Code	High	High	High	High
Statically Linked	High	High	High	High
Dynamically Linked	High	High	High	High
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
Unspecified	High	High	High	High

Managing deep license data

Black Duck displays declared licenses for the components in your BOM. However, deep licenses (also known as sub-licenses or embedded licenses) may also exist in your open source components. Managing this deep license data reduces the risk of license infringement and makes it easier to understand and report on deep licenses and their risks in the open source being used.

Deep license data is not enabled by default; you must enable including deep license data to your BOM components. Once enabled, any deep licenses, as determined by the Black Duck KnowledgeBase, are automatically active.

Note: Depending upon the number of components and number of deep licenses, enabling the viewing of deep license data can impact the BOM calculation scan time. Adding deep license data to your BOM can affect your license risk and can trigger policy violations.

To manage your deep license data:

1. Enable deep level license data. As this feature is enabled at the project level, deep license data will be enabled and active for all project versions.

In your project version BOM, the deep license data icon () identifies the components with deep level licenses.

2. View the deep license data. You can:

- Review the evidence as determined by the Black Duck KnowledgeBase.

Evidence consists of the list of files and file content which you can view to confirm the inclusion of deep license data.

- Activate or deactivate the license. By default, deep license data is activated for all origins. If there are multiple origins, deep license data is activated for all origins.
- Add licenses.
- Read the license text.

Enabling deep license data

Deep license data is enabled at the project level.

To enable deep license data

1. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the **Settings** tab.

Project Details

- Members
- Groups
- Activity

Settings

Project Name * Sample Project 1

Description

Owner start typing to select owner...

Tier select tier...

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

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

Custom Scan Signature Custom Scan Signature can identify third-party and proprietary software used in your code. There may be performance issues seen when using this feature.
 Enable Custom Scan Signature

Depth, as measured in the number of levels in the directory structure, from root, to perform custom signature scanning for this project. The initial value is the default value defined by the System Administrator.

Deep License Data Enabling this checkbox will apply deep license data to your components and allow visibility to embedded licenses which may exist in your components beyond declared licenses. Please note, this can affect the license risk and policy violation for components. It can also impact the bill of materials calculation time depending upon the number of components and amount of deep licenses.
 Apply Deep License Data to bill of materials

License Conflicts Enabling this checkbox will apply license conflicts data to your components
 Apply License Conflicts Data to bill of materials

Additional Fields

Data Sensitivity Does the application contain or access important confidential information or personally identifiable data.

Application ID A field that can be used to store an external mapping id for the project to an external system, like an asset management system or application catalog.

Clone Project Clone selected versions of this project as well as existing project settings, members, groups, additional fields, component edits and application ID.

Delete Project Once you delete a project, you cannot restore it and you lose all information and versions related to the project. Scans will be unmapped from all versions and not deleted.

3. Enable the **Apply Deep License Data to bill of materials** option in the **Deep License Data** section.

Clear the option to disable this feature.

4. Click **Save**.

Reviewing deep license data

1. Open the project version BOM to view the components which have deep license data.

Component	Source	Match Type	Usage	License	Security Risk	Operational Risk
abbrev 1.1.1	D 1 Match	Transitive Dependency	Dynamically Linked	ISC		
amdefine 1.0.1	D 1 Match	Transitive Dependency	Dynamically Linked	MIT or 1 more...	High	
ansi-regex 2.1.1	D 1 Match	Transitive Dependency	Dynamically Linked	MIT	High	
anymatch 1.3.2	D 1 Match	Transitive Dependency	Dynamically Linked	ISC	High	
Apache Commons FileUpload 1.1	D 1 Match	Exact Directory	Dynamically Linked	Apache-2.0	1 1 3	High
Apache Commons Logging 1.0.4	D 1 Match	Exact Directory	Dynamically Linked	Apache-2.0		High
Apache Lucene 1.4.3	D 1 Match	Exact Directory	Dynamically Linked	Apache-2.0		High
Apache Tomcat 6.0.26	Manually Added	Dynamically Linked	M Apache-2.0	8 41 4		High

2. Components with have deep license data. Click to open the Component Name Version Deep License page.

License	Active	License Family	Status	Last Updated
Apache License 2.0	✓	Permissive	Approved	Apr 7, 2021 by System User
Bzip2 License	✓	Permissive	Unreviewed	Apr 7, 2021 by System User
Common Public License 1.0	✓	Weak Reciprocal	Unreviewed	Apr 7, 2021 by System User
Eclipse Public License 1.0	✓	Weak Reciprocal	Unreviewed	Apr 7, 2021 by System User
libpng License	✓	Permissive	Unreviewed	Apr 7, 2021 by System User
zlib License	✓	Permissive	Unreviewed	Apr 7, 2021 by System User

This page displays the following information:

Column	Description
License	<p>License name.</p> <p>Select the name to display the <i>License Name</i> page which displays the license text.</p> <p>Click > to view the origins for this license.</p>
Active	<p>Indicates whether this license is active.</p> <p>Active licenses are included in the calculation of license risk and policy violations.</p>

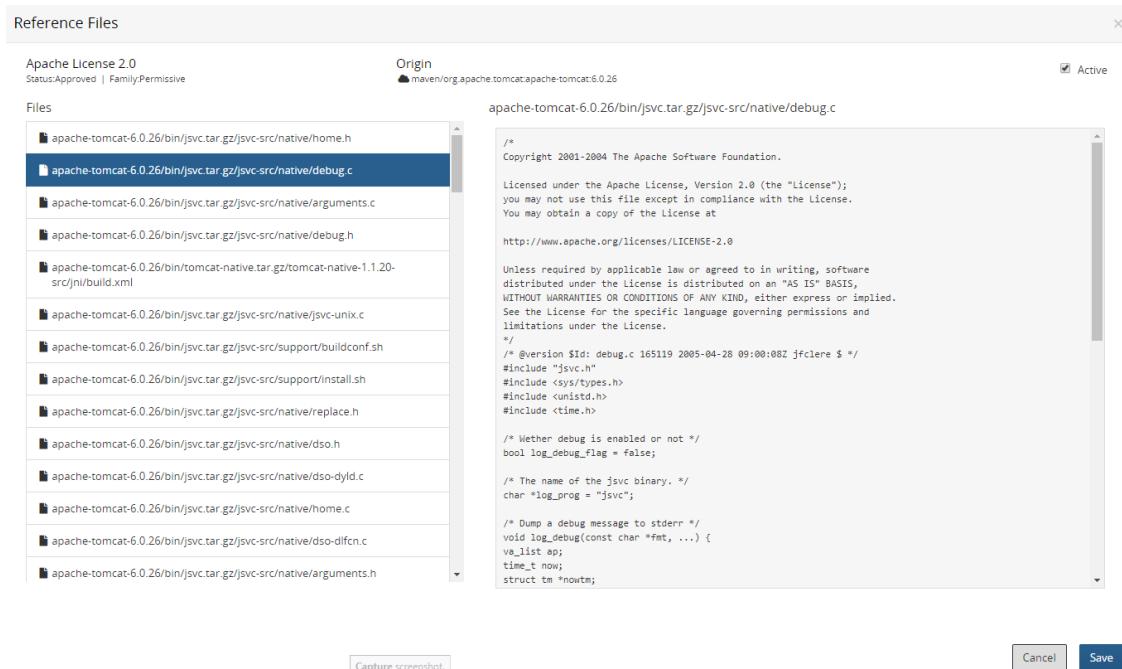
Column	Description
License Family	The license family for this license.
Last Updated	Date and user who last updated the information on this page.
Status	<p>The review status for the license. Possible values are:</p> <ul style="list-style-type: none"> • Unreviewed • In Review • Reviewed • Approved • Limited Approval • Rejected • Deprecated

3. From this page:

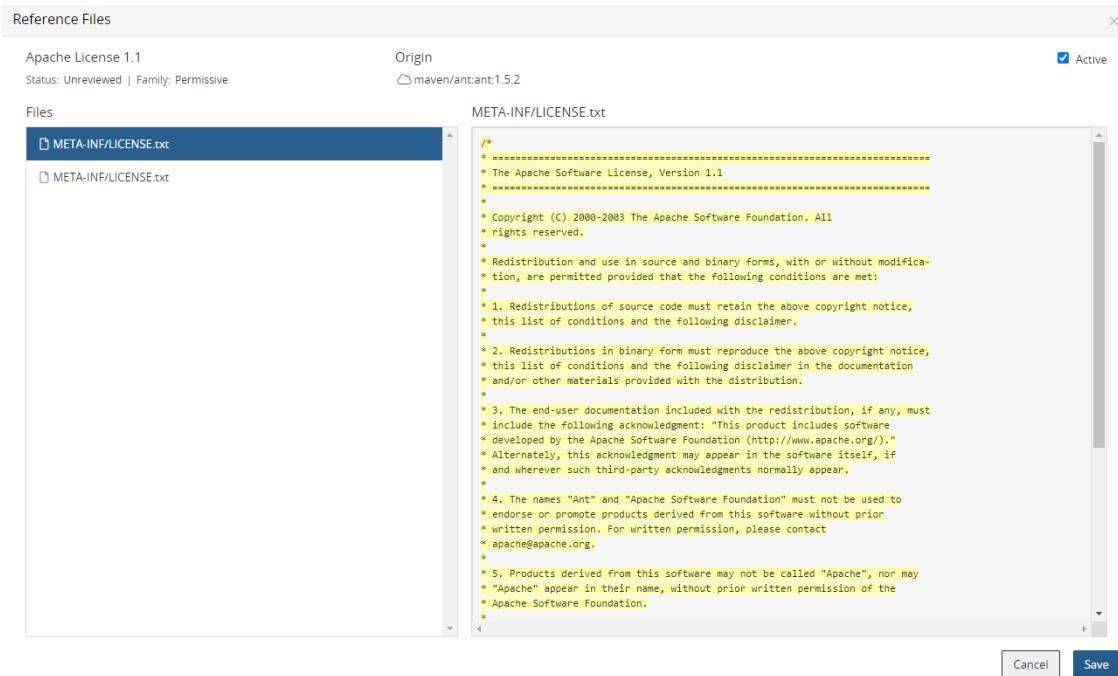
- View the evidence for the inclusion of this deep license.

The Black Duck KnowledgeBase determines deep license data at the origin level. Therefore, click > to display the origins for this license.

Select an origin to open the Reference Files dialog box which displays the files and corresponding evidence for inclusion of this license.



The **Files** section lists the files found containing deep license data. Select a file to view the contents of that file. Deep license data is highlighted.

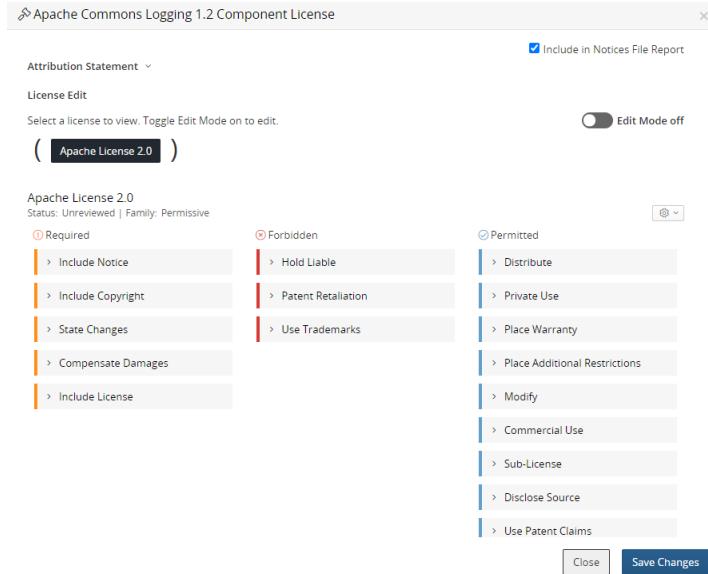


If the file cannot be determined, the file name and path display "Unknown."

- Activate or deactivate the deep license. By default, all deep licenses are active.

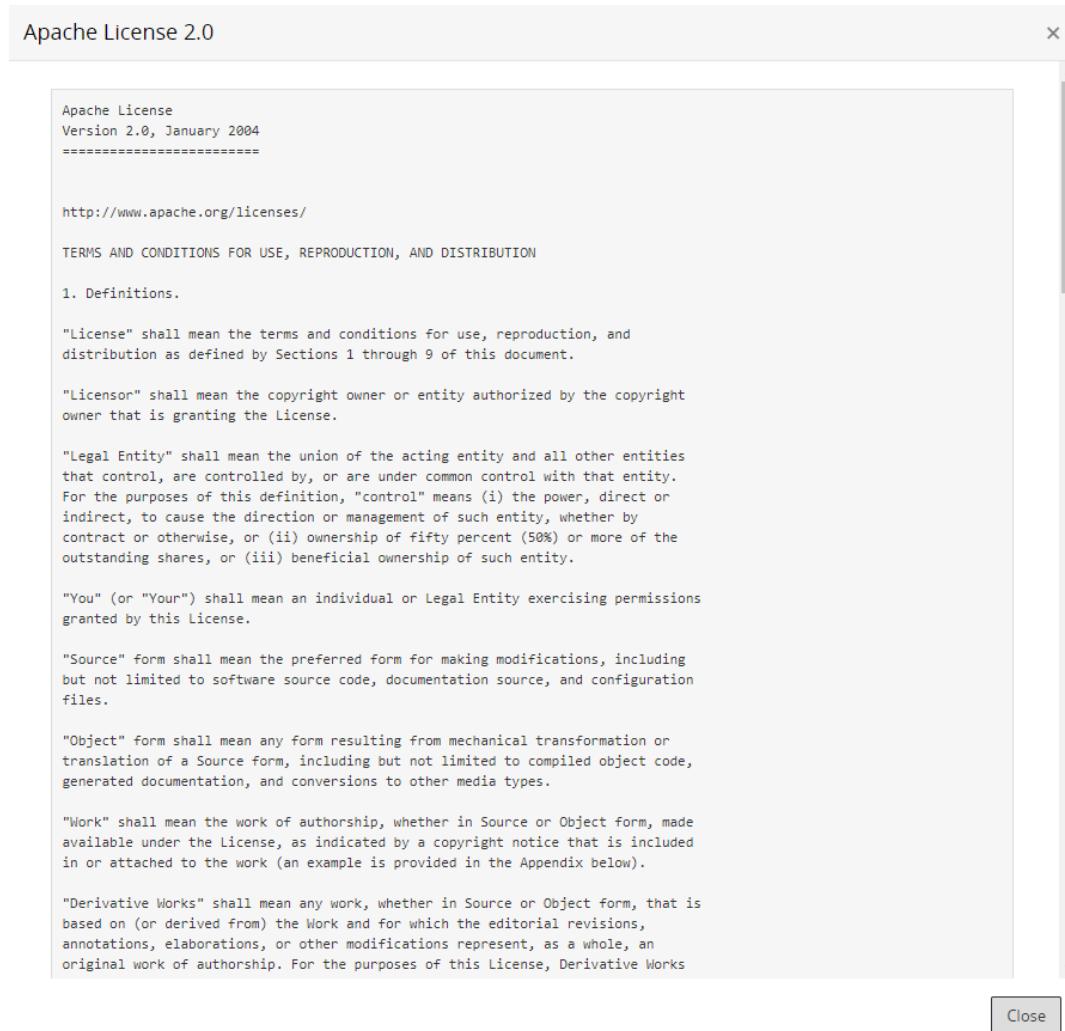
You can activate or deactivate a deep license by:

- Selecting a license in the *Component Name Version* Deep License page and clicking **Activate or Deactivate**.
- Selecting or clearing the **Active** option located in the upper right-corner of the Reference Files dialog box.
- Add a license or remove a manually added license.
 - To add a license, click **Add License**, select the license, and click . The new license appears in the table.
 - To remove a license, click in the row of the manually added license you want to delete and select **Remove** in the confirmation dialog box.
- View license text:
 - View the declared license text and obligation information. Select the license name in the header to open the *Component Name Version* Component License dialog box.



Note that you can [modify the declared license](#).

- o View deep license text by selecting the license name from the table.



Detecting embedded licenses

Black Duck can detect instances of embedded open source licenses not declared by the Black Duck KnowledgeBase for a component.

By enabling detection of deep license data when scanning code, users focused on license compliance can view the licenses that were detected in their open source to ensure there are no problematic licenses and that all licenses are accounted for in their BOM.

With this feature, Black Duck performs a search for license string text and displays the licenses found in the **Source** tab.

By displaying this information in the **Source** tab, you can easily find the files and directories that interest you and determine if embedded licenses are located there.

The screenshot shows the Black Duck Software interface for a project named "Sample Project > 1.0". The left sidebar displays a file tree with directories like "hh 2/6 TT scan", "blowfish.c", "external", "lib", "lib - Copy", "licenses", "new", "samplefile1.h", "samplefile2.c", "src" (which is selected), "src_jo", "src_ourfaces", "src_psl", and "tools". The main content area has tabs for "Files" and "Discoveries" (which is selected). A search bar at the top right contains the text "src/# » src". Below the search bar are sections for "License Searches" and "Licenses". The "Licenses" section shows a list of detected licenses with their counts: Apache License Version 2.0 (224 hits, 219 files), Eclipse Public License Version 1.0 (3 hits), and GNU General Public License Version 2 (1 hits). The "License References" section shows a list of licenses with their counts: GNU General Public License (8 hits, 3 files), Apache License (5 hits, 2 files), and BSD (5 hits, 1 file). There is also a checked checkbox for "All Subfolders".

Black Duck groups the detected licenses into one of two categories:

- **Licenses**. An exact match to a license and version.
- **License References**. A "fuzzy" match to a license; license version information was not found.

For each license statement found, Black Duck displays the number of:

- "Hits". The number of instances that license text was found in all files.
- Files where these "hits" were found.

In the example shown above, there were five instances of Apache License text found in two files, while there were 224 instances of Apache License Version 2.0 found in 219 files.

Black Duck also lists the total number of files affected for each category. Note that this value may not equal the total number of files shown for each license in that category as a file can have multiple different licenses, as shown above for the **Licenses** category.

Optionally, to help you review this information, upload your source files so that BOM reviewers can view discovered license text from within the **Source** tab. When source files are uploaded, Black Duck provides a list of embedded licenses and displays the highlighted license text in the file. This can help BOM reviewers evaluate the license text.

The screenshot shows the 'Discoveries' window with the title 'Discoveries'. A message at the top says 'We found these discoveries in this file: 1 License, 2 License References, 2 Copyrights'. Below this, there are three main categories: 'Licenses', 'License References', and 'Copyrights', each with a 'Hits' column. The 'Licenses' section has one hit for 'GNU General Public License v2.0 or later'. The 'License References' section has two hits for 'GNU General Public License' and 'GNU Lesser General Public License'. The 'Copyrights' section has one hit for 'Copyright (C) yyyy name of author' and another for 'Copyright (C) 1989, 1991 Free Software Foundation, Inc.'. To the right of these lists is a large text area showing the full text of the 'GNU GENERAL PUBLIC LICENSE' version 2, starting with the Preamble and continuing through the terms and conditions for copying, distribution, and modification.

If you do not upload the source files, the Black Duck UI only displays the location of the discovered license text in the file, by line number:

The screenshot shows the 'Discoveries' window with the title 'Discoveries'. A message at the top says 'We found these discoveries in this file: 1 License, 0 License Reference, 0 Copyright'. Below this, there is one entry in the 'Licenses' section: 'Apache License 1.1' with 1 hit. To the right of this list is a large text area with the heading 'No File to Display' and the sub-instruction 'You need to upload your files in order to display them'. Below this, it says 'We found hits in these lines:' followed by 'Hit 1: Line 5 to Line 6'.

To include your source files, after your administrator has enabled source uploads, as described in the installation guide, include the upload source parameter when scanning.

Note: Regardless whether you upload your source files or not, embedded license detection cannot be completed offline as it requires communication with the Black Duck server.

Supported file extensions/ file names

Embedded license search occurs in file extensions such as `.bat` or `.js` and for these file names, or file names that include the following text, regardless of case:

- `bds!`
- `copying`
- `copyright`
- `control`

- dad
- gpl
- install
- legal
- lGPL
- license
- licence
- licenses
- licences
- notice
- readme

License detection process

The process to view embedded licenses is:

1. Enable detecting of deep license data when scanning and optionally, enable uploading source files for viewing embedded licenses within the file.
2. Review embedded licenses.

Enable detecting of deep license data

All scanning methods have an option to enable license string search:

- Signature Scanner command line
- Synopsys Detect (Desktop) Synopsys Detect
- Synopsys Detect

Using the Signature Scanner command line

Use the **--license-search** parameter to enable embedded licenses.

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

Using Synopsys Detect (Desktop) or Synopsys Detect

Use the **--detect.blackduck.signature.scanner.license.search** property to enable deep license data detection. This property is available in Synopsys Detect version 6.2 and later.

Reviewing embedded licenses

Black Duck displays the location of these licenses in your code tree.

To review embedded licenses

1. After enabling license search, select the **Source** tab from your project version BOM page.
2. Select a folder in the code tree that you want to determine if there are embedded licenses.

Optionally, select **All Subfolders** to view information for all subfolders.

The table displays information in the table for the selected location. By default the **Files** option is selected.

Name	Component	Match Type	License	Usage	Discovery Types
travis.yml	Apache Commons FileUpload commons-fileupload-1.4	Exact Directory	Apache-2.0	Dynamically Linked	
Base64Decoder.html	Apache Commons FileUpload commons-fileupload-1.4	Exact Directory	Apache-2.0	Dynamically Linked	License
Base64Decoder.java	Apache Commons FileUpload commons-fileupload-1.4	Exact Directory	Apache-2.0	Dynamically Linked	License </>
Base64DecoderTestCase.html	Apache Commons FileUpload commons-fileupload-1.4	Exact Directory	Apache-2.0	Dynamically Linked	License
Base64DecoderTestCase.java	Apache Commons FileUpload commons-fileupload-1.4	Exact Directory	Apache-2.0	Dynamically Linked	License </>
CONTRIBUTING.md	Apache Commons FileUpload commons-fileupload-1.4	Exact Directory	Apache-2.0	Dynamically Linked	
Closeable.html	Apache Commons FileUpload commons-fileupload-1.4	Exact Directory	Apache-2.0	Dynamically Linked	
Closeable.html	Apache Commons FileUpload commons-fileupload-1.4	Exact Directory	Apache-2.0	Dynamically Linked	
Closeable.html	Apache Commons FileUpload commons-fileupload-1.4	Exact Directory	Apache-2.0	Dynamically Linked	License
Closeable.java	Apache Commons FileUpload commons-fileupload-1.4	Exact Directory	Apache-2.0	Dynamically Linked	</>

3. Select **Discoveries** to view the list of embedded licenses for this location.

Licenses	Count
Apache License Version 2.0 224 hits	219
Eclipse Public License Version 1.0 3 hits	1
GNU General Public License Version 2 1 hits	1

License References	Count
GNU General Public License 8 hits	3
Apache License 5 hits	2
BSD 5 hits	1

4. Select a license to view the **Source** tab filtered to display the files that contain the selected embedded

license text.

The screenshot shows the Black Duck Project interface for a 'Sample Project'. On the left, there's a file tree with various directories like 'external', 'lib', 'licenses', 'new', and 'src'. Under 'src', there are sub-directories for 'commons-fileupload-1.4-src...' and 'commons-fileupload-1....'. The 'commons-fileupload-1....' directory contains files such as '.travis.yml', 'CONTRIBUTING.md', 'LICENSE.txt', 'NOTICE.txt', 'pom.xml', 'README.md', and 'RELEASE-NOTES.txt'. On the right, there's a search bar with 'src/#> src' and a 'Discoveries' tab selected. A table lists a single discovery result: 'Name' is 'customsorttypes.js', 'Component' is 'Apache Commons FileUpload commons-fileupload-1.4', 'Match Type' is 'Exact Directory', 'License' is 'Apache-2.0', and 'Usage' is 'Dynamically Linked'. A note at the bottom says 'Displaying 1-1 of 1'.

Optionally, select a file name to view the location of the file in the code tree. If you uploaded your source files, the file contents appears on the page.

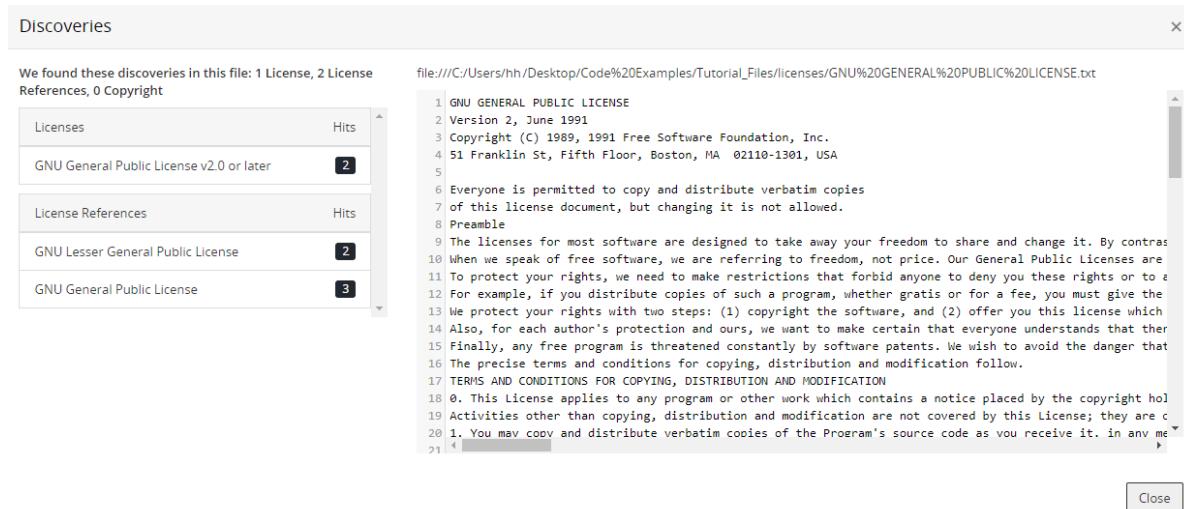
This screenshot shows the same project interface after selecting the 'customsorttypes.js' file from the tree. The right panel now displays the file's content. The code is as follows:

```

1 /*
2 * Cobertura - http://cobertura.sourceforge.net/
3 *
4 * Copyright (C) 2005 Mark Doliner
5 * Copyright (C) 2005 Olivier Parent
6 *
7 * Cobertura is free software; you can redistribute it and/or modify
8 * it under the terms of the GNU General Public License as published
9 * by the Free Software Foundation; either version 2 of the License,
10 * or (at your option) any later version.
11 *
12 * Cobertura is distributed in the hope that it will be useful, but
13 * WITHOUT ANY WARRANTY; without even the implied warranty of
14 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU
15 * General Public License for more details.
16 *
17 * You should have received a copy of the GNU General Public License
18 * along with Cobertura; if not, write to the Free Software
19 * Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307
20 * USA
21 */
22

```

- Select a type of discovery (**License** or **License Reference**) from the **Discovery Type** column to open the Discoveries dialog box.

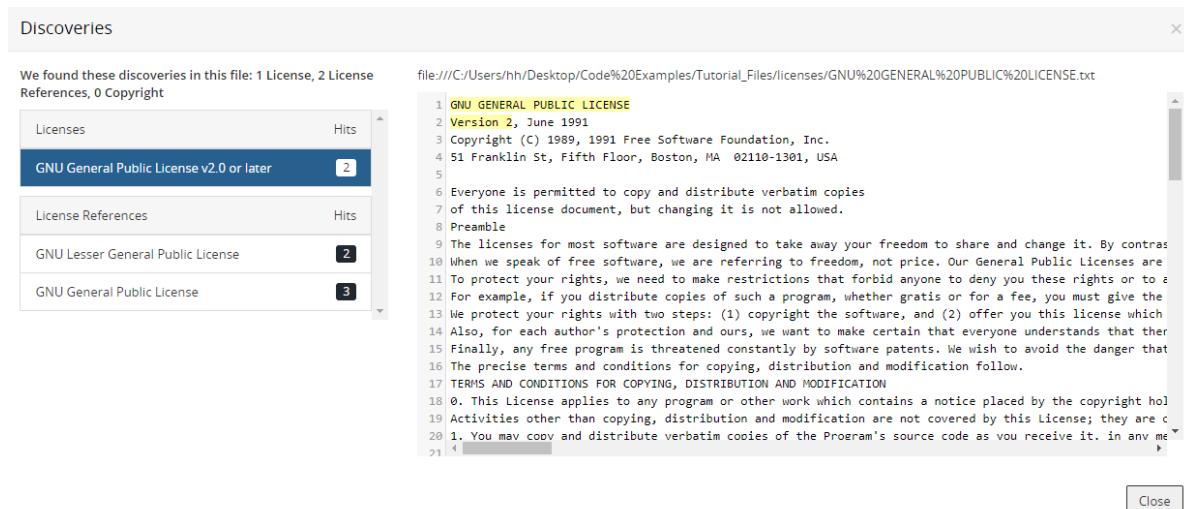


The Discoveries dialog box shows all licenses and license references found for the selected file.

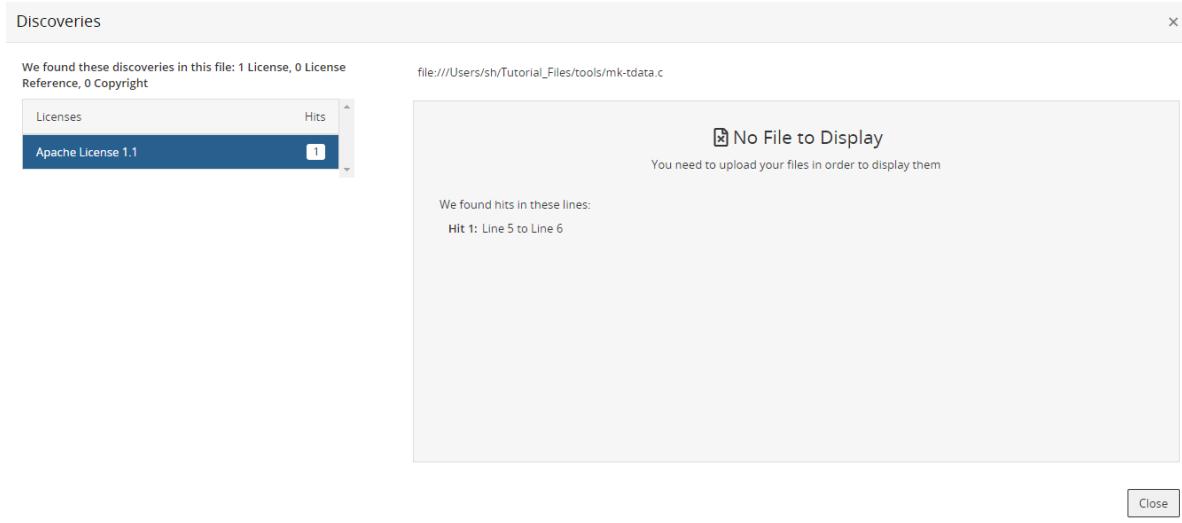
The information that appears here depends on whether you uploaded source files.

In the example shown above, source files were uploaded in the scan.

6. Select a license to view the highlighted license text indicating the embedded license text found.



If you did not upload source files, the Discoveries dialog box displays the location of the discovered license text in the file, by line number:



Modifying licenses for a component

So that you can successfully manage license risk, you may need to edit the license(s) for a component version so that it is different from the component version's declared license identified in the Black Duck KB or the license originally selected for the version of the custom component.

You can modify a single license or include multi-license scenarios, such as "License A AND License B" or "License A OR License B". This lets you accurately represent the licenses in Black Duck for the components in your projects.

Note the following:

- Edits made to a license in the BOM are *local* edits. These edits apply to this version of the component for this BOM only.
- Edits made to a license from the [Black Duck KnowledgeBase component version page](#) or the [custom component version page](#) are *global* edits. These edits apply to all instances of this version of the component. However, edits made at the BOM level will override these edits.

To modify licenses

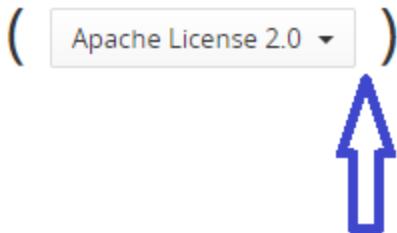
1. To modify a single license:
 - a. Click ▾ located next to the license name and select the license from the list of suggestions.
 - b. Do one of the following:
 - Click ✅ to confirm this selection.
 - Click 🗑 to delete this license and the operand.
2. To add a license to the existing license(s):
 - a. Click **Add License**. Black Duck adds the following at the root level:

AND ▾ △ Select a license... ▾

For example, when added to a single license, the following appears:

((Apache License 2.0 ▾) AND ▾ △ Select a license... ▾)

- To add a license at the original license level, select the license by placing the cursor within the parentheses of that license.



Click **Add License**. The license is added at the level of the original license:

((Apache License 2.0 ▾ AND ▾ △ Select a license... ▾))

For example, when added to an existing multi-license scenario, the following appears:

((Common Development and Distribution License 1.1 ▾ OR ▾ Sun GPL With Classpath Exception v2.0 ▾)
AND ▾ △ Select a license... ▾)

- To add a license at the same level as the existing multi-licenses, select the license by placing the cursor within the parentheses of the existing group.

((Sun JavaMail 1.4 License ▾ AND ▾ Common Development and Distribution License 1.0 ▾))

Click **Add License**. The license is added at the same level as the existing licenses:

((Sun JavaMail 1.4 License ▾ AND ▾ Common Development and Distribution License 1.0 ▾ AND ▾ △ Select a license... ▾))

- Optionally, click ▾ next to the operand to change it. Possible values are AND or OR.
- Click ▾ located next to the license name and select the license from the list of suggestions.
 - Click to confirm this selection.
 - Click to delete this license and the operand.
- Repeat as necessary.

3. To add a multi-license scenario (for example, License A AND (License B OR License C)):

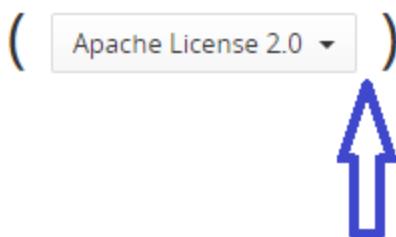
- Click **Add Group**. Black Duck adds the following at the root level:



When added to a single license, the following appears:

((Apache License 2.0) AND (Select a license...))

- To add a group at the original license level, select the license by placing the cursor within the parentheses.



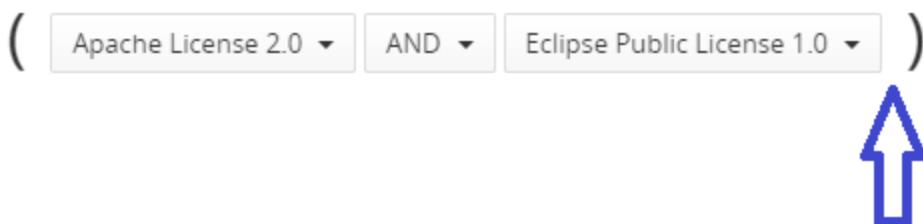
Click **Add Group**. The following appears:

((Apache License 2.0) AND (Select a license...)))

When added to an existing multi-license scenario, the following appears:

((Apache License 2.0) AND (Eclipse Public License 1.0)) AND (Select a license...))

- To add a group at the original license level, select the license by placing the cursor within the parentheses:



Click **Add group**. The following appears:

((Apache License 2.0) AND (Eclipse Public License 1.0) AND (Select a license...)))

- b. Optionally, add additional licenses as described in step 6a.
 - c. Optionally, click ▾ next to the operand to change it. Possible values are AND or OR.
 - d. Click ▾ located next to the license name and select the license from the list of suggestions.
 - Click ✓ to confirm this selection.
 - Click ✎ to delete this license and operand.
 - e. Repeat as necessary.
4. Optionally:
- Select **Reset Changes** to display the license(s) that appeared when you initially opened this dialog box.
 - Select a group and select **Delete Selected Group** to remove this group.
5. Click **Save Changes** if editing the license in the BOM or **Save**.

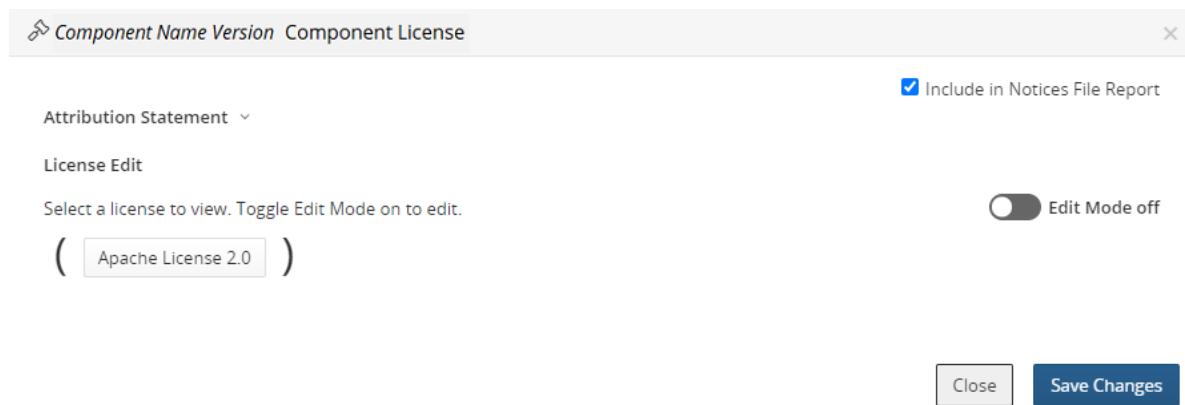
When viewed in the BOM, the license obligations for the revised license(s) will appear when you re-open the *Component Name Version* Component License dialog box.

Reverting BOM-level license edits

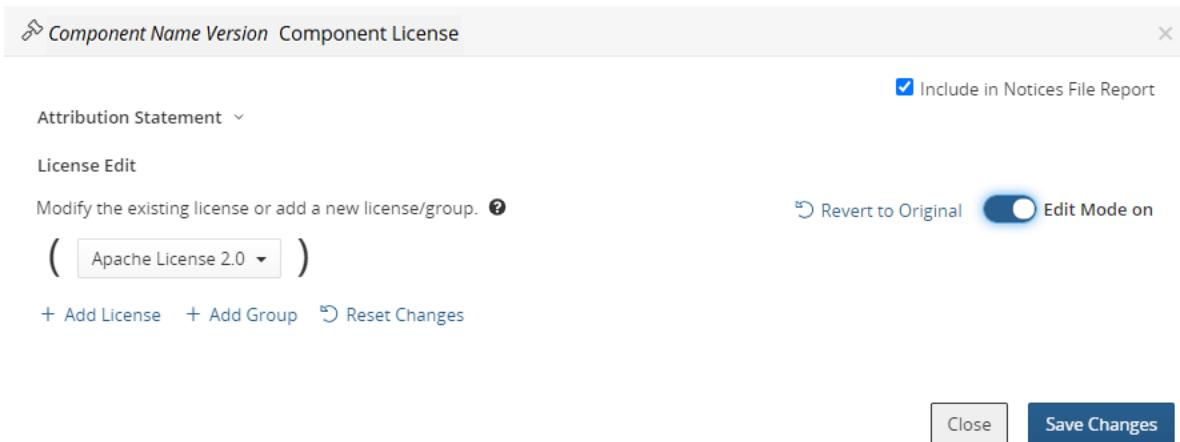
If you selected a different license for a component when editing licenses in the BOM, you can revert the license to its original license as defined in the Black Duck KnowledgeBase.

To revert to an original license

1. Log in to Black Duck.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab and view the BOM.
4. Select the to open the *Component Name Version* Component License dialog box.



5. Select the Edit Mode option to enable editing.



6. Select **Revert to Original** to revert the license

7. Click **Save Changes**.

Note that the license obligations for the revised license(s) will appear when you re-open the *Component Name Version Component License* dialog box.

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:

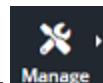
- Consist of a name, a [license family](#), and license text.
- Can be used to [create policy rules](#).
- Use the same [rules to determine license risk](#) as licenses from the Black Duck KnowledgeBase.
- Can be [modified locally in a BOM](#) by users with the appropriate [role](#). That user cannot 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 **Manage** > **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/filter bar with 'In Use' checked, a 'Filter licenses...' dropdown, and an '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. 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. Click **Create License** to open the Create a Custom License dialog box.

The screenshot shows the 'Create a Custom License' dialog box. It has fields for 'Name' (a required field with a red asterisk), 'License Family' (set to 'Nothing Selected'), 'License Text' (a large text area), 'Status' (set to 'Nothing Selected'), 'Notes' (a text area), and 'Expiration Date' (a date picker). At the bottom are 'Cancel' and 'Create' buttons.

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.

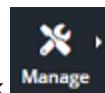
These edits only apply to the version of the custom license used in the BOM.

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.

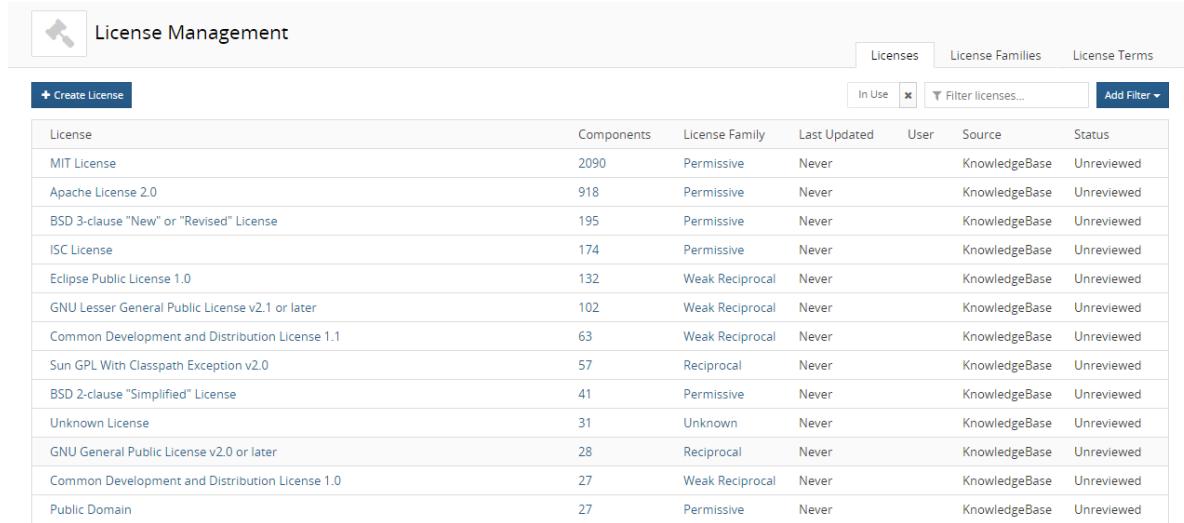
To edit a custom license

1. Log in to Black Duck with the License Manager role.



2. Click **Manage** > **License Management**.

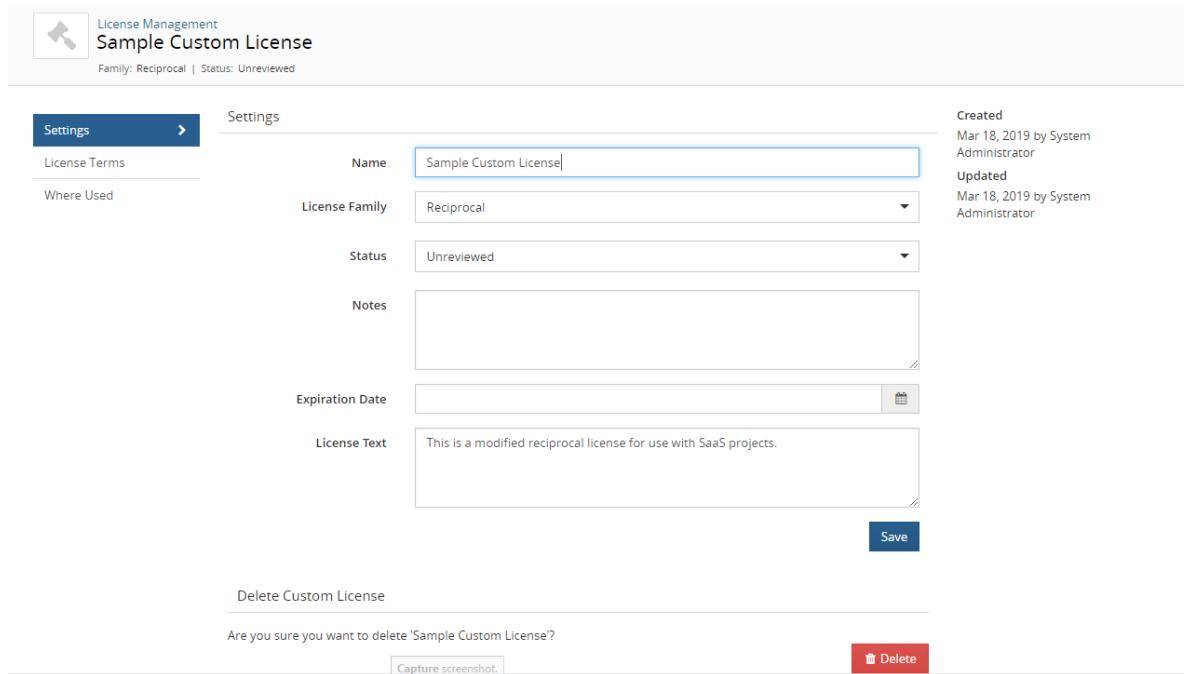
The License Management page appears.



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, ISC, Eclipse, and GPL, along with a "Public Domain" entry.

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 form fields include: Name (Sample Custom License), License Family (Reciprocal), Status (Unreviewed), Notes (empty), Expiration Date (calendar icon), and License Text (This is a modified reciprocal license for use with SaaS projects). On the right, there are "Created" and "Updated" logs. At the bottom, there's a "Save" button, a "Delete Custom License" section with a confirmation message, and a "Delete" button.

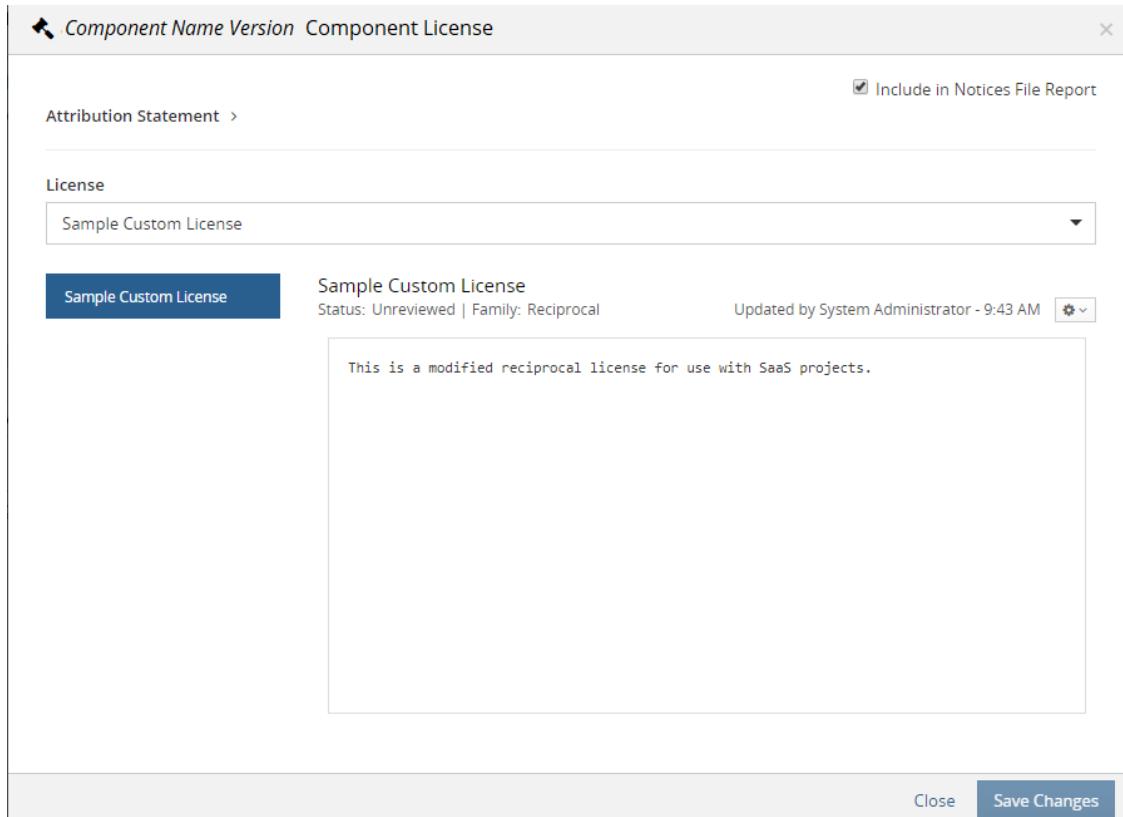
4. Modify the information shown for this custom license.

- 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.

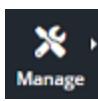


Deleting custom licenses

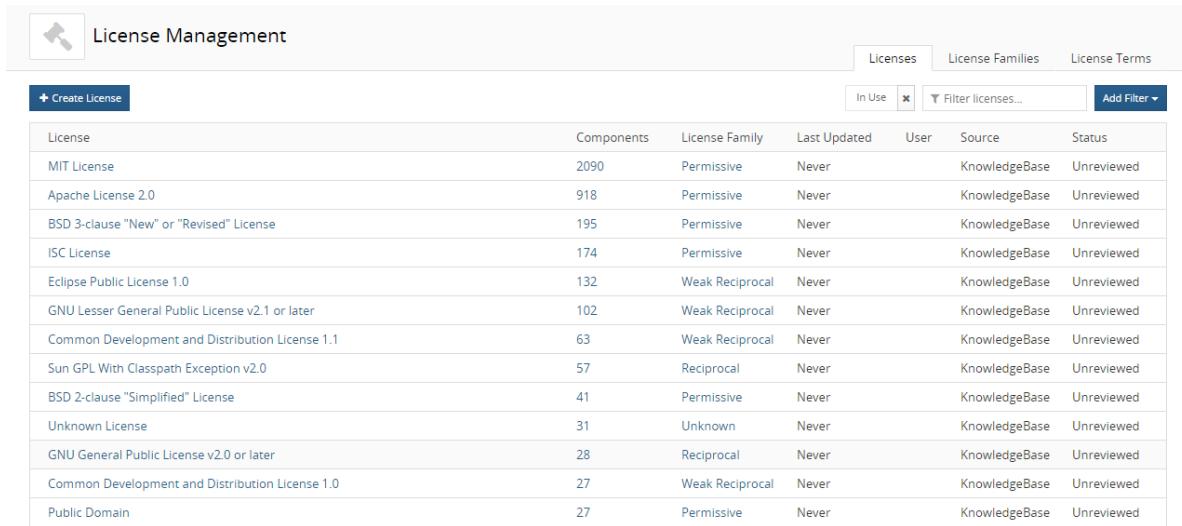
You cannot delete a license that is being used in a BOM.

You also cannot delete licenses provided by the Black Duck KnowledgeBase.

1. Log in to Black Duck with the License Manager [role](#).

2. Click  **Manage** > **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 tabs for 'Licenses' (which is selected), 'License Families', and 'License Terms'. Below the header is a search bar with filters for 'In Use' (unchecked), '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. Some rows have a blue background, indicating they are custom licenses.

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.

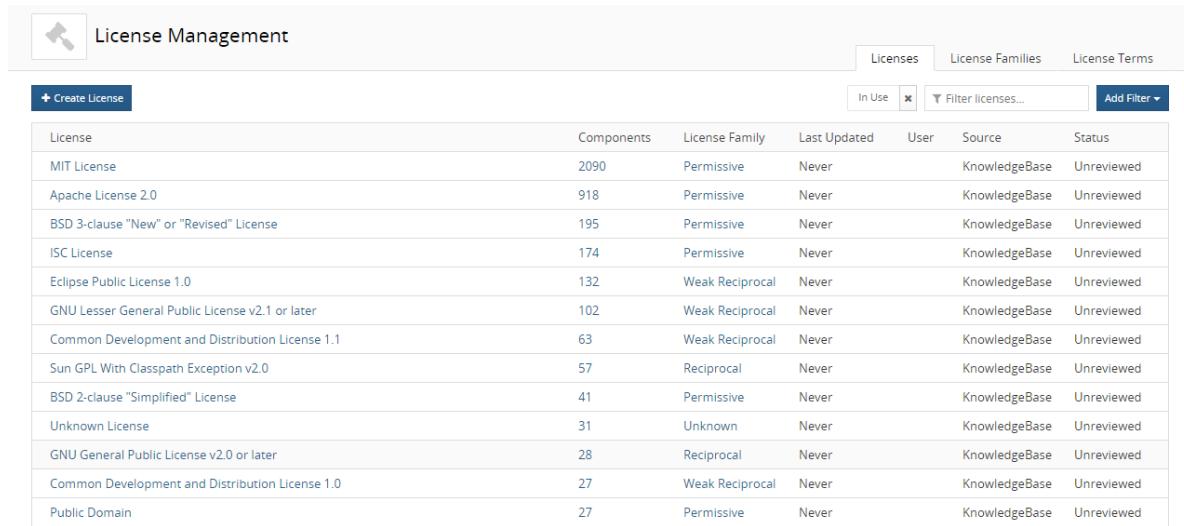
The **License Terms** tab shows all license terms for custom and KnowledgeBase licenses.

To view the License Terms tab

1. Log in to Black Duck with the License Manager [role](#).

- 
2. Click **Manage** > **License Management**.

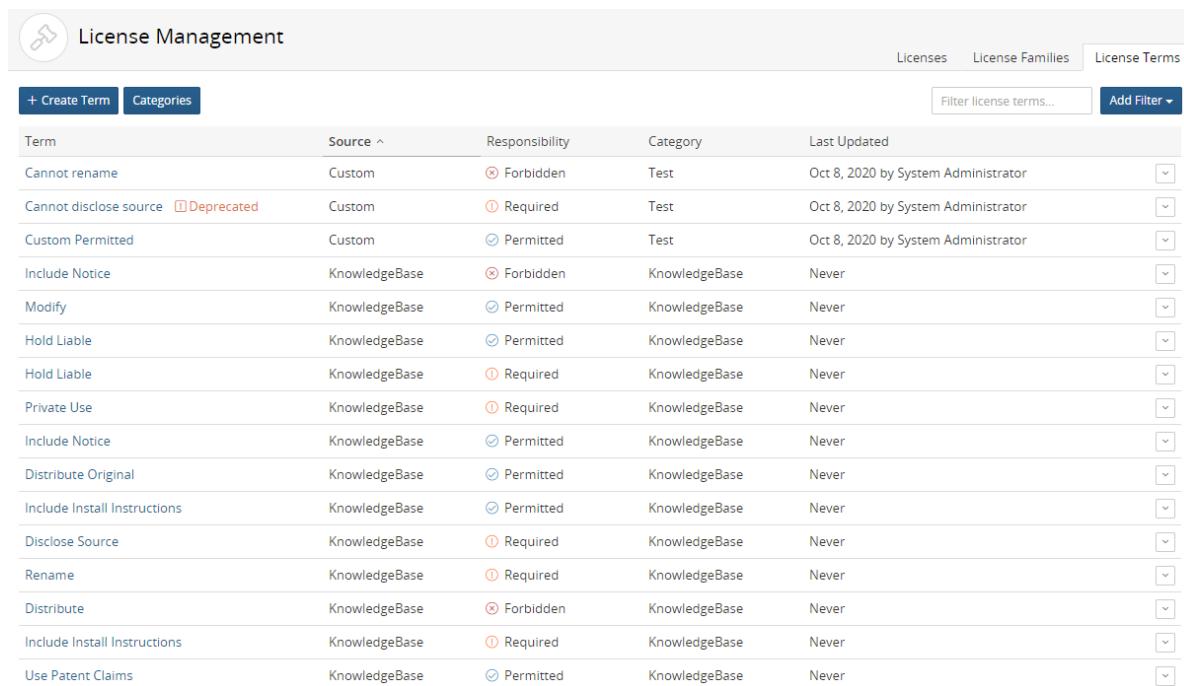
The License Management page appears.



The screenshot shows the '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/filter bar with fields for '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. Some rows have small icons next to them.

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's a navigation bar with tabs: 'Licenses' (selected), 'License Families', and 'License Terms'. Below the navigation bar is a search/filter bar with fields for 'Filter license terms...' and 'Add Filter'. A large table lists various license terms with columns for Term, Source, Responsibility, Category, and Last Updated. Some rows have small icons next to them.

Term	Source ^	Responsibility	Category	Last Updated
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small>Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never

The table provides the following information:

Column	Description
Term	<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>
Source	<p>The source for this term. Possible values are:</p> <ul style="list-style-type: none"> KnowledgeBase. This is a standard term from the Black Duck KnowledgeBase. Custom. A license term you created.
Responsibility	<p>Responsibility for this license term. Possible values are:</p> <ul style="list-style-type: none"> Permitted Forbidden Required
Category	<p>Category for this license term.</p> <p>License terms from the Black Duck KnowledgeBase have KnowledgeBase as the category. Custom license terms list the category defined when adding the term.</p>
Last Updated	<p>Date that the license term was last updated and the username of the user who updated this term.</p> <p>The column lists Never 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.

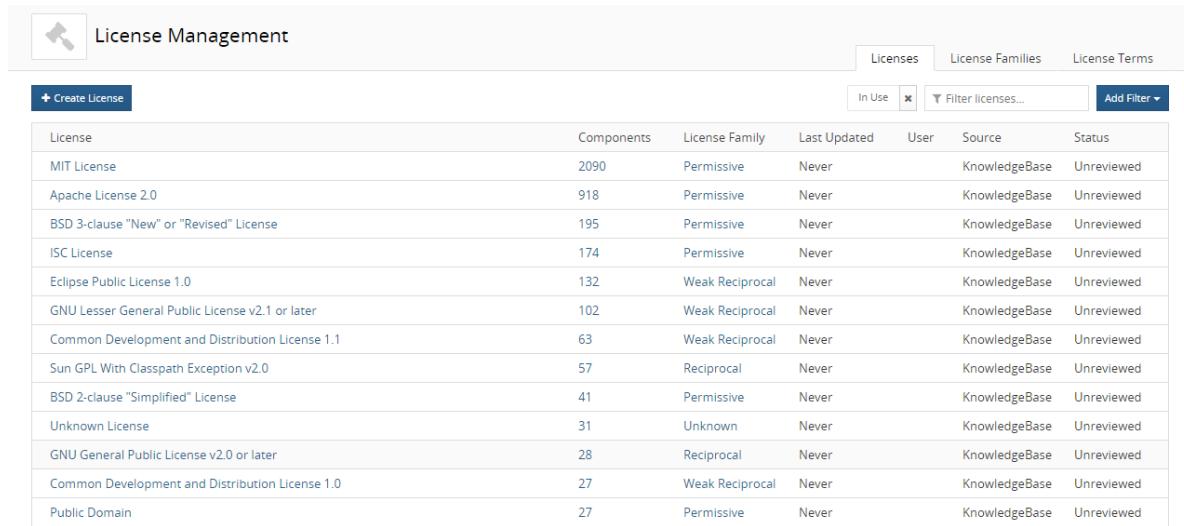
Note: License obligation will not appear in the UI, if the information is unavailable from [OpenHub](#).

To view the license terms from the License Management page

1. Log in to Black Duck with the License Manager [role](#).

- 
2. Click **Manage** > **License Management**.

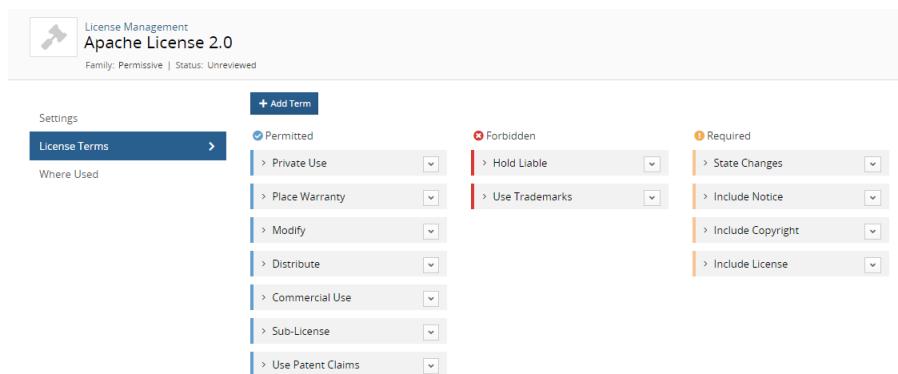
The License Management page appears.



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, ISC, Eclipse, and GPL, along with a "Public Domain" entry.

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.



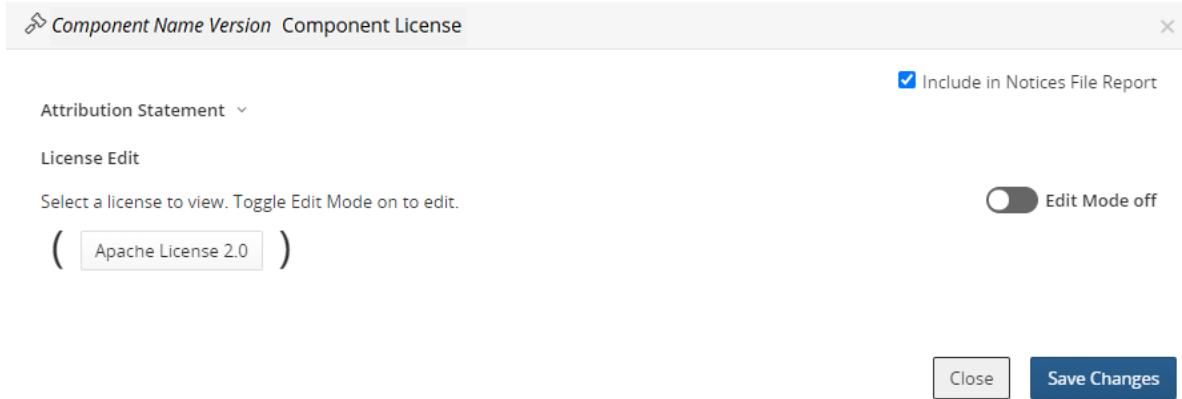
The screenshot shows the "Apache License 2.0" settings page. It includes tabs for "Settings", "License Terms", and "Where Used". Under "License Terms", there is a "Add Term" button and three sections: "Permitted" (with options like Private Use, Place Warranty, Modify, Distribute, Commercial Use, Sub-License, and Use Patent Claims), "Forbidden" (with Hold Liable and Use Trademarks), and "Required" (with State Changes, Include Notice, Include Copyright, and 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. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the version name to open the **Components** tab and view the BOM.
3. Select the license name to open the *Component Name Version* Component License dialog box.



4. Select the license you wish to view the license obligation information. The dialog box expands to show the obligations and license text for the selected license.

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 **Term Fulfillment** tab on 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.
- Policy managers can [create a policy rule](#) that will trigger a violation when there are unfulfilled license terms.

Note that the **Term Fulfillment** tab on 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](#).

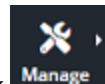
- A new project version report, `license_term_fulfillment_date_time.csv` lists the license terms and fulfillment status for a project version.

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.

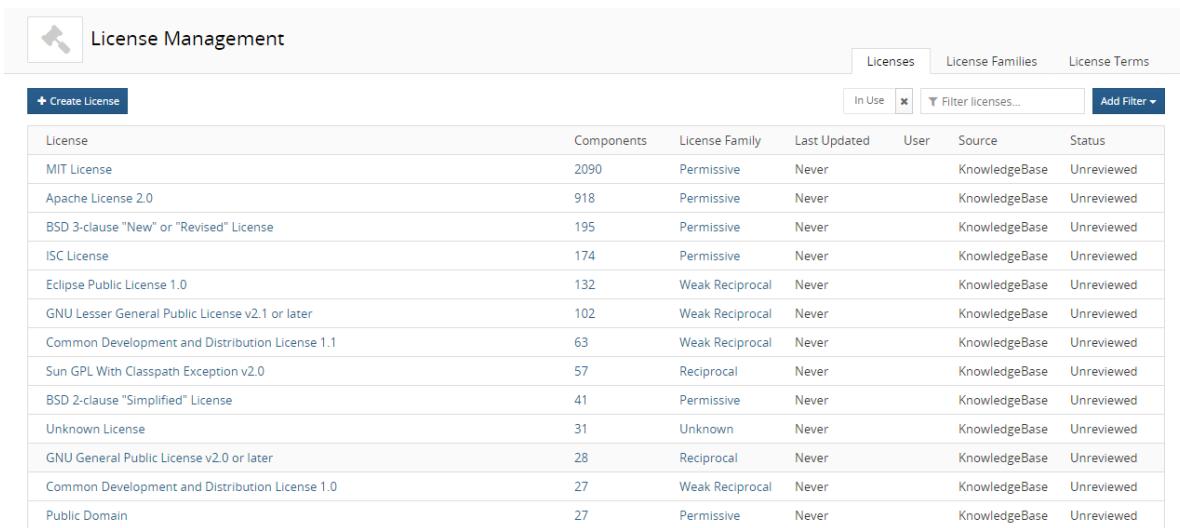
To define the fulfillment requirement when viewing a license

1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **License Management**.

The License Management page appears.



A screenshot of the Black Duck License Management interface. The title bar says "License Management". Below it is a toolbar with a "Create License" button, a search bar labeled "In Use" with a clear button, a "Filter licenses..." button, and an "Add Filter" dropdown. The main area is a table with the following columns: License, Components, License Family, Last Updated, User, Source, and Status. The table lists various open source licenses:

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: 'Name' (Apache License 2.0), 'License Family' (Permissive), 'Status' (Unreviewed), 'Notes' (empty), 'Expiration Date' (empty), and 'License Text' (containing the Apache License text). A 'Save' button is located at the bottom right. On the left, there's a sidebar with tabs for 'Settings' (selected) and 'License Terms'. Below the sidebar, there's a section for 'Where Used'.

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. It features a grid of license terms categorized into three groups: 'Permitted' (blue border), 'Forbidden' (red border), and 'Required' (orange border). The 'Permitted' group includes: Private Use, Place Warranty, Modify, Distribute, Commercial Use, Sub-License, and Use Patent Claims. The 'Forbidden' group includes: Hold Liable and Use Trademarks. The 'Required' group includes: State Changes, Include Notice, Include Copyright, and Include License. A 'Add Term' button is located at the top left of the grid. On the left, there's a sidebar with tabs for 'Settings' and 'License Terms' (selected). Below the sidebar, there's a section for 'Where Used'.

5. 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 page in the License Management section. At the top, there's a key icon, the title "Apache License 2.0", and a status message "Family: Permissive | Status: Limited Approval". Below this, there are tabs for "Settings", "License Terms" (which is selected and highlighted in blue), and "Where Used". A "Add Term" button is located at the top right. The main area is divided into three columns: "Permitted" (blue background), "Forbidden" (red background), and "Required" (orange background). Each column contains several license terms with dropdown menus.

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		

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

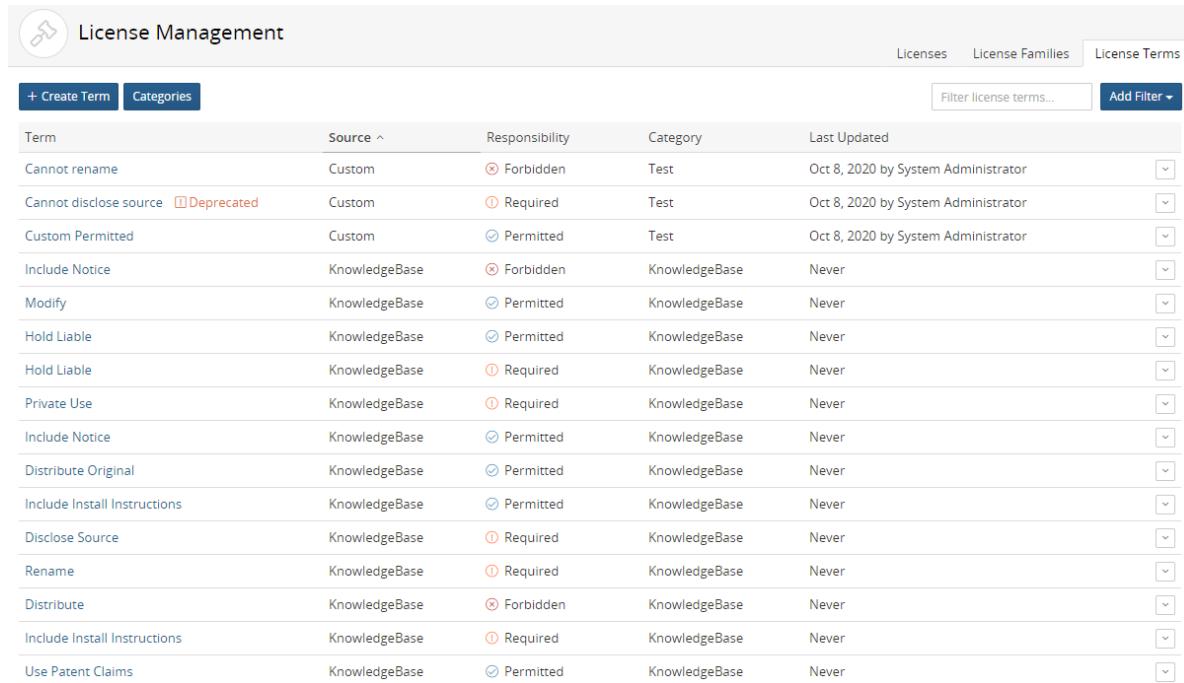
1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **License Management**.

The License Management page appears.

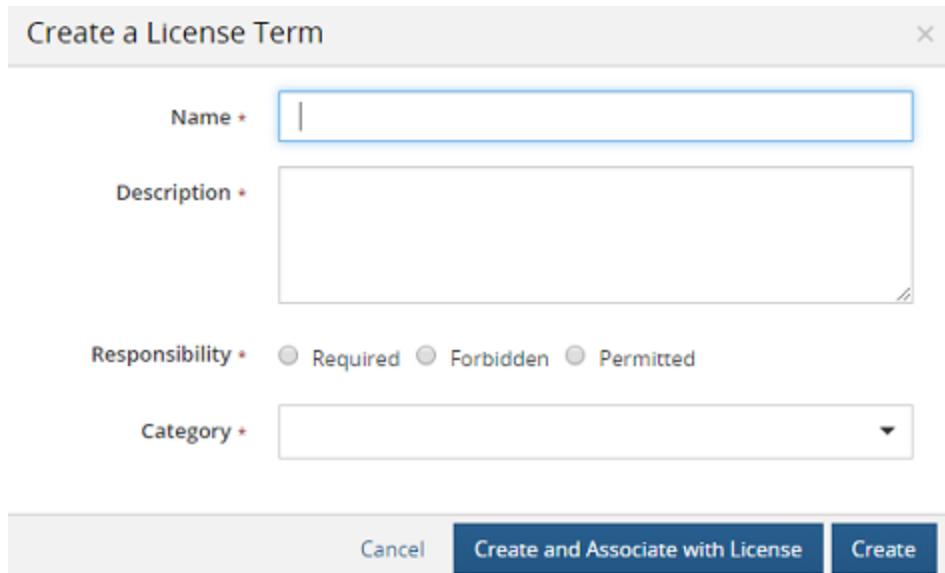
Select the **License Terms** tab to display all license terms.



The screenshot shows a table titled "License Management" with the following columns: Term, Source, Responsibility, Category, and Last Updated. There are 18 rows of data, each representing a different license term. The "Source" column includes entries like "Custom", "KnowledgeBase", and "Deprecated". The "Responsibility" column uses icons to represent "Forbidden", "Required", and "Permitted". The "Category" column contains mostly "Test" or "KnowledgeBase". The "Last Updated" column shows dates from October 8, 2020, to "Never".

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small> Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never

3. Click **Create Terms** to open the Create a License Term dialog box.



The dialog box has the following fields:

- Name ***: An input field with a placeholder icon.
- Description ***: A large text area for entering a description.
- Responsibility ***: Radio buttons for "Required", "Forbidden", and "Permitted".
- Category ***: A dropdown menu.

At the bottom are three 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 **Manage** > **License Management**.

The License Management page appears.

A screenshot of the Black Duck License Management interface. The top navigation bar includes a wrench and gear icon, the title "License Management", and tabs for "Licenses", "License Families", and "License Terms". A "Create License" button is located in the top-left corner of the main content area. The main content is a table listing various open source licenses. The columns are: License, Components, License Family, Last Updated, User, Source, and Status. The table lists the following data:

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 'Settings' tab is selected. The page contains several input fields: 'Name' (Apache License 2.0), 'License Family' (Permissive), 'Status' (Unreviewed), 'Notes' (empty text area), 'Expiration Date' (empty date field with a calendar icon), and 'License Text' (a rich text area containing the Apache License text). On the right side, there are 'Created' and 'Updated' timestamps both set to 'never'. A 'Save' button is located 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' License Terms page. The 'License Terms' tab is selected. It lists various terms categorized into three groups: '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). A 'Where Used' section is also present. A '+ Add Term' button is located at the top right of the term list.

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

+ Add Term

Settings License Terms > Where Used

Permitted Forbidden Required
No term associated

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

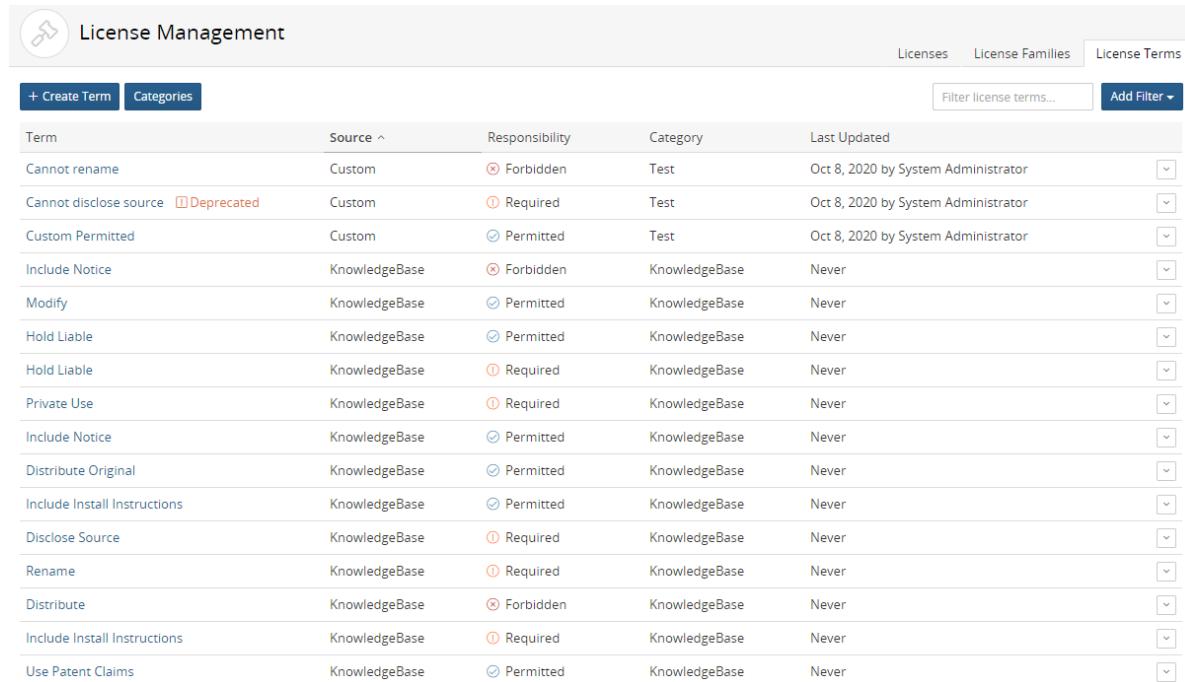
You can also create a category when [creating a license term](#).

1. Log in to Black Duck with the License Manager role.

2. Click  > License Management.

The License Management page appears.

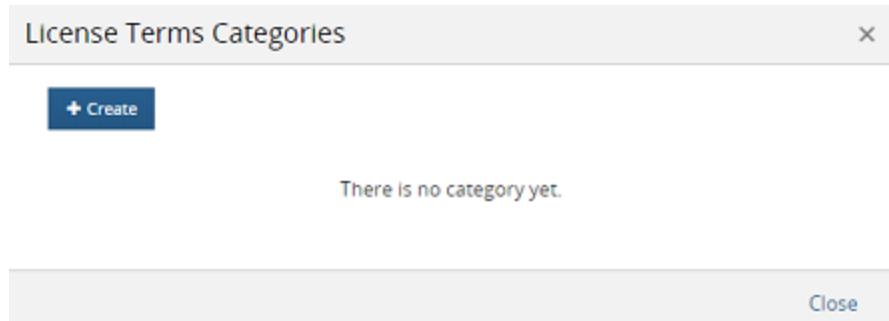
Select the **License Terms** tab to display all license terms.



License Management				
		Licenses	License Families	License Terms
+ Create Term Categories		Filter license terms... Add Filter		
Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small> ⓘ Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never

3. Click **Categories**.

The License Terms Categories dialog box appears.



The screenshot shows a dialog box titled "License Terms Categories". At the top right is a close button (an "X"). Below the title is a blue button with a white plus sign and the word "Create". Underneath the button, a message reads "There is no category yet.". At the bottom right of the dialog is another "Close" button.

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

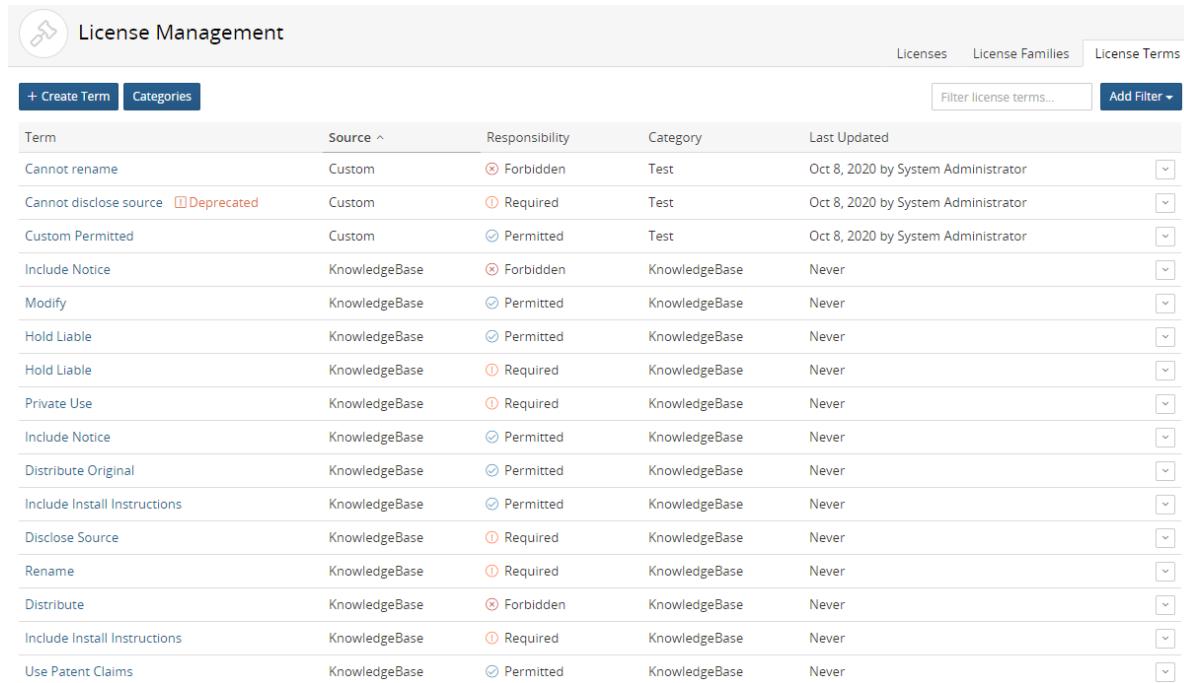
You cannot delete a category that is in use.

1. Log in to Black Duck with the License Manager role.

2. Click  > **License Management**.

The License Management page appears.

Select the **Terms** tab to display all license terms.



The screenshot shows a table titled "License Management" with the following columns: Term, Source, Responsibility, Category, and Last Updated. There are 18 rows of data, each representing a different license term. The "Source" column includes entries like "Custom", "KnowledgeBase", and "Deprecated". The "Responsibility" column uses icons to indicate whether a term is "Forbidden" (red X), "Required" (orange circle), or "Permitted" (blue circle). The "Category" column lists categories such as "Test", "KnowledgeBase", and "Never". The "Last Updated" column shows dates like "Oct 8, 2020 by System Administrator". At the top of the table, there are buttons for "+ Create Term" and "Categories", and filters for "Filter license terms..." and "Add Filter".

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small> Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never

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

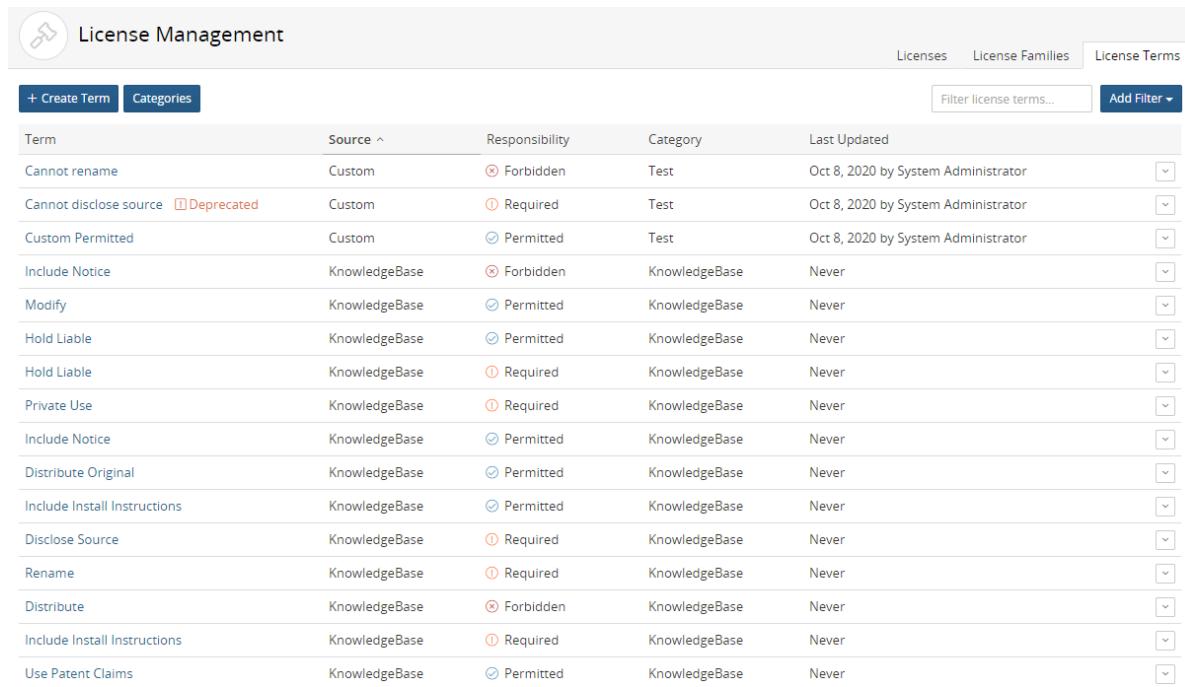
You can associate a new license term you created or an existing KnowledgeBase term to one or more custom or KnowledgeBase licenses.

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:

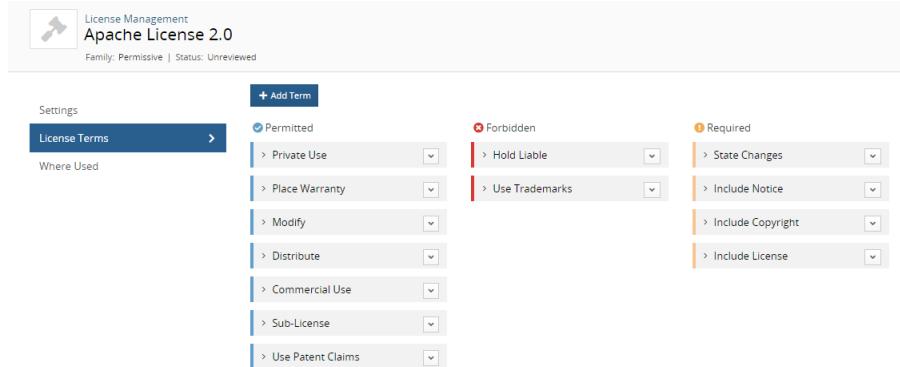
- Creating a license term. Click [here](#) for more information about creating a new term.
- Using the **License Terms** tab which lists all license terms:



The screenshot shows the Black Duck License Management interface. At the top, there's a header with a wrench icon, the title "License Management", and tabs for "Licenses", "License Families", and "License Terms". Below the header is a search bar labeled "Filter license terms..." and a "Add Filter" button. A table lists various license terms with columns for "Term", "Source", "Responsibility", "Category", and "Last Updated". Each row has a small dropdown arrow at the end.

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	☒ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small>_DEPRECATED</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	☑ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	☒ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	☒ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	☑ Permitted	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 "Apache License 2.0" and "Family: Permissive | Status: Unreviewed". Below that is a "License Terms" tab. On the left, there's a "Settings" section with a "Where Used" dropdown. The main area shows three groups of terms: "Permitted" (selected), "Forbidden", and "Required". Each group has several items listed with dropdown menus.

Setting	Value
Where Used	Apache License 2.0
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

To associate a license term to one or more licenses

Use these procedures to associate a license term to one or more licenses.

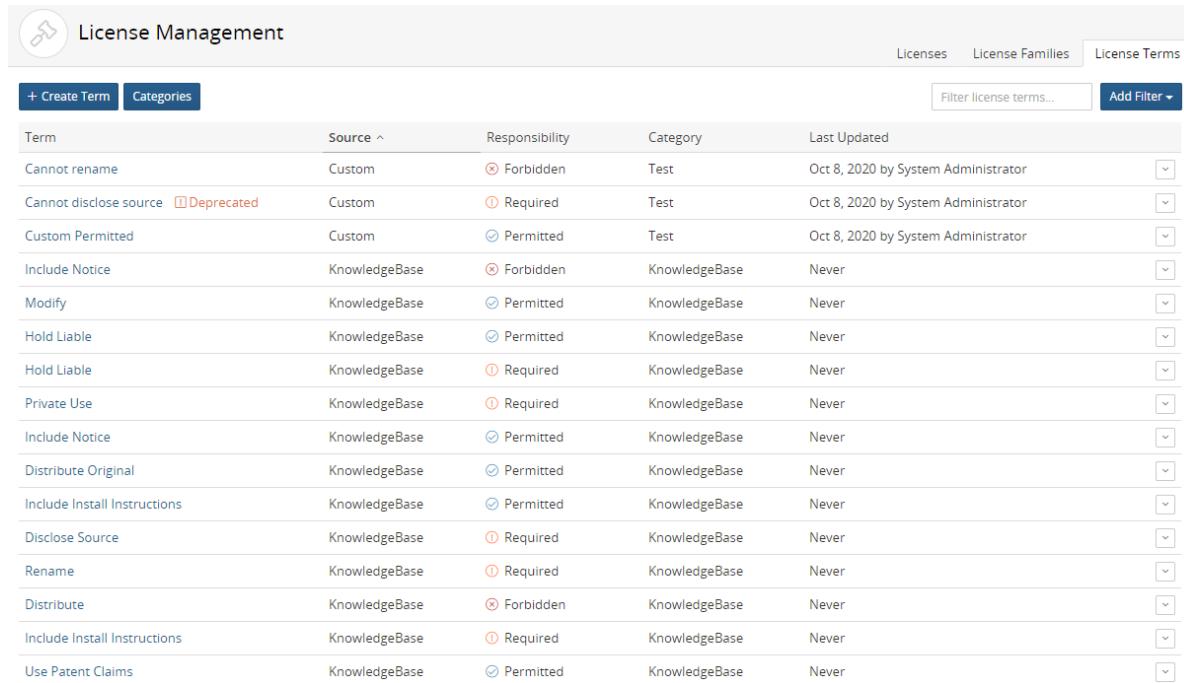
1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.

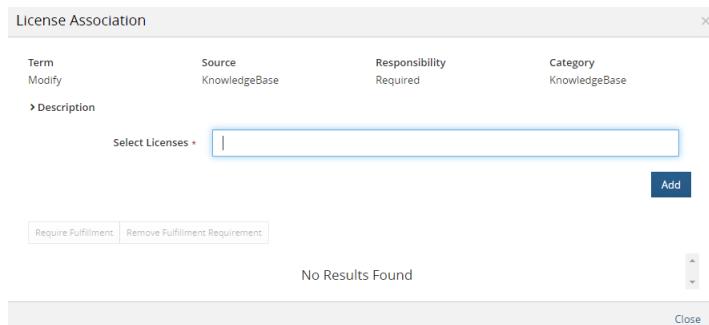


The screenshot shows a table titled "License Management" with the following columns: Term, Source, Responsibility, Category, and Last Updated. The table lists various license terms such as "Cannot rename", "Cannot disclose source", "Custom Permitted", etc., each with its source (Custom, KnowledgeBase), responsibility (Forbidden, Required, Permitted), category (Test, KnowledgeBase), and last updated date (Oct 8, 2020 by System Administrator). There are also dropdown arrows next to each row.

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	☒ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small> Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	☑ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	☒ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	☒ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	☑ Permitted	KnowledgeBase	Never

3. Click  in the row of the license term and select **License Association**.

The License Association dialog box appears.



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:
- Select the check box next to the license where fulfillment of this term is required.
 - 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 **Manage** > **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 a wrench and gear icon, the title "License Management", and tabs for "Licenses", "License Families", and "License Terms". Below the tabs are buttons for "+ Create License", "In Use", "Filter licenses...", and "Add Filter". 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:

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 'Settings' tab is selected. The form includes fields for 'Name' (Apache License 2.0), 'License Family' (Permissive), 'Status' (Unreviewed), 'Notes' (empty), 'Expiration Date' (empty), and a large 'License Text' area containing the text 'Apache License Version 2.0, January 2004' followed by several '=' characters. A 'Save' button is located at the bottom right. On the right side of the form, there are 'Created' and 'Updated' status indicators.

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. The 'License Terms' tab is selected. It features a 'Where Used' section and a main area titled '+ Add Term' with three categories: 'Permitted' (selected), 'Forbidden', and 'Required'. Under 'Permitted', there is a list of terms: Private Use, Place Warranty, Modify, Distribute, Commercial Use, Sub-License, and Use Patent Claims. Under 'Forbidden', there are two terms: Hold Liable and Use Trademarks. Under 'Required', there are four terms: State Changes, Include Notice, Include Copyright, and Include License. A 'Save' button is located at the bottom right.

5. Click **Add Term** to open the Add Term dialog box.
6. Select **Existing** to add an existing license term.

The screenshot shows the 'Add Term' dialog box. At the top, there are radio buttons for 'Existing' and 'New'. Below that is a 'Name' field with a dropdown menu containing 'Start typing to add a term...'. There is also a 'Description' field, a 'Responsibility' section with radio buttons for 'Required', 'Forbidden', and 'Permitted', a 'Category' field, and a 'Fulfillment' section with a checkbox labeled 'Required'. At the bottom of the dialog are 'Cancel' and 'Add' buttons.

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 () 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.

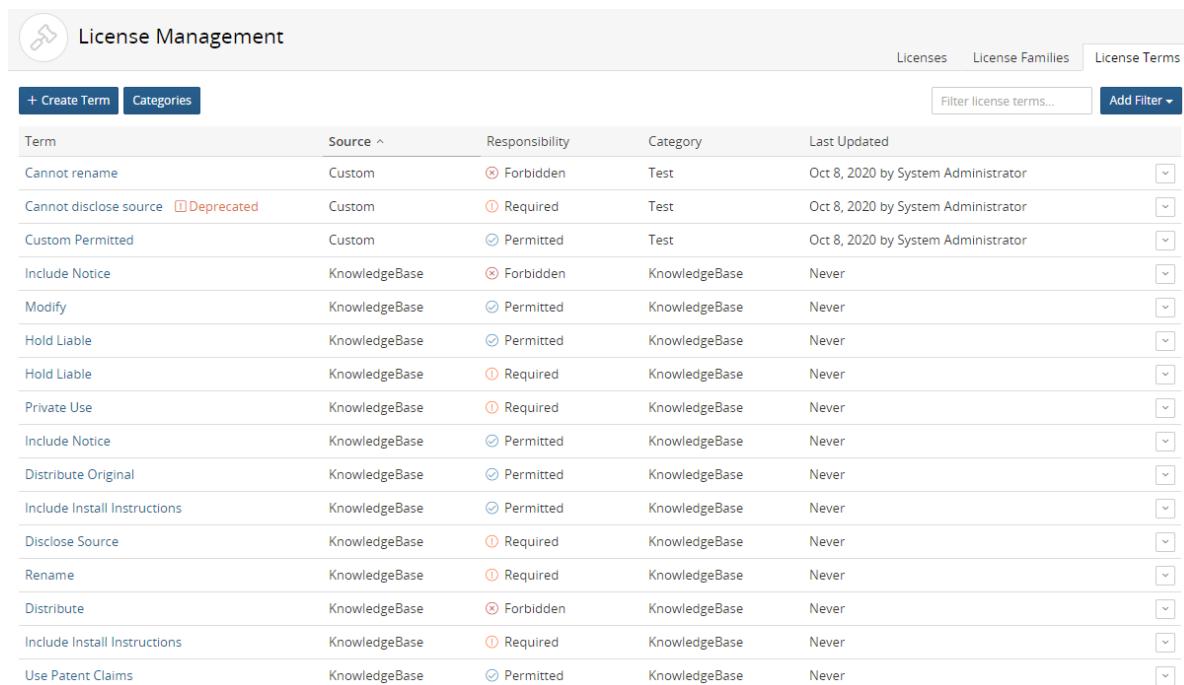
To edit a license term

1. Log in to Black Duck with the License Manager [role](#).

-
- The screenshot shows the Black Duck navigation bar. It includes a logo, a search bar, and links for 'Manage' and 'License Management'. The 'Manage' link is highlighted in blue.
2. Click **Manage** > **License Management**.

The License Management page appears.

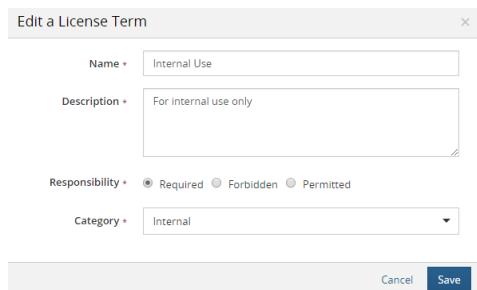
Select the **License Terms** tab to display all license terms.



The screenshot shows a table titled "License Management" with the following columns: Term, Source, Responsibility, Category, and Last Updated. There are 19 rows of data, each representing a license term. The "Source" column includes entries like "Custom", "KnowledgeBase", and "Deprecated". The "Responsibility" column uses icons to represent "Forbidden", "Required", and "Permitted". The "Category" column contains mostly "Test" or "KnowledgeBase". The "Last Updated" column shows dates from October 8, 2020, to "Never". A search bar and filter button are at the top right.

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small> Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never

3. Select the license term to open the Edit a License Term dialog box.



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 **Manage** > **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	☒ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	☑ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	☒ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	☒ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	☑ Permitted	KnowledgeBase	Never

3. Click in the row of the license term and select **Delete**.

The Delete a License Term dialog box appears.

4. 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.

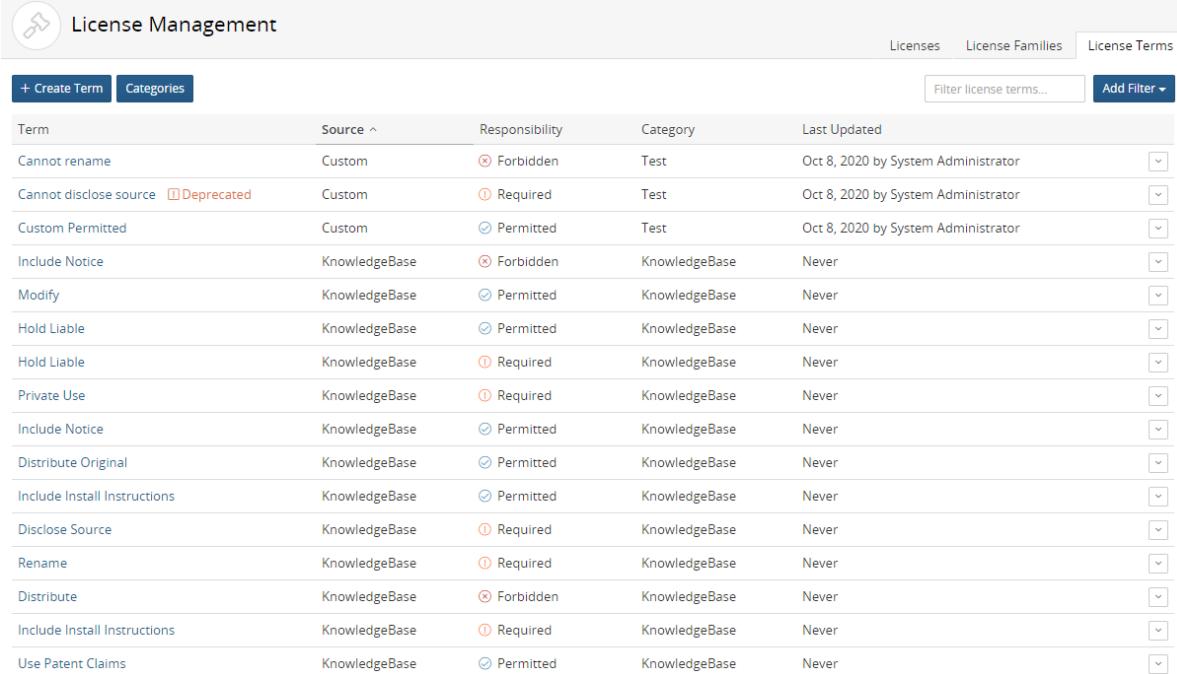
1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.



A screenshot of the Black Duck License Management interface. At the top, there's a navigation bar with icons for Home, Manage, and Help. Below the navigation bar, the title "License Management" is displayed next to a key icon. To the right of the title are three tabs: "Licenses", "License Families", and "License Terms", with "License Terms" being the active tab. Underneath 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 "Cannot rename", "Cannot disclose source", "Custom Permitted", etc. Some terms like "Cannot disclose source" have a small red "Deprecated" label next to them. The "Last Updated" column shows dates like "Oct 8, 2020 by System Administrator".

Term	Source	Responsibility	Category	Last Updated	
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator	<input checked="" type="checkbox"/>
Cannot disclose source	Custom	✗ Required	Test	Oct 8, 2020 by System Administrator	<input checked="" type="checkbox"/>
Custom Permitted	Custom	✓ Permitted	Test	Oct 8, 2020 by System Administrator	<input checked="" type="checkbox"/>
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Modify	KnowledgeBase	✓ Permitted	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Hold Liable	KnowledgeBase	✓ Permitted	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Hold Liable	KnowledgeBase	✗ Required	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Private Use	KnowledgeBase	✗ Required	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Include Notice	KnowledgeBase	✓ Permitted	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Distribute Original	KnowledgeBase	✓ Permitted	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Include Install Instructions	KnowledgeBase	✓ Permitted	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Disclose Source	KnowledgeBase	✗ Required	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Rename	KnowledgeBase	✗ Required	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Include Install Instructions	KnowledgeBase	✗ Required	KnowledgeBase	Never	<input checked="" type="checkbox"/>
Use Patent Claims	KnowledgeBase	✓ Permitted	KnowledgeBase	Never	<input checked="" type="checkbox"/>

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.

The label appears next to the license term where the term appears in the **License Terms** tabs in License Management.

Note that the 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 **Manage** > **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	☒ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	☑ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	☒ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	☒ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	☑ Permitted	KnowledgeBase	Never

3. Click in the row of the license term and select **Remove Deprecated Status**.

The Deprecate a License Term dialog box appears.

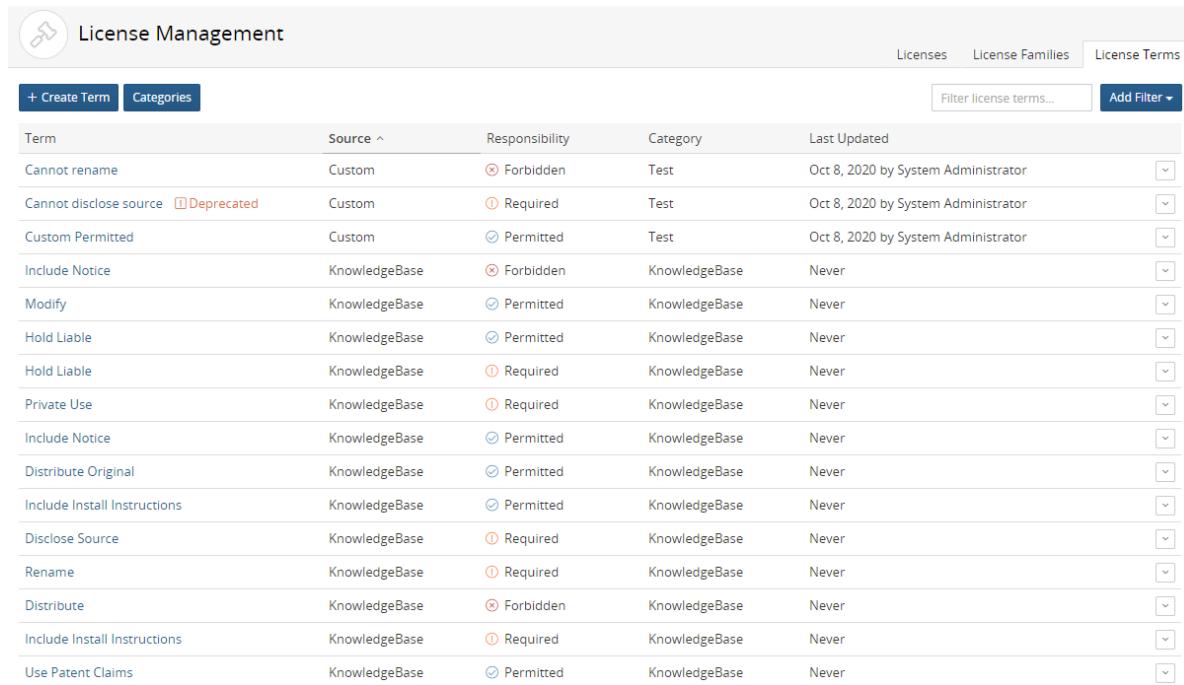
4. Click **Remove Deprecated Status** to confirm. The **Deprecated** label is removed from the license term.

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:

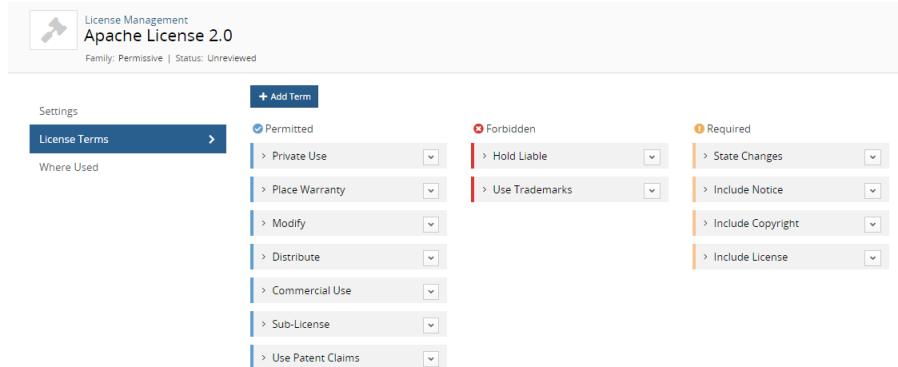
■ Using the **License Terms** tab which lists all license terms



The screenshot shows the 'License Management' interface with the 'License Terms' tab selected. The page title is 'License Management'. There are tabs for '+ Create Term' and 'Categories'. A search bar says 'Filter license terms...' and a button says 'Add Filter'. The main area is a table with columns: Term, Source, Responsibility, Category, and Last Updated. The table lists various license terms like 'Cannot rename', 'Cannot disclose source', etc., with their respective details.

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small>Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never

■ Using the **License Terms** tab for an individual license

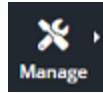


The screenshot shows the 'Apache License 2.0' settings page. The 'License Terms' tab is selected. It shows a list of terms categorized into 'Permitted', 'Forbidden', and 'Required'. The 'Permitted' section includes 'Private Use', 'Place Warranty', 'Modify', 'Distribute', 'Commercial Use', 'Sub-License', and 'Use Patent Claims'. The 'Forbidden' section includes 'Hold Liable' and 'Use Trademarks'. The 'Required' section includes 'State Changes', 'Include Notice', 'Include Copyright', and 'Include License'.

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



> License Management.

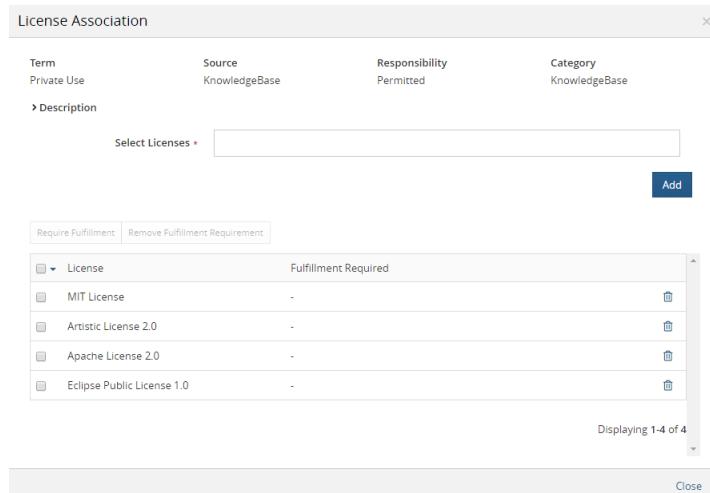
The License Management page appears.

Select the **License Terms** tab to display all license terms.

Term	Source	Responsibility	Category	Last Updated	
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator	<input type="button" value="▼"/>
Cannot disclose source <small>⚠ Deprecated</small>	Custom	⚠ Required	Test	Oct 8, 2020 by System Administrator	<input type="button" value="▼"/>
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator	<input type="button" value="▼"/>
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Hold Liable	KnowledgeBase	⚠ Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Private Use	KnowledgeBase	⚠ Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Disclose Source	KnowledgeBase	⚠ Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Rename	KnowledgeBase	⚠ Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Install Instructions	KnowledgeBase	⚠ Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Use Patent Claims	KnowledgeBase	ⓘ Permitted	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.



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 **Manage** > **License Management**.

The License Management page appears.

License Management						
+ Create License		Licenses	License Families	License Terms		
		In Use	<input type="button" value="Filter licenses..."/>	<input type="button" value="Add Filter"/>		
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" and its family "Permissive". Below this, the "Settings" tab is selected. The page contains several input fields and dropdown menus:

- Name:** Apache License 2.0
- License Family:** Permissive
- Status:** Unreviewed
- Notes:** (empty text area)
- Expiration Date:** (empty text field)
- License Text:** Apache License
Version 2.0, January 2004
=====

On the right side, there are "Created" and "Updated" status indicators. A "Save" button is located 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 license terms page. The "License Terms" tab is selected. On the left, there is a list of terms under "Where Used". On the right, there is a grid of terms categorized into three groups: Permitted (blue), Forbidden (red), and Required (orange). Each group has a "Add Term" button at the top. The terms listed are:

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

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	☒ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small>Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	☑ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	☒ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	☑ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	☒ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	☑ Permitted	KnowledgeBase	Never

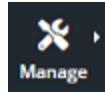
- Using the **License Terms** tab for an individual license

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



> License Management.

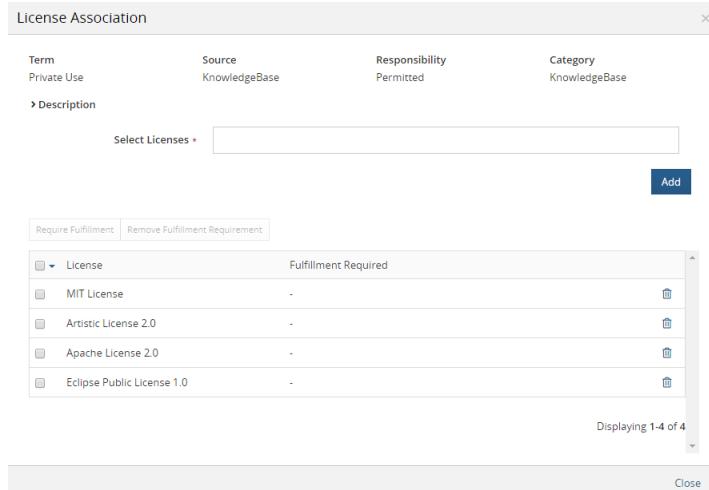
The License Management page appears.

Select the **License Terms** tab to display all license terms.

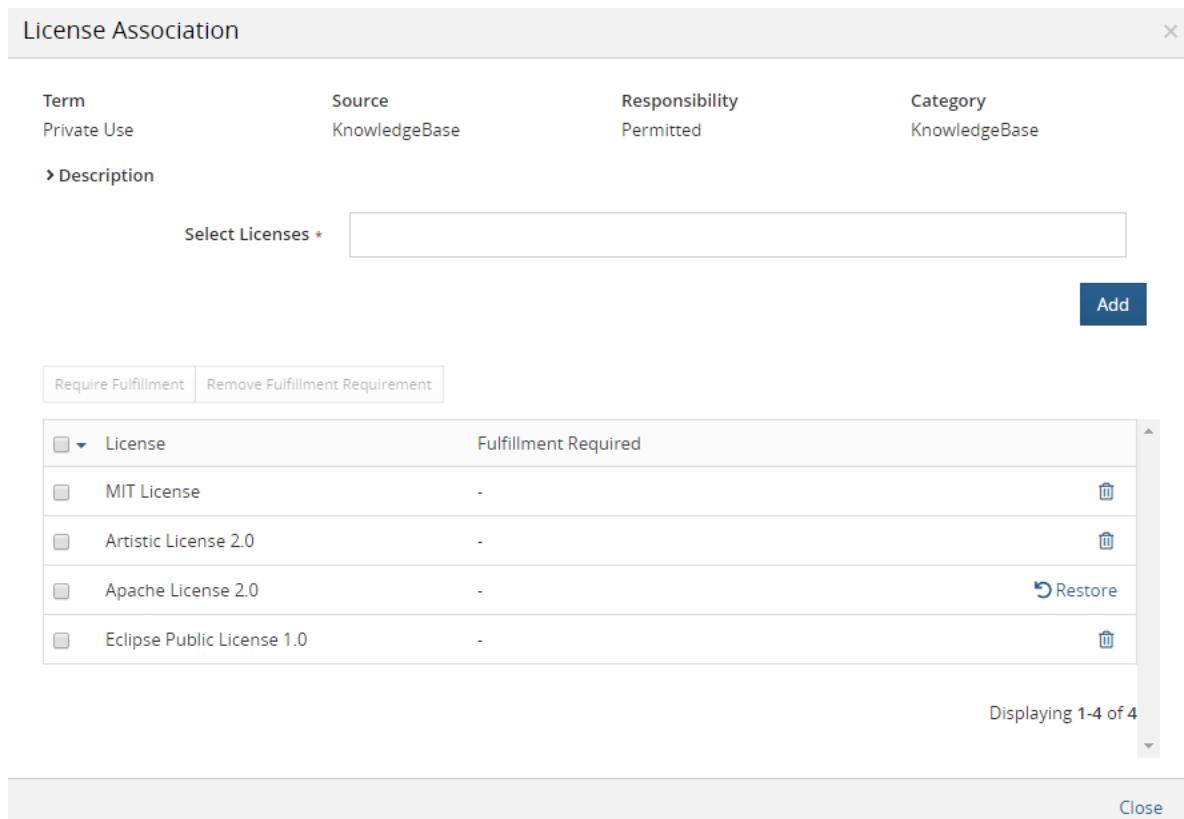
Term	Source	Responsibility	Category	Last Updated	
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator	<input type="button" value="▼"/>
Cannot disclose source <small>⚠ Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator	<input type="button" value="▼"/>
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator	<input type="button" value="▼"/>
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never	<input type="button" value="▼"/>
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never	<input type="button" value="▼"/>
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never	<input type="button" value="▼"/>
Use Patent Claims	KnowledgeBase	ⓘ Permitted	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.



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.



To deactivate a KnowledgeBase license term when viewing a license

1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **License Management**.

The License Management page appears.

License Management			Licenses	License Families	License Terms
		+ Create License	In Use	Filter licenses...	Add Filter
License		Components	License Family	Last Updated	User
MIT License		2090	Permissive	Never	KnowledgeBase
Apache License 2.0		918	Permissive	Never	KnowledgeBase
BSD 3-clause "New" or "Revised" License		195	Permissive	Never	KnowledgeBase
ISC License		174	Permissive	Never	KnowledgeBase
Eclipse Public License 1.0		132	Weak Reciprocal	Never	KnowledgeBase
GNU Lesser General Public License v2.1 or later		102	Weak Reciprocal	Never	KnowledgeBase
Common Development and Distribution License 1.1		63	Weak Reciprocal	Never	KnowledgeBase
Sun GPL With Classpath Exception v2.0		57	Reciprocal	Never	KnowledgeBase
BSD 2-clause "Simplified" License		41	Permissive	Never	KnowledgeBase
Unknown License		31	Unknown	Never	KnowledgeBase
GNU General Public License v2.0 or later		28	Reciprocal	Never	KnowledgeBase
Common Development and Distribution License 1.0		27	Weak Reciprocal	Never	KnowledgeBase
Public Domain		27	Permissive	Never	KnowledgeBase

3. In the **Licenses** tab, select the license name to display the *License Name Settings* tab.

Apache License 2.0	
Family: Permissive Status: Unreviewed	
Settings	Settings
License Terms	Name: Apache License 2.0
Where Used	License Family: Permissive
	Status: Unreviewed
Notes	
Expiration Date	
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. In the 'Permitted' section, 'Private Use' is checked. In the 'Forbidden' section, 'Hold Liable' and 'Use Trademarks' are checked. In the 'Required' section, 'State Changes', 'Include Notice', 'Include Copyright', and 'Include License' are checked.

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.

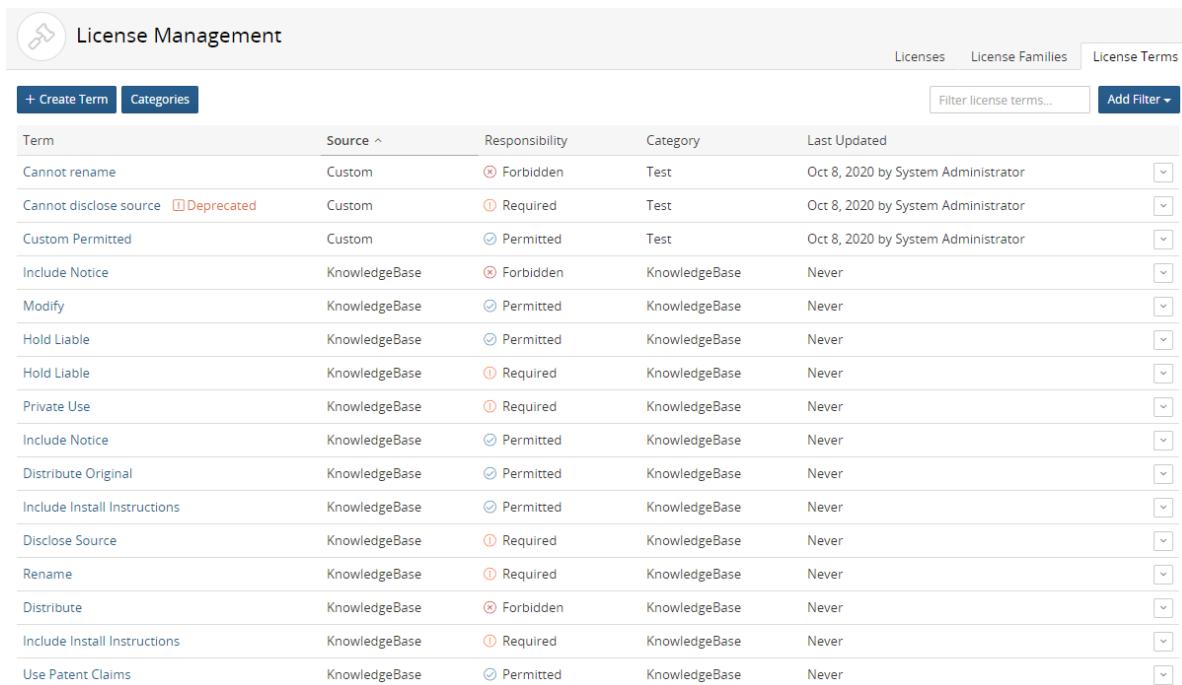
The screenshot shows the Apache License 2.0 settings page. The 'License Terms' tab is active. In the 'Permitted' section, 'Private Use' is checked. In the 'Forbidden' section, 'Hold Liable' and 'Use Trademarks' are checked. In the 'Required' section, 'State Changes', 'Include Notice', 'Include Copyright', and 'Include License' are checked.

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:

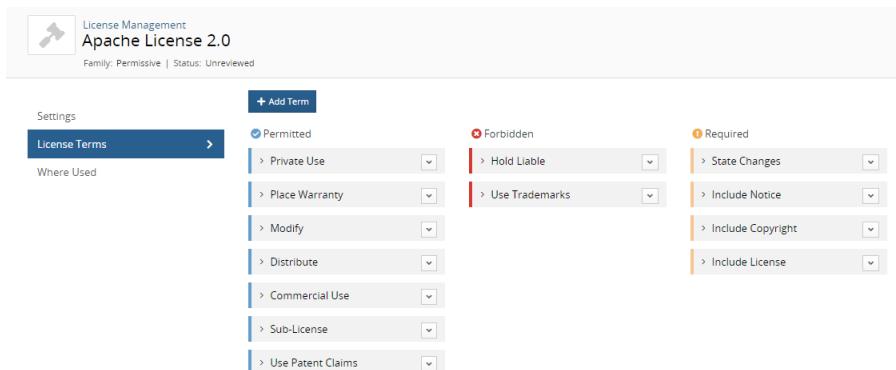
■ Using the **License Terms** tab which lists all license terms



The screenshot shows the Black Duck License Management interface. The title bar says "License Management". Below it, there are tabs: "Licensing", "License Families", and "License Terms" (which is highlighted). There are also buttons for "+ Create Term" and "Categories". A search bar says "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 18 rows of license terms, each with a dropdown arrow icon at the end of the last column.

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small>Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never

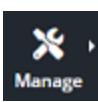
■ Using the **License Terms** tab for an individual license



The screenshot shows the Apache License 2.0 settings page. The title bar says "Apache License 2.0". Below it, it says "Family: Permissive | Status: Unreviewed". There are tabs: "Settings", "License Terms" (which is highlighted), and "Where Used". On the left, there's a sidebar with "Settings" and "License Terms". The "License Terms" section has a "Where Used" dropdown and a list of terms: "Private Use", "Place Warranty", "Modify", "Distribute", "Commercial Use", "Sub-License", and "Use Patent Claims". To the right, there are three groups of radio buttons: "Permitted" (selected), "Forbidden" (disabled), and "Required" (disabled). Under "Permitted", there are dropdown menus for each term, showing options like "Hold Liable", "Use Trademarks", etc.

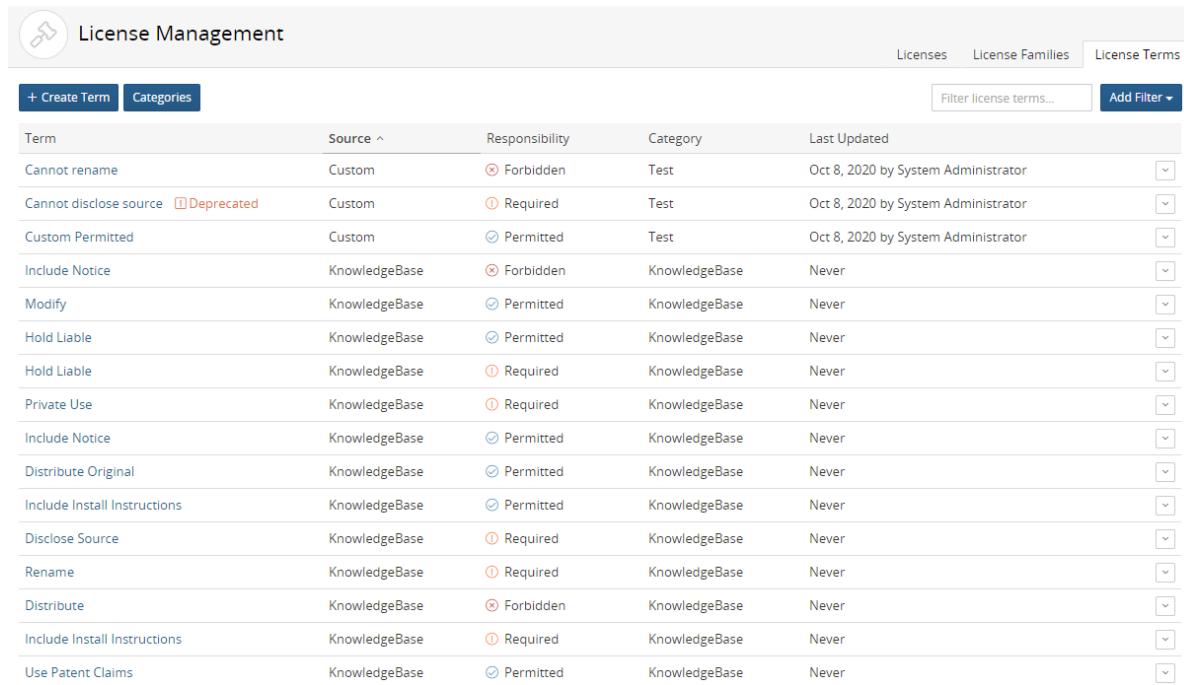
To restore a KnowledgeBase license term when viewing all terms

1. Log in to Black Duck with the License Manager [role](#).

2. Click  > 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 the "License Terms" tab selected. The table has columns: Term, Source, Responsibility, Category, and Last Updated. There are 18 rows of data, each representing a license term. The "Source" column includes entries like "Custom", "KnowledgeBase", and "Deprecated". The "Responsibility" column uses icons to indicate whether a term is "Forbidden" (red circle with a slash), "Required" (orange circle with a dot), or "Permitted" (blue circle with a dot). The "Category" column contains mostly "Test" and "KnowledgeBase". The "Last Updated" column shows dates like "Oct 8, 2020 by System Administrator". A search bar at the top right says "Filter license terms..." and an "Add Filter" button is also present.

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small>⚠ Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never

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 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 the 'License Association' dialog. 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 *'. A large list area contains a table with 'License' and 'Fulfillment Required' columns. The 'Apache License 2.0' row has a 'Restore' button next to it. At the bottom right of the dialog is a 'Close' button.

4. 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

This screenshot shows the 'License Association' dialog after step 4. The list of licenses is identical to the previous one, but the 'Restore' button next to the 'Apache License 2.0' row is no longer visible, indicating the action has been completed.

To restore a KnowledgeBase license term when viewing a license

1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **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', 'License Families', and 'License Terms'. Below the navigation is a search bar with filters for 'In Use' and 'Filter licenses...'. 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 selected or being viewed.

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 (containing the Apache License text). On the right, there are 'Created' and 'Updated' timestamps. A 'Save' button is at the bottom right.

4. Select the **License Terms** tab to view the terms associated with this tab.

Category	Term
Permitted	Private Use
	Place Warranty
	Modify
	Distribute
	Commercial Use
	Sub-License
	Use Patent Claims
Forbidden	Hold Liable
	Use Trademarks
	Required
Include Notice	
Include Copyright	
Include License	

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.

Category	Term
Permitted	Private Use
	Place Warranty
	Modify
	Distribute
	Commercial Use
	Sub-License
	Use Patent Claims
Forbidden	Hold Liable
	Use Trademarks
	Required
Include Notice	
Include Copyright	
Include License	

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 **Manage** > **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 and the title "License Management". Below the header is a search bar with filters for "Licenses", "License Families", and "License Terms". A "Create License" button is also visible. The main area is a table listing various licenses with columns for License Name, Components, License Family, Last Updated, User, Source, and Status. Some rows are collapsed, indicated by a minus sign icon.

Licenses	License Families	License Terms				
+ Create License	In Use	x				
	Filter licenses...	Add Filter ▾				
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 'License Management' interface for the 'Apache License 2.0'. At the top, it displays 'Family: Permissive | Status: Unreviewed'. Below this, there are two tabs: 'Settings' (selected) and 'License Terms'. The 'Settings' tab contains several input fields: 'Name' (Apache License 2.0), 'License Family' (Permissive), 'Status' (Unreviewed), 'Notes' (empty), 'Expiration Date' (with a calendar icon), and 'License Text' (containing the text 'Apache License Version 2.0, January 2004'). To the right of these fields, there are 'Created' and 'Updated' status indicators: 'Created never' and 'Updated 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.

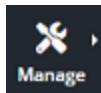
KnowledgeBase licenses can be [restored](#) to their original values.

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



> License Management.

The License Management page appears.

License Management							
		Licenses	License Families	License Terms			
+ Create License		In Use	Filter licenses...	Add Filter ▾			
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.

Viewing detected copyright statements

Black Duck can detect instances of copyright statements for a component. By enabling detection of copyright data when scanning code, users focused on license compliance can reduce license compliance risks by detecting and managing open source software and proprietary copyrights statements.

With this feature, Black Duck performs a search for copyright string text and displays the text found in the **Source** tab.

By displaying this information in the **Source** tab, you can easily find the files and directories that interest you and determine if copyright text is located there.

The screenshot shows the Black Duck interface with the 'Source' tab selected. On the left, a tree view of a project named 'copyright-license' shows various file and directory structures. The 'Tools' folder is currently selected. The main pane displays 'Discoveries' under 'License Searches' and 'Copyright Searches'. In 'License Searches', there are three hits for 'GNU General Public License Version 2' across 3 files. In 'Copyright Searches', there are seven files containing copyright statements: 'Copyright (C) 1998, 1999, 2000, 2001 Free Software Foundation, Inc.' (5 hits), 'Copyright (C) 2001, 2002, 2003 Free Software Foundation, Inc.' (1 hit), and 'Copyright 1997 Werner Koch (dd9jn)' (1 hit).

Black Duck groups the detected copyright statements into the **Copyright Searches** section.

For the copyright text found, Black Duck displays the number of:

- "Hits". The number of instances that copyright text was found in all files.
- Files where these "hits" were found.

In the example shown above, there were three instances of copyright text found in seven files.

Black Duck also lists the total number of files. Note that this value may not equal the total number of files shown for the copyright text as a file can have multiple different copyright statements.

Optionally, to help you review this information, upload your source files so that reviewers can view discovered copyright text from within the **Source** tab. When source files are uploaded, Black Duck provides a list of copyright statements. Select a copyright statement to highlight the text in the file. This can help reviewers evaluate the copyright text.

Discoveries

We found these discoveries in this file: 1 License, 1 License Reference, 1 Copyright

Licenses	Hits
GNU General Public License Version 2	1
License References	Hits
GNU General Public License	2
Copyrights	Hits
Copyright (C) 1998, 1999, 2000, 2001 Free Software Foundation, Inc.	1

```
file:///Users/sh/Downloads/Tutorial_Files/tools/bftest.c
1 /* bftest.c - Blowfish test program
2  * Copyright (C) 1998, 1999, 2000, 2001 Free Software Foundation, Inc.
3  *
4  * This file is part of GnuPG.
5  *
6  * GnuPG is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU General Public License as published by
8  * the Free Software Foundation; either version 2 of the License, or
9  * (at your option) any later version.
10 *
11 * GnuPG is distributed in the hope that it will be useful,
12 * but WITHOUT ANY WARRANTY; without even the implied warranty of
13 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
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 02111-1307, USA
19 */
20
21 #include <config.h>
22 #include <stdio.h>
23 #include <stdlib.h>
24 #include <string.h>
25 #ifdef HAVE_DOSISH_SYSTEM
mc_m...da...da...da...
```

Close

If you do not upload the source files, the Black Duck UI only displays the location of the discovered text in the file, by line number:

Discoveries

We found these discoveries in this file: 0 License, 0 License Reference, 1 Copyright

Copyrights	Hits
Copyright (C) 1998, 1999, 2000, 2001 Free Software Foundation, Inc.	1

No File to Display
You need to upload your files in order to display them

We found hits in these lines:
Hit 1: Line 2

Close

To include your source files, after your administrator has enabled source uploads, as described in the installation guide, include the upload source parameter when scanning.

Note: Regardless whether you upload your source files or not, copyright detection cannot be completed offline as it requires communication with the Black Duck server.

Supported file extensions/ file names

Copyright text search occurs in file extensions such as .bat or .js and for these file names, or file names that include the following text, regardless of case:

- bds!
- copying
- copyright
- control
- dad
- gpl
- install
- legal
- lGPL
- license
- licence
- licenses
- licences
- notice
- readme

Copyright detection process

The process to view copyright text is:

1. Enable detecting of copyright data when scanning and optionally, enable uploading source files for viewing copyright text within the file.
2. Review the copyright text.

Enable detecting of copyright data

All scanning methods have an option to enable copyright string search:

- Signature Scanner command line
- Synopsys Detect (Desktop)Synopsys Detect
- Synopsys Detect

Using the Signature Scanner command line

Use the **-copyright-search** parameter to enable copyright text search.

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

Using Synopsys Detect (Desktop) or Synopsys Detect

A property to enable copyright detection will be available in Synopsys Detect version 6.4 and later.

Reviewing copyright data

Black Duck displays the location of these copyright statements in your code tree.

To review copyright text

1. After enabling copyright text search, select the **Source** tab from your project version BOM page.
2. Select a folder in the code tree that you want to determine if there is copyright text.

Optionally, select **All Subfolders** to view information for all subfolders.

The table displays information in the table for the selected location. By default the **Files** option is selected.

Name	Component	Match Type	License	Usage	Discovery Types
bfest.c	GnuPG 0.2.1	Manually Identified	Unknown License	Dynamically Linked	Copyright, License Reference, License
clean-sat.c	GnuPG 0.2.1	Manually Identified	Unknown License	Dynamically Linked	Copyright
fileone.c					
filetwo.c					
gpgsplit.c					Copyright, License Reference, License
mk-tdata.c	GnuPG 0.2.1	Manually Identified	Unknown License	Dynamically Linked	Copyright
mpicalc.c	GnuPG 0.2.1	Manually Identified	Unknown License	Dynamically Linked	Copyright, License Reference, License
shmttest.c	GnuPG 0.2.1	Manually Identified	Unknown License	Dynamically Linked	Copyright

3. Select **Discoveries** to view the list of copyright text, shown in the **Copyright Searches** section.

- Select a copyright statement to view the **Source** tab filtered to display the files that contain the selected copyright text.

Name	Component	Match Type	License	Usage	Discovery Types
bfest.c					Copyright, License Reference, License
clean-sat.c					Copyright
mk-tdata.c					Copyright
mpicalc.c					Copyright, License Reference, License
shmtest.c					Copyright

Optionally, select a file name to view the location of the file in the code tree. If you uploaded your source files, the file contents appears on the page.

```

1 /* clean-sat.c
2 * Copyright (C) 1998, 1999, 2000, 2001 Free Software Foundation, Inc.
3 *
4 * This file is free software; as a special exception the author gives
5 * unlimited permission to copy and/or distribute it, with or without
6 * modifications, as long as this notice is preserved.
7 *
8 * This program is distributed in the hope that it will be useful, but
9 * WITHOUT ANY WARRANTY; to the extent permitted by law; without even the
10 * implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
11 */
12
13 #include <stdio.h>
14
15 int
16 main(int argc, char **argv)
17 {
18     int c;
19
20     if( argc > 1 ) {
21         fprintf(stderr, "no arguments, please\n");
22     }

```

- Select **Copyright** from the **Discovery Type** column to open the Discoveries dialog box.

```

1 /* bftest.c - Blowfish test program
2 * Copyright (C) 1998, 1999, 2000, 2001 Free Software Foundation, Inc.
3 *
4 * This file is part of GnuPG.
5 *
6 * GnuPG is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU General Public license as published by
8 * the Free Software Foundation; either version 2 of the License, or
9 * (at your option) any later version.
10 *
11 * GnuPG is distributed in the hope that it will be useful,
12 * but WITHOUT ANY WARRANTY; without even the implied warranty of
13 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
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 02111-1307, USA
19 */
20
21 #include <config.h>
22 #include <stdio.h>
23 #include <stdlib.h>
24 #include <string.h>
25 #ifdef HAVE_DOSISH_SYSTEM
26 #include <sys/types.h>

```

Close

The Discoveries dialog box shows all copyright text found for the selected file. If embedded licenses and license references were also found, that text is also shown.

The information that appears here depends on whether you uploaded source files.

In the example shown above, source files were uploaded in the scan.

6. Select the copyright text to view the highlighted text.

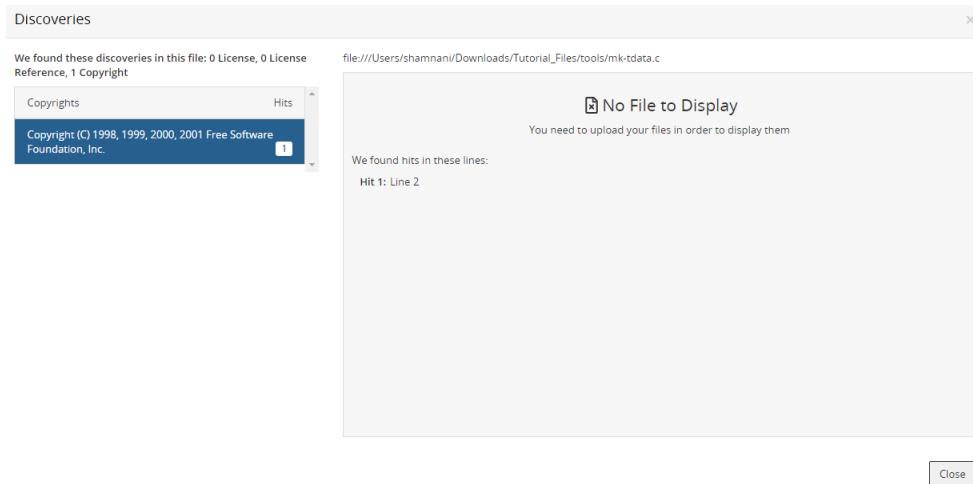
```

1 /* bftest.c - Blowfish test program
2 * Copyright (C) 1998, 1999, 2000, 2001 Free Software Foundation, Inc.
3 *
4 * This file is part of GnuPG.
5 *
6 * GnuPG is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU General Public License as published by
8 * the Free Software Foundation; either version 2 of the License, or
9 * (at your option) any later version.
10 *
11 * GnuPG is distributed in the hope that it will be useful,
12 * but WITHOUT ANY WARRANTY; without even the implied warranty of
13 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
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 02111-1307, USA
19 */
20
21 #include <config.h>
22 #include <stdio.h>
23 #include <stdlib.h>
24 #include <string.h>
25 #ifdef HAVE_DOSISH_SYSTEM
26 #include <sys/types.h>

```

Close

If you did not upload source files, the Discoveries dialog box displays the location of the discovered copyright text in the file, by line number:



Managing copyrights

Users with the Copyright Editor [role](#) can manage open source copyright statements for their organization. Using this feature makes it easier for you to include the full list of copyright holders for the open source components you use in your notices file report.

Users with the Copyright Editor role can:

- View all copyright statements for a component version.
- Create or edit custom copyright statements.
- Edit Black Duck KnowledgeBase copyright statements
- Revert an edited Black Duck KnowledgeBase copyright statement to its original text.
- Activate or deactivate copyright statements.

Black Duck manages copyright statements by the origin name/id for a component version. Therefore, edits made to copyright statements for an origin for a component version apply to all BOMs that use that component version origin. This enables you to reuse data across your organization and reduce your workload.

To manage copyright statements in Black Duck:

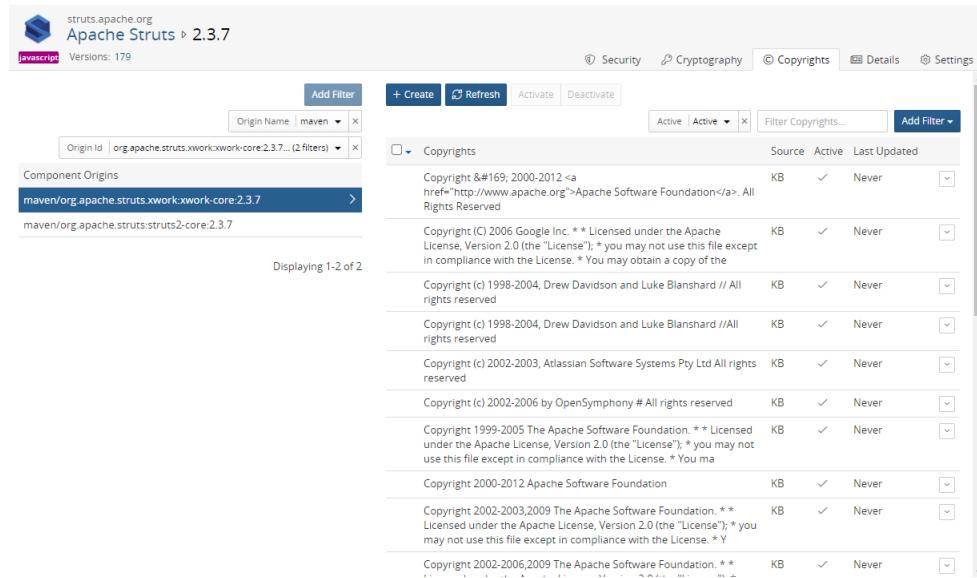
1. Review the existing Black Duck KnowledgeBase copyright statements.
2. If necessary, edit the existing KnowledgeBase copyright statements and/or create custom copyright statements.
3. Deactivate any copyright statements that do not apply.
4. Create the [Notices file report](#) and select the **Copyright Data** option to include copyright statements in your report.

Viewing and managing copyright statements

To view and manage the copyright statements, do one of the following:

- In the project version BOM, click  in the row of the component version you wish to view copyright statements and select **Copyrights**.

The component version **Copyrights** tab appears filtered to display the copyright statements for the origin used in your BOM.

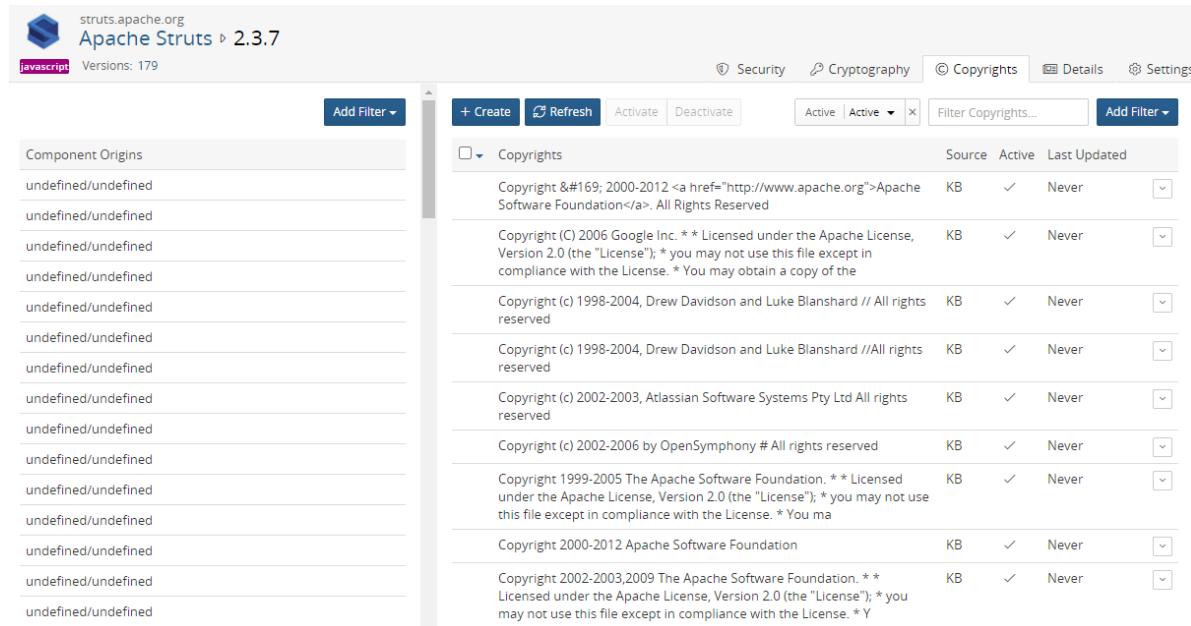


This screenshot shows the Apache Struts 2.3.7 project version BOM. The left sidebar lists component origins, including 'maven/org.apache.struts.xwork.xwork.core:2.3.7'. The right panel displays the 'Copyrights' tab, which is filtered for the selected origin. The table shows various copyright statements from different sources and years, such as Google Inc., Apache Software Foundation, and Atlassian Software Systems Pty Ltd. Each statement includes a link to the original document and details about its license status and last update.

Source	Active	Last Updated
Copyright © 2000-2012 Apache Software Foundation. All Rights Reserved	✓	Never
Copyright (C) 2006 Google Inc. ** Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License. * You may obtain a copy of the	✓	Never
Copyright (c) 1998-2004, Drew Davidson and Luke Blanshard // All rights reserved	✓	Never
Copyright (c) 1998-2004, Drew Davidson and Luke Blanshard //All rights reserved	✓	Never
Copyright (c) 2002-2003, Atlassian Software Systems Pty Ltd All rights reserved	✓	Never
Copyright (c) 2002-2006 by OpenSymphony # All rights reserved	✓	Never
Copyright 1999-2005 The Apache Software Foundation. ** Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License. * You ma	✓	Never
Copyright 2000-2012 Apache Software Foundation	✓	Never
Copyright 2002-2003,2009 The Apache Software Foundation. ** Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License. * Y	✓	Never
Copyright 2002-2006,2009 The Apache Software Foundation. **	✓	Never

Note if a component version is not defined in the BOM (as shown by (?) for the version), the **Copyrights** option is not available.

- Select to view an open source component version and select the **Copyrights** tab.



This screenshot shows the Apache Struts 2.3.7 project version BOM. The left sidebar lists component origins, all of which are currently undefined. The right panel displays the 'Copyrights' tab, which is filtered for the selected origin. The table shows various copyright statements from different sources and years, similar to the previous screenshot.

Source	Active	Last Updated
Copyright © 2000-2012 Apache Software Foundation. All Rights Reserved	✓	Never
Copyright (C) 2006 Google Inc. ** Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License. * You may obtain a copy of the	✓	Never
Copyright (c) 1998-2004, Drew Davidson and Luke Blanshard // All rights reserved	✓	Never
Copyright (c) 1998-2004, Drew Davidson and Luke Blanshard //All rights reserved	✓	Never
Copyright (c) 2002-2003, Atlassian Software Systems Pty Ltd All rights reserved	✓	Never
Copyright (c) 2002-2006 by OpenSymphony # All rights reserved	✓	Never
Copyright 1999-2005 The Apache Software Foundation. ** Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License. * You ma	✓	Never
Copyright 2000-2012 Apache Software Foundation	✓	Never
Copyright 2002-2003,2009 The Apache Software Foundation. ** Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License. * Y	✓	Never

The page is unfiltered and lists all origins for this component version.

Select an origin to view the copyright statements for that origin.

Use the component origin name and ID filters to limit the origins displayed on the page.

For each copyright statement, the following information appears:

Column	Description
Copyrights	Copyright text.
Source	Source for this copyright statement. Possible values are: <ul style="list-style-type: none"> KB. An unmodified, active copyright statement from the Black Duck KnowledgeBase. KB Modified. A copyright statement from the Black Duck KnowledgeBase that has been edited, deactivated, or reactivated. Custom. Copyright statement created by a user with the Copyright Editor role.
Active	One of the following icons appears: <ul style="list-style-type: none"> Active copyright statement which will appear in your Notices File report. Inactive copyright statement which will not appear in your Notices File report.
Last Updated	One of the following appears: <ul style="list-style-type: none"> Never indicates that the statement from the Black Duck KnowledgeBase has never been modified. Date and username. <ul style="list-style-type: none"> For Black Duck KnowledgeBase copyright statements, the date when this copyright statement was modified and the responsible user. A date and username also appears for Black Duck KnowledgeBase copyright statements that have been deactivated or reverted back to their original text. For custom copyright statements, the date when this statement was either created or last edited and the responsible user.

Creating custom copyright statements

To create a custom copyright statement

- As copyright statements are based by component origin, select the origin for this copyright statement from the **Component Origins** section.
- Click **Create**. The Create Copyright dialog box appears.



- Enter the copyright text and click **Save**.

Copyright statements are active by default. See below to deactivate this statement.

Editing custom copyright statements

To edit a custom copyright statement

1. In the row of the copyright statement you want to edit, select  and select **Edit**.

The Edit Copyright dialog box appears.



2. Edit the text and/or select or clear the **Active** option and click **Save**.

Deactivating copyright statements

By default, all copyright statements are active.

To deactivate a copyright statement

1. Do one of the following:

- Click  in the row of the copyright statement you wish to deactivate and select **Deactivate**.
- Select one or more checkboxes located to the left of the copyright statement and click **Deactivate**.

You can also deactivate a copyright statement when editing it.

Activating copyright statements

To activate copyright statements

1. Do one of the following:

- Click  in the row of the copyright statement you wish to activate and select **Activate**.
- Select one or more checkboxes located to the left of the copyright statement and click **Activate**.

You can also activate a copyright statement when editing it.

Editing KnowledgeBase copyright statements

You can modify an existing Black Duck KnowledgeBase copyright statement.

To edit a KnowledgeBase copyright statement

1. Click  in the row of the copyright statement you wish to edit and select **Edit**.



If this is the initial attempt to edit a KnowledgeBase copyright statement, the option to revert to the original text is not available.

2. Edit the text and/or clear or select the **Active** option and click **Save**.

Reverting KnowledgeBase copyright statements

If you edited a KnowledgeBase copyright statement, you can revert to the original text of the KnowledgeBase copyright statement.

To revert a KnowledgeBase copyright statement

1. Click  in the row of the copyright statement you wish to edit and select **Edit**.

The dialog box displays the edited text and the original copyright text from the KnowledgeBase.



Note: Reverted text may include poorly formatted and extraneous text not shown in the original copyright statement, which was edited to make it more readable.

2. Click **Revert to Original**.
3. Click **Save**.

Updating KnowledgeBase copyright statements

The Black Duck KnowledgeBase may have updated copyright information.

You can refresh the copyright statements for a component origin: if there is new or updated data, Black

Duck updates the information shown while keeping any edits that you made.

To update KnowledgeBase copyright statements for an origin

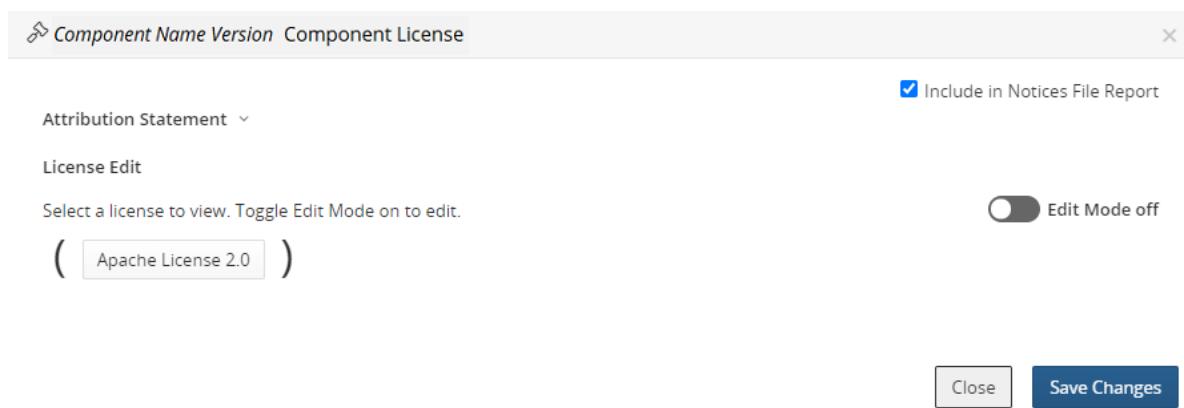
1. Open the **Copyrights** tab, as described previously.
2. Select a component origin.
3. Click **Refresh**.

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.

To add an attribution statement

1. Log in to Black Duck.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab and view the BOM.
4. Select the license to open the *Component/Subproject Name Version* Component License dialog box.



5. Click **>** to open the **Attribution Statement** field and enter the text.
Delete the text in this field to remove an attribution statement.
6. Click **Save Changes**.

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 log4j 1.2.15	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.

Copyright Data

Apache log4j 1.2.15

- Copyright 1999-2005 The Apache Software Foundation
- Copyright 2007 The Apache Software Foundation

Licenses

Apache License 2.0

Apache log4j 1.2.15

```
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.
```

About license conflicts

License terms are the provisions in the license which grant rights or impose restrictions on the use of the software under that license. They indicate the things you can do (permitted), cannot do (forbidden), and must do (required) to comply with the license.

License terms can be in conflict with each other because different licenses can have contradictory requirements. Although license terms for permitted actions cannot be in conflict with other license terms, forbidden or required license terms can be in conflict with each other.

Black Duck has identified those KnowledgeBase license terms that are in conflict with other

KnowledgeBase terms that have the same name but opposing responsibilities (a required license term that is incompatible with a forbidden license term).

You can now [manage conflicts](#) between open source component licenses that have terms which conflict with the project license. You can also [define incompatible terms](#) for your custom license terms.

Defining conflicts for customer license terms

Black Duck has identified those KnowledgeBase license terms that are in conflict with other KnowledgeBase terms that have the same name but opposing responsibilities.

You can define the custom license terms for forbidden or required actions that are in conflict with Black Duck KnowledgeBase terms or with your custom license terms.

Defining an incompatible term

You can define incompatible terms for your custom license terms with a forbidden or required responsibility, including deprecated custom license terms.

- A required license term can only be defined as incompatible to a forbidden license term.
- A forbidden license term can only be defined as incompatible to a required license term.

You cannot define incompatible terms for:

- Black Duck KnowledgeBase license terms
- Custom license terms with a permitted responsibility

To define an incompatible term

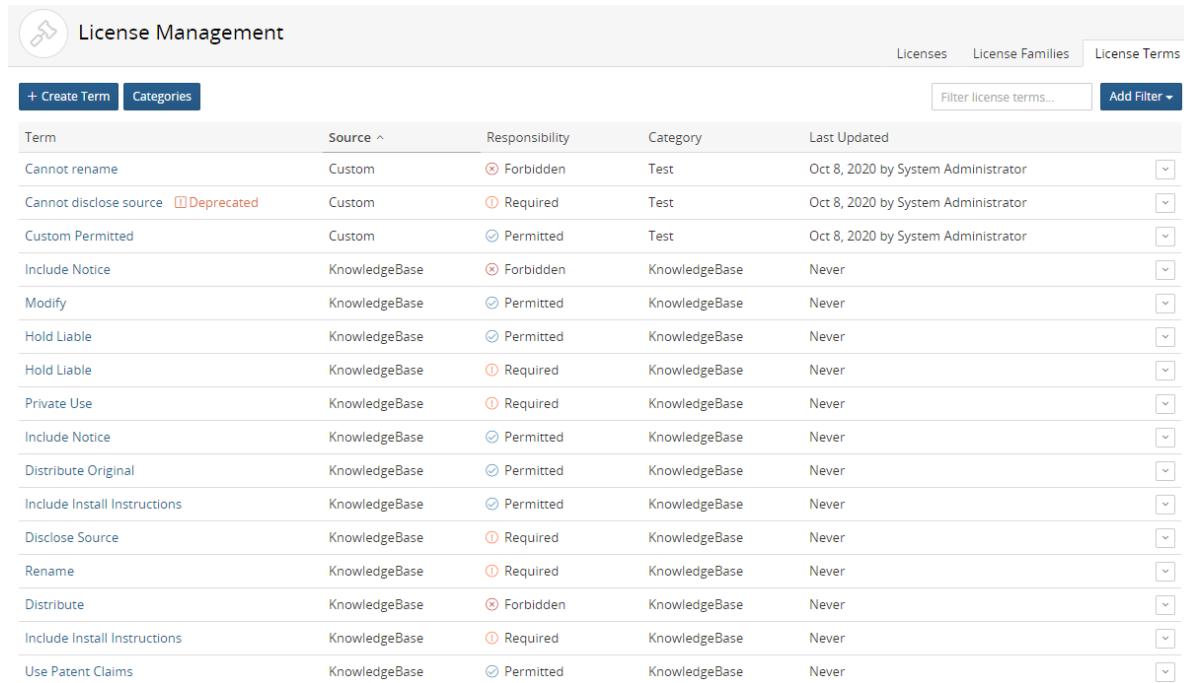
1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **License Management**.

The License Management page appears.

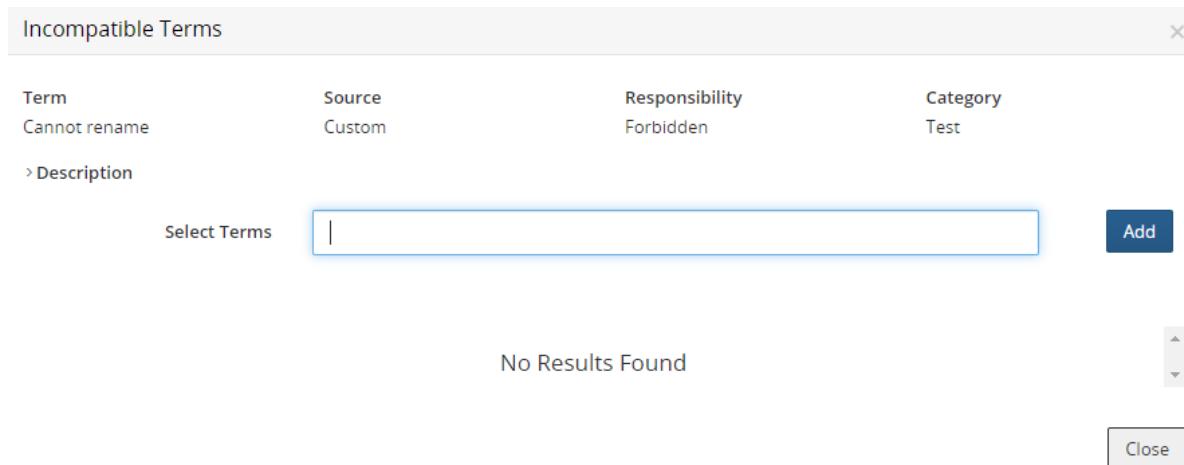
Select the **License Terms** tab to display all license terms.



The screenshot shows the Black Duck License Management interface. At the top, there's a navigation bar with icons for Home, License Management, Licenses, License Families, and License Terms. Below the navigation is a search bar labeled "Filter license terms..." and a "Add Filter" button. A toolbar at the top left includes "+ Create Term" and "Categories". The main area is a table with columns: Term, Source, Responsibility, Category, and Last Updated. The table lists various license terms like "Cannot rename", "Cannot disclose source", "Custom Permitted", etc., each with its source (Custom or KnowledgeBase), responsibility (Forbidden, Required, Permitted), category (Test, KnowledgeBase), and last update date.

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small>Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never

3. Click  in the row of the custom license term and select **Incompatible Terms** to open the Incompatible Terms dialog box.



The screenshot shows the "Incompatible Terms" dialog box. It displays a single row of incompatible terms: "Cannot rename" (Source: Custom, Responsibility: Forbidden, Category: Test). Below this row is a section titled "Description" with a link "› Description". At the bottom, there's a "Select Terms" input field containing a placeholder "Select Terms" and an "Add" button. A message "No Results Found" is displayed below the input field. A "Close" button is located at the bottom right of the dialog.

Term	Source	Responsibility	Category
Cannot rename	Custom	Forbidden	Test

> Description

Select Terms | Add

No Results Found

Close

4. Type the incompatible license term name in the **Select Terms** field.

Black Duck displays a list of terms that have the opposite responsibility as possible incompatible license terms; for example if you are defining conflicts for a forbidden license term, only required terms appear in the list.

Select a term and click **Add**.

5. Optionally, repeat step 4 to add additional incompatible license terms.

6. Click Close.

Viewing incompatible terms

1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small>Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never

3. Click in the row of the license term and select **Incompatible Terms** to open the Incompatible Terms dialog box which lists the incompatible terms for this license term.

Term	Source	Responsibility	Description
Include Notice	KnowledgeBase	Forbidden	You are required to include a notice in product documentation with certain requirements as specified by the license

Note that if a Black Duck KnowledgeBase license term does not have any incompatible license terms, the **Incompatible Terms** option is not available.

Tip: Use the **Has Incompatible Term(s)** filter to easily view all those license terms for which incompatible terms have been identified.

Deleting incompatible license terms

You cannot delete incompatible terms defined for Black Duck KnowledgeBase license terms. You can only delete incompatible terms that you have defined for your custom license terms.

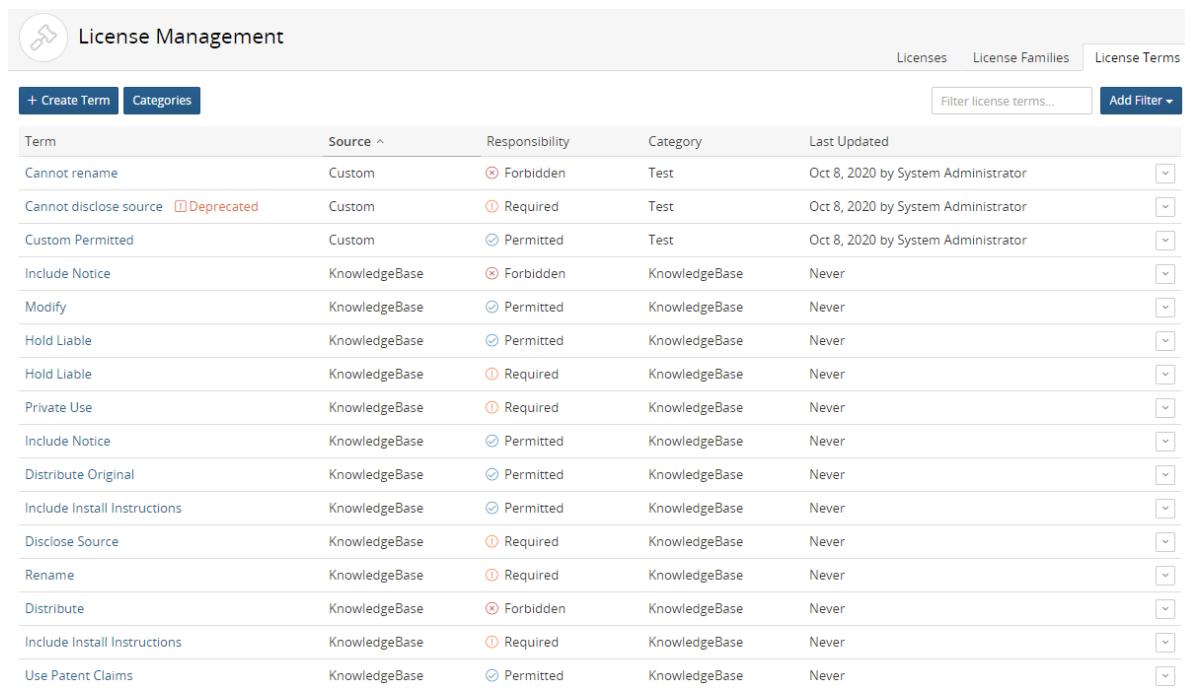
1. Log in to Black Duck with the License Manager [role](#).



2. Click **Manage** > **License Management**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.



A screenshot of the Black Duck License Management interface. The title bar says "License Management". Below it is a navigation bar with "Licenses", "License Families", and "License Terms" tabs, where "License Terms" is selected. There are two buttons: "+ Create Term" and "Categories". A search bar says "Filter license terms..." and an "Add Filter" button. The main area is a table with columns: Term, Source, Responsibility, Category, and Last Updated. The table lists various custom license terms like "Cannot rename", "Cannot disclose source", etc., with details such as "Custom" or "KnowledgeBase" as the source, and "Forbidden", "Required", or "Permitted" as the responsibility level. The last updated column shows dates like "Oct 8, 2020 by System Administrator".

Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	✗ Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source <small>Deprecated</small>	Custom	ⓘ Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	ⓘ Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Modify	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Hold Liable	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Private Use	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Include Notice	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Distribute Original	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never
Disclose Source	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Rename	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Distribute	KnowledgeBase	✗ Forbidden	KnowledgeBase	Never
Include Install Instructions	KnowledgeBase	ⓘ Required	KnowledgeBase	Never
Use Patent Claims	KnowledgeBase	ⓘ Permitted	KnowledgeBase	Never

3. Click  in the row of the custom license term and select **Incompatible Terms** to open the Incompatible Terms dialog box.

The screenshot shows a modal dialog titled "Incompatible Terms". It contains a table with one row:

Term	Source	Responsibility	Category
Cannot rename	Custom	Forbidden	Test

Below the table is a section labeled "Description" with the text: "If modified, you are required to rename the software to indicate it is not the original work". There is an "Add" button at the bottom right of the dialog.

4. Click in the row of the custom term that you want to remove.
5. Click **Remove** to confirm.

Enabling management of license term conflicts

BOM Managers, and other users with the appropriate [role](#), can view conflicts for a license term using the **License Conflicts** tab in the *Project Version's Legal* tab.

By default, these tabs are disabled. To enable this feature:

1. Use the System Setting page to enable the feature for all *future* projects.
2. Once future projects are enabled, use a project's **Setting** tab to enable the feature for a *current* project.

To enable license conflicts for future projects

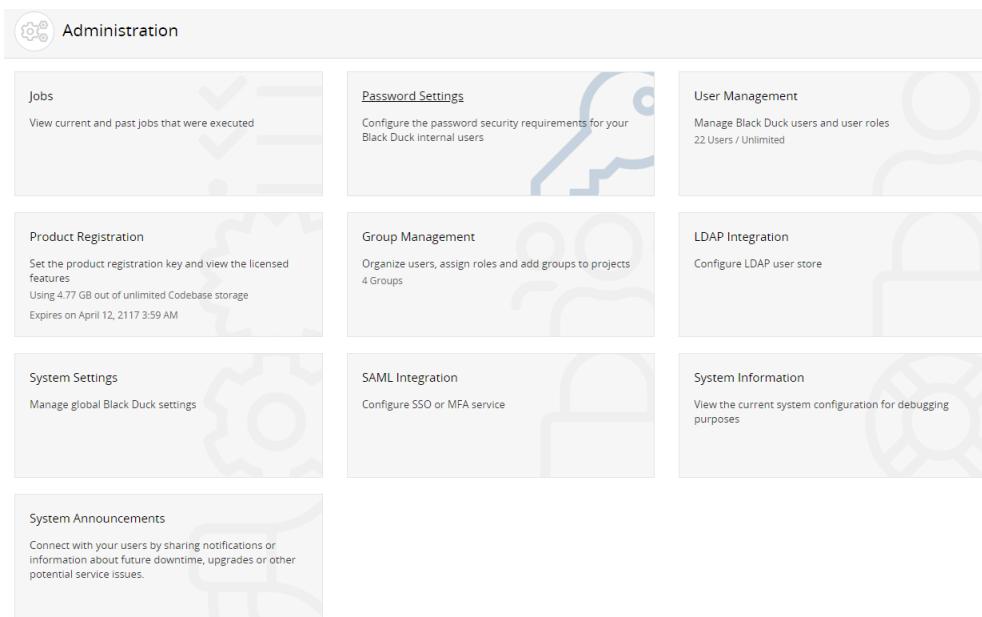
Use the System Settings page to enable the **Legal** and **License Conflicts** tabs for all future projects.

1. Log into Black Duck with the System Administrator role.



2. Click **Admin**.

The Administration page appears.

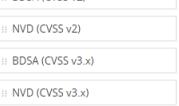


3. Select System Settings.

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.
[Disable](#)

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.
 [Save](#)

Custom Scan Signature Level
Depth, as measured in the number of levels in the directory structure, from root, to perform custom signature scanning. This is the default value for each project.
 [Save](#)

Snippet Max File Size
Enter a value from 1 to 16 to set the maximum file size in MB for snippet scanning. The default value is 2MB.
 [Save](#)

4. Click **Enable** located in the **License Conflicts** section to display the **Legal** and **License Conflicts** tabs.

Click **Disable** to remove the **Legal** and **License Conflicts** tabs. Note that if you select to enable license term fulfillment, the **Legal** tab will appear, but the **License Conflicts** tab will not appear.

Enabling or disabling license conflicts for a specific project

Once the system setting is enabled, you can enable this feature for current projects, or when necessary, disable the feature for a current project.

To enable or disable license term conflicts

1. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the **Settings** tab.

The screenshot displays the 'Settings' tab of the 'Sample Project 1' configuration page. It includes sections for Project Details (Members, Groups, Activity), Component Adjustments, Cloning, Custom Scan Signature, Deep License Data, License Conflicts, Additional Fields, Application ID, Clone Project, and Delete Project. Each section contains specific configuration options and checkboxes. A 'Save' button is located at the bottom of each section.

3. Do one of the following:

- Enable the **Apply license conflicts data to bill of materials** option in the **License Conflicts** section.
 - Clear the option to disable this feature for this project.
4. Click **Save**.

Managing project license conflicts

As a BOM reviewer, you need to understand when a component in your BOM has a license with terms that are incompatible with the declared license of a project. Black Duck identifies the specific license and term that is causing the incompatibility, thereby letting you manage this conflict and reducing the risk of license infringement.

Black Duck identifies the Black Duck KnowledgeBase conflicts (license terms that have the same name but [opposing responsibilities](#)) and the custom license terms that [you defined as in conflict](#) with Black Duck KnowledgeBase terms for a project version.

Note the following:

- License conflict information is not automatically enabled. System Administrators must [enable the Legal and License Conflicts tab](#) for all *future* project versions. Use the project's **Settings** tabs to enable the feature for *current* projects.
 - Note that Black Duck only determines license conflicts for component versions with high license risk. For the Black Duck license risk model, "high risk" means that licenses in this family tend to have license conflicts under this business scenario (combination of distribution type and component usage) making them incompatible. Medium or low risks means it may have risks if the business scenario changes (or is defined incorrectly) or due to other, non-license conflicts factors.
 - Black Duck calculates license risk during a scan or if you select to enable the feature for a current project.
- Manual edits to a BOM, including changing the usage for a component or the license of the project version using the **License Conflict** or **Components** tab will trigger a recalculation of the license conflict.
- License conflicts for snippets are not shown until the [snippet is confirmed](#).
 - You can [create a policy rule](#) that is triggered when a component's license conflicts with the license for a project version.

Viewing project license conflicts

1. From a project version BOM, select the **Legal** tab, and if necessary, select the **License Conflicts** tab to view a list of components that have a license that conflicts with the project license.

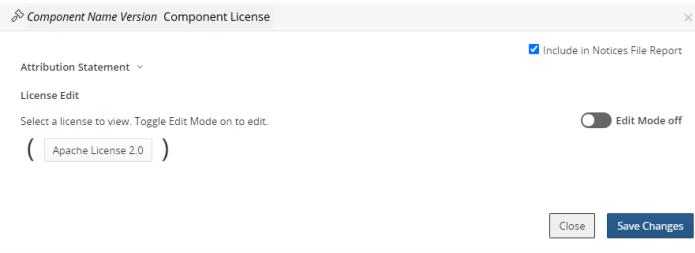
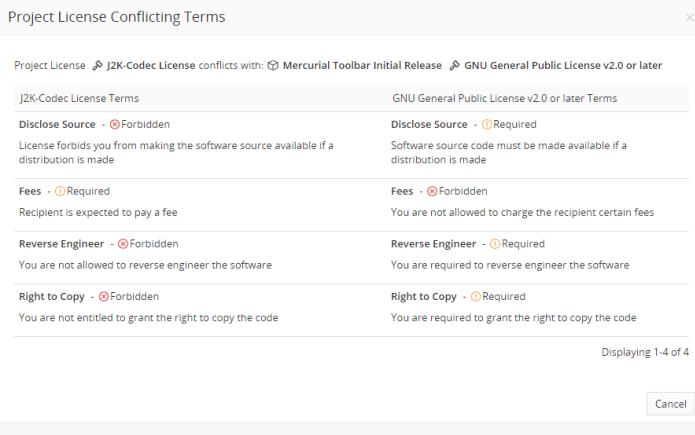
The screenshot shows the Black Duck Projects interface for the 'Tutorial_Files' project. The 'License Conflicts' tab is selected. A message at the top states: 'These are the component licenses in conflict with your project license.' Below this, a note says: 'J2K-Codec License conflicts with:'. A table lists the components and their details:

Component	Usage	License	Conflict
AMX Mod X - 1.8.1	Source Code	GNU General Public License v2.0 or later	Conflict
GnuPG - 1.0.5	Source Code	GNU General Public License v2.0 or later	Conflict
GnuPG - 1.3.5	Source Code	GNU General Public License v2.0 or later	Conflict
Mercurial Toolbar - Initial Release	Dynamically Linked	GNU General Public License v2.0 or later	Conflict
OurFaces - 20040830_2321	Source Code	Sun Public License v1.0	Conflict
petris - 1.11	Source Code	GNU General Public License v2.0 or later	Conflict

Displaying 1-6 of 6

The table displays the following information:

Column	Description
Ø	<u>Policy violation.</u> Hover over the icon to view the policy rule(s) this component violates. Select the icon to open the Policy Violations dialog box.
Component	Component name.
Usage	Indicates the <u>usage</u> of this component.

Column	Description
License	<p>The license for this component.</p> <p>Select the license name to open the <i>Component Name Version Component License</i> dialog box.</p>  <p>Use this dialog box to edit the existing license(s), view obligations, and view/edit the license text.</p>
Conflict	<p>Indicates there is a conflict. Select Conflict to open the Project License Conflicting Terms dialog box.</p>  <p>Use this dialog box to view the list of project license terms and conflicting component version license terms.</p>

2. Optionally, to edit the component, click  in the row of the component and select **Edit** to open the Edit Component dialog box.
3. Optionally, to add a comment, click  in the row of the component and select **Comment** to open the *Component/Subproject Name Version Comment* dialog box.

Enter the comment and click **Add Comment**. The comment appears for this component in the BOM.

Chapter 13: Running a report

Black Duck provides the following reports:

- [Notices File](#)
- [Project version](#). Detail and vulnerability reports.
- Global or project(s) reports:
 - [Vulnerability Remediation](#)
 - [Vulnerability Status](#)
 - [Vulnerability Update](#)

Tip: Reporting schemas in the PostgreSQL database provide access to Black Duck data for reporting purposes. See the Report Database guide which contains information on using the report database.

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](#).*

Notices File report

The Notices File report provides a list of open source components, versions, the associated license text, and optionally, copyright statements. 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, including [deep license data](#).
You can [exclude a component or subproject](#) or add an [attribution statement](#),
- Licenses. Provides the license text for all licenses listed in the **Components** section.

You can [edit the license text](#) shown here.

- Copyright data. Provides a report section that contains the copyright statements obtained from the Black Duck KnowledgeBase, edited KnowledgeBase copyright statements, and/or custom copyright statements for the open source components you use.

User with the Copyright Editor role can [create or edit](#) copyright statements for an open source component version origin.

This feature is optional.

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 log4j 1.2.15	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.

Copyright Data

Apache log4j 1.2.15

- Copyright 1999-2005 The Apache Software Foundation
- Copyright 2007 The Apache Software Foundation

Licenses

Apache License 2.0

Apache log4j 1.2.15

```
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.
```

Note that licenses from the [Unknown license family](#) are not included in the Notices File report, however the component with the unknown license is included in the report unless you select to remove it.

Note: If you notice omissions or errors in the license text, contact Synopsys Support and provide the correct information so that the Black Duck KnowledgeBase can be updated.

To run a Notices File report

1. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the version of the project for which you want to run the report.
3. Select the **Reports** tab.
4. Click **+ Create Notices File** and select the format for the report:
 - Text. This is the default format for the report.
 - HTML.

5. Optionally, select whether to include copyright data.

The Notices File report may take more time to run if this option is selected.

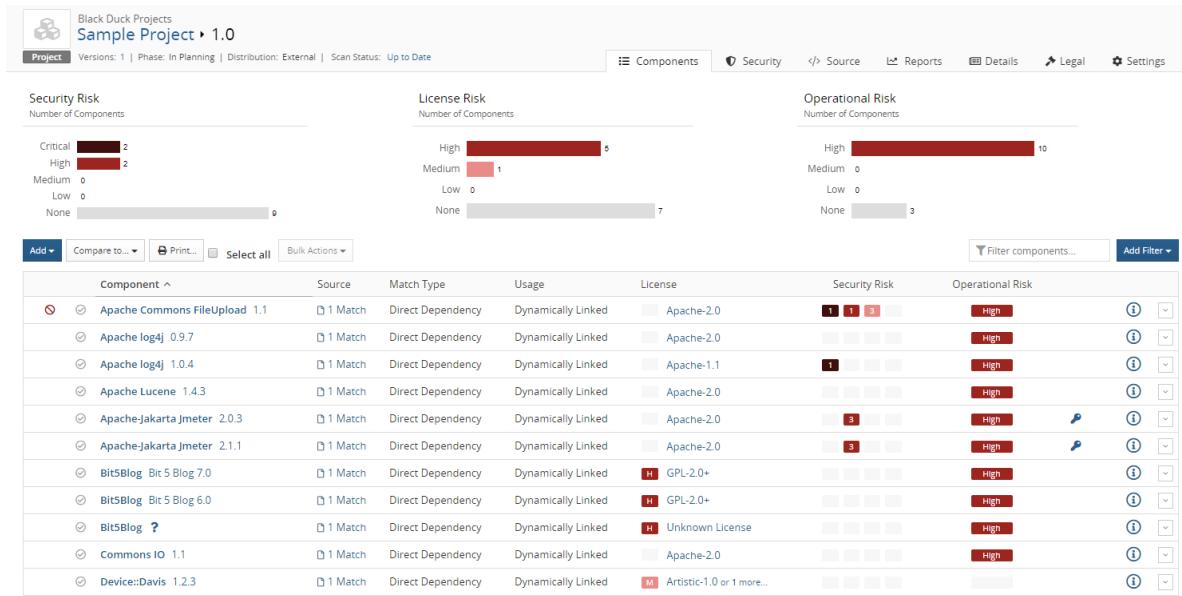
6. Click **Create** to run the report.
7. A link that includes the project, version name, and date appears when the report completes. Any user who is a member of the project can access the link.
 - If you selected the text format, download the report and extract the zip file locally.
 - If you selected the HTML format, select the link to open the report in a new tab.

Excluding a component or subproject from the Notices File report

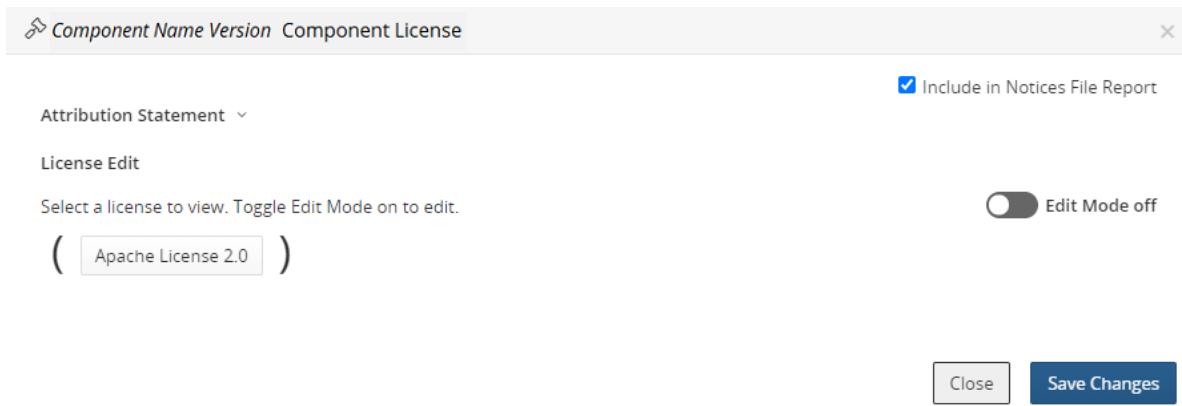
By default, all components and subprojects are included in the [Notices File report](#).

To exclude a component in the Notices File report

1. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the version name to display the **Components** tab and view the BOM.



3. Select the existing license from the **License** column to open the *Component/Subproject Name Version* Component License dialog box.



4. 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.

5. Click **Save Changes**

Project version reports

Use project version reports to export and share the content of a single project version. You can run detail reports or a vulnerability report.

Detail reports

Depending on the categories you select, running a project version report creates these comma-separated

files:

- `bom_component_custom_fields_date_time.csv` lists the same information as the `components_date_time.csv` report, but also includes BOM component, component, and component version custom field labels and the values selected for this project version.
- `components_date_time.csv` lists each component in the project version, including the respective licensing, usage, match type, operation risk information, policy violation information, and review status.
- `crypto_date_time.csv` lists the cryptography information for each component in the project version, including the algorithm ID, algorithm name, key length type, and key length.
- `license_conflicts_date_time.csv` lists the license conflicts for this project version.
- `license_term_fulfillment_date_time.csv` lists the license terms and fulfillment status for this project version.
- `project_version_custom_fields_date_time.csv` lists the project version custom field labels and the values selected for this project version.
- `project_version_upgrade_guidance_date_time.csv` lists the upgrade guidance information for all components for this project version.

As Black Duck caches this data, the information shown in this report may lag the Black Duck KnowledgeBase up to 24 hours.

- `scans_date_time.csv` lists the mapped scans.
- `security_date_time.csv` lists the security risk associated with each component, including the vulnerability ID and description, vulnerability scores, and remediation information.
- `source_date_time.csv` lists the individual files and dependencies associated with each component, including match type, usage information, and policy violation information.
- `version_date_time.csv` lists the name and details of the project version, including the release date, phase, method of release, and policy violation information.
- `vulnerability_matches_date_time.csv` lists the component, vulnerability data, and [vulnerability impact analysis](#) data (called function, qualified name, and line number) for each component potentially reached by a vulnerability.

This report is empty if there are no components that are potentially reachable.

For these project version reports:

- The archive file name is <ProjectName-ProjectVersion>_<YYYY-MM-DD>_<HHMMSS>.zip (time stamp in system timezone).
- 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 (_).

To run a project version detail report

1. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the version of the project for which you want to run the report.
3. Select the **Reports** tab.
4. Click **Create > Create Version Detail Report** and select the categories you would like to include in the report:
 - Component
 - Component Additional Fields
 - Cryptography
 - License Conflicts
 - License Terms
 - Project Version Additional Fields
 - Scans
 - Source
 - Upgrade Guidance
 - Version Details
 - Vulnerabilities
 - Vulnerability Matches
5. 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.

6. Download the report and extract the zip locally.

Vulnerability reports

You can create a [vulnerability remediation report](#), [vulnerability status report](#), or [vulnerability update report](#) for a specific project version.

To run a vulnerability report at the project version level

1. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the version of the project for which you want to run the report.
3. Select the **Reports** tab.
4. Click **Create > Create Vulnerability Report**.
5. Select one of the following from the **Report Type** list:

- Vulnerability Remediation Report
 - Vulnerability Status Report
 - Vulnerability Update Report
6. 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.
7. Select dates for the Vulnerability Remediation and Vulnerability Update reports.
- For the Vulnerability Remediation report, the date represents the day when the vulnerability was published.
 - For the Vulnerability Update 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.
8. Optionally, for the Vulnerability Remediation Report, select one or more remediation statuses.
9. Click **Confirm** to run the report.

One of the following links appears when the report completes:

- vulnerability-remediation-report_<ProjectName>-<VersionName>_<YYYY-MM-DD>_<HHMMSS> (time stamp in system timezone)
- vulnerability-status-report_<ProjectName>-<VersionName>_<YYYY-MM-DD>_<HHMMSS> (time stamp in system timezone)
- vulnerability-update-report_<ProjectName>-<VersionName>_<YYYY-MM-DD>_<HHMMSS> (time stamp in system timezone)

10. Select the link to view the report.

If you selected CSV as the report format, download the report and extract the zip file.

For reports in CSV format:

- The archive file name is <ReportName>-<ProjectName>-<ProjectVersion>_<YYYY-MM-DD>_<HHMMSS>.zip (time stamp in system timezone).
- The directory and filename are <ReportName>-<ProjectName>-<ProjectVersion>_<YYYY-MM-DD>_<HHMMSS>/<ReportName>-<ProjectName>-<ProjectVersion>_<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 (_).

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.

This report can be run at the global level (for all projects to which you have access) or for one or more

projects to which you have access. It can also be run at the [project version level](#) to view this information for a specific project version.

To run a Vulnerability Remediation report at the global or project level

1. Log in to Black Duck.



2. Click **Reports**.

3. Click **+ Create new report**. The Create New Report dialog box appears.

4. Select **Vulnerability Remediation Report** from the **Report Type** list.

5. To run the report for selected projects, enter one or more project names in the **Projects** field. Leave the field blank to create the report for all projects to which you have access.

6. 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.

7. 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.

8. Optionally, select one or more remediation statuses.

9. Click **Confirm** to run the report.

One of the following links appears when the report completes:

- vulnerability-remediation-report_all_assigned_projects_YYYY-MM-DD_HHMMSS (time stamp in system timezone) for a global version of the report
- vulnerability-remediation-report_YYYY-MM-DD_HHMMSS (time stamp in system timezone) for one or more projects

Reports for a specific project can be accessed by any user who is a member of the project. However, if the report contains multiple projects, the user must be a member of all projects to access the report.

10. 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.

This report can be run at the global level (for all projects to which you have access) or for one or more projects to which you have access. It can also be run at the [project version level](#) to view this information for a specific project version.

To run a Vulnerability Status report at the global or project level

1. Log in to Black Duck.



2. Click **Reports**.
3. Click **+ Create new report**. The Create New Report dialog box appears.
4. Select **Vulnerability Status Report** from the **Report Type** list.
5. To run the report for selected projects, enter one or more project names in the **Projects** field. Leave the field blank to create the report for all projects to which you have access.
6. 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.

7. Click **Confirm** to run the report.

One of the following links appear when the report completes:

- vulnerability-status-report_all_assigned_projects_YYYY-MM-DD_HHMMSS (time stamp in system timezone) for a global version of the report
- vulnerability-status-report_YYYY-MM-DD_HHMMSS (time stamp in system timezone) for one or more projects

Reports for a specific project can be accessed by any user who is a member of the project. However, if the report contains multiple projects, the user must be a member of all projects to access the report.

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 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.

This report can be run at the global level (for all projects to which you have access) or for one or more projects to which you have access. It can also be run at the [project version level](#) to view this information for a specific project version.

To run a Vulnerability Update report at the global or project level

1. Log in to Black Duck.



2. Click **Reports**.

3. Click **+ Create new report**. The Create New Report dialog box appears.

4. Select **Vulnerability Update Report** from the **Report Type** list.

5. To run the report for selected projects, enter one or more project names in the **Projects** field. Leave the field blank to create the report for all projects to which you have access.

6. 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.

7. 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.

8. Click **Confirm** to run the report.

One of the following links appear when the report completes:

- `vulnerability-update-report_all_assigned_projects_YYYY-MM-DD_HHMMSS` (time stamp in system timezone) for a global version of the report
- `vulnerability-update-report_YYYY-MM-DD_HHMMSS` (time stamp in system timezone) for one or more projects

Reports for a specific project can be accessed by any user who is a member of the project. However, if the report contains multiple projects, the user must be a member of all projects to access the report.

9. 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.

Note that you can also create external user accounts.

Users with the System Administrator or Super User role can also configure the [password requirements](#) for user accounts.

Configuring password requirements

Users with the System Administrator role can set password requirements for *local* Black Duck accounts. If enabled, Black Duck ensures that the new password meets your requirements and also rejects passwords that are considered weak, such as "password", "blackduck", or a user's username or email address.

Note: These requirements do not apply to external (LDAP or SAML) accounts.

System Administrators can:

- define the minimum password length (from 8 to 25 characters). The maximum length is 128 characters.

- define the minimum number of character types for the password (from one to four character types). Possible character types are lowercase letters, uppercase letters, numbers, or special characters.
- select whether to enforce the password requirements on current users when they log in to Black Duck.

If you select this option, current users who try to log in with a password that does not meet the requirements will be forced to create a new password before they can access the system.

Note that when using the Black Duck APIs, users with a password that does not meet your requirements will receive a 412 response code which will include the reason why the current password does not meet requirements.

If password requirements are enabled, all new passwords must satisfy the requirements. Password requirements are still enforced on current users when they attempt to change their password.

Administrators must also create passwords that meet these requirements when resetting a current user's password or when they make any changes to a user's detail information (such as their first name).

By default, password requirements are enabled and have these settings:

- The minimum password length is eight characters.
- Only one character type is required.
- Password requirements are not enforced on current users when logging in to Black Duck.

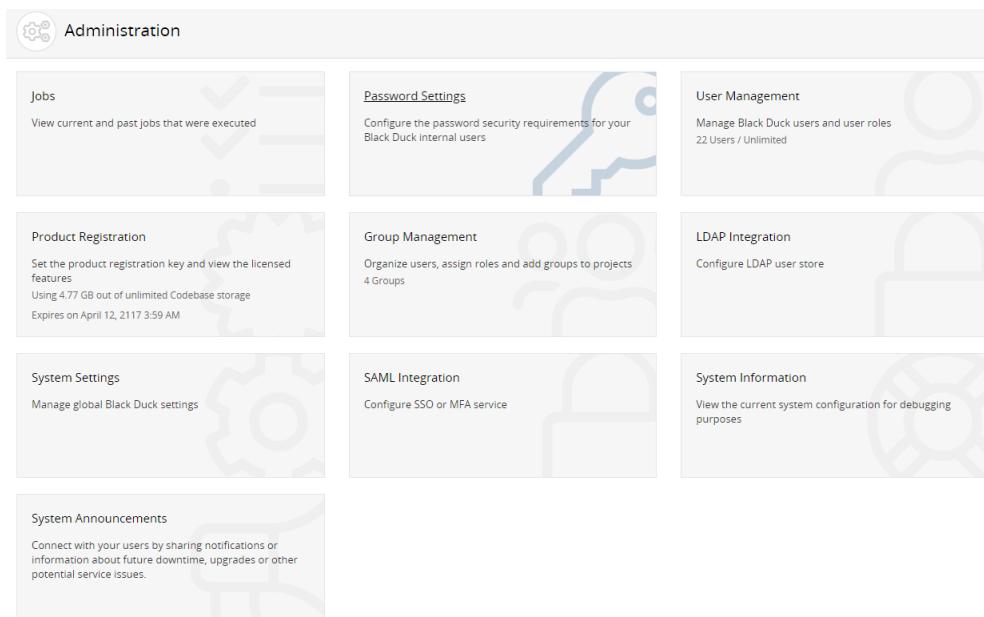
To manage password requirements

1. Log in to Black Duck with the System Administrator role.



2. Click **Admin**.

The Administration page appears.



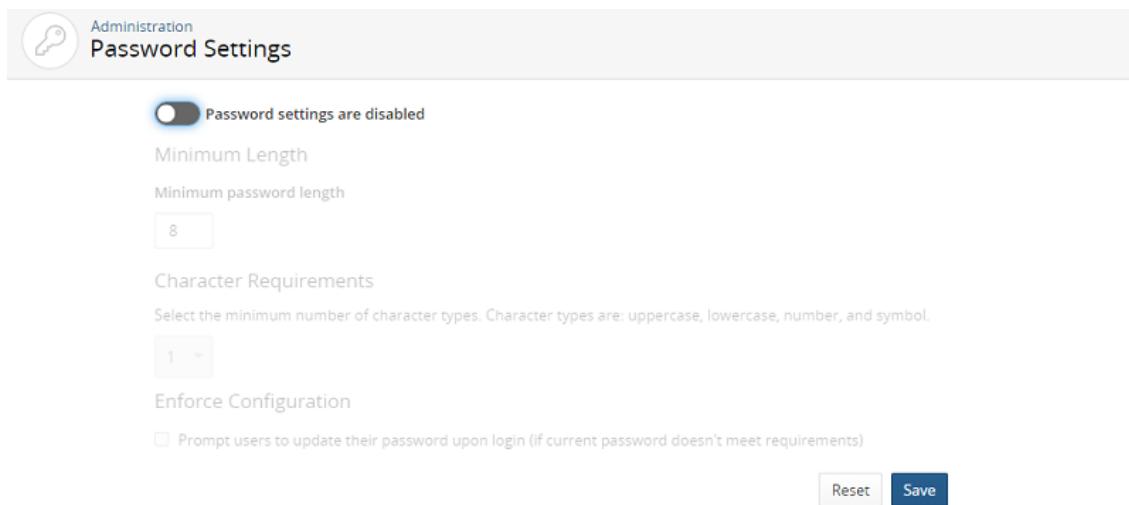
3. Select Password Settings.

The Password Settings page appears.

The screenshot shows the "Password Settings" configuration page. At the top left is a key icon labeled "Administration" and "Password Settings". A toggle switch is set to "On" with the label "Password settings are enabled". Under "Character Requirements", a dropdown menu is set to "1". There is also an unchecked checkbox for "Enforce Configuration" and a note about prompting users to update their password if it doesn't meet requirements. At the bottom are "Reset" and "Save" buttons.

4. Select to enable or disable password settings.

- To disable, select the password settings option so that **Password settings are disabled** appears on the page.



- To enable, ensure that the **Password settings are enabled** option is enabled: if **Password settings are disabled** appears on the page, select the option to enable settings.
5. If you enabled password settings:
- a. Select the following:
 - **Minimum length.** Minimum number of characters in the password.
 - **Character requirements.** Select the minimum number of character types.
For example, if you select the value 2, passwords must include at least two of the following: lowercase letters, uppercase letters, numbers, or special characters.
 - **Enforce configuration.** Select this option to enforce the password requirements on your current users when logging in to Black Duck.
 - b. Optionally, click **Reset** to undo your current edits and display the previous values.
 - c. Click **Save**.

Creating a user account

You can create a Black Duck user account for a local user (an internal user account) for an external user (such as a user managed by an external source, such as LDAP).

If you have enabled LDAP, you can 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.

Note that with external user accounts:

- You can create users and assign roles without the user logging in to Black Duck.
- User information, such as the first or last name, can be changed in Black Duck, however passwords

are not managed by Black Duck.

- The first name, last name, and email address of the external user will be overridden with the information present on the external server, (such as an LDAP server), at the time of login.
- An external user is only created when an administrator configures either SAML or LDAP in Black Duck. If both SAML and LDAP are enabled, or *both* are disabled, the external user will not be created.

To create a user account

1. Log in to Black Duck.



2. Click **Admin**.

The Administration page appears.

The screenshot shows the Black Duck Administration interface. At the top left is the "Administration" logo. Below it are several cards:

- Jobs**: View current and past jobs that were executed.
- Password Settings**: Configure the password security requirements for your Black Duck internal users.
- User Management**: Manage Black Duck users and user roles. (22 Users / Unlimited)
- Product Registration**: Set the product registration key and view the licensed features. (Using 4.77 GB out of unlimited Codebase storage, Expires on April 12, 2017 3:59 AM)
- Group Management**: Organize users, assign roles and add groups to projects. (4 Groups)
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- System Announcements**: Connect with your users by sharing notifications or information about future downtime, upgrades or other potential service issues.

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

The screenshot shows the User Management page. At the top left is the "User Management" logo. Below it is a search bar with filters for "User Status: Active" and "Add Filter". A table lists users:

Username	First Name	Last Name	Email	Roles	Status
sysadmin			noreply@blackducksoftware.com	BOM Manager, Code Scanner, License Manager, Policy Manager, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23	test123@bds.com		Active

At the bottom right, it says "Displaying 1-3 of 3".

4. Click **+ Create User**. The Create a New User dialog box appears.

The screenshot shows a 'Create a New User' dialog box. At the top left is the title 'Create a New User' and a close button 'X'. Below it is a 'Type' section with two radio buttons: 'Internal' (selected) and 'External (LDAP, SAML)'. A 'User Name *' field contains the placeholder 'l'. Below it are 'First Name *' and 'Last Name *' fields, both empty. An 'Email' field is also empty. Underneath these fields is a section titled 'Passwords must:' with three options: 'Contain between 8 and 128 characters', 'Be difficult to guess', and another option partially visible. Below this are 'Password *' and 'Confirm Password *' fields, both empty. A checked checkbox labeled 'Active user' is present. At the bottom right are 'Cancel' and 'Create' buttons, with 'Create' being blue.

5. Select whether this user is an internal (managed within Black Duck) or external (managed by LDAP, SAML) account.
6. Do one of the following:
 - For an internal user, enter the following information
 - Username.
 - First Name.
 - Last Name.
 - Email. This field is optional.
 - Password.If there are [password requirements](#), those requirements are listed in this dialog box. Black Duck notes when each requirement is met. You will not be able to create the user account unless the password meets *all* requirements.
 - Confirm password: This must match the password you entered.
 - For an external user, enter the following information:
 - Username.
 - First Name.
 - Last Name.
 - Email. This field is optional.Note that the passwords for external accounts are managed by the external source such as LDAP, not by Black Duck.
7. Select whether this user is active or inactive. Clearing this check box inactivates this user.

8. Click **Create**.

Black Duck creates the user account with the password you specified.

After creating a user, you can:

- [assign roles to this user](#)
- [assign groups to this user](#)
- [add this user to a project team](#)

If you [created default groups](#), this user is automatically added to the default group and is granted all roles and access to all projects configured for that group.

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.

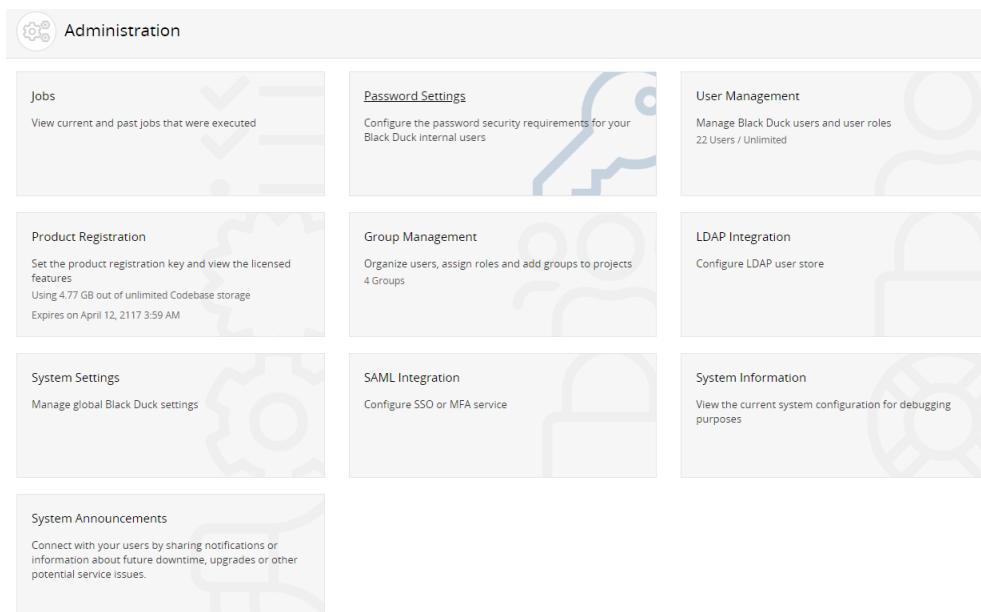
To disable a user account

1. Log in to Black Duck.



2. Click **Admin**.

The Administration page appears.



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

- Jobs**: View current and past jobs that were executed.
- Password Settings**: Configure the password security requirements for your Black Duck internal users.
- User Management**: Manage Black Duck users and user roles (22 Users / Unlimited).
- Product Registration**: Set the product registration key and view the licensed features (Using 4.77 GB out of unlimited Codebase storage, Expires on April 12, 2117 3:59 AM).
- Group Management**: Organize users, assign roles and add groups to projects (4 Groups).
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- System Announcements**: Connect with your users by sharing notifications or information about future downtime, upgrades or other potential service issues.

3. Select User Management to display the User Management page.

Username	First Name	Last Name	Email	Roles	Status
sysadmin			noreply@blackducksoftware.com	BOM Manager, Code Scanner, License Manager, Policy Manager, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23	test123@bds.com		Active

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.

User Details

Username *	SampleUser
First Name *	Sample
Last Name *	User
Email *	username@company.com

Active user Active user

Save

Roles

<input type="checkbox"/> BOM Manager BOM manager is granted the privilege of editing BOM
<input type="checkbox"/> Code Scanner Manages code-related scans.
<input type="checkbox"/> Policy Manager Policy Manager is granted the privilege of editing policy rules.
<input checked="" type="checkbox"/> System Administrator System Administrator are granted the privilege of editing users and other system settings.

User Groups

Group Name	Source	Status	Roles
Sample Group	Internal	Active	Policy Manager

6. Clear the Active user check box in the Internal or External User Details section and click Save.

Converting a user account

You can convert an internal account to an external account or an external account to an internal account.

Note: Converting an internal user account to an external user account requires that an administrator has configured *either* SAML or LDAP in Black Duck. If *both* SAML and LDAP are enabled, or both are disabled, you will be unable to convert the internal user account to an external user account.

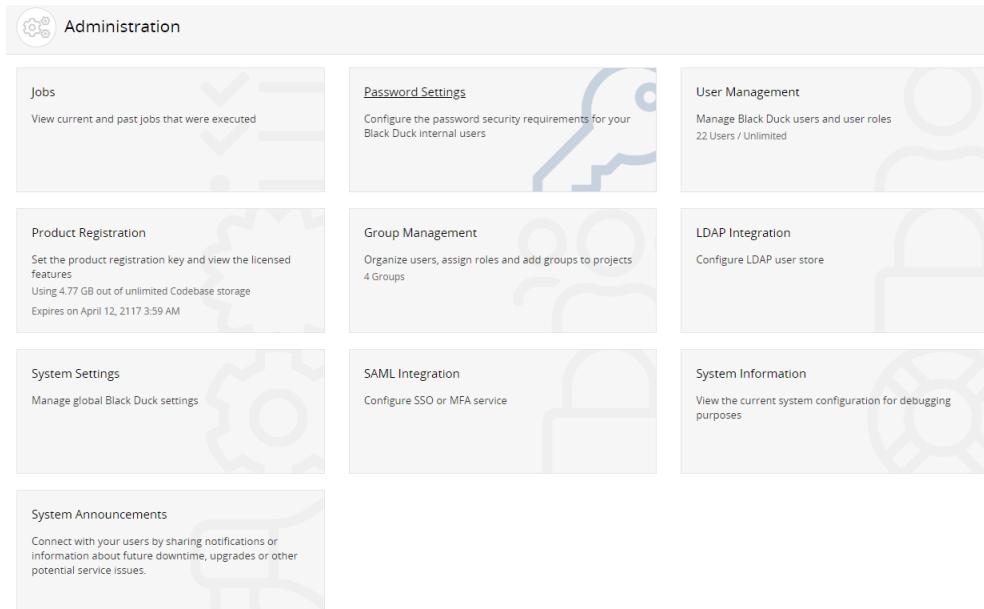
To convert an account

1. Log in to Black Duck.

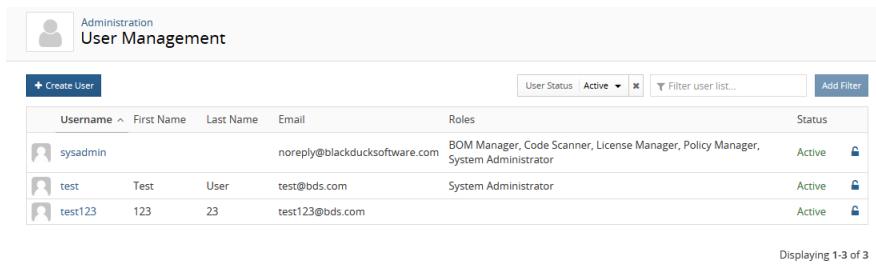


2. Click **Admin**.

The Administration page appears.

A screenshot of the Black Duck Administration page. The top navigation bar says "Administration". Below it are several cards: "Jobs" (View current and past jobs that were executed), "Password Settings" (Configure the password security requirements for your Black Duck internal users), "User Management" (Manage Black Duck users and user roles, 22 Users / Unlimited), "Product Registration" (Set the product registration key and view the licensed features, Using 4.77 GB out of unlimited Codebase storage, Expires on April 12, 2117 3:59 AM), "Group Management" (Organize users, assign roles and add groups to projects, 4 Groups), "LDAP Integration" (Configure LDAP user store), "System Settings" (Manage global Black Duck settings), "SAML Integration" (Configure SSO or MFA service), and "System Information" (View the current system configuration for debugging purposes). A "System Announcements" card at the bottom provides information about connecting with users via notifications.

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

A screenshot of the User Management page under the Administration section. The top navigation bar says "User Management". Below it is a table showing user details:Username, First Name, Last Name, Email, Roles, and Status. There are three rows:

Username	First Name	Last Name	Email	Roles	Status
sysadmin			noreply@blackducksoftware.com	BOM Manager, Code Scanner, License Manager, Policy Manager, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23	test123@bds.com		Active

At the bottom right, it says "Displaying 1-3 of 3".

4. Select the username of the account you wish to convert. The *Username* page appears.

The screenshot shows the 'User Management / User Details' page for a user named 'Sample User'. The 'User Details' section contains fields for Username ('SampleUser'), First Name ('Sample'), Last Name ('User'), and Email ('username@company.com'). Below these fields are two checkboxes: 'Active user' (unchecked) and 'Inactive user' (checked). A 'Save' button is located to the right of the email field. The 'Roles' section lists several roles with checkboxes: 'BOM Manager' (unchecked), 'Code Scanner' (unchecked), 'Policy Manager' (unchecked), and 'System Administrator' (checked). The 'User Groups' section shows a single group named 'Sample Group' with 'Internal' source and 'Active' status, assigned the 'Policy Manager' role. A 'Delete' icon is shown next to the group entry.

Depending on whether the account you selected is an internal or external account, do one of the following:

- To convert an existing external account to an internal account, click **Convert to Internal Account (Black Duck)**.
 - To convert an existing internal account to an external account, click **Convert to External Account (LDAP, SAML)**.
5. Do one of the following:
- To convert from an external account to an internal account, enter the following information:
 - Username. Enter a username.
 - First Name. The existing first name is shown. You can retain the existing name or enter a new first name.
 - Last Name. The existing last name is shown. You can retain the existing name or enter a new last name
 - Email. This field is optional.
 - Password
 - Confirm password: This must match the password you entered. Black Duck validates this when you create the user account.
 - To convert an internal account to an external account, enter the following information:
 - Username. Enter a username.
 - First Name. The existing first name is shown. You can retain the existing name or enter a new first name.

- Last Name. The existing first name is shown. You can retain the existing name or enter a new first name.
- Email. This field is optional.

Note that the passwords for external accounts are managed by LDAP, not by Black Duck.

6. Select whether this user is active or inactive. Clearing this check box inactivates this user.
7. 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.

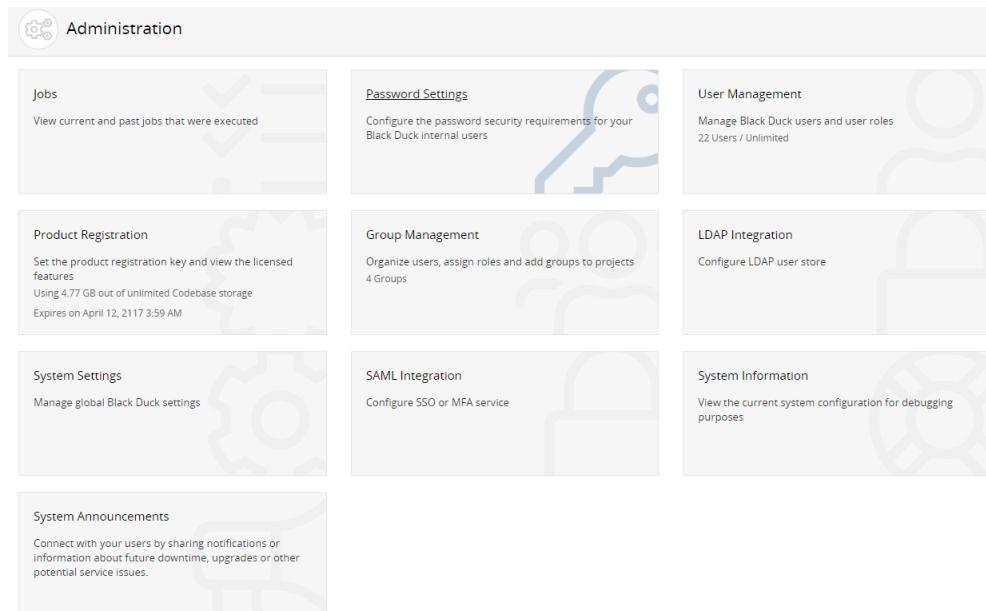
To view a user's groups

1. Log in to Black Duck.



2. Click **Admin**.

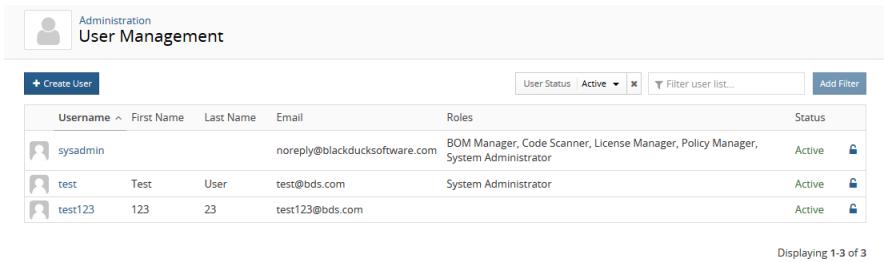
The Administration page appears.



A screenshot of the Black Duck Administration page. The page has a header with the title "Administration". Below the header are several cards representing different system components:

- Jobs**: View current and past jobs that were executed.
- Password Settings**: Configure the password security requirements for your Black Duck internal users.
- User Management**: Manage Black Duck users and user roles. (22 Users / Unlimited)
- Product Registration**: Set the product registration key and view the licensed features. (Using 4.77 GB out of unlimited Codebase storage. Expires on April 12, 2117 3:59 AM)
- Group Management**: Organize users, assign roles and add groups to projects. (4 Groups)
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- System Announcements**: Connect with your users by sharing notifications or information about future downtime, upgrades or other potential service issues.

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



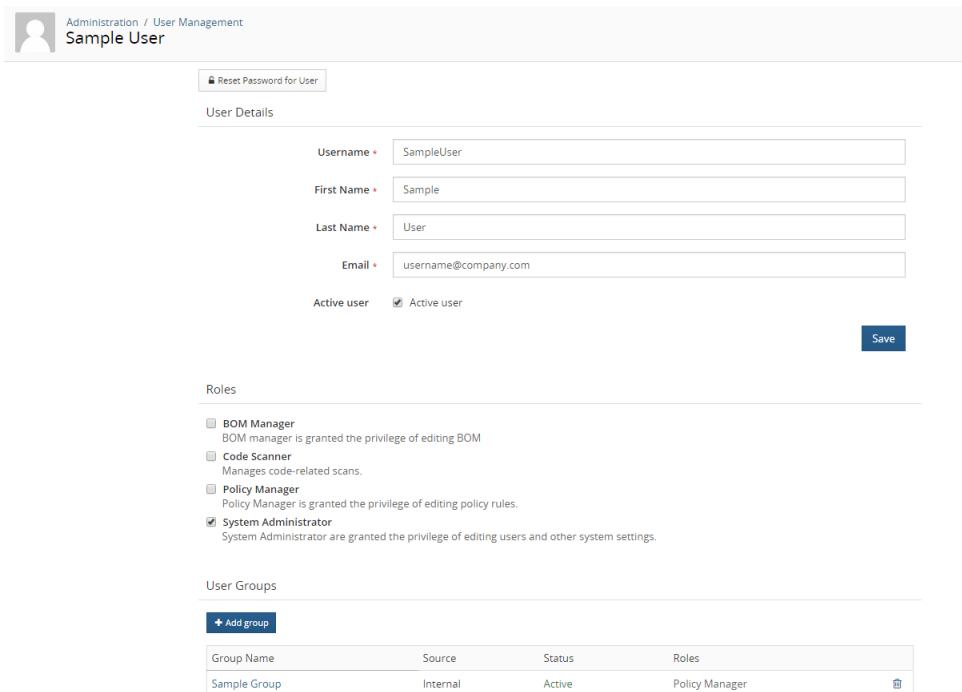
User Management				Status	
Username	First Name	Last Name	Email	Roles	Status
sysadmin			noreply@blackducksoftware.com	BOM Manager, Code Scanner, License Manager, Policy Manager, System Administrator	Active 
test	Test	User	test@bds.com	System Administrator	Active 
test123	123	23	test123@bds.com		Active 

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.



User Details

Username *	SampleUser
First Name *	Sample
Last Name *	User
Email *	username@company.com

Active user Active user **Save**

Roles

<input type="checkbox"/> BOM Manager BOM manager is granted the privilege of editing BOM
<input type="checkbox"/> Code Scanner Manages code-related scans.
<input type="checkbox"/> Policy Manager Policy Manager is granted the privilege of editing policy rules.
<input checked="" type="checkbox"/> System Administrator System Administrator are granted the privilege of editing users and other system settings.

User Groups

Add group
Group Name Source Status Roles
Sample Group Internal Active Policy Manager 

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.

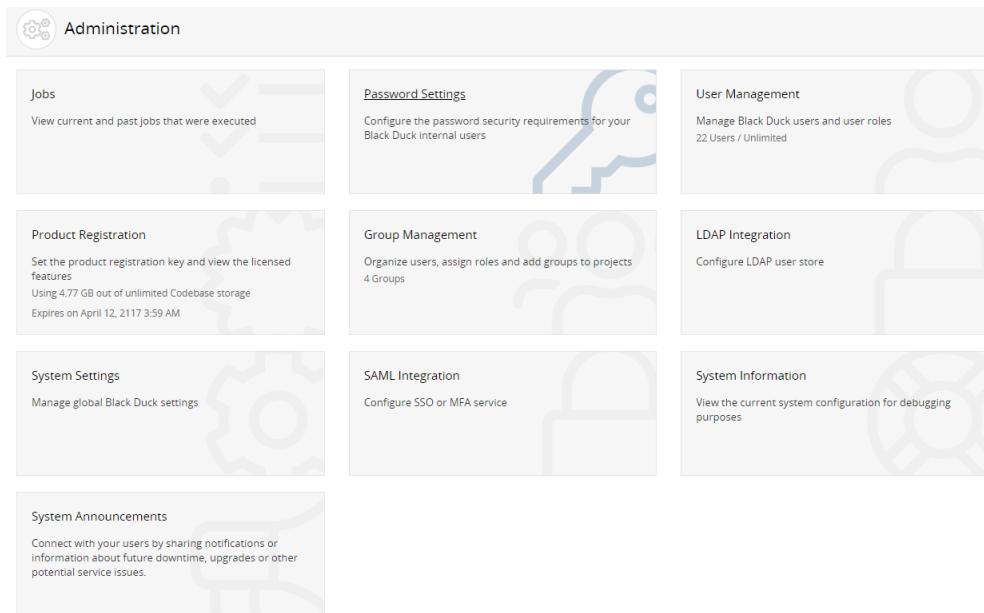
To view a user's projects

1. Log in to Black Duck.



2. Click **Admin**.

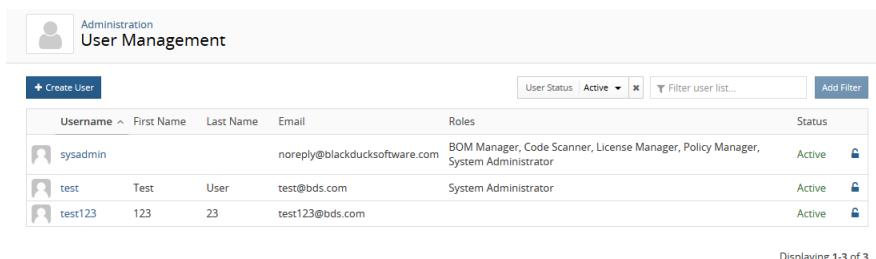
The Administration page appears.



The screenshot shows the Black Duck Administration interface. The main header is "Administration". Below it are several cards:

- Jobs**: View current and past jobs that were executed.
- Password Settings**: Configure the password security requirements for your Black Duck internal users.
- User Management**: Manage Black Duck users and user roles. (22 Users / Unlimited)
- Product Registration**: Set the product registration key and view the licensed features. (Using 4.77 GB out of unlimited Codebase storage. Expires on April 12, 2117 3:59 AM)
- Group Management**: Organize users, assign roles and add groups to projects. (4 Groups)
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- System Announcements**: Connect with your users by sharing notifications or information about future downtime, upgrades or other potential service issues.

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



The screenshot shows the "User Management" page. The header includes a "Create User" button and filters for "User Status" (Active), "Filter user list...", and "Add Filter". The main table lists users:

Username	First Name	Last Name	Email	Roles	Status
sysadmin			noreply@blackducksoftware.com	BOM Manager, Code Scanner, License Manager, Policy Manager, System Administrator	Active 
test	Test	User	test@bds.com	System Administrator	Active 
test123	123	23	test123@bds.com		Active 

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.

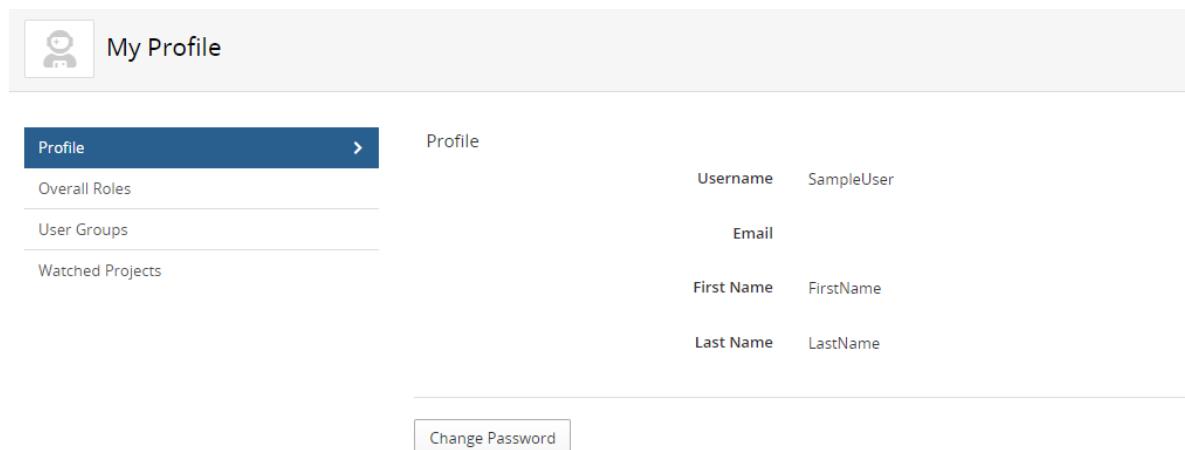
If your Black Duck server does not use LDAP to manage user accounts, your username and initial password were created by your Black Duck administrator. You can change your password on your profile page.

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. From the user menu located on the top navigation bar, select **My Profile**.

The My Profile page appears.



3. Click **Change Password**.

The Change Password dialog box appears.

Change Password

Passwords must:

Contain between 8 and 128 characters
 Be difficult to guess

Current Password *

New Password *

Confirm Password *

Cancel Save

4. Type your current password in the **Current Password** field.

5. Type your new password in the **New Password** field.

If there are [password requirements](#), those requirements are listed in the dialog box. Black Duck notes when each requirement is met as you type your new password. You will not be able to save this password if it does not meet *all* requirements.

6. Type the same new password in the **Confirm Password** field.

7. Click **Save**.

Changing user account information

You can modify the information for internal or external user accounts.

Note: If you have enabled LDAP, you can manage user account information on the LDAP server or, in Black Duck (for *external* Black Duck user accounts only). Note that any changes you make to user account information in Black Duck for *external* Black Duck user accounts will be overwritten the next time user information is synchronized with the data on the LDAP server.

Note: You can only update the information for an external user if an administrator has configured either SAML or LDAP in Black Duck. If *both* SAML and LDAP are enabled, or both are disabled, you cannot modify the information for an external user.

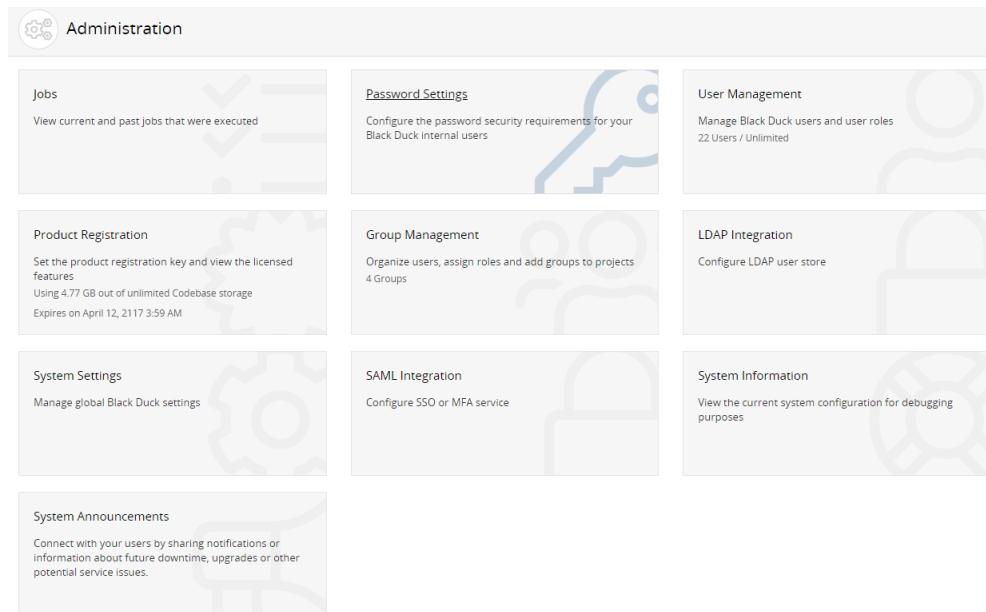
To change user account information:

1. Log in to Black Duck.



2. Click **Admin**.

The Administration page appears.



A screenshot of the Black Duck Administration page. The page has a header with the title "Administration". Below the header are nine cards arranged in a grid:

- Jobs**: View current and past jobs that were executed.
- Password Settings**: Configure the password security requirements for your Black Duck internal users.
- User Management**: Manage Black Duck users and user roles. (22 Users / Unlimited)
- Product Registration**: Set the product registration key and view the licensed features. (Using 4.77 GB out of unlimited Codebase storage, Expires on April 12, 2117 3:59 AM)
- Group Management**: Organize users, assign roles and add groups to projects. (4 Groups)
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- System Announcements**: Connect with your users by sharing notifications or information about future downtime, upgrades or other potential service issues.

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

User Management				
+ Create User User Status: Active <input type="button" value="x"/> Filter user list... <input type="button" value="Add Filter"/>				
Username	First Name	Last Name	Email	Roles
sysadmin			noreply@blackducksoftware.com	BOM Manager, Code Scanner, License Manager, Policy Manager, System Administrator
test	Test	User	test@bds.com	System Administrator
test123	123	23	test123@bds.com	

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 "Administration / User Management" section for a "Sample User".

User Details:

- Username: SampleUser
- First Name: Sample
- Last Name: User
- Email: username@company.com
- Active user: Active user
- Save button

Roles:

- BOM Manager**: BOM manager is granted the privilege of editing BOM
- Code Scanner**: Manages code-related scans.
- Policy Manager**: Policy Manager is granted the privilege of editing policy rules.
- System Administrator**: System Administrator are granted the privilege of editing users and other system settings.

User Groups:

- + Add group

Group Name	Source	Status	Roles
Sample Group	Internal	Active	Policy Manager

6. Enter the updated information in the **Internal** or **External User Details** section.

Note: If you are updating information for internal users in the **Internal User Details** section and [password requirements](#) have been defined, you will not be able to save the updated information if this user's password does not meet the password requirements; an error message appears notifying you of which password requirements are not met. Update the user's password to meet the password requirements and then update the information in this 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.

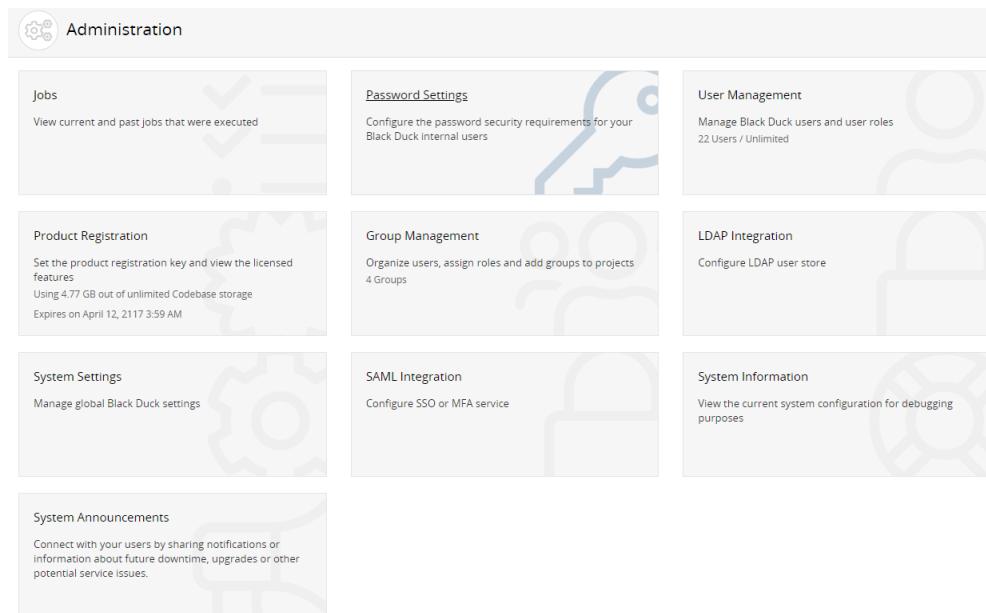
To change a user's password

1. Log in to Black Duck.



2. Click **Admin**.

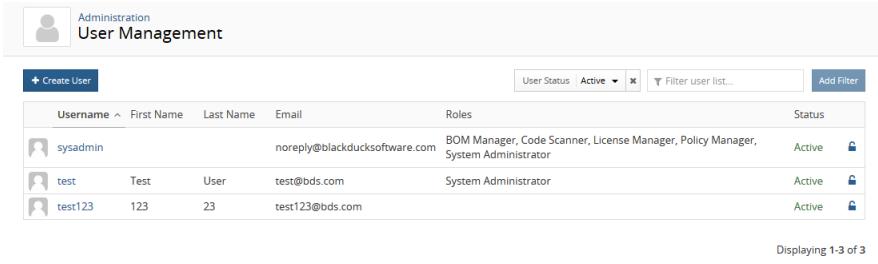
The Administration page appears.



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:

- Jobs**: View current and past jobs that were executed.
- Password Settings**: Configure the password security requirements for your Black Duck internal users. This card features a large blue gear icon.
- User Management**: Manage Black Duck users and user roles. 22 Users / Unlimited.
- Product Registration**: Set the product registration key and view the licensed features. Using 4.77 GB out of unlimited Codebase storage. Expires on April 12, 2117 3:59 AM.
- Group Management**: Organize users, assign roles and add groups to projects. 4 Groups.
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- System Announcements**: Connect with your users by sharing notifications or information about future downtime, upgrades or other potential service issues.

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, the text 'Administration', and 'User Management'. Below the header is a search bar with 'User Status: Active' and a 'Filter user list...' dropdown, along with a 'Add Filter' button. A table lists three users: 'sysadmin' (Email: noreply@blackducksoftware.com, Roles: BOM Manager, Code Scanner, License Manager, Policy Manager, System Administrator), 'test' (Email: test@bds.com, Roles: System Administrator), and 'test123' (Email: test123@bds.com). Each user row has a status column showing 'Active' and a lock icon. At the bottom 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 (🔒) icon for that user. Note that 🔒 indicates that you cannot change this user's password.
 - 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.
If there are [password requirements](#), those requirements are listed in this dialog box. Black Duck notes when each requirement is met. You will not be able to save this password if it does not meet *all* requirements.
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 and read-only access to the BOM and projects that user is assigned

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 and reviewing Black Duck KnowledgeBase 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.

- Copyright Editor

The Copyright Editor is responsible for creating or editing copyright statements for components.

This role is often assigned to someone within the Legal department.

- Global Code Scanner

The Global Code Scanner has access to all scans in Black Duck and can run, map, or delete scans for any existing 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 Release Creator

The Global Release Creator can create releases or versions of projects.

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.

- Global Security Manager

This Security Manager can create, edit, or delete global remediation statuses 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.

- License Manager

The License Manager is responsible for approving and/or rejecting licenses and managing the

licenses that can be used in applications. Users with this role can create, edit, and delete custom licenses, custom license terms, and custom license families. They can also manage BlackDuck KnowledgeBase licenses and license terms.

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.

- Super User

The Super User can:

- (Create/Edit/Delete) all project groups, projects, and versions
- (Create/Edit/Deactivate) users and user groups
- (Assign/Remove) members of user groups
- (Assign/Remove) users and user groups to/from projects and project groups

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.

- Project Creator

The Project Creator can create projects and can edit project and 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.

- 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 Annotator

The BOM Annotator can add or edit comments in a BOM for a specific project, but cannot edit the BOM. Users with this role can also update BOM component [custom fields](#).

- **BOM Manager**

The BOM Manager can modify the BOM for projects in which they are members or have project-group privileges, including modifying component identifications, ignoring components, updating the review status, adding comments, and running project version reports.

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, and modify BOM entries.

By default Project Managers can manage policy violations and remediate security vulnerabilities. However, the system administrator can [disable these capabilities](#).

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.

Project Group roles

The following project group roles have the same permissions as their project-only counterparts, except they apply for every project in their assigned project group:

- Project Group BOM Annotator
- Project Group BOM Manager
- Project Group Code Scanner
- Project Group Manager
- Project Group Policy Violation Reviewer
- Project Group Security Manager

Direct Access vs Indirect Access

New concepts used in Project Groups are Direct Access and Indirect Access to a project. Direct Access refers to a user being directly linked to a project. This has been the normal behavior and remains unchanged with the advent of Project Groups. Introduced with Project Groups, Indirect Access means that a user is linked to a project as a result of being in a user group that is linked to a project group, or because the project is in a project group to which this user is associated.

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.

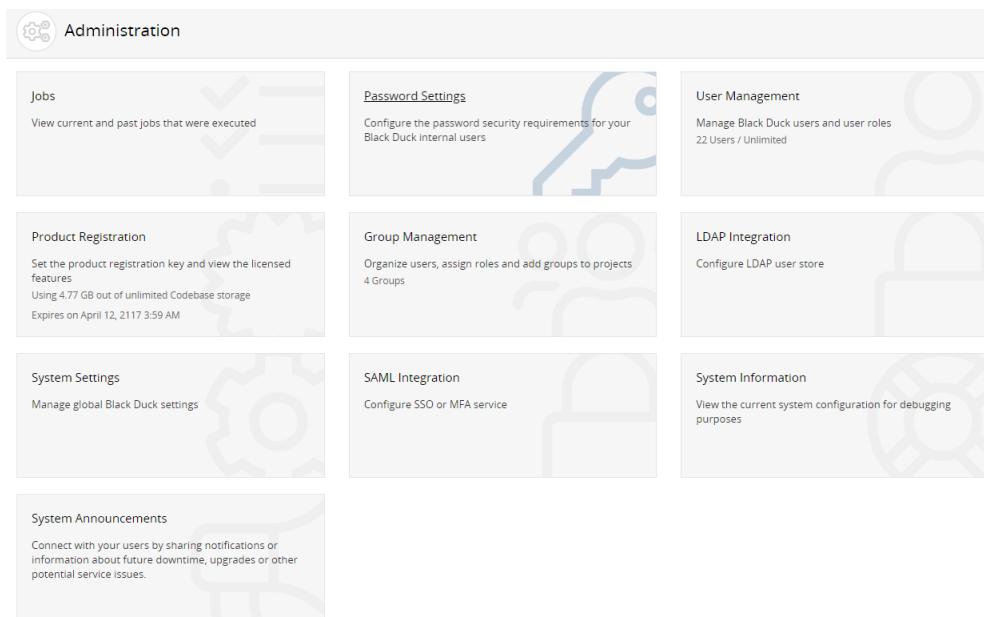
Note: If you do not assign a role to a user, that user has read-only access to Black Duck; this user cannot create projects. However, if a user with no roles is added as a project member, that user will be able to update and delete that project.

To assign an overall role to a user



1. Click **Admin**.

The Administration page appears.



2. Select **User Management** to display the User Management page.

Username	First Name	Last Name	Email	Roles	Status
sysadmin			noreply@blackducksoftware.com	BOM Manager, Code Scanner, License Manager, Policy Manager, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23	test123@bds.com		Active

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 'Sample User' edit page. At the top, there's a navigation bar with a user icon and the text 'Administration / User Management'. Below it, the page title is 'Sample User'. A 'Reset Password for User' button is visible. The main area is divided into sections: 'User Details' (Username: SampleUser, First Name: Sample, Last Name: User, Email: username@company.com), 'Active user' checkboxes, and a 'Save' button. The 'Roles' section lists several roles with checkboxes: BOM Manager (unchecked), Code Scanner (unchecked), Policy Manager (unchecked), and System Administrator (checked). The 'User Groups' section shows a table with one row: Group Name: Sample Group, Source: Internal, Status: Active, Roles: Policy Manager, and a delete icon.

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.

Note: Users can also obtain roles via user groups. Roles obtained through user groups are not shown in the **Overall Roles** section; instead the **User Groups** section lists roles for each user group.

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. From the user menu located on the top navigation bar, select **My Profile**.

The My Profile page appears.

The screenshot shows the 'My Profile' page. The left sidebar has three items: 'Overall Roles' (selected), 'User Groups', and 'Watched Projects'. The main content area displays user information: Username: SampleUser, Email, First Name: FirstName, and Last Name: LastName. At the bottom is a 'Change Password' button.

Select **Overall Roles** to view the roles assigned to your user account. Note that this section includes all roles that were assigned to you via user groups.

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 or a project group.

Global roles by task

Task	Super User role	Component Manager role	Copyright Editor	Global Code Scanner role	Global Project Viewer	Global Release Creator	Global Security Manager	License Manager role	Policy Manager role	Project Creator role	System Administrator role
Manage code scans/Protex BOM files:				√							
<ul style="list-style-type: none"> Scan code. Upload scans to Black Duck. Map or unmap scans to projects. 											
Create, edit, delete projects.		√								√	
Manage projects:	√					√ Create permission only				√ See Project Manager role for permissions obtained when creating a project	
Manage custom components.			√								
Manage licenses:								√ Can manage licenses for all			
<ul style="list-style-type: none"> Create, edit, delete custom licenses. Manage Knowledge 											

Task	Super User role	Component Manager role	Copyright Editor	Global Code Scanner role	Global Project Viewer	Global Release Creator	Global Security Manager	License Manager role	Policy Manager role	Project Creator role	System Administrator role
<ul style="list-style-type: none"> Base licenses. Create, edit, delete custom license families. Manage KB and custom license terms. 								projects.			
View BOMs: <ul style="list-style-type: none"> View BOM. Add/edit/view comments. Print BOM. Compare BOMs. 	√				√	Can view all projects.					
Manage BOMs: <ul style="list-style-type: none"> Manually add components; delete manually added components. Ignore components. Review components. Remediate security vulnerabilities. Override policy violations. 	√										

Task	Super User role	Component Manager role	Copyright Editor	Global Code Scanner role	Global Project Viewer	Global Release Creator	Global Security Manager	License Manager role	Policy Manager role	Project Creator role	System Administrator role
<ul style="list-style-type: none"> Remove override of policy violations. Edit licenses, including excluding license from Notices File report, adding an attribution statement, or selecting a different license for a component version. Indicate license term fulfillment status. Manage deep license data. View license conflicts. 											
Manage policy rules: create, edit, or delete policy rules.									✓		
Update custom field values.	✓ Can update any type of custom	✓ Can only update Component and Component									

Task	Super User role	Component Manager role	Copyright Editor	Global Code Scanner role	Global Project Viewer	Global Release Creator	Global Security Manager	License Manager role	Policy Manager role	Project Creator role	System Administrator role
Version custom fields.	field.										
Create, edit, or delete global remediation statuses.							√ Must be assigned to a project to view data.				
Run project vulnerability reports from the Reports menu.	√	√	√	√	√		√ Must be assigned to a project to view data.	√ Must be assigned to a project to view data.	√ Must be assigned to a project to view data.	√ Must be assigned to a project to view data.	√ Must be assigned to a project to view data.
Create and modify copyright statements.				√ Must be assigned to a project to view data.							
Project version reports:	√			√ Must be assigned to a project	√						
• Run Notices File report.											

Task	Super User role	Component Manager role	Copyright Editor	Global Code Scanner role	Global Project Viewer	Global Release Creator	Global Security Manager	License Manager role	Policy Manager role	Project Creator role	System Administrator role
• Run Project Version report.			to view data.								
View information in Dashboard pages.	√		√ Must be assigned to a project to view data.		√		√ Must be assigned to a project to view data.				
Access the Tools page from which user can:	√	√	√	√	√		√	√	√	√	√
• Download the scanner. • Access links to the Community and Customer Education.											
Search	√	√	√	√	√		√	√	√	√	√
Administer Black Duck. Use the Admin menu to:											√
• View jobs. • Register Black Duck. • Configure LDAP. • Configure											

Task	Super User role	Component Manager role	Copyright Editor	Global Code Scanner role	Global Project Viewer	Global Release Creator	Global Security Manager	License Manager role	Policy Manager role	Project Creator role	System Administrator role
SAML.											
<ul style="list-style-type: none"> Manage system settings. Manage system announcements. Configure password requirements. 											
Administer users and groups. Use the Admin menu to:	✓										
<ul style="list-style-type: none"> Manage users, including resetting passwords. Manage groups. 											
Manage snippets	✓										
View issues	✓										
Manage project groups:	✓										
<ul style="list-style-type: none"> Create/Edit/Delete project groups Add/Remove members and user groups from project groups 											

Project roles

These project-level roles only apply to the projects a user is assigned.

Task	Project Manager role	Security Manager role	BOM Annotator	BOM Manager role	Project Code Scanner role	Policy Violation Reviewer	No roles/Project Viewer
Manage code scans/Protex BOM files:	<ul style="list-style-type: none"> Scan code. Upload scans to Black Duck. Map or unmap scans to projects. 	✓ 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:	✓				✓ Can only create project versions.		
Manage custom licenses:				✓ Can only edit custom license text in BOM.			
View BOMs:	✓	✓	✓	✓	✓	✓	✓
• View BOM.							

Task	Project Manager role	Security Manager role	BOM Annotator	BOM Manager role	Project Code Scanner role	Policy Violation Reviewer	No roles/Project Viewer When a user is created and assigned to a project they have a read only/project viewer role.
<ul style="list-style-type: none">• View notifications.• Add/edit/view comments.• Print BOM.• Compare BOMs.			Can only view the BOM and add or edit comments.				

Task	Project Manager role	Security Manager role	BOM Annotator	BOM Manager role	Project Code Scanner role	Policy Violation Reviewer	No roles/Project Viewer When a user is created and assigned to a project they have a read only/project viewer role.
Manage BOMs:	√			√			
<ul style="list-style-type: none"> Manually add components; delete manually added components. Ignore components. Review components. Edit licenses, including excluding license from Notices File report, adding an attribution statement, or selecting a different license for a component version. Indicate license term fulfillment status. Manage deep license data. Update custom field information. View license conflicts. 							
Manage policy violations:	√ Can only manage policy					√	

Task	Project Manager role	Security Manager role	BOM Annotator	BOM Manager role	Project Code Scanner role	Policy Violation Reviewer	No roles/Project Viewer When a user is created and assigned to a project they have a read only/project viewer role.
• Remove override of policy violations.	violations if enabled by the system administrator .						
Remediate security vulnerabilities.	✓ Can only remediate security vulnerabilities if enabled by the system administrator .	✓ Can only modify remediation for vulnerabilities associated with components.					
Update custom field values.	✓ Can only update BOM Component, Project, and Project Version custom fields.		✓ Can only update BOM Component custom field.	✓ Can only update BOM Component custom field.			
Manage policy rules: create, edit, or delete policy rules.							
Run project vulnerability reports from the Report menu.	✓	✓	✓	✓	✓	✓	✓

Task	Project Manager role	Security Manager role	BOM Annotator	BOM Manager role	Project Code Scanner role	Policy Violation Reviewer	No roles/Project Viewer
Project version reports: <ul style="list-style-type: none">• Run Notices File report.• Run Project Version report.	√	√	√	√	√	√	√
View information in Dashboard pages.	√	√	√	√	√	√	√
Access the Tools page from which user can: <ul style="list-style-type: none">• Download the scanner.• Access API documentation.	√	√	√	√	√	√	√
Search	√	√	√	√	√	√	√

Task	Project Manager role	Security Manager role	BOM Annotator	BOM Manager role	Project Code Scanner role	Policy Violation Reviewer	No roles/Project Viewer When a user is created and assigned to a project they have a read only/project viewer role.
Administer Black Duck. Use the Administration menu to:							
<ul style="list-style-type: none"> • View jobs. • Register Black Duck. • Configure LDAP. • Configure SAML. • Manage system settings. • Manage system announcement s. • Configure password requirements. 							
Administer users and groups. Use the Administration menu to:							
<ul style="list-style-type: none"> • Manage users, including resetting passwords. • Manage groups. 							
Manage snippets	✓			✓			

Project Group roles

The following project group roles have the same permissions as their project-only counterparts, except they

apply for every project in their assigned project group:

Task	Project Group Manager	Project Group Security Manager	Project Group BOM Annotator	Project Group BOM Manager	Project Group Code Scanner	Project Group Policy Violation Reviewer
Manage code scans/Protex BOM files: <ul style="list-style-type: none">• Scan code.• Upload scans to Black Duck.• Map or unmap scans to projects.	√ Can unmap scans from their projects.				√ Can map or unmap a code scan to/from projects for which they have access.	
Create, edit, delete projects.						
Manage projects: <ul style="list-style-type: none">• Create, edit, delete project versions.• Edit project or version settings, including tags.	√				√ Can only create project versions.	
Manage custom licenses: <ul style="list-style-type: none">• Create, edit, delete custom licenses.				√ Can only edit custom license text in BOM.		
View BOMs: <ul style="list-style-type: none">• View BOM.• View notifications.• Add/edit/view comments.• Print BOM.• Compare BOMs.	√	√	√ Can only view the BOM and add or edit comments.	√	√	√

Task	Project Group Manager	Project Group Security Manager	Project Group BOM Annotator	Project Group BOM Manager	Project Group Code Scanner	Project Group Policy Violation Reviewer
Manage BOMs:	√			√		
<ul style="list-style-type: none"> • Manually add components; delete manually added components. • Ignore components. • Review components. • Edit licenses, including excluding license from Notices File report, adding an attribution statement, or selecting a different license for a component version. • Indicate license term fulfillment status. • Manage deep license data. • Update custom field information. • View license conflicts. 						
Manage policy violations:	√ Can only manage policy violations if enabled by the system administrator .					√

Task	Project Group Manager	Project Group Security Manager	Project Group BOM Annotator	Project Group BOM Manager	Project Group Code Scanner	Project Group Policy Violation Reviewer
Remediate security vulnerabilities.	√ Can only remediate security vulnerabilities if enabled by the system administrator .	√ Can only modify remediation for vulnerabilities associated with components.				
Update custom field values.	√ Can only update BOM Component, Project, and Project Version custom fields.		√ Can only update BOM Component custom field.	√ Can only update BOM Component custom field.		
Manage policy rules: create, edit, or delete policy rules.						
Run project vulnerability reports from the Report menu.	√	√	√	√	√	√
Project version reports: • Run Notices File report. • Run Project Version report.	√	√	√	√	√	√
View information in Dashboard pages.	√	√	√	√	√	√
Access the Tools page from which user can: • Download the scanner. • Access API documentation.	√	√	√	√	√	√

Task	Project Group Manager	Project Group Security Manager	Project Group BOM Annotator	Project Group BOM Manager	Project Group Code Scanner	Project Group Policy Violation Reviewer
Search	✓	✓	✓	✓	✓	✓
Administer Black Duck. Use the Administration menu to:						
<ul style="list-style-type: none"> • View jobs. • Register Black Duck. • Configure LDAP. • Configure SAML. • Manage system settings. • Manage system announcements. • Configure password requirements. 						
Administer users and groups. Use the Administration menu to:						
<ul style="list-style-type: none"> • Manage users, including resetting passwords. • Manage groups. 						
Manage snippets	✓			✓		

Managing the Project Manager role

System administrators can define whether users with the Project Manager role can manage policy violations (override policy violations or remove overrides) or remediate security vulnerabilities for a project.

Note: This is a global setting: all users with the Project Manager role are affected by any changes you make to the role.

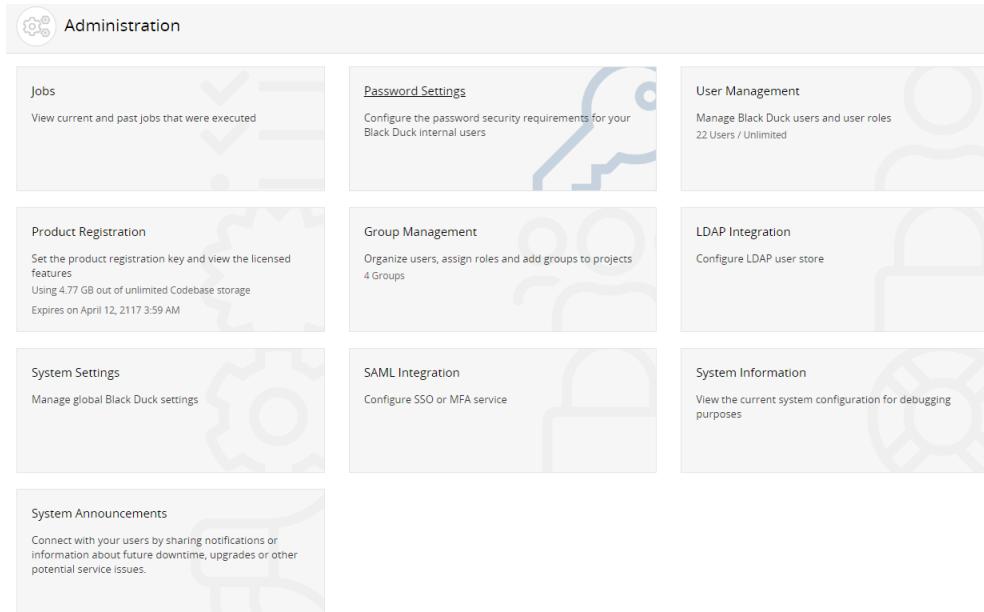
To modify the project manager role

1. Log in to Black Duck with the System Administrator role.



2. Click **Admin**.

The Administration page appears.



A screenshot of the Black Duck Administration page. The page has a header titled 'Administration' with a gear icon. Below the header are several cards:

- Jobs**: View current and past jobs that were executed.
- Password Settings**: Configure the password security requirements for your Black Duck internal users.
- User Management**: Manage Black Duck users and user roles (22 Users / Unlimited).
- Product Registration**: Set the product registration key and view the licensed features (Using 4.77 GB out of unlimited Codebase storage, Expires on April 12, 2017 3:59 AM).
- Group Management**: Organize users, assign roles and add groups to projects (4 Groups).
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- System Announcements**: Connect with your users by sharing notifications or information about future downtime, upgrades or other potential service issues.

3. Select **System Settings**.

The System Settings page appears.

The screenshot shows the 'System Settings' section of the Black Duck Administration interface. It includes fields for 'Logo' (with a placeholder for 'SYNOPSYS'), 'System Logs' (with a download link), 'Legal Tab Visibility' (with a 'Disable' button), 'Security Risk Configuration Ranking' (with a drag-and-drop interface for CVSS priority), 'Custom Scan Signature Level' (set to 5), and 'Snippet Max File Size' (set to 2).

4. In the **Project Manager Role Settings** section, select or clear the **Policy Violation Reviewer** and/or **Security Manager** options.
5. Click **Save**.

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.

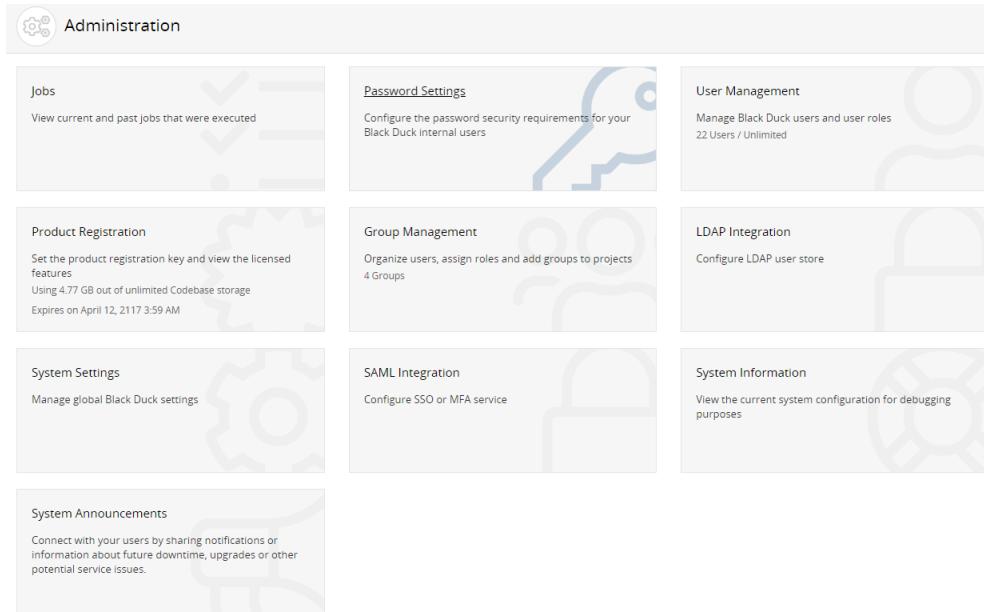
Example: `cn`

2. Log in to Black Duck as a system administrator.



3. Click **Admin**.

The Administration page appears.



A screenshot of the Black Duck Administration page. The page has a light gray header with the title "Administration". Below the header is a grid of nine cards, each with a title, a brief description, and a small icon. The cards are arranged in three rows of three. The first row contains "Jobs" (View current and past jobs that were executed), "Password Settings" (Configure the password security requirements for your Black Duck internal users), and "User Management" (Manage Black Duck users and user roles). The second row contains "Product Registration" (Set the product registration key and view the licensed features), "Group Management" (Organize users, assign roles and add groups to projects), and "LDAP Integration" (Configure LDAP user store). The third row contains "System Settings" (Manage global Black Duck settings), "SAML Integration" (Configure SSO or MFA service), and "System Information" (View the current system configuration for debugging purposes). A "System Announcements" card is also visible at the bottom left, which is currently empty.

4. Select **LDAP Integration** to open the LDAP Integration page.

The screenshot shows the 'LDAP Server Details' configuration page. It includes the following sections:

- LDAP Server Details:** Contains fields for 'Enable LDAP' (checked), 'Server URL' (ldaps://ldap1.blackducksoftware.com:389/), 'Authentication Type' (Simple), 'Manager DN' (CN=Fred.Smith,OU=IT,DC=blackducksoftware,DC=com), and 'Manager Password'.
- LDAP User Attributes:** Contains fields for 'Create user accounts automatically in Black Duck' (checked), 'User Search Base' (dc=blackducksoftware,dc=com), 'User Search Filter' (sAMAccountName=), and 'User DN Pattern'.
- LDAP Attribute Mappings:** Contains fields for 'First Name', 'Last Name', and 'Email'.
- LDAP Groups:** Contains fields for 'Synchronize LDAP groups' (checked), 'Group search base' (dc=blackducksoftware,dc=com), 'Group filter' ((&(objectClass=group)(member=*)(cn=Administrators)(cn=Development)(cn=Domain Admins)(cn=QA Admin Group))], and 'Group name attribute' (cn).
- Test Connection, User Authentication and Field Mapping:** Contains fields for 'Test Username' and 'Test Password', and a 'Test Connection' button.

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.

Optionally, clear the **Create user accounts automatically in Black Duck** check box to turn off the automatic creation of users when they authenticate with LDAP. This check box is selected by default so users that do not exist in Black Duck are created automatically when they log into Black Duck using LDAP. This applies to new installs and upgrades.

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 **Admin**.

The Administration page appears.

3. Select **LDAP integration** to display the LDAP Integration page.

The screenshot shows the 'LDAP Server Details' configuration page. It includes sections for:

- LDAP Server Details:** Contains fields for 'Server URL' (ldap://bdad1.blackducksoftware.com:389), 'Authentication Type' (Simple), 'Manager DN' (CN=Fred Smith,OU=IT,DC=blackducksoftware,DC=com), and 'Manager Password' (Password already set, click to change it).
- LDAP User Attributes:** Contains fields for 'User Search Base' (dc=blackducksoftware,dc=com), 'User Search Filter' (sAMAccountName=(0)), and 'User DN Pattern'.
- LDAP Attribute Mappings:** Contains fields for 'First Name', 'Last Name', and 'Email'.
- LDAP Groups:** Contains fields for 'Group search base' (dc=blackducksoftware,dc=com), 'Group filter' (&(objectClass=group)(member=(0))(cn=Administrators)(cn=Development)(cn=Domain Admins)(cn=QA Admin Group))], and 'Group name attribute' (cn).
- Test Connection, User Authentication and Field Mapping:** Includes fields for 'Test Username' and 'Test Password', and a 'Test Connection' button.

A 'Save' button is located at the bottom right of the main form area.

4. Select the **Enable LDAP** option and complete the information in the **LDAP Server Details** section, as

described above. In the **Server URL** field, ensure that you have configured the secure LDAP server: the protocol scheme is `ldaps://`.

5. Complete the information in the **LDAP User Attributes** section, as described above.

Optionally, clear the **Create user accounts automatically in Black Duck** check box to turn off the automatic creation of users when they authenticate with LDAP. This check box is selected by default so users that do not exist in Black Duck are created automatically when they log into Black Duck using LDAP. This applies to new installs and upgrades.

6. Enter the user credentials in the **Test Connection, User Authentication and Field Mapping** section and click **Test Connection**.
7. 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 *	flast						
Test Password *	*****						
Test Connection	⚠️ Test Connection ✓ Connection Test Succeeded						
<table border="1"><tr><td>✓ First Name</td><td>First</td></tr><tr><td>✓ Last Name</td><td>Last</td></tr><tr><td>✓ Email</td><td>flast@company.com</td></tr></table>		✓ First Name	First	✓ Last Name	Last	✓ Email	flast@company.com
✓ First Name	First						
✓ Last Name	Last						
✓ Email	flast@company.com						

8. If there is an issue with the certificate, a dialog box listing details about the certificate appears:

Certificate Problem

Details about the certificates are below. If you'd like to accept this certificate, press "Save".

Certificate Details	<p>Issuer: CN=www.blackducksoftware.com, OU=Engineering, O="Black Duck Software, Inc.", L=Burlington, ST=Massachusetts, C=US</p> <p>Subject: CN=www.blackducksoftware.com, OU=Engineering, O="Black Duck Software, Inc.", L=Burlington, ST=Massachusetts, C=US</p> <p>Alt Subjects: blackducksoftware.com, ldap.blackducksoftware.com, skrib, *updates.blackducksoftware.com</p> <p>Begins On: Jun 19, 2017</p> <p>Expires On: Jun 19, 2019</p> <p>Algorithm: SHA1withRSA</p>
Cancel Save	

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. From the user menu located on the top navigation bar, select **My Profile**.

The My Profile page appears.

The screenshot shows the 'My Profile' page. On the left, there's a sidebar with 'Profile' selected, followed by 'Overall Roles', 'User Groups', and 'Watched Projects'. The main area is titled 'Profile' and contains fields for 'Username' (SampleUser), 'Email' (Email), 'First Name' (FirstName), and 'Last Name' (LastName). At the bottom is a 'Change Password' button.

3. Select **User Groups** to view the group in which you are a member.

The screenshot shows the 'User Groups' page. The sidebar has 'User Groups' selected. The main table lists one group: 'Sample Group' with 'Internal' source and 'Active' status. A note at the bottom says 'Displaying 1-1 of 1'.

Group Name	Source	Status	Roles
Sample Group	Internal	Active	

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.

If you create a default group, subsequent new users are automatically added to this group and are granted all roles and access to all projects configured for this group. Note that:

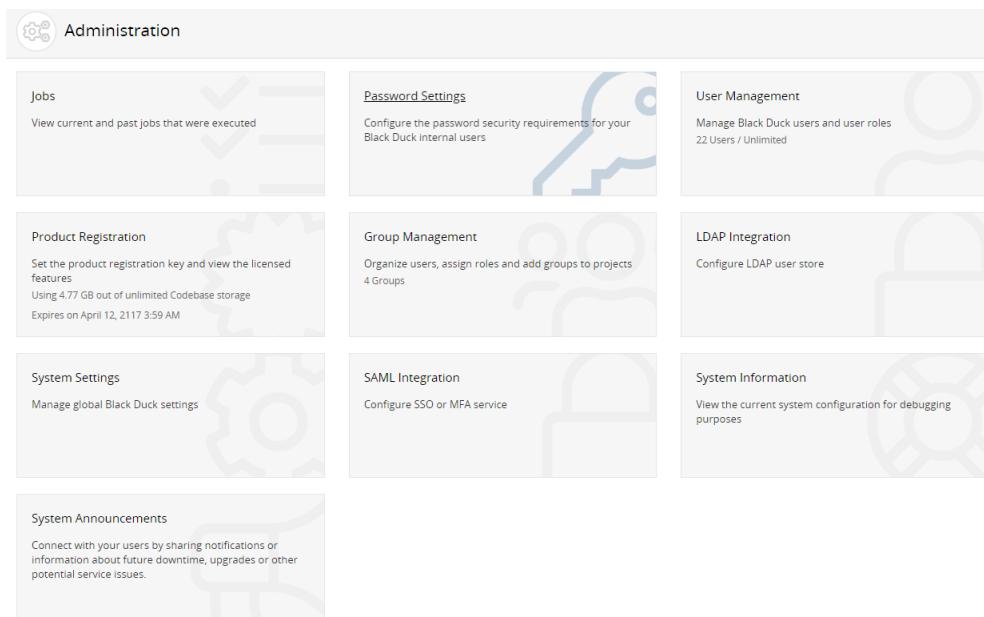
- You can have more than one default group.
- Default groups have a status of *Status - Default*.

To create a group

1. Log in to Black Duck.

2. Click  Admin.

The Administration page appears.



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. Click **Create Group to display the Create a New Group dialog box.**

Create a New Group

Group Name *

Active Group Active Group

Default Group Default Group
All new users will be added to this group. Users will have the roles and access to projects configured for this group.

5. In the Create a New Group dialog box:

- Type the name of the group in the **Group Name** field.

- b. Select whether this group is active or inactive.
- c. Select whether this group is a default group.
- d. Click **Create**. 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, status (active/inactive), and/or whether this is a default group.

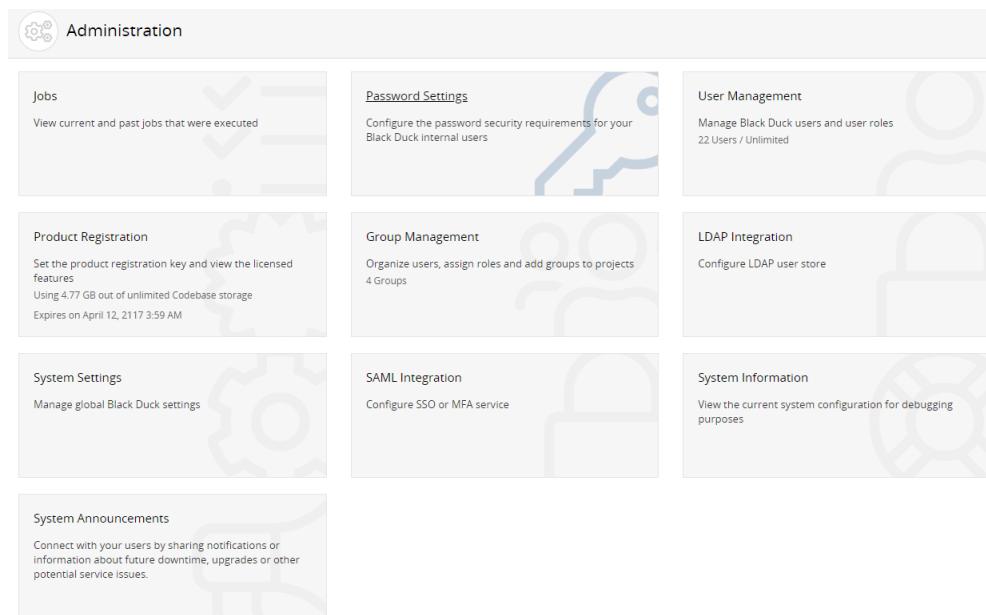
To manage group information

1. Log in to Black Duck.



2. Click **Admin**.

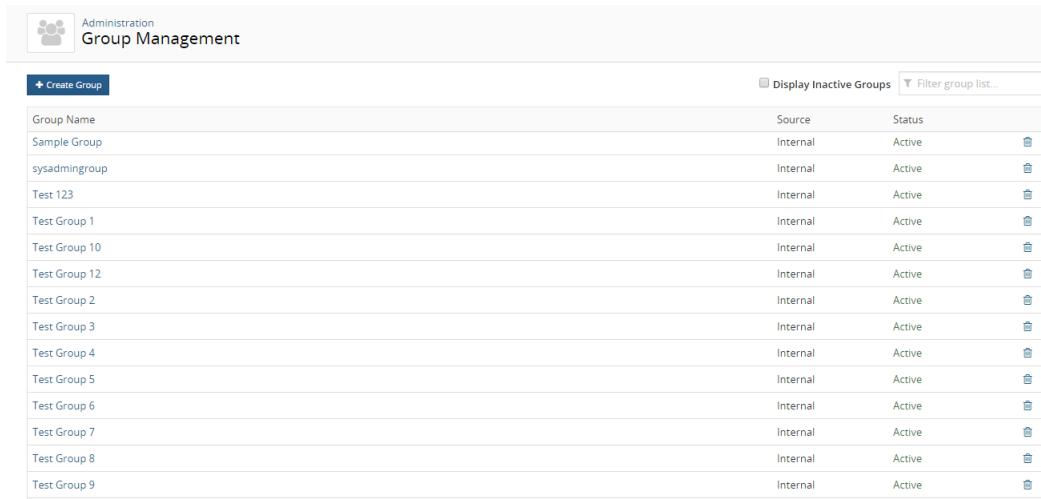
The Administration page appears.



A screenshot of the Black Duck Administration page. The page has a light gray header with the title "Administration". Below the header is a grid of nine cards, each representing a different administrative function. The cards are arranged in three rows of three. The first row contains "Jobs" (with a checkmark icon), "Password Settings" (with a padlock icon), and "User Management" (with a user icon). The second row contains "Product Registration" (with a gear icon), "Group Management" (with a person icon), and "LDAP Integration" (with a lock icon). The third row contains "System Settings" (with a gear icon), "SAML Integration" (with a lock icon), and "System Information" (with a gear icon). Each card has a title, a brief description, and some small text at the bottom.

Jobs	Password Settings	User Management
View current and past jobs that were executed	Configure the password security requirements for your Black Duck internal users	Manage Black Duck users and user roles 22 Users / Unlimited
Product Registration	Group Management	LDAP Integration
Set the product registration key and view the licensed features Using 4.77 GB out of unlimited Codebase storage Expires on April 12, 2117 3:59 AM	Organize users, assign roles and add groups to projects 4 Groups	Configure LDAP user store
System Settings	SAML Integration	System Information
Manage global Black Duck settings	Configure SSO or MFA service	View the current system configuration for debugging purposes
System Announcements		
Connect with your users by sharing notifications or information about future downtime, upgrades or other potential service issues.		

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

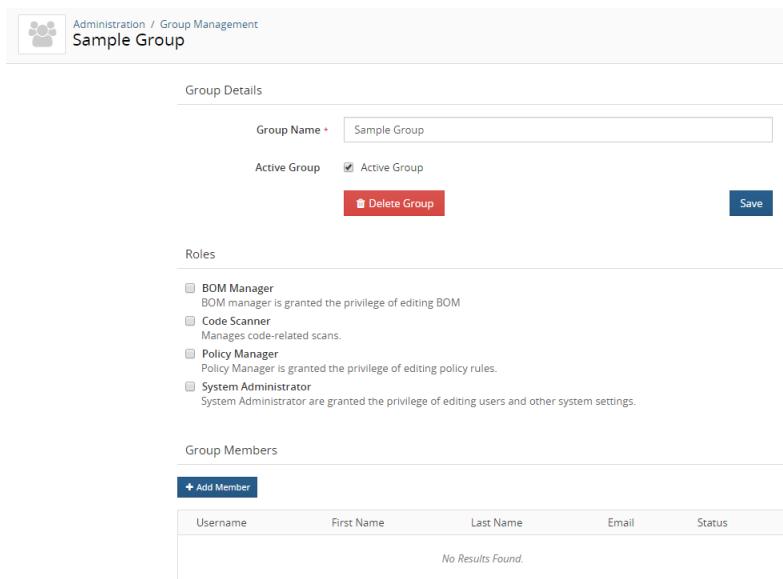


Group Name	Source	Status
Sample Group	Internal	Active
sysadmingroup	Internal	Active
Test 123	Internal	Active
Test Group 1	Internal	Active
Test Group 10	Internal	Active
Test Group 12	Internal	Active
Test Group 2	Internal	Active
Test Group 3	Internal	Active
Test Group 4	Internal	Active
Test Group 5	Internal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

4. Find the 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.



Group Details

Group Name	Sample Group
Active Group	<input checked="" type="checkbox"/> Active Group

Delete Group **Save**

Roles

<input type="checkbox"/> BOM Manager BOM manager is granted the privilege of editing BOM
<input type="checkbox"/> Code Scanner Manages code-related scans.
<input type="checkbox"/> Policy Manager Policy Manager is granted the privilege of editing policy rules.
<input type="checkbox"/> System Administrator System Administrator are granted the privilege of editing users and other system settings.

Group Members

Add Member	Username	First Name	Last Name	Email	Status
No Results Found					

6. In the **Group Details** section, type the new group name, change the status, or change whether this is a

default group.

Note that if you enabled group synchronization when configuring LDAP or SAML, the name of this group in the external authentication system (LDAP or SSO) appears in the **External Group Name** field. Black Duck uses the external name to synchronize the group and its members with integrated authentication/authorization systems. Generally, the two group names are the same when created automatically by synchronization. However, if the group name changes on the external system, you can edit the name to keep the Black Duck group name in sync with the external authentication system group name.

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.

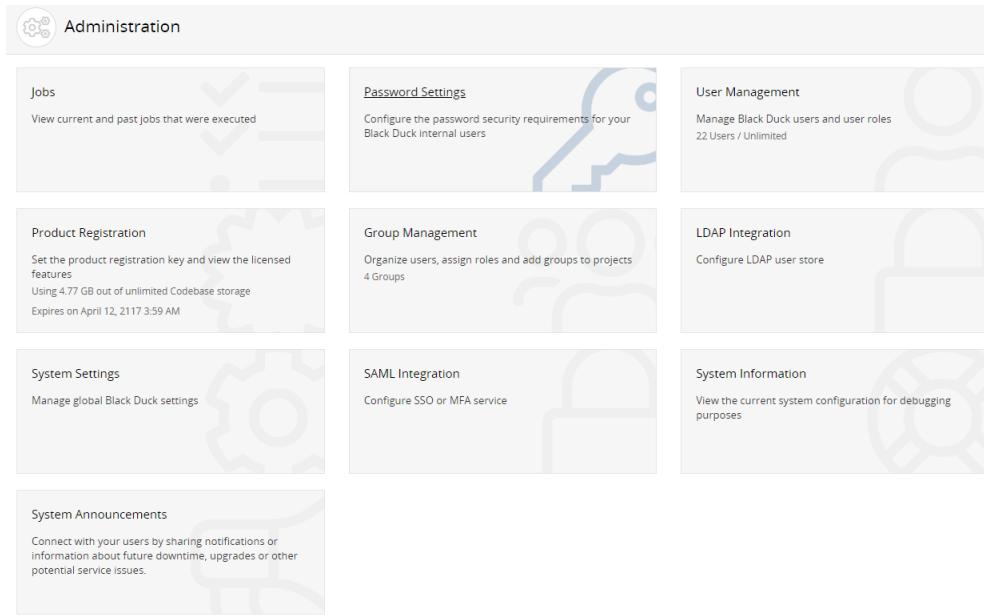
To assign a project to a group

1. Log in to Black Duck.



2. Click **Admin**.

The Administration page appears.



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

Group Management		
+ Create Group		<input type="checkbox"/> Display Inactive Groups <input type="text"/> Filter group list...
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 'Administration / Group Management' interface. A group named 'Sample Group' is selected. The 'Group Details' section includes a 'Group Name' field set to 'Sample Group' and an 'Active Group' checkbox which is checked. Below this are sections for 'Roles' (listing BOM Manager, Code Scanner, Policy Manager, and System Administrator) and 'Group Members' (which currently shows 'No Results Found.'). At the bottom are 'Delete Group' and 'Save' buttons.

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**.

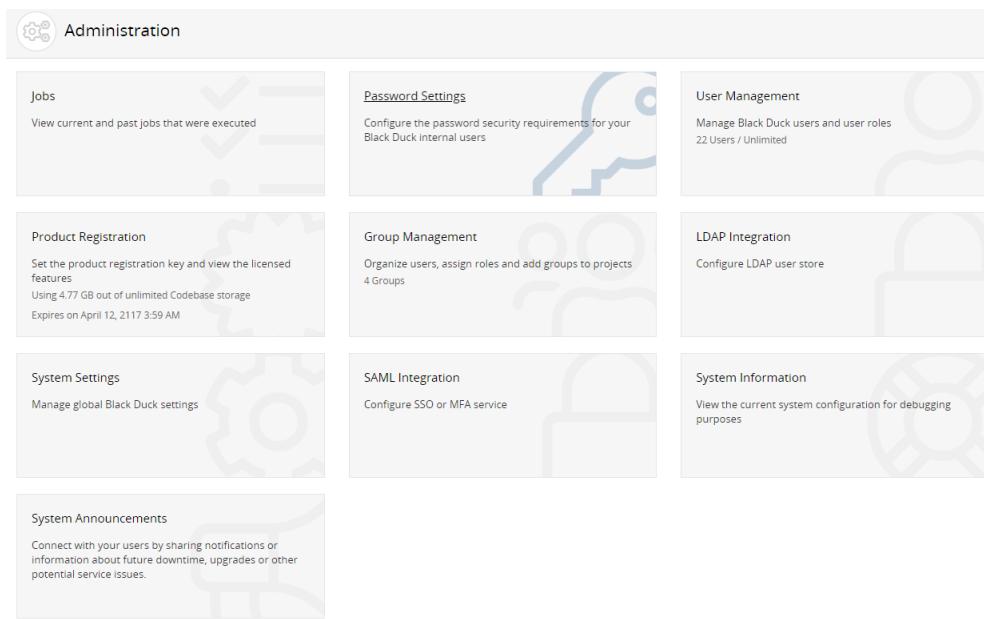
To remove a project from a group

1. Log in to Black Duck.



2. Click **Admin**.

The Administration page appears.



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

Group Management		
+ Create Group		<input type="checkbox"/> Display Inactive Groups <input type="text"/> Filter group list...
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 'Administration / Group Management' page for a group named 'Sample Group'. The 'Group Details' section includes a 'Group Name' field set to 'Sample Group', an 'Active Group' checkbox checked, and a 'Delete Group' button. Below this is a 'Roles' section listing four options: 'BOM Manager' (BOM manager is granted the privilege of editing BOM), 'Code Scanner' (Manages code-related scans), 'Policy Manager' (Policy Manager is granted the privilege of editing policy rules), and 'System Administrator' (System Administrator are granted the privilege of editing users and other system settings). The 'Group Members' section shows a table with columns: Username, First Name, Last Name, Email, and Status. A 'No Results Found.' message is displayed. A '+ Add Member' button is located at the top of this section.

5. Click in the row of the group you want to remove in the **Group Projects** section.
6. Click **Remove** to confirm.

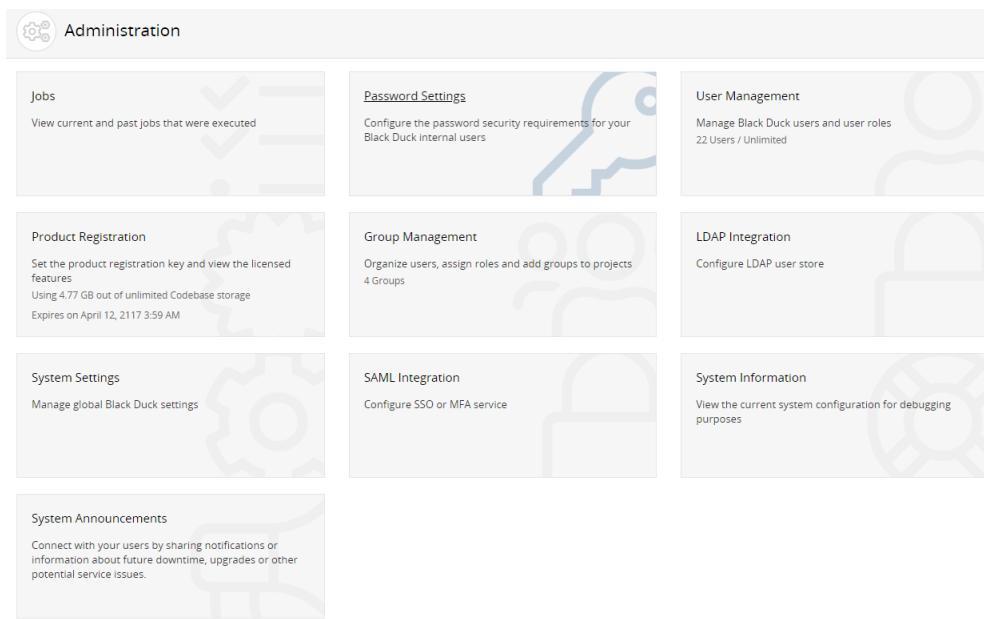
Managing group roles

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 Administration page appears.



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

The screenshot shows the Group Management page with the following table:

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

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' page for a group named 'Sample Group'. The 'Group Details' section includes a 'Group Name' field set to 'Sample Group' and an 'Active Group' checkbox checked. Below this are sections for 'Roles' and 'Group Members'. The 'Roles' section lists four options: 'BOM Manager' (selected), 'Code Scanner', 'Policy Manager', and 'System Administrator'. The 'Group Members' section has a table with columns: Username, First Name, Last Name, Email, and Status. A button '+ Add Member' is at the top left of the table area.

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

Note that subsequent users are automatically added to [default groups](#).

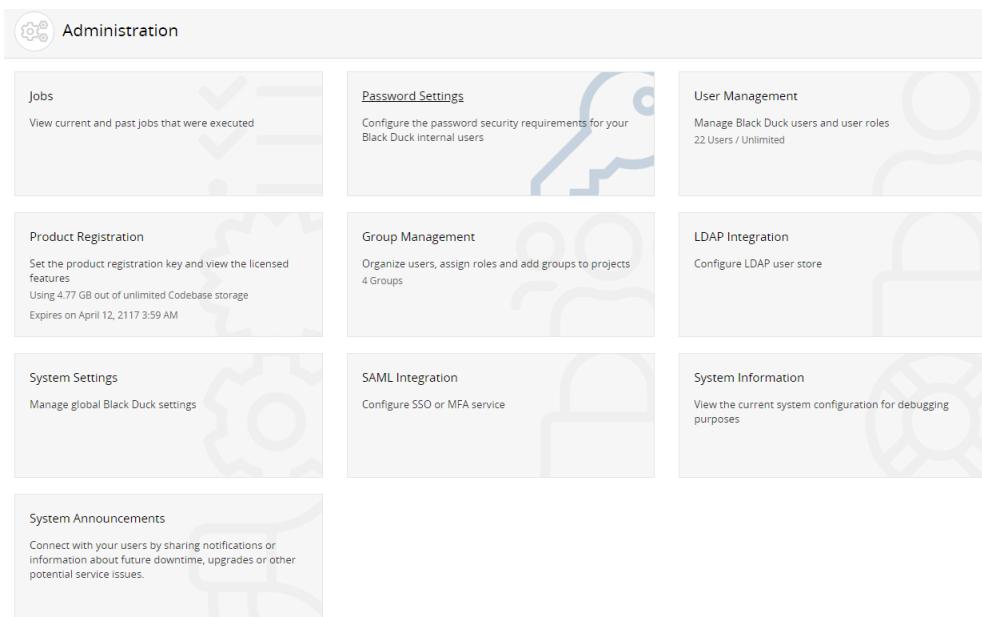
To add a member to a group by managing a group

1. Log in to Black Duck.



2. Click **Admin**.

The Administration page appears.



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

Administration		
Group Management		
<input type="button" value="Create Group"/> <input type="checkbox"/> Display Inactive Groups <input type="text" value="Filter group list..."/>		
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 'Administration / Group Management' section for a group named 'Sample Group'. The 'Group Details' section includes a 'Group Name' field set to 'Sample Group' and an 'Active Group' checkbox checked. Below are sections for 'Roles' (listing BOM Manager, Code Scanner, Policy Manager, and System Administrator) and 'Group Members' (with a '+ Add Member' button and a table showing 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.

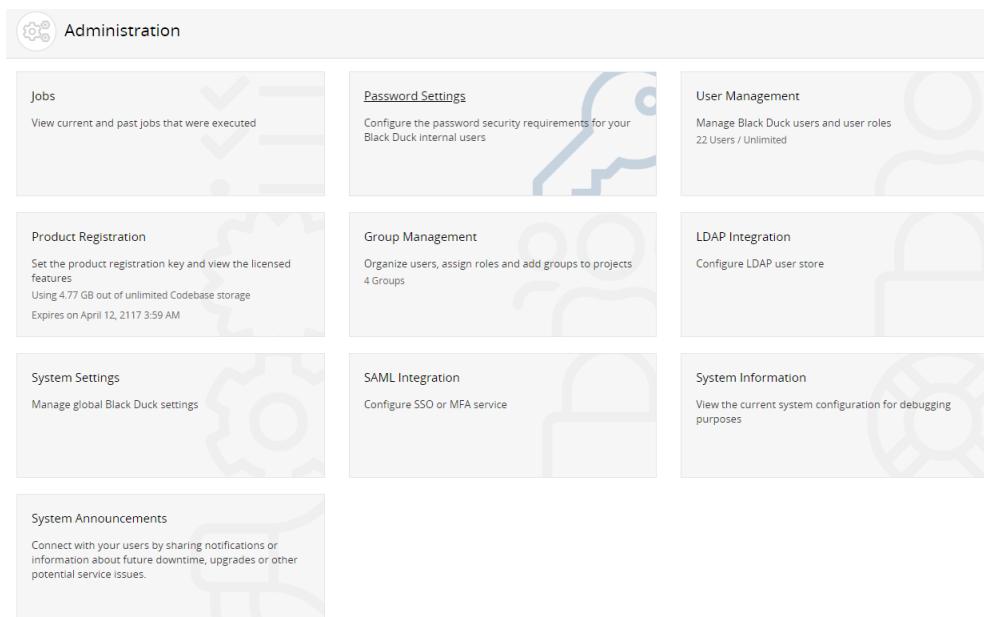
To add a member to a group by managing a user

1. Log in to Black Duck.



2. Click **Admin**.

The Administration page appears.



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 "User Status: Active" and a "Filter user list..." button. Below the search bar is a "Create User" button. The main area displays a table of users:

Username	First Name	Last Name	Email	Roles	Status
sysadmin			noreply@blackducksoftware.com	BOM Manager, Code Scanner, License Manager, Policy Manager, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23	test123@bds.com		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.

Administration / User Management
Sample User

User Details

Username * SampleUser

First Name * Sample

Last Name * User

Email * username@company.com

Active user Inactive user

Save

Roles

- BOM Manager**
BOM manager is granted the privilege of editing BOM
- Code Scanner**
Manages code-related scans.
- Policy Manager**
Policy Manager is granted the privilege of editing policy rules.
- System Administrator**
System Administrator are granted the privilege of editing users and other system settings.

User Groups

Add group

Group Name	Source	Status	Roles
Sample Group	Internal	Active	Policy Manager

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.
- Select the **Active only** check box to see active groups only.
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 Administration page appears.

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

Group Name	Source	Status
Sample Group	Internal	Active
sysadminingroup	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 'Administration / Group Management' page for a group named 'Sample Group'. The 'Group Details' section includes a 'Group Name' input field containing 'Sample Group', an 'Active Group' checkbox which is checked, and two buttons: a red 'Delete Group' button and a blue 'Save' button. Below this is a 'Roles' section listing four roles with descriptions: BOM Manager, Code Scanner, Policy Manager, and System Administrator. At the bottom is a 'Group Members' section with a table header ('Username', 'First Name', 'Last Name', 'Email', 'Status') and a 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.

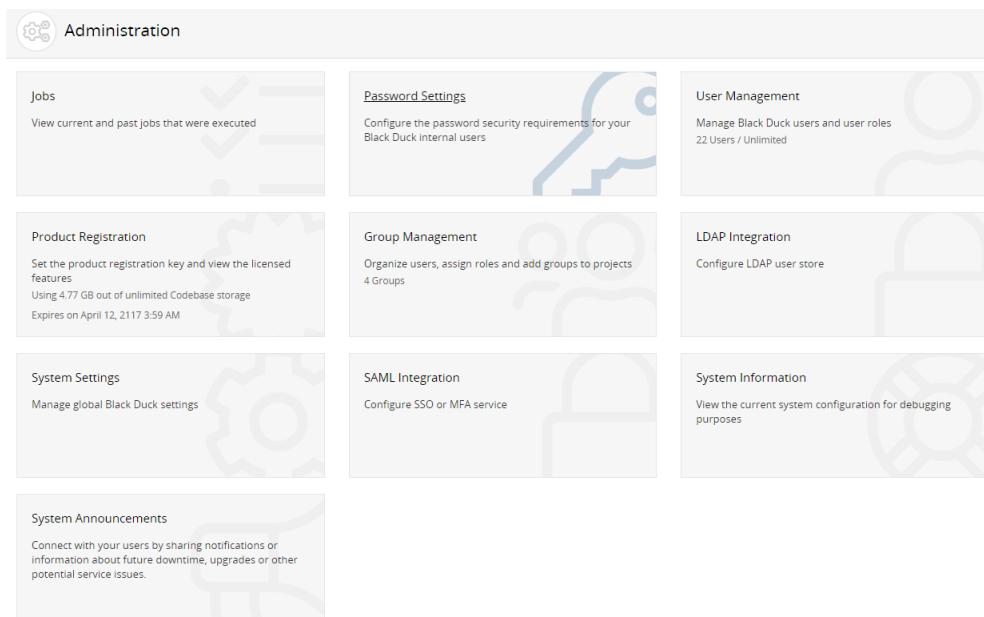
To remove a member from a group while managing a user

1. Log in to Black Duck.



2. Click .

The Administration page appears.



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

Username	First Name	Last Name	Email	Roles	Status
sysadmin			noreply@blackducksoftware.com	BOM Manager, Code Scanner, License Manager, Policy Manager, System Administrator	Active
test	Test	User	test@bds.com	System Administrator	Active
test123	123	23	test123@bds.com		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 'User Management' section of the Black Duck interface. At the top, there's a navigation bar with a user icon and the text 'Administration / User Management'. Below it, the user profile 'Sample User' is displayed. The 'User Details' section contains fields for 'Username' (SampleUser), 'First Name' (Sample), 'Last Name' (User), and 'Email' (username@company.com). There are two checkboxes: 'Active user' (unchecked) and 'Inactive user' (checked). A 'Save' button is located at the bottom right of this section. Below 'User Details' is a 'Roles' section with a list of checkboxes for 'BOM Manager', 'Code Scanner', 'Policy Manager', and 'System Administrator'. The 'System Administrator' checkbox is checked. Under 'User Groups', there's a table with one row: 'Sample Group' (Source: Internal, Status: Active, Roles: Policy Manager). A 'Delete' icon is shown next to the group name.

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.

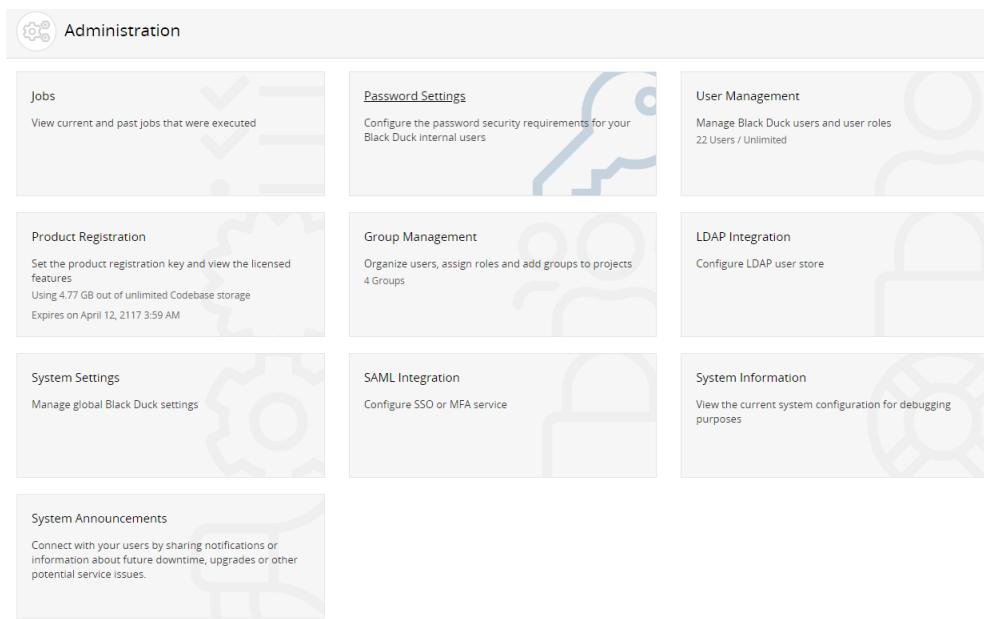
To delete a user group

1. Log in to Black Duck.



2. Click .

The Administration page appears.



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

Administration		
Group Management		
<input type="button" value="Create Group"/> <input checked="" type="checkbox" value="Display Inactive Groups"/> <input type="text" value="Filter group list..."/>		
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.

About Project Groups

Black Duck provides the ability to logically group all your projects in the Hub, allowing you to organize which projects belong to which business unit making it easier for you to view risk across the organization. Project groups can contain both projects and other project groups to provide a multi-level hierarchy.

To view the Project Group Management page:

- ## 1. Log in to Black Duck.



2. Click **Manage** and select **Project Group Management**.

From the Project Group Management page, you will see the root project group for all projects and groups.

Project Group Management

Black Duck Project Groups

Project Group 1

Project Group 2

Project Group 3

Project Group 4

Project Group 5

Black Duck Project Groups

Groups & Projects

Description: No description.

Content: 3 Subgroups | 7 Projects

Created: Updated:

+ Add Group ▾ ▾ Move ▾ Show All Descendants Sort by... Filter results... ▾

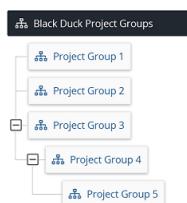
Select All Projects

Project Group	Black Duck Project Groups ▸ Project Group	Subgroups	Projects	Created Date
Project Group 1	Black Duck Project Groups ▸ Project Group 1	0 Subgroups	0 Projects	Created Date: 8/23/2021
Project Group 2	Black Duck Project Groups ▸ Project Group 2	0 Subgroups	0 Projects	Created Date: 8/23/2021
Project Group 3	Black Duck Project Groups ▸ Project Group 3	1 Subgroup	0 Projects	Created Date: 8/23/2021
Monkey - Itacis madoqua	Black Duck Project Groups ▸ Monkey - Itac...		1 Version	Created Date: 8/23/2021
Monkey - melebius kitcheneri	Black Duck Project Groups ▸ Monkey - mel...		1 Version	Created Date: 8/23/2021
Monkey - presbytis calamitorum	Black Duck Project Groups ▸ Monkey - pre...		1 Version	Created Date: 8/23/2021
Monkey - volat trachogrammaticus	Black Duck Project Groups ▸ Monkey - vol...		1 Version	Created Date: 8/23/2021
Monkey - wham pylepheliticus	Black Duck Project Groups ▸ Monkey - wh...		1 Version	Created Date: 8/23/2021
apache-cxf	Black Duck Project Groups ▸ apache-cxf		1 Version	Created Date: 8/23/2021
apache-cxf-2.7.15-addingFunnyChars	Black Duck Project Groups ▸ apache-cxf-2...		1 Version	Created Date: 8/23/2021

Displaying 1-10 of 10

Project Group Hierarchy

The top level of the hierarchy, or the root level, is the main level for all subsequent projects and project groups for your organization. By default, it is named "Black Duck Project Groups", but can be changed at any time.



Ancestor, Parent, and Child Projects and Project Groups

In the example above, you can see a number of levels which may act as individual projects or other project groups. Projects 1, 2, and 3 are considered children of the root level project group. The same can be said for Project 4 and Project 5 in relation to Project 3. Project 5 is also a child of Project 4.

The inverse relationship is called a parent. For example, Project 4's parent is Project 3 and Project 5's parent is Project 4.

An ancestor is any project group existing above the parent for that project group. Project 3 is an ancestor of Project 5.

Members and User Groups

The relationship between projects and project groups matters when assigning members and user groups to project groups. Members and user groups can be assigned to project groups with any number of roles. That assignment will give those users access to the project or project group they are directly assigned to (Direct Access), and to all child projects and project groups of that group with the specified roles unless that assignment is explicitly overridden at the lower levels (Indirect Access). This concept allows for setting users with default access to projects that haven't been created yet. For more details regarding user roles, see [Understanding roles](#).

Black Duck Projects > Project 3			
Members			
+ Add Member			
User Name	Direct Access	Indirect Access	Status:
Bom	Yes	No	<input checked="" type="checkbox"/> Active 
Copyright	Yes	No	<input checked="" type="checkbox"/> Active 
sysadmin	No	Yes	<input checked="" type="checkbox"/> Active 

Displaying 1-3 of 3

In the example above, users Bom and Copyright have been added as members with Direct Access. The sysadmin user is a member of the root level project group, therefore has Indirect Access to all child projects and project groups, including Project 3.

Black Duck Projects > Project 3 > Project 4			
Members			
+ Add Member			
User Name	Direct Access	Indirect Access	Status:
Bom	No	Yes	<input checked="" type="checkbox"/> Active 
Copyright	No	Yes	<input checked="" type="checkbox"/> Active 
sysadmin	No	Yes	<input checked="" type="checkbox"/> Active 

Displaying 1-3 of 3

As a result, users Bom and Copyright now have Indirect Access to Project 4 as seen above. This extends to all children of Project 3. They will hold the same role for all child projects and project groups of Project 3.

Creating a Project Group

You can create a project group by following these steps:

1. Log in to Black Duck.



2. Click **Manage**.

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

Description	Content	Created	Updated
No description.	3 Subgroups 7 Projects	Sort by...	Filter results...
+ Add Group	+ More		
Select All Projects			
Project Group 1	Black Duck Project Groups ▸ Project Grou... 0 Subgroups 0 Projects	Created Date: 8/23/2021	...
Project Group 2	Black Duck Project Groups ▸ Project Grou... 0 Subgroups 0 Projects	Created Date: 8/23/2021	...
Project Group 3	Black Duck Project Groups ▸ Project Grou... 1 Subgroup 0 Projects	Created Date: 8/23/2021	...
Monkey - itasca madagascariensis	Black Duck Project Groups ▸ Monkey - ita... 1 Version	Created Date: 8/23/2021	...
Monkey - meleagris gallopavo	Black Duck Project Groups ▸ Monkey - mel... 1 Version	Created Date: 8/23/2021	...
Monkey - presbytis calamita	Black Duck Project Groups ▸ Monkey - pre... 1 Version	Created Date: 8/23/2021	...
Monkey - volvulus tragicomicus	Black Duck Project Groups ▸ Monkey - vol... 1 Version	Created Date: 8/23/2021	...
Monkey - white-cheeked gibbon	Black Duck Project Groups ▸ Monkey - whi... 1 Version	Created Date: 8/23/2021	...
apache-cxf	Black Duck Project Groups ▸ apache-cxf 1 Version	Created Date: 8/23/2021	...
apache-cxf-2.7.15-edn@fuzzyChar	Black Duck Project Groups ▸ apache-cxf-2... 1 Version	Created Date: 8/23/2021	...

Displaying 1-10 of 10

4. Click [Manage](#) and select **Groups and Projects** from the dropdown menu.
5. Click [+ Add Group](#) and select **Create New...** from the dropdown menu.
6. In the Create New Project Group dialog box:
 - a. Type the name of the group in the **Group Name** field. This field is mandatory.
 - b. Type a description for the Project Group in the **Description** field. This field is optional.
 - c. Click **Save**. The Project Group Management page updates to display the new group.

You can now:

- [Add members and user groups](#) to the project group.

Editing a Project Group

Once you have created a project group, you can add project and project group children, individual members, and/or user groups. You can also change the project group's name or description. To do so, follow the steps listed below:

1. Log in to Black Duck.



2. Click [Manage](#).

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

The screenshot shows the 'Project Group Management' interface. On the left, a tree view displays 'Black Duck Project Groups' with several subgroups: 'Project Group 1', 'Project Group 2', 'Project Group 3', 'Project Group 4', and 'Project Group 5'. The 'Project Group 1' node is expanded. On the right, a table lists project groups and their details:

Project Group	Description	Content	Created	Updated
Project Group 1	No description.	3 Subgroups 7 Projects		
Project Group 2				
Project Group 3				
Monkey - itasca madoqua	Black Duck Project Groups > Project Group 1 > Monkey - itasca madoqua	0 Subgroups 0 Projects	Created Date: 8/23/2021	
Monkey - melobiso kitchenet	Black Duck Project Groups > Project Group 1 > Monkey - melobiso kitchenet	0 Subgroups 0 Projects	Created Date: 8/23/2021	
Monkey - presbytis calamitoria	Black Duck Project Groups > Project Group 1 > Monkey - presbytis calamitoria	0 Subgroups 0 Projects	Created Date: 8/23/2021	
Monkey - vollette tragicomicaria	Black Duck Project Groups > Project Group 1 > Monkey - vollette tragicomicaria	0 Subgroups 0 Projects	Created Date: 8/23/2021	
Monkey - sham pyrophilic	Black Duck Project Groups > Project Group 1 > Monkey - sham pyrophilic	0 Subgroups 0 Projects	Created Date: 8/23/2021	
apache-odf	Black Duck Project Groups > Project Group 1 > apache-odf	0 Subgroups 0 Projects	Created Date: 8/23/2021	
apache-odf2.7.15-addingFunnyChars	Black Duck Project Groups > Project Group 1 > apache-odf2.7.15-addingFunnyChars	0 Subgroups 0 Projects	Created Date: 8/23/2021	

At the bottom, a note says 'Displaying 1-10 of 10'.

Editing the name or description

By default the root level project group is called "Black Duck Project Groups" but it can be renamed.

1. Click **Manage** and select **Settings** from the dropdown menu.
2. Edit the name of the project group in the **Group Name** field. This field is mandatory.
3. Edit the description for the project group in the **Description** field. This field is optional.
4. Click **Save**. The Project Group Management page updates to display the new group.

Adding project sub-groups

1. Click **Manage** and select **Groups and Projects** from the dropdown menu.
2. Click **+ Add Group** and select **Create New...** from the dropdown menu.
3. In the Create New Project Group dialog box:
 - a. Type the name of the project group in the **Group Name** field. This field is mandatory.
 - b. Type a description for the project group in the **Description** field. This field is optional.
 - c. Click **Save**. The Project Group Management page updates to display the new group.

Removing project sub-groups

1. Select the desired project group from the project group tree in the left-hand panel. This displays all child project groups in the right-hand panel.
2. Click **⊖**.

3. Select **Delete** from the dropdown menu.
4. Click **Delete** from the confirmation dialog box.

If the project group is a child of a parent group:

1. Select the parent of the desired project group from the project group tree in the left-hand panel. This displays all project sub-groups for that project group in the right-hand panel.
2. Click 
3. Select **Delete** from the dropdown menu.
4. Click **Delete** from the confirmation dialog box.

Moving a project group to a different project group

1. Select the parent of the desired project group from the project group tree in the left-hand panel. This displays all child project groups for that project group in the right-hand panel.
2. Click 
3. Select **Move**
4. Select a project group game from the Group Name dropdown menu presented in the **Move Selected Group to...** dialog box.
5. Click **Save** to confirm the move.

Moving another project group into the selected group

1. Select the project group from the project group tree in the left-hand panel. This will display the details for the project group itself.
2. Click  
3. Select **Move existing...**
4. Select a project group game from the Group Name dropdown menu presented in the **Move Selected Group to...** dialog box. Please note, a project group cannot be moved into the selected project group if it is an ancestor of the selected project group.
5. Click **Save** to confirm the move.

Adding a member to a project group

1. Select the desired project group from the project group tree in the left-hand panel.
2. Click   and select **Members** from the dropdown menu.
3. Click  
4. Type or select a user name from the Users dropdown menu to open a list of members.
5. Select any role(s) that user will have for that project group. For more details regarding user roles, see

[Understanding roles.](#)

6. Click **Save**.

Removing a member from a project group

1. Select the desired project group from the project group tree in the left-hand panel.
2. Click  and select **Members** from the dropdown menu.
3. Click .
4. Select **Delete Direct Access**.
5. Click **Delete** from the confirmation dialog box.

Editing a member's roles in a project group

1. Select the desired project group from the project group tree in the left-hand panel.
2. Click  and select **Members** from the dropdown menu.
3. Click .
4. Select **Edit Direct Access**.
5. Add or remove any role(s) that user will have for that project group. For more details regarding user roles, see [Understanding roles.](#)
6. Click **Save**.

Adding a user group to a project group

1. Select the desired project group from the project group tree in the left-hand panel.
2. Click  and select **User Groups** from the dropdown menu.
3. Click  **+ Add User Group**.
4. Type or select a user name from the User Group dropdown menu to open a list of user groups.
5. Select any role(s) that user group will have for that project group. For more details regarding user roles, see [Understanding roles.](#)
6. Click **Save**.

Removing a user group from a project group

1. Select the desired project group from the project group tree in the left-hand panel.
2. Click  and select **User Groups** from the dropdown menu.
3. Click .

4. Select **Delete Direct Access**.
5. Click **Delete** from the confirmation dialog box.

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 super user [role](#) can:

- [Create, edit, or delete](#) 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:

- Custom fields can be optional or required.

When creating a custom field, you can select whether the custom field is required.

For required custom fields:

- You can determine the enforcement of required custom fields as they can be mandatory or not mandatory. By default, If you select that a custom field is required, it is not mandatory: users can still view and save non-custom field information and information for non-required custom fields on the page if data is not entered for the required custom field. With mandatory custom fields, users *must* enter values when editing objects which have required custom fields.

To enable mandatory required custom fields:

1. Select **Custom Fields** from the System Settings page.

2. Enable the **Force Entry of Required Custom Fields** option.
- A warning message "Additional fields are required" appears when viewing custom field information, as described in the next section.
 - An asterisk (*) next to the custom field label on the page indicates the required custom field.
 - Use the "Missing Custom Field Data" filter in the BOM to view those components in the project version BOM which are missing information.
- 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.
 - You can create a policy rule for project or BOM component custom fields for any 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 Information icon (ⓘ) can indicate that there are custom fields for this component and information for the custom field has been added. Hover over the icon to see whether **Has Additional Fields** appears which indicates custom field information is available.

Users with the BOM Manager, Super User, or Project Manager role can update or edit the values for a BOM Component custom field.

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**.

Click ⓘ 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:

The screenshot shows the Apache Lucene component details page. At the top, there's a navigation bar with the Apache logo, the text "lucene.apache.org", "Apache Lucene", and "Versions: 1049". Below this is a "Component Details" tab. The main area contains fields for "Component Name" (set to "Apache Lucene"), "Description" (a text area containing a brief description of Apache Lucene), "Url" (set to "http://lucene.apache.org/"), "Notes" (an empty text area), and "Status" (set to "Unreviewed"). A "Save" button is located at the bottom right. Below the main form, there's a section titled "Additional Fields" with a note "* Additional fields are required". Underneath this note, there's a question "Select the team that originally added this component" followed by three radio buttons: "Architecture", "Maintenance", and "New Development". The third option, "Empty Value (Unselect option)", is selected. Another "Save" button is located at the bottom right of this section.

Users with the Super User or Component Manager role can update or edit the values for a Component custom field.

- Component version custom field information is shown in the **Additional Fields** section of the *Component Name Version Name Settings* tab:

The screenshot shows the OpenSSL component version details page. At the top, there's a navigation bar with the OpenSSL logo, the text "http://www.openssl.org/", "OpenSSL • 0.8.1b", and "Versions: 360". Below this is a "Component Details" tab. The main area contains fields for "Version" (set to "0.8.1b"), "License" (set to "SSLeay License"), "Release Date" (set to "12/21/1998"), "Notes" (an empty text area), and "Status" (set to "Unreviewed"). A "Save" button is located at the bottom right. Below the main form, there's a section titled "Additional Fields" with a note "* Additional fields are required". Underneath this note, there's a question "Select the team who originally added this component version" followed by three radio buttons: "Architecture", "Maintenance", and "New Development". The second option, "Maintenance", is selected. Another "Save" button is located at the bottom right of this section.

Users with the Super User or Component Manager role can update or edit the values for a Component Version custom field.

Once you have selected values for the custom fields, the information appears on the *Component Name Version Name Details* tab:

Description
Apache Lucene is a high-performance, full-featured text search engine library written entirely in Java. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform.

Released	Newer Versions	Status	Updated
Dec 9, 2008	986	Unreviewed	Jun 23, 2020

Activity	Community
Last 12 Months: 1833 commits ↑ increasing Last commit: Jun 23, 2020	Last 12 Months: 136 contributors

Where Used	Project	Version	Released	Phase
tut-ir	1	Never	In Development	

Displaying 1-1 of 1

Security 0 Vulnerabilities

Licenses
Apache License 2.0

Open Hub
<https://www.openhub.net/p/3564>

Component Links
<http://lucene.apache.org/>

Tags
apache, apache_software_foundation, documents, fulltext_search, index, indexer, indexing, information_retrieval, java, lucene, search, search_engine, searchengine

Additional Fields
Select the team that originally added this component version
Architecture

- Project custom field information is shown in the **Additional Fields** section of the *Project Name Settings* tab:

Additional Fields

Has the attorney approved all licenses?

Select the team responsible for this project
 New features team
 Architecture team
 Maintenance development team

Save

Application ID
A field that can be used to store an external mapping id for the project to an external system, like an asset management system or application catalog.

Application ID

Save

Users with the Super User or Project Manager role can update or edit the values for a Project custom field.

Once you have selected values for the custom fields, the information appears on the *Project Name Overview* tab:

Black Duck Projects
bom2
Project Not Watching Project | Versions: 2

+ Create Version Filter versions... Add Filter ▾

Version	Phase	Last Updated	License	Security Risk	License Risk	Operational Risk
1	In Development	Nov 18, 2020	Unknown License	██████	██████	██████
2	In Development	Nov 18, 2020	Unknown License	██████	██████	██████

Displaying 1-2 of 2

Description
No description.

Created
Aug 17, 2020 by sysadmin

Updated
Aug 17, 2020 by sysadmin

Tags
No Tags

Additional Fields
Has Legal approved this project?
true

- Project version custom field information is shown in the **Additional Fields** section of the *Project Name Version Name Settings* tab:

Black Duck Projects
Custom Sample ▸ 1.0
Project Versions: 1 | Phase: In Planning | Distribution: External

Components Security Source Reports Details Settings

Version Details >

Scans	Version +	1.0
	License	Unknown License
	Notes	
	Nickname	
	Release Date	
	Phase	In Planning
	Distribution	External

Save

Additional Fields

Enter the GA date for this project version 02/14/2019

Save

Users with the Super User or Project Manager role can update or edit the values for a Project Version custom field.

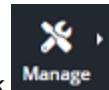
Creating a custom field

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



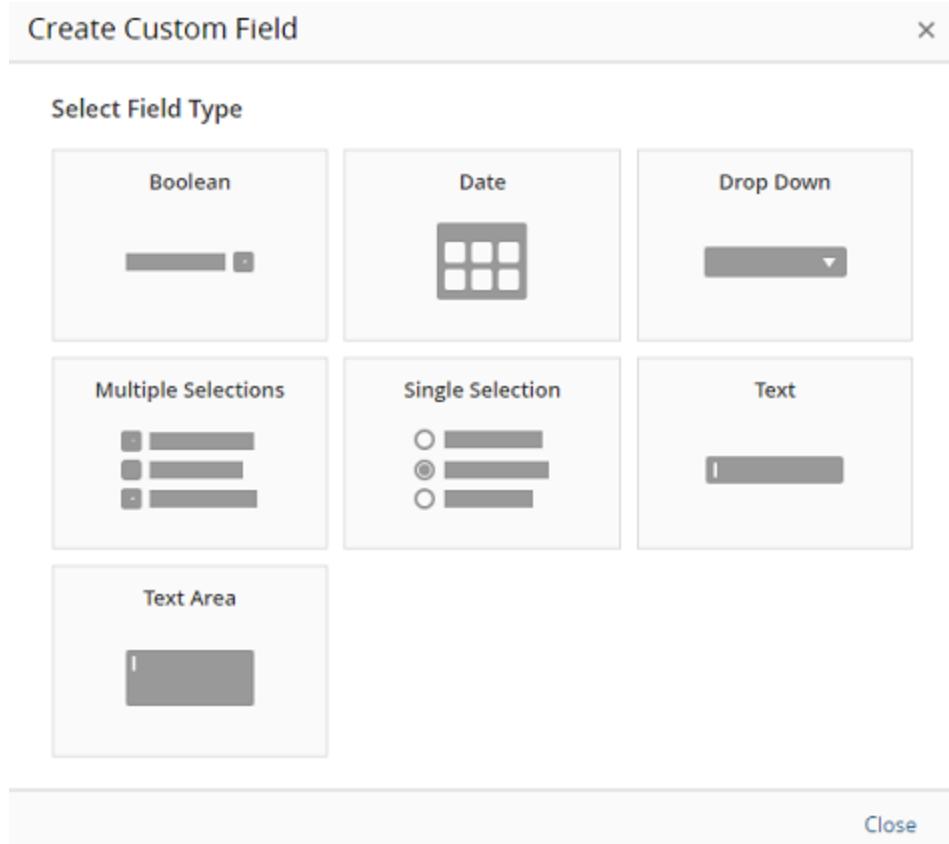
1. Click **Manage** > **Custom Fields Management**.

The Custom Fields Management page appears.

A screenshot of a web-based application interface. At the top left is a wrench and screwdriver icon next to the word "Administration". Below that is the title "Custom Fields Management". A navigation bar at the top has tabs for "Project" (which is selected, indicated by a blue background) and "Project Version". To the right of the tabs is a button labeled "+ Create". On the right side of the page, the text "No Results Found" is displayed, followed by the message "It looks like you haven't created custom fields yet.".

By default, the **BOM Component** tab is selected.

2. 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.



3. Select the type of custom field. The types of custom fields are:

- **Boolean.** A drop-down list appears to the user from which they can select **True** or **False**. The user can clear the option after one is selected.
- **Date.** A calendar appears to the user from which they can select a 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. There is also an option to clear the selected value.
- **Text.** A field appears to the user where they can enter text. There is no limit to the number of characters the user can enter for this field.
- **Text Area.** A field appears to the user where they 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.

4. 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.
- **Make this field required/recommended.** Select this option to indicate to users that information for this custom field is required/recommended.

Note that the display for this option changes depending on whether the **Force Entry of Required Custom Fields** is enabled. If Force Entry of Required Custom Fields is enabled, this option displays **Make this field required**. If **Force Entry of Required Custom Fields** is disabled, this option displays **Make this field recommended**.

While this indicates that the custom field is required, it acts more as a warning, as users can still view and save non-custom field information and information for non-required custom fields on the page without entering information for the required custom field.

- 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.
5. 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.
6. 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.

Tip: If you cannot [delete a custom field](#), deactivate it so that the field no longer appears to your users.

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 **Manage** > **Custom Fields Management**.

The Custom Fields Management page appears.

Project Version	Label	Type	Last Modified	Active
	Make sure that the attorney has approved all licenses.	Text	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>
	Has Development VP signed off on this project?	Boolean	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>
	Select the team responsible for this project	Single Selection	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>

2. Select the tab which contains the custom field.
 3. 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.

Project Version	Label	Type	Last Modified	Active
	Make sure that the attorney has approved all licenses.	Text	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>
	Has Development VP signed off on this project?	Boolean	02/13/2019 - sysadmin	<input type="checkbox"/>
	Select the team responsible for this project	Single Selection	02/13/2019 - sysadmin	<input type="checkbox"/>

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

For all custom fields, you can edit the label, description, and the designation whether this custom field is required. For drop down, single, and multiple selection custom fields, you can:

- rearrange the order of options
- edit existing options
- add new options

You cannot delete existing options.

Edits made to options will propagate to any policies.

Note that you cannot change the type of custom field once it has been created. For example, suppose you created a multiple selections custom field. If, after you created the field, you want to change that custom field to a single selection custom field, you must create a new custom field.

To edit a custom field



1. Click **Manage** > **Custom Fields Management** to display the Custom Fields Management page.

Project	Label	Type	Last Modified	Active
Project Version	Make sure that the attorney has approved all licenses.	Text	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>
	Has Development VP signed off on this project?	Boolean	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>
	Select the team responsible for this project	Single Selection	02/13/2019 - sysadmin	<input checked="" type="checkbox"/>

2. Select the tab which contains the custom field you want to edit.

3. Click and select **Edit** In the row of the custom field.

4. In the Edit Custom Field dialog box, modify the custom field, and click **Save**.

Deleting a custom field

Deleting a custom field removes the custom field and all data associated with it.

Note: You can [deactivate a field](#) so that it retains its data but no longer appears to your end users.

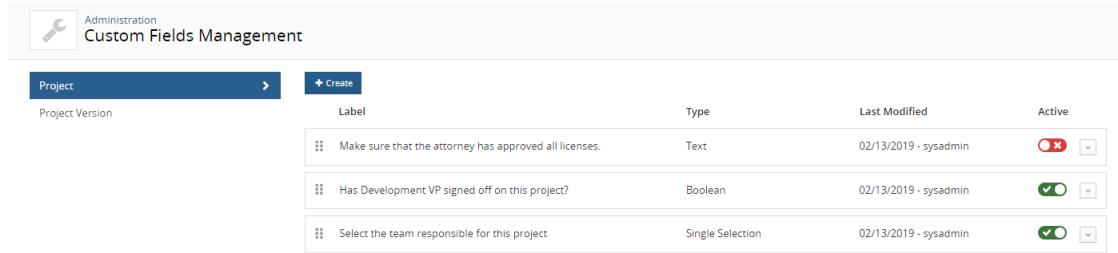
You must have the system administrator role to delete a custom field.

To delete a custom field



1. Click **Manage** > **Custom Fields Management**.

The Custom Fields Management page appears.



Label	Type	Last Modified	Active
Make sure that the attorney has approved all licenses.	Text	02/13/2019 - sysadmin	<input checked="" type="checkbox"/> <input type="checkbox"/>
Has Development VP signed off on this project?	Boolean	02/13/2019 - sysadmin	<input checked="" type="checkbox"/> <input type="checkbox"/>
Select the team responsible for this project	Single Selection	02/13/2019 - sysadmin	<input checked="" type="checkbox"/> <input type="checkbox"/>

2. Select the tab which contains the custom field you want to delete.
3. Click  and select **Delete** in the row of the custom field.
4. In the Delete Custom Field dialog box, confirm that you have selected the correct custom field to delete, and click **Delete**.

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.](#)

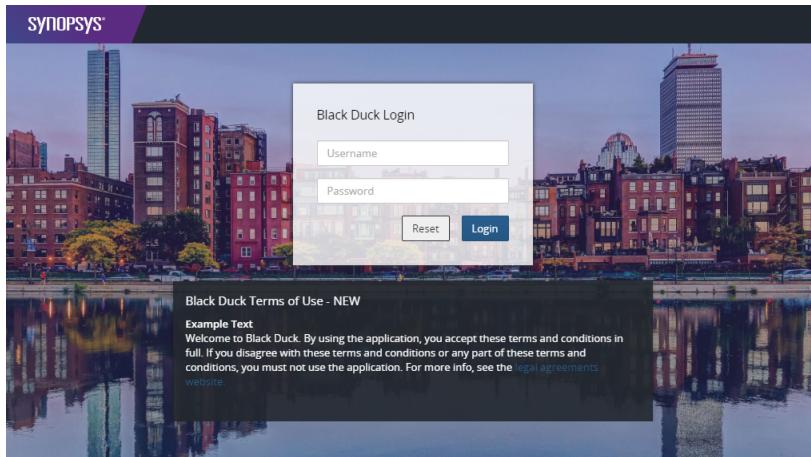
Creating system announcements

System Administrators can create custom sign-on and post sign-on messages to your Black Duck users.

For example, use system announcements to tell your users about upcoming events or if you need to show a disclaimer indicating what happens for unauthorized use.

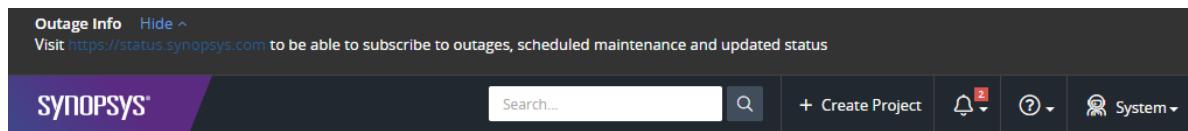
There are four types of messages that you can create:

- Login. A message that appears to the user when they are logging in to Black Duck.

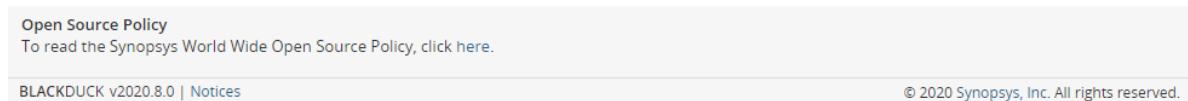


Since this message appears for all users - including unauthenticated users - Synopsys recommends that you do not use this type of system announcement to display sensitive information.

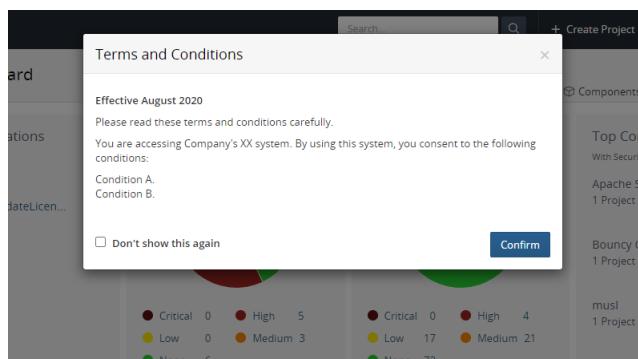
- **Banner.** A message that appears at the top of every page.



- **Footer.** A message that appears in the footer of every page.



- **Welcome.** A message that appears after the user logs in to Black Duck.



Unlike other announcements, you can provide an option so that users can suppress this message: users will not see this message again unless you edit the message.

Note that you can only create one announcement of each type.

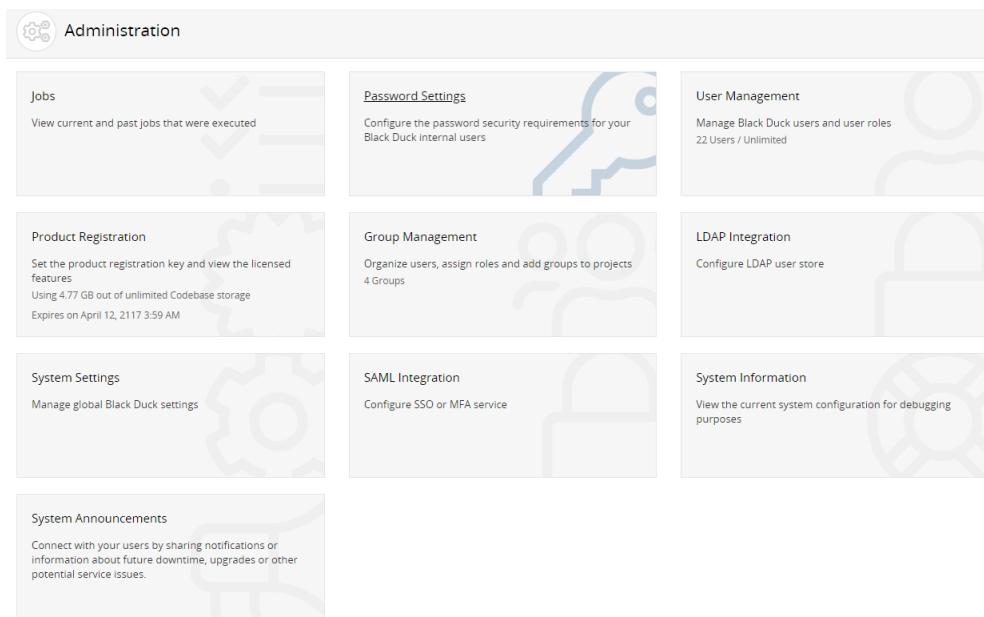
To create a system announcement

1. Log in to Black Duck with the System Administrator role.



2. Click **Admin**.

The Administration page appears.



3. Select **System Announcements** to display the System Announcements page.

A screenshot of the "System Announcements" edit page. It shows a sidebar with announcement types: Login (selected), Banner, Footer, and Welcome. The main area has fields for "Title" (with a placeholder "Title *") and "Announcement Text" (with a rich text editor, "Edit" button, and "Preview" button). Below these are "Start Date" (8/6/2020) and "End Date" (No End Date) fields, a note about leaving dates empty for persistent announcements, and a checked "Enabled" checkbox. At the bottom are "Reset" and "Save" buttons.

4. Select the type of announcement.
5. Enter a title for this announcement.
6. Enter the announcement text. If not selected, click **Edit** and enter the text. Click **Preview** to view your announcement as it will appear to your users.

You must use markdown language when creating the announcement. See the next section for more information.

7. For the Welcome announcement, select whether the user can suppress the announcement.
If you select to suppress the announcement, the user will see the following option ("Don't show this again") in the announcement. However, the announcement will reappear to the user if you make any changes to the announcement.
8. Optionally, enter the start and end date for this announcement. By default the start date is today, with no end date, indicating a persistent announcement. Dates are inclusive: the announcement will appear for the date range you select here, including the start and end dates.
9. Select whether to enable the message. Once enabled, the announcement will appear to users once you click **Save**.
10. Click **Save**.

Markdown language

You must use markdown language when creating the announcement.

Click [here](#) for more information on the syntax for markdown language.

The following is the list of allowable tags for system announcements:

- h1, h2, h3, h4, h5, h6

Note that h1 and h2 tags are converted to h3 tags.

- blockquote
- p
- a
- ul
- ol
- nl
- li
- b
- i
- strong
- em
- strike
- abbr
- code
- hr
- b
- table
- thead
- caption

- tbody
- tr
- th
- td
- pre
- iframe
- Anchor tags <a> are only allowed with the following attributes:
 - href
 - name
 - target

Note that images are not allowed.

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.

- Log in to Black Duck.
- Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
- 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 in the sidebar. The main area displays a table of events:

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:

Change	Component Added	User: blackduck_system	Mon, Mar 4, 2019 12:33 PM
Source: KnowledgeBase Type: Component Version: 1.4 Is Modified: false Origin External Namespace: maven Origin External Id: javax.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
 - Maximum scan size
- Licensed modules. Available modules are:
 - Black Duck Binary Analysis
 - Black Duck Security Advisory
 - Component Scanning
 - Cryptography
 - Dependency Scanning
 - Enhanced Vulnerability Analysis
 - License Management

- Notifications
- OpsSight
- OSS Notices Report
- Policy Management
- Risk Management
- Signature Scanning
- Snippets

Updating your product registration

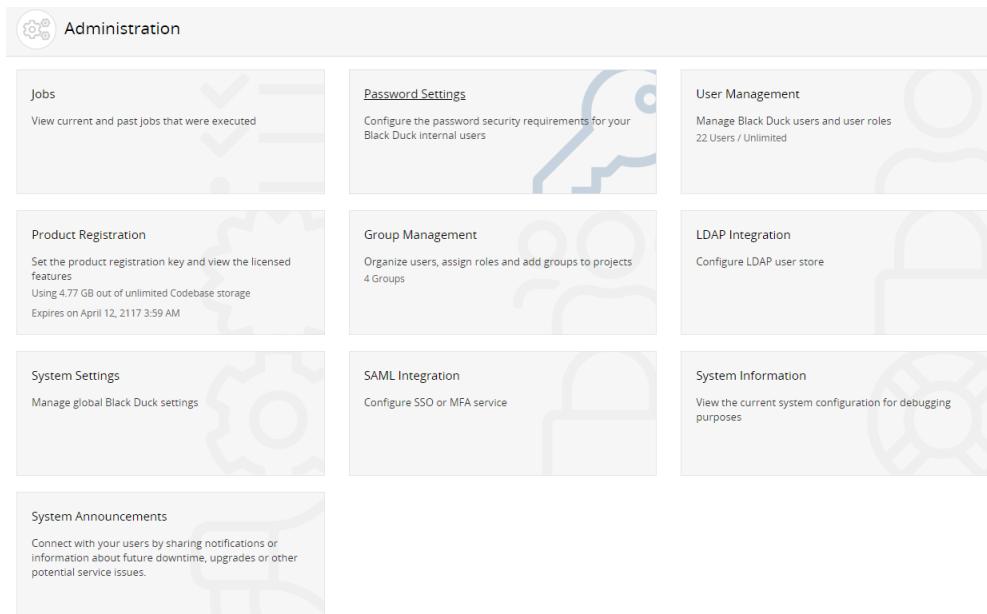
Your Black Duck license may restrict the number of users, projects, and/or project versions. If you need more capacity, you can purchase a new license. Once you receive a new license from Black Duck Software, enter the new registration ID information in Black Duck to activate your newly-licensed capacity.

1. Log in to Black Duck as a system administrator.



2. Click **Admin**.

The Administration page appears.



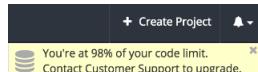
A screenshot of the Black Duck Administration page. The page has a header with the title "Administration". Below the header are nine cards arranged in a grid:

- Jobs**: View current and past jobs that were executed.
- Password Settings**: Configure the password security requirements for your Black Duck internal users.
- User Management**: Manage Black Duck users and user roles. (22 Users / Unlimited)
- Product Registration**: Set the product registration key and view the licensed features. (Using 4.77 GB out of unlimited Codebase storage. Expires on April 12, 2117 3:59 AM)
- Group Management**: Organize users, assign roles and add groups to projects. (4 Groups)
- LDAP Integration**: Configure LDAP user store.
- System Settings**: Manage global Black Duck settings.
- SAML Integration**: Configure SSO or MFA service.
- System Information**: View the current system configuration for debugging purposes.
- System Announcements**: Connect with your users by sharing notifications or information about future downtime, upgrades or other potential service issues.

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 project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
 2. Select the version name which displays the **Components** tab.
 3. Select the **Settings** tab.
 4. Select **Scans** to view the scans mapped to this project version.

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. Click 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 page. At the top, it displays the project name "ComplexBomMainProject" and the scan timestamp "2015-12-04 10:28:23". Below this, the "Scan Details" section provides summary statistics: Path "/", Host "scorpion.blackducksoftware.com", Created on "Mon, Aug 15, 2016 6:06 PM", Scan Size "1.19 MB", Match Count "74", Folders "22", and Files "73". A red "Delete Scan" button is visible. To the right, there's a "Mapped to Project Version" section with a "Sample Project v 3.0" link and a "Unmap from Project" button. A large "Delete" icon is also present. The "Scan History" section shows one entry: "Complete" status, "74 Matches", "scorpion.blackducksoftware.com" host, Path "/", Scan Size "1.19 MB", Last Updated "Tue, Sep 29, 2020 1:17 PM", Initiated By "sysadmin", and a "View BOM Import Log" link. A note at the bottom says "Displaying 1-1 of 1".

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:

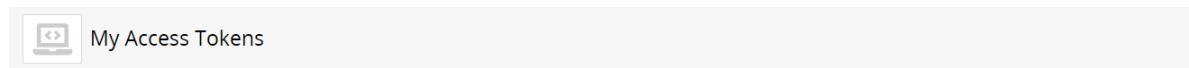
- Access tokens can only be created by the current user.
- Access tokens are tied to a user’s account; therefore, an access token has the same role as the user who created the token.
- A user can have multiple tokens. Each token must have a unique name.
- Access tokens do not expire.
- If a user is inactivated, their tokens are invalidated.

Refer to the *Getting Started with the SDK* guide for information on using the API keys.

To generate an access token

1. Log in to Black Duck.
2. From the user menu located on the top navigation bar, select **My Access Tokens**.

The My Access Tokens page appears.



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.

+ Create New Token

There are currently no user access tokens. Feel free to generate one above!

3. Click **Create New Token**. The Create New Token dialog box appears.



4. Enter a name, description (optional), and select the scope for this token (read or read and write access). You can only select one access for a token.

5. Click **Create**.

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. From the user menu located on the top navigation bar, select **My Access Tokens**.

The My Access Tokens page appears.

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.

+ Create New Token

ReadWrite has been generated.

Name	Description	Scopes
ReadWrite	Read and write access	read, write

Displaying 1-1 of 1

3. Click in the row of the token you want to revise and select **Edit**.

The Edit User Access Token dialog box appears.

4. 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. From the user menu located on the top navigation bar, select **My Access Tokens**.

The My Access Tokens page appears.

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.

+ Create New Token

ReadWrite has been generated.

Name	Description	Scopes
ReadWrite	Read and write access	read, write

Displaying 1-1 of 1

3. Click in the row of the token you want to regenerate and select **Regenerate**.

The Regenerate User Access Token dialog box appears.

4. Click **Regenerate** to confirm.

The *Access Token Name* dialog box appears with the new access token.

5. 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.

6. Click **Close**.

To delete an access token

1. Log in to Black Duck.
2. From the user menu located on the top navigation bar, select **My Access Tokens**.

The My Access Tokens page appears.

The screenshot shows the 'My Access Tokens' page. At the top, there's a header with a key icon and the title 'My Access Tokens'. Below the header, a message states: '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.' A blue button labeled '+ Create New Token' is visible. The main table lists one token:

Name	Description	Scopes
ReadWrite	Read and write access	read, write

A green banner at the bottom of the table area says 'ReadWrite has been generated.' In the bottom right corner, it says 'Displaying 1-1 of 1'.

3. Click in the row of the token you want to remove and select **Delete**.

The Delete User Access Token dialog box appears.

4. 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 **Terms Fulfillment** tab in the *Project Version's Legal* tab.

By default, these tabs are disabled. System Administrators must enable these tabs for them to appear to these users.

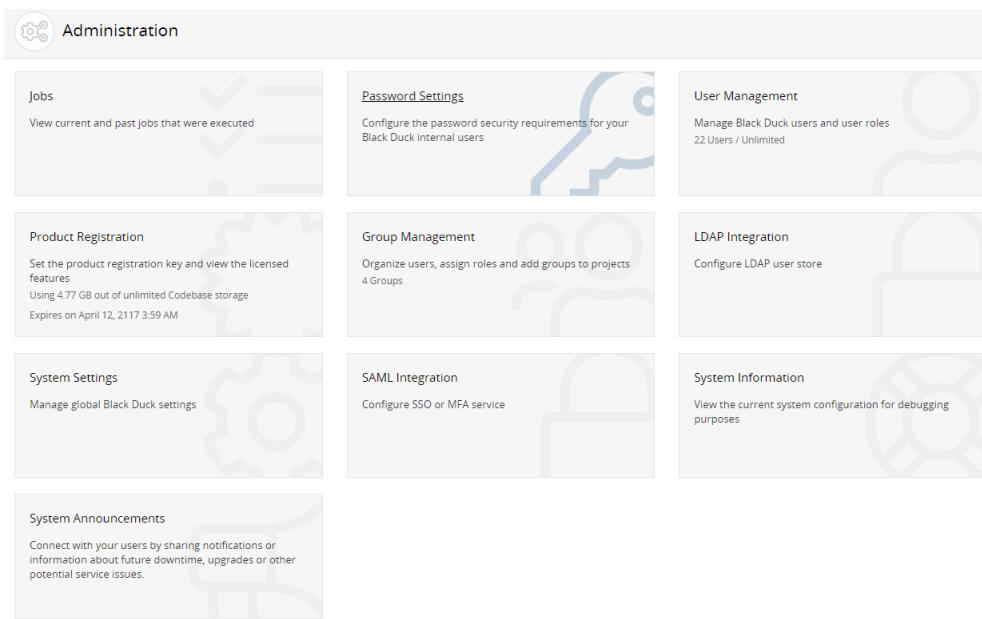
Note: Enabling the **Legal** tab is a global setting. Once enabled, all project versions will display the **Term Fulfillment** tab in the **Legal** tab.

To display the Legal and Term Fulfillment tabs

1. Log into Black Duck with the System Administrator role.

2. Click Admin.

The Administration page appears.



3. Select System Settings.

The System Settings page includes the following sections:

- Logo**: A placeholder for a logo with a "Upload logo" button.
- System Logs**: A link to download logs in a zip file.
- Legal Tab Visibility**: A section with a "Disable" button.
- Security Risk Configuration Ranking**: A list of configurations with a "Save" button.
- Custom Scan Signature Level**: A setting for "Levels" with a value of 5, and a "Save" button.
- Snippet Max File Size**: A setting for "Maximum Snippet File Size" with a value of 2, and a "Save" button.

4. Click **Enable** located in the **Terms Fulfillment** section to display the **Legal** and **Term Fulfillment** tabs.

Click **Disable** to remove the **Legal** and **Term Fulfillment** tabs. Note that if you select to enable license conflicts, the **Legal** tab will still appear.

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 **Admin**.

The Administration page appears.

A screenshot of the Black Duck Administration page. The top navigation bar has a gear icon and the word "Administration". Below it is a grid of nine cards: "Jobs" (View current and past jobs), "Password Settings" (Configure password security requirements), "User Management" (Manage users and roles), "Product Registration" (Set registration key, 4.77 GB storage used), "Group Management" (Organize users, 4 groups), "LDAP Integration" (Configure LDAP user store), "System Settings" (Manage global settings), "SAML Integration" (Configure SSO or MFA), and "System Information" (View current system configuration). A sidebar on the left shows "System Announcements" with a message about future downtime.

3. Select **System Settings** to display the System Settings page.

The screenshot shows the 'System Settings' page under 'Administration'. It includes sections for 'Logo' (with a placeholder for 'SYNOPSYS'), 'System Logs' (with a 'Download Logs (.zip)' button), 'Legal Tab Visibility' (with a 'Disable' button), 'Security Risk Configuration Ranking' (with a list of items: BDSA (CVSS v2), NVD (CVSS v2), BDSA (CVSS v3.x), and NVD (CVSS v3.x), and a 'Save' button), 'Custom Scan Signature Level' (with a 'Levels' input set to 5 and a 'Save' button), 'Snippet Max File Size' (with a note about setting a value from 1 to 16 MB and a 'Save' button), and 'Maximum Snippet File Size' (with a note about setting a value from 1 to 2 and a 'Save' button).

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 **Admin**.

3. Select **Diagnostics**.

4. Select the **System Information** tab.

5. 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
BdioDataTransferJob	Processes scan data and prepares it for the matching process.
BomAggregatePurgeOrphansCheckJob	Checks to see if any BOM data is not associated with a project version and starts the necessary jobs.
BomAggregatePurgeOrphansJob	Deletes any BOM data not associated with a project version.
BomVulnerabilityDataRecomputationCheckJob	Checks if BOM computations are required when certain settings change and starts the necessary jobs.
BomVulnerabilityDataRecomputationJob	Updates component information received from the KnowledgeBase.
BomVulnerabilityNotificationJob	Does the following: <ul style="list-style-type: none"> Creates vulnerability notifications for users. Removes old audit events records from the database that have triggered notifications.
CollectScanStatsJob	Collects scan statistics shown on the usage: scan completion section on the System Information page.
HierarchicalVersionBomJob	Creates and updates the hierarchical version BOM .
HierarchicalVersionBomCheckJob	Checks if hierarchical BOM computations are required and starts the necessary jobs to process them
JobHistoryStatsJob-Calculate Daily Statistics	Calculates daily statistics based on job activity.
JobHistoryStatsJob-Calculate Five Minute Statistics	Calculates statistics in 5-minute intervals based on job activity.
JobHistoryStatsJob-Calculate Hourly Statistics	Calculates statistics in one-hour periods based on job activity.
JobHistoryStatsJob-Prune Job History	Prunes old records from the job history based on the retention settings.
KBUpdateCheckJob	Initiates updates received from the KnowledgeBase.
JobMaintenanceJob	Manages data retention and cleanup for existing jobs.
KbUpdateWorkflowJob-BDSA Vulnerability Update	Updates BDSA vulnerability information received from the KnowledgeBase.
KbUpdateWorkflowJob-Component Update	Updates component information received from the KnowledgeBase.
KbUpdateWorkflowJob-Component Version Update	Processes component version updates received from the KnowledgeBase.
KbUpdateWorkflowJob-License Update	Updates license information received from the KnowledgeBase.

Job Name	Description
KbUpdateWorkflowJob-NVD Vulnerability Update	Updates NVD vulnerability information received from the KnowledgeBase.
KbUpdateWorkflowJob-Summary	Issues a summary report about the most recent KnowledgeBase update.
LicenseDashboardRefreshJob	Updates Black Duck with the latest licenses and use counts.
LicenseTermDataPopulatorJob	Updates Black Duck with the latest Black Duck KB license term data.
LicenseTermFulfillmentJob	Does the following: <ul style="list-style-type: none"> Applies license term fulfillment requirements to all BOMs. Removes old license term association audit events from the database.
LicenseTermFulfillmentCheckJob	Checks if license fulfillment processing is required and starts the necessary jobs.
MigratedKbObjectPurgeJob	Cleans up records of KnowledgeBase Migrations that were detected by the system if Quartz is Disabled.
NotificationPurgeJob	Manages data retention for existing notifications.
NotificationPurgeCheckJob	Checks if there are notifications that need cleanup and starts the necessary jobs.
PolicyRuleModificationBomComputationJob	Computes version BOMs affected by changes to policy rules.
QuartzMigratedKbObjectPurgeJob	Cleans up records of KnowledgeBase Migrations that were detected by the system when Quartz is Enabled.
QuartzVersionBomEventCleanupJob	Cleans up BOM events based on the retention policy.
ReportingDatabaseTransferJob	Migrates Black Duck data to the Black Duck reporting warehouse.
ReportPurgeJob	Manages data retention for existing reports.
ScanAutoBomJob	Manages matching and BOM computation for a scan.
ScanPurgeJob	Deletes old scans or BOM imports.
ScanStatisticsPurgeJob	Computes signatures for all the scanned files.
SearchDashboardRefreshJob	Updates the information shown in saved searches shown on the Dashboard page.
SnippetScanAutoBomJob	Manages the matching process for snippet scan signatures.
SystemMaintenanceJob	Maintains system-related activities.
VersionBomComputationCheckJob	Checks if BOM computations are required and starts the necessary jobs to process them.

Job Name	Description
VersionBomComputationJob	Manages version BOM computation.
VersionBomEventCleanupJob	Clears BOM events that may be stuck due to processing errors or topology changes.
VersionBomNotificationCheckJob	Issues notifications for BOM computation results.
VersionLicenseReportJob	Creates the Notices File report .
VersionReportJob	Creates the Project Version report .
VulnerabilityRemediationReportJob	Creates the Vulnerability Remediation Report .
VulnerabilityReprioritizationJob	Recomputes all BOMs with the new vulnerability priority setting.
VulnerabilityStatusReportJob	Creates the Vulnerability Status Report .
VulnerabilitySummaryFetchJob	Locates missing CVSS v3.x data.
VulnerabilityUpdateReportJob	Creates the Vulnerability Update Report .
WatchdogJob	Monitors recurring jobs to ensure they are running properly and reports on or fixes issues as they are determined.

To view jobs

1. Log in to Black Duck with the System Administrator role.



2. Click **Admin**.
3. Click **Diagnostics**.
4. Click the **Jobs** tab to display the Jobs page which is divided into a **Summary** and **Details** section.
 - The **Summary** section lists the jobs for the number of days you are retaining logs (30 days by default). It also provides a description and indicates those jobs that have failed.
 - 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 filters to view specific data:
 - **Job Status**. Select one or more of the following:
 - Pending
 - In progress
 - Complete
 - Error

Periodic jobs that are scheduled in the future have a Pending status in the Black Duck UI.

- **Job Type.**

Note that you cannot filter based on job type for jobs with a Pending status.

- **Schedule Type.** You can filter jobs based on their schedule.

- Periodic. Runs jobs that repeat using a CRON schedule or a repeating interval.
- On Demand. Handles work created by various events in the system, including incoming scans, check jobs, user interactions, and other items.

Appendix A: Understanding how to search in Black Duck

You can search using the following categories:

- **Projects**: These are the 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.
Note: The information that you provide about your projects is not shared outside of your company.
- **Components** and the **Black Duck KnowledgeBase**: These are the components that comprise your projects and are viewable in your BOM or it is the components in the Black Duck KnowledgeBase, which is a comprehensive database of open source software (OSS) components.
- **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.

You can also perform a global search which searches for your term in all categories. Enter the term in the search field located at the top of the page and press **Enter** or click . Select a category to view the search results.

Note that entering a global search term initiates a new search and resets any filters you previously selected on the **Projects**, **Components**, **Vulnerabilities**, or **Black Duck KnowledgeBase** tabs.

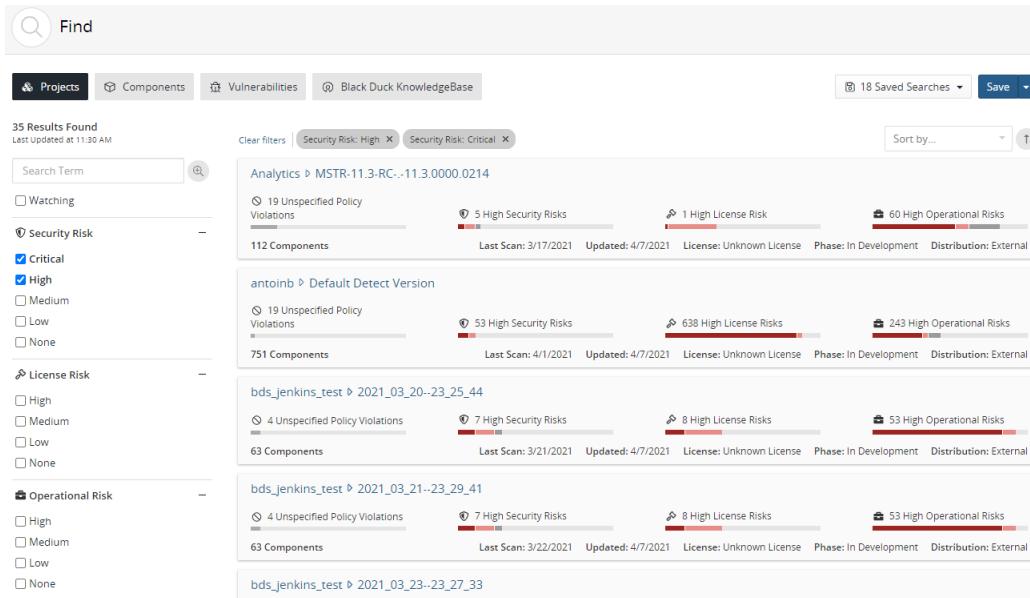
Searching for projects

You can search for project versions that meet your search criteria.

To search for projects

1. Click  to open the Find page and select the **Projects** tab.
2. Type your search term in the Search field.
3. Optionally, select any filters, as described in the next section, "Using search filters."
4. Optionally, [save this search](#), so that you can easily view them on your dashboard.

The Find page displays the project versions that meet your search criteria.



You can also type your search term in the Search field located at the top of the application and press **Enter** or click . The Find page appears displaying the search results. Note that entering a global search term initiates a new search and resets any filters you previously selected. Select the **Projects** tab and filters to refine the results, as described below.

About the search results

Search results show all project versions that meet your search criteria. The following information is shown for each project version:



- Use the bars to quickly view the number of components with the highest level of security, license, or operational risk.

For example, the following shows that while there is a component with lower risk, the highest security risk for this project version is High and that four components in this project version have a high level of security risk as their highest risk level:



- Hover over the bar to see the number of components for each risk category.

Security Risk

by Component



* Each component is counted once by its highest severity risk

In this example, there is one component that has a high risk level as their highest risk. 10 components that have medium risk as their highest risk level, and six components that have low risk as their highest risk level.

Note: Each component is only counted once and is shown with its highest risk severity level.

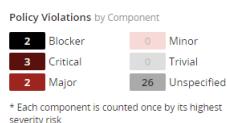
- Use the bar to see the number of components with the highest policy severity level for this project version.

For example, the following shows that while there are components with lower severity levels, the highest policy severity level for this project version is Blocker and there is one component that has Blocker as its highest policy severity level.

1 Blocker Policy Violation

Note: The text shown states the number of components with the highest policy severity level for this project version, not all policy severity levels affecting this project version.

- Hover over the bar to see the number of components with policy violations by severity level:



- View the number of results found and the time the database was last updated:

14 Results Found
Last Updated at 1:27 PM

- For each project version, the search results also show:

- Number of components in this project version.
- Last scan date.
- When this project version was last updated.
- License of this project version.
- Phase for this project version.

- Distribution of this project version.
- Select the project or version name to view the BOM.

Using search filters

If your search query returns many projects, use filters to narrow your results.

Note that:

- Where necessary, click + to display the filter values; click - to hide them.
- If you select more than one type of filter, Black Duck displays items that match *all* values. If you select more than one value for a specific filter, Black Duck displays items that match either value.

For example, if you use the License Risk filter and select high and medium, the search results display all projects that have high *or* medium license risk. If you select a high License Risk filter and a critical Security Risk filter, the search results display only those projects that meet have a high license risk *and* critical security risks.

Possible project filters are:

- Never Scanned. Select whether this project has never been scanned.
- Watching. Select whether this project is a watched project.
- Security Risk. Select one or more security risk levels.
- License Risk. Select one or more license risk levels.
- Operational Risk. Select one or more operational risk levels.
- Policy Rule. Select a policy rule from the list to find the projects that violate this policy.
- Policy Violation. Severity level of the policy rule.
- Distribution. Select one or more distribution methods.
- Last Scanned Date. Select a time period when this project was last scanned.
- Not Scanned Since. Select the time period since this project was last scanned.
- Release Phase. Select one or more release phases.
- Tier. Select one or more tiers.

Sorting the search results

Optionally, you can sort the results that appear on the page by selecting a value from the **Sort by** list:



Note that if you sort the results and save this search, the Dashboard page displays the saved search in the sorted order.

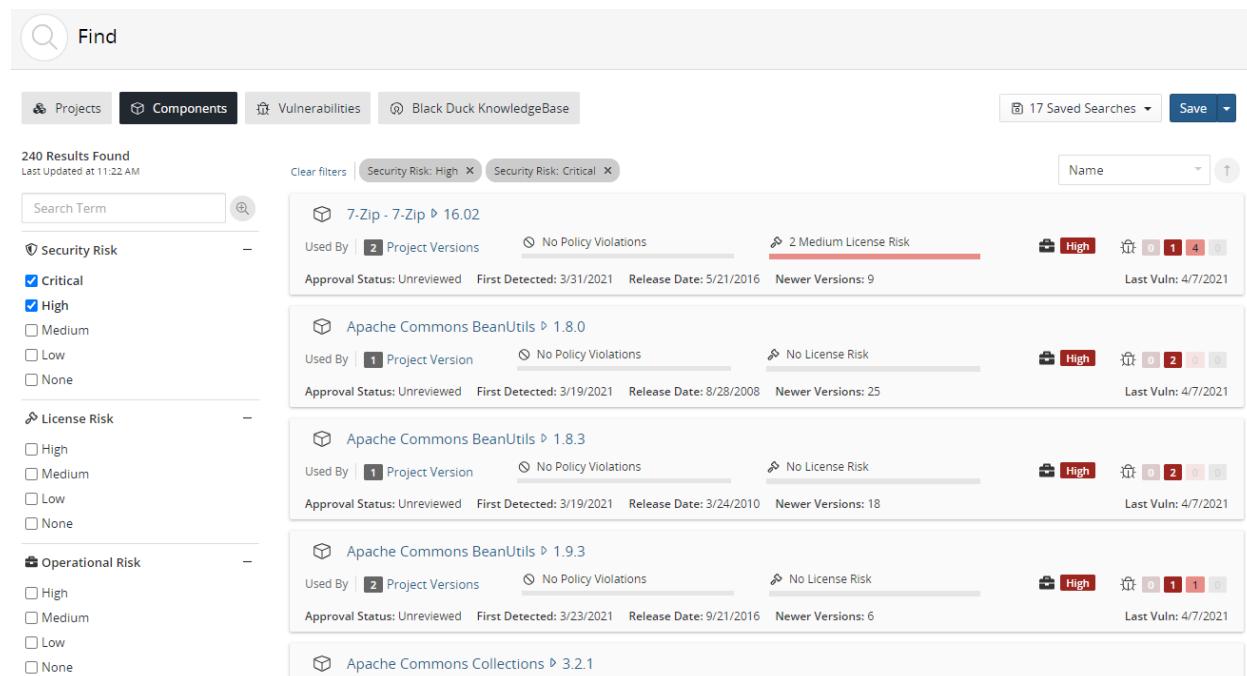
Searching for components

You can search for component versions used in your BOMs and/or components in the [Black Duck KnowledgeBase \(KB\)](#).

To search for components

1. Click  to open the Find page.
2. Do one of the following:
 - Select the **Components** tab to find component versions used in your projects.
 - Select the **Black Duck KnowledgeBase** tab to search for Black Duck KnowledgeBase components.
3. Type your search term in the Search field and/or optionally, select any filters, as described in the next section, "Using search filters.".
4. Optionally, for component searches, [save this search](#), so that the results appear on the Dashboard page.

The Find page displays the components that meet your search criteria.



Component	Version	Used By	No. of Project Versions	No. of Policy Violations	License Risk	Approval Status	First Detected	Release Date	Newer Versions	Last Vuln
7-Zip - 7-Zip	16.0.2	Used By	2	0	2 Medium	Unreviewed	3/31/2021	5/21/2016	9	4/7/2021
Apache Commons BeanUtils	1.8.0	Used By	1	0	0 No	Unreviewed	3/19/2021	8/28/2008	25	4/7/2021
Apache Commons BeanUtils	1.8.3	Used By	1	0	0 No	Unreviewed	3/19/2021	3/24/2010	18	4/7/2021
Apache Commons BeanUtils	1.9.3	Used By	2	0	0 No	Unreviewed	3/23/2021	9/21/2016	6	4/7/2021
Apache Commons Collections	3.2.1									

You can also type your search term in the Search field located at the top of the application and press **Enter** or click . The Find page appears displaying the search results. Note that entering a global search term initiates a new search and resets any filters you previously selected. Select the **Components** or **Black Duck KnowledgeBase** tab and filters to refine the results, as described below.

Using search filters

Filters that appear depend on whether you are searching for components used in your BOMs or searching the Black Duck KnowledgeBase.

For each filter:

- Where necessary, click + to display the filter values; click - to hide them.
- If you select more than one type of filter, Black Duck displays items that match *all* values. If you select more than one value for a specific filter, Black Duck displays items that match either value.

For example, if you use the License Risk filter and select high and medium, the search results display all components that have high *or* medium license risk. If you select a high License Risk filter and a critical Security Risk filter, the search results display only those projects that meet have a high license risk *and* critical security risks.

KnowledgeBase filters

Use the following filters to narrow your results when searching the Black Duck KnowledgeBase:

- Primary Language. Primary language in which the component is written. The filter displays the list of available languages in descending order of frequency of use in components.
- Tags. Available for all components that have tags applied to them to provide additional metadata about the component.
- Commit Activity. Represents the trending commit activity level for the open source component over time.
- Component Source. Defines the source of this component. Possible values are **Black Duck Custom Component** or **Black Duck Projects**.

Note: Primary language, tags, and commit activity information is provided by [Black Duck Open Hub](#).

Component filters

Use the following filters to narrow your results when searching components used in your BOM:

- Security Risk.
- License Risk.
- Operational Risk.
- Policy Rule. Select a policy rule from the list to find the components that violate this policy.
- Policy Violations. Severity level of the policy rule.
- Review Status. Select whether the [component has been reviewed](#) in the BOM.
- Component Approval Status. Select an [approval status](#) for a component.
- First Detected. Date when the component was first detected by Black Duck (such as by scanning, being manually added to a BOM, and so on).
- License Family. Select a license family from the list.
- Missing Custom Field Data. Select to view the components and/or component versions which have [required custom fields](#) and are missing data.
- Released. Date when the component was released according to the Black Duck KnowledgeBase.
- License. Select a license from the list.
- Vulnerability CWE. Select a vulnerability CWE from the list.
- Vulnerability Reported. Select when a vulnerability for this component was last updated.

About the search results

Search results show all components that meet your search criteria.

Black Duck KB component search results

The following information is shown for each KnowledgeBase component that meets your search criteria:

- Select the component name to open the [Black Duck KB Component Name page](#).
- View the number of project versions that use this component as shown by the value next to **Used By**.

Used By | 2 Project Versions

Select **Project Versions** to open the Where Used dialog box.

This dialog box lists the projects that use a version of this component.

Column	Description
Project	Name of the project and version that uses this component. Select the project name to display the project version's Components tab.
Phase	Project Phase .
Component Version	Version of this component used in this project version.
Security Risk	<p>Lists the vulnerabilities for each severity level, from left to right: Critical, High, Medium, and Low.</p> <p>0 3 28 11</p> <p>Select a value to display the Security tab of the <i>Black Duck KB Component Name Version</i> page, which lists the vulnerabilities associated with this version of the component.</p>

- For each component, the search results show:
 - Commit Activity.

- Last commit date.
- Total number of versions for this component.
- Select **Tags** to view the tags for this component.
- The URL in the upper right corner is the URL for this component.

Components search results

The following information is shown for each component in your BOM that meets your search criteria.

- Select the component name/version to display the [Component Name Version page](#).
- View the number of project versions that use this component version as shown by the value next to **Used By**.

Used By | **2** Project Versions

Select **Project Versions** to open the Where Used dialog box.

Project Name	Phase	License	Review Status	Security Risk
Sample Project - 4.0	In Planning	Apache License 2.0	Not Reviewed	0 3 6 0

This dialog box shows the project versions that use this version of the component.

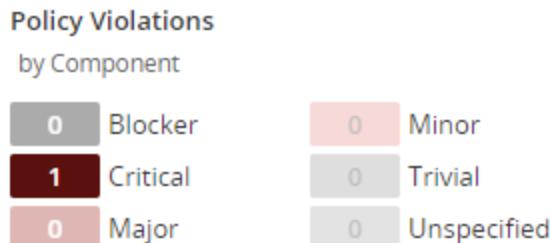
Column	Description
Project Name	Name of the project and version that uses this component version. Select the project name to display the project version's Components tab.
Phase	Project Phase .
License	License for this component version.

Column	Description
Review Status	Whether this component has been reviewed in this project version.
Security Risk	<p>Lists the vulnerabilities for each severity level, from left to right: Critical, High, Medium, and Low.</p> <p>Select a value to display the Security tab of the Black Duck KnowledgeBase <i>Component Name Version</i> page, which lists the vulnerabilities associated with this version of this component.</p>

- Use the bar to quickly see the number of components with the highest policy severity level.



Select the bar to see the number of components with policy violations by severity level:



* Each component is counted once by its highest severity risk

Note: A component is only counted once with the highest policy severity level, not all policy severity levels affecting this component.

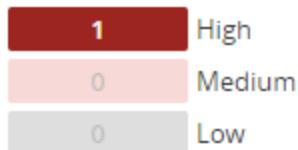
- Use the bar to quickly view the number of components with the highest level of license risk.



Select the bar to view the number of components in each risk category.

License Risk

by Component



* Each component is counted once by its highest severity risk

- View the operational risk for this component version:



- View the number of vulnerabilities by severity associated with this component version. The **Last Vuln** date is when a vulnerability for this component was last updated in Black Duck (by the Black Duck KnowledgeBase or a user).



Select a value to display the **Security** tab of the the Black Duck KB *Component Name Version* page, which lists the vulnerabilities associated with this version of this component.

Identifier	Published	Overall Score
> NVD CVE-2016-3081	Apr 26, 2016	9.3 High
> BDSC BDSC-2018-2905 (CVE-2018-11776)	Aug 22, 2018	8.3 High
> BDSC BDSC-2013-0027 (CVE-2013-4316)	Oct 8, 2018	7.4 High
> BDSC BDSC-2014-0067 (CVE-2013-2251)	May 24, 2018	6.5 Medium
> BDSC BDSC-2017-0903 (CVE-2017-9805)	Sep 6, 2017	6.2 Medium
> BDSC BDSC-2014-0117 (CVE-2014-0094)	Feb 19, 2019	6.2 Medium
> BDSC BDSC-2013-0028 (CVE-2013-1966)	Oct 9, 2018	6.2 Medium
> BDSC BDSC-2017-0367 (CVE-2017-9791)	Aug 9, 2017	6.2 Medium
> BDSC BDSC-2017-0031 (CVE-2017-5638)	Mar 10, 2017	6.2 Medium
> BDSC BDSC-2017-0954 (CVE-2017-12611)	Sep 11, 2017	6.2 Medium
> BDSC BDSC-2013-0052 (CVE-2013-2115)	Aug 14, 2019	6.2 Medium
> BDSC BDSC-2020-2097 (CVE-2019-0230)	Aug 18, 2020	5.9 Medium

- For each component version, the search results also show:

- Approval status. Status indicates whether this component version has been reviewed.
- First detected date.
- Date this component version was released.

- Number of newer versions.
 - Date when a vulnerability for the component was last updated in Black Duck (such as updates from the Black Duck KnowledgeBase, a user manually changing the associated vulnerability, and so on).
- View the number of results found and the time the database was last updated:

14 Results Found
Last Updated at 1:27 PM

Sorting the search results

Optionally, you can sort the results that appear on the page by selecting a value from the **Sort by** list:

Sort by... 

Note that if you sort the results and save this search, the Dashboard page displays the saved search in the sorted order.

Searching for 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.
- Create a custom vulnerability dashboard so that you can focus on the vulnerabilities that are important to you.

To search for vulnerabilities

1. Click  to open the Find page and select the **Vulnerabilities** tab.
2. Optionally, type your search term in the Search Term field.
3. Optionally, select any filters, as described in the next section, "Using search filters."

Note that you can enter a search term only, include filters with the search term, or just search using filters.

4. Optionally, [save this search](#), so that the results appear on the Dashboard page.

The Find page displays the vulnerabilities that meet your search criteria.

The screenshot shows the Black Duck KnowledgeBase interface for searching vulnerabilities. At the top, there's a search bar with a magnifying glass icon labeled "Find". Below it, a navigation bar includes "Projects", "Components", "Vulnerabilities" (which is selected), and "Black Duck KnowledgeBase". There are also buttons for "3 Saved Searches" and "Save".

On the left, a sidebar lists various filters with "+" and "-" buttons to expand or collapse them:

- Affecting Projects
- Default Remediation
- Reachable
- Exploit** (selected, indicated by a plus sign)
- First Detected
- Remediation Status
- Duplicate
- Ignored
- Mitigated
- Needs Review
- New** (selected, indicated by a checked checkbox)
- Patched
- Remediation Complete
- Remediation Required
- Solution
- Base Score

At the top center, there are buttons for "Clear filters" and "Remediation Status: New".

The main area displays a list of vulnerabilities with the following details:

Vulnerability ID	Description	Used By	Overall Risk	Solution	Workaround	Exploit
BDSA-2019-1853 (CVE-2019-11272)	BDSA-2019-1853 (CVE-2019-11272)	9 Project Versions	5.5 Medium	✓ Solution	No Workaround	No Exploit
BDSA-2018-4975	BDSA-2018-4975	1 Project Version	3.2 Low	✓ Solution	No Workaround	No Exploit
BDSA-2013-0030 (CVE-2013-1965)	BDSA-2013-0030 (CVE-2013-1965)	9 Project Versions	3.9 Medium	✓ Solution	No Workaround	⚠ Exploit
BDSA-2018-1901 (CVE-2018-11040)	BDSA-2018-1901 (CVE-2018-11040)	9 Project Versions	3.2 Low	✓ Solution	✓ Workaround	No Exploit
BDSA-2017-3875 (CVE-2019-1010266)	BDSA-2017-3875 (CVE-2019-1010266)	1 Project Version	3.8 Low	✓ Solution	No Workaround	⚠ Exploit

Each row shows the vulnerability ID, description, number of versions used, overall risk level, availability of solution and workaround, and whether an exploit is available.

You can also perform a global search by typing your search term in the Search field located at the top of the application and pressing **Enter** or clicking . If not displayed, select the **Vulnerabilities** tab to view your results. Note that entering a global search term initiates a new search and resets any filters you previously selected.

Using search filters

For each filter:

- Where necessary, click **+** to display the filter values; click **-** to hide them.
- If you select more than one type of filter, Black Duck displays items that match *all* values. If you select more than one value for a specific filter, Black Duck displays items that match either value.

For example, if you use the remediation status filter and select new and needs review, the search results display all vulnerabilities that have a remediation status or new *or* needs review. If you select a remediation status of new and a security filter of high, the search results display only those vulnerabilities that meet have a remediation status of new *and* a high security level.

Use the following filters to narrow your results when searching for vulnerabilities:

- Affecting projects. Selecting this filter searches for vulnerabilities in your projects only. Clearing this filter searches the Black Duck KnowledgeBase and your projects.
- Default Remediation. Selecting displays vulnerabilities that are [automatically remediated](#).
- Reachable. Vulnerability is [reachable](#).
- Exploit. Select whether an exploit is available for a vulnerability.
- First Detected. When the vulnerability was first appeared in a BOM.
- Remediation Status. Select one or more remediation statuses.
- Solution. Select whether a solution is available for a vulnerability.
- Base Score. Enter the minimum base score value; Black Duck displays vulnerabilities that have this

score or higher.

- Exploitability Score. Enter the minimum exploitability score value; Black Duck displays vulnerabilities that have this score or higher.
- Impact Score. Enter the minimum impact score value; Black Duck displays vulnerabilities that have this score or higher.
- Overall Score. Enter the minimum overall score value; Black Duck displays vulnerabilities that have this score or higher.
- Published Year. Year the vulnerability was published.
- Severity. The severity levels shown depend on the [selected security configuration](#) as CVSS v2 does not have a critical security level.
- Source. BDSA or NVD.
- Temporal Score. Enter the minimum base score value; Black Duck displays vulnerabilities that have this score or higher.
- Workaround. Select whether a workaround is available for a vulnerability.

About the search results

Search results show all vulnerabilities that meet your search criteria. The following information is shown for each vulnerability:

BDSA BDSA-2020-1234 (CVE-2020-13430)
Used By 0 Project Versions Overall Risk 8.1 High
First Detected: Never Published: 5/27/2020 Last Modified: 7/27/2020
✓ Solution ✓ Workaround No Exploit
CWE-79

- Select the vulnerability ID to view more information on the vulnerability, such as additional score values. 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](#).
- View the number of project versions that affected by this vulnerability next to **Used By**.

Used By 2 Project Versions

Select **Project Versions** to open the **Affected Projects** tab for the vulnerability which lists the project versions affected by this vulnerability.

Black Duck Security Advisory
Apache HttpClient Vulnerable to Man-In-The-Middle (MITM) Attack via SSL Hostname Verification Bypass
BDSA BDSA-2014-0126 | CVE-2014-3577 | Published May 30, 2019 | Updated Feb 7, 2020
Overview Affected Projects Technical CVE References Settings
Remediate Filter projects...

Project	Component	Component Origin	Status	Target date	Actual date
cloudfoundry-identity-parent 3.6.3	Apache HttpClient 3.1	maven/commons-httpclient:commons-httpclient:3.1	New	Never	Never
cloudfoundry-identity-parent 3.6.3	Apache HttpClient 4.3.3	maven/org.apache.httpcomponents:httpclient:4.3.3	New	Never	Never
cloudfoundry-identity-parent 3.6.3	Apache HttpComponents Core 4.3.2	maven/org.apache.httpcomponents:httpcore:4.3.2	New	Never	Never
cloudfoundry-identity-parent 3.6.3	Apache HttpComponents Core 4.3.3	maven/org.apache.httpcomponents:httpcore:4.3.3	New	Never	Never

Displaying 1-4 of 4

- View the overall risk score. The search results show the Temporal Score for BDSA vulnerabilities or

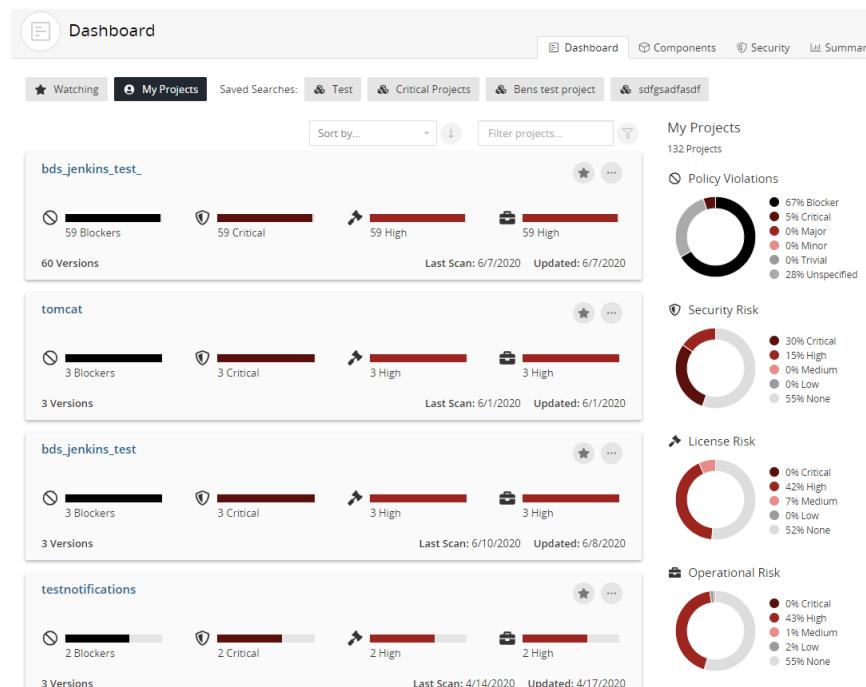
the Base Score for NVD vulnerabilities and the associated risk level. Note that the score shown and risk level depends on the [selected security rankings](#).

Select the score to view individual scores: temporal, base, exploitability, and impact for BDSA; base, exploitability, and impact for NVD.

- View whether a solution, workaround, or exploit is available:
 - ✓ indicates that there is a solution or workaround available for this vulnerability.
 - △ indicates there is an exploit for this vulnerability.
- For each vulnerability, the search results also show:
 - First Detected.
 - Published date.
 - Last modified date.
 - Common Weakness Enumeration (CWE) number for this security vulnerability.

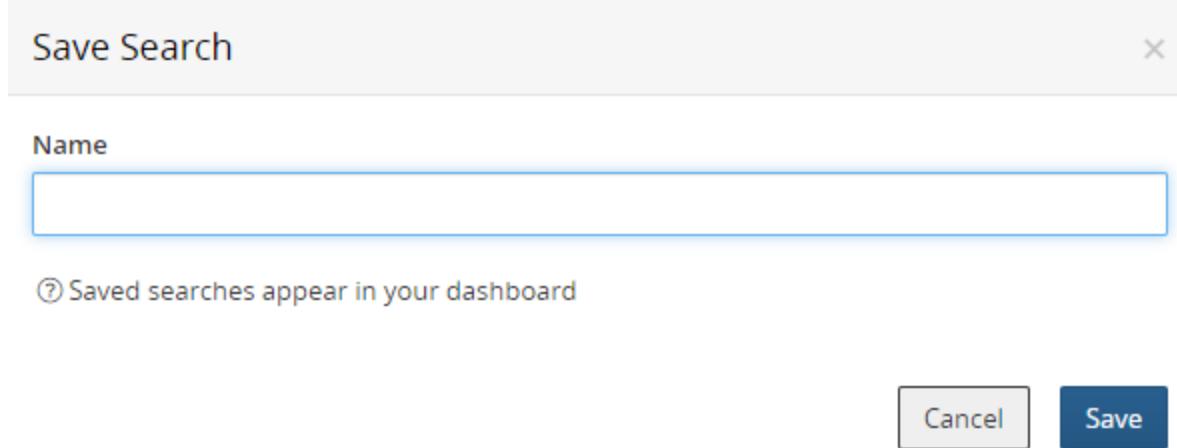
Saving and managing search results

Black Duck gives you the ability to save your search results. This lets you search for projects, components, or vulnerabilities using a variety of attributes, save those searches, and then view dashboards of those saved searches so that you can quickly view the information that is relevant to you.



Saving search results

1. After using the Find page to create the search results you wish to view on your Dashboard, click **Save** on the Find page. The Save Search dialog box appears.



2. Enter a name for these search results and click **Save**.

This saved search now appears in the list of saved searches on your Dashboard page.

Tip: You can also use an existing saved search as a basis for a new saved search. Select a search from the list of saved searches and optionally modify any filters. Click **Save New** and specify a new name for this search. This gives you a new saved search while your existing saved search is not modified.

Editing saved searches

1. Click . The Find page appears.
2. Select the saved search you wish to modify from the list of saved searches. The Find page displays the search results for that saved search.
3. Optionally, modify the filters for this saved search.
4. Click **Update**.

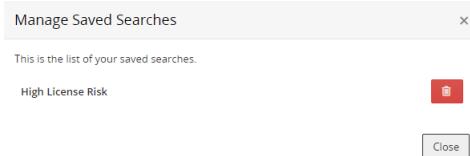
Renaming saved searches

1. Click . The Find page appears.
2. Select the saved search you wish to modify from the list of saved searches. The Find page displays the search results for that saved search.
3. Optionally, modify the filters for this saved search.
4. Click **Update** and select **Rename** from the menu. The Rename Saved Search dialog box appears.

5. Enter the new name of this saved search.
6. Click **Save**.

Deleting saved searches

1. Click . The Find page appears.
2. Select **Manage** from the list of saved searches. The Manage Saved Searches dialog box appears.



3. Click  in the row of the saved search you wish to delete. The saved search is removed from the list of saved searches and from your Dashboard page.
4. Click **Close**.

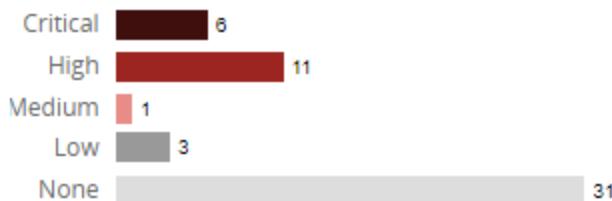
Filtering the data shown in tables

Risk graphs and/or advanced filters are available on some pages to help you filter the information shown in tables.

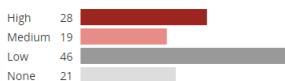
Risk Graphs

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.

For security risk:



For license and operational risk:



- **Critical** risk: 50% black and 50% red (security risk only)
- **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:

Component Name	Versions	Used count	Security Risk	License Risk	Operational Risk
> Kerberos	2 Versions	2	██████	██████	██████
Angular	1.3.0-beta.11	1	██████	██████	██████
Python programming language	2.7.16	1	██████	██████	██████
SQLite	3.28.0	1	██████	██████	██████
> Lo-Dash	4 Versions	5	██████	██████	██████
> GNU Compiler Collection	2 Versions	3	██████	██████	██████
> PostgreSQL Database Server	4 Versions	41	██████	██████	██████
morris.js	0.4.3	1	██████	██████	██████
> mixin-deep	2 Versions	3	██████	██████	██████
> JS-YAML Native JS port of PyYAML	2 Versions	3	██████	██████	██████
> Growl	2 Versions	2	██████	██████	██████
> Scala	2 Versions	2	██████	██████	██████
> minimatch	3 Versions	7	██████	██████	██████
> randomatic	3 Versions	3	██████	██████	██████
> set-value	3 Versions	5	██████	██████	██████

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 a filter (▼) icon in the graph and a field appears above the table displaying the values you selected. Click ▼ 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](#).

Tip: You can [remove projects you are watching](#) so that you do not receive notifications for those projects or components in those projects.

Viewing notifications

1. Open the notifications list by selecting .

The list displays all new and previously viewed notifications.

2. To manage the notifications, 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

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.

Hiding notifications

You can hide notifications so that they no longer appear in the drop-down list and appeared grayed out on the All Notifications page.

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.

You can also use the same method in the All Notifications page.

Click  located in the upper right corner of a notification in the All Notifications page to redisplay the notification.

Appendix C: About the Tools page

The Tools page provides download links and links to Black Duck integrations on GitHub, the Synopsys Software Integrity Community, and Synopsys Software Integrity, Customer Education web pages. It is divided into these sections: **Downloads**, **Black Duck Open Source Integrations**, and **Community and Education**, as described below.

To access the Tools page, from the user menu located on the top navigation bar, select **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 10.10 or later. A minimum of 8 GB of RAM.
 - Windows. Windows 10. 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 10.10 or later. A minimum of 8 GB of RAM.
- Windows. Windows 10. 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.

Black Duck Open Source Integrations

Clicking this link opens the Black Duck pages on [GitHub](#).

To view Black Duck integrations documentation, from the help menu () located on the top navigation bar, select **Integrations Documentation**.

Community and Education

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 (SIG Edu).** Clicking this link on the Tools page opens a web page that provides more information on education courses for Synopsys Integrity Group products.

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
-?, --help	Shows help for this tool.
-A, ---dest <host: port>	Specifies Black Duck host name and port.
-P, --hub-project <name>	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.
-R, --hub-release <name>	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 hub-project , hub-release is optional. If you do not specify hub-release , the version defaults to the value of the release parameter.
-S, --secure-dest	Uses HTTPS to connect to the server hosting Black Duck. If you do not include this parameter, HTTP is used.
-U, --dest-user <user>	Specifies the username to log in to the Black Duck server.
-W, --dest-password	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

Parameter	Description
	dest-password parameter is optional: the tool prompts the user for the password; it does not check the password against the variable.
-a, ---address <host:port>	Specifies the Protex host name and port.
-r --release<name>	Specifies a value to use to identify the current state of the Protex BOM. You can use any value for <name>. Use this parameter to enable viewing multiple "versions" of a Protex BOM in Black Duck. Click here or more information.
--list-projects <SearchQuery>	<p>Lists all Protex project identifiers for all projects to which you have access, one per line, on the console.</p> <p><SearchQuery> 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>
--data <path>	Specifies the path to the Protex BOM file.
--output <path>	Writes the BOM out to a file or directory with the project name.
-p, --project <id or name>	Specifies the Protex project identifier or project name.
-s, --secure	Uses HTTPS to connect to the server hosting Protex. If you do not specify this parameter, HTTP is used.
-u, --user <user>	Specifies the username to log in to the Protex server.
-w, --password	<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 password parameter is optional.</p>
-V, --version	Shows the version information of this tool.
-v, --verbose	Sets the logging level to verbose.
--dryRunWriteDir <dryRunWriteDir>	Specifies the directory to which the Protex BOM Tool outputs a JSON file with the original file metadata used for scanning.
--debug	Shows debug output.
--include-files	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
```

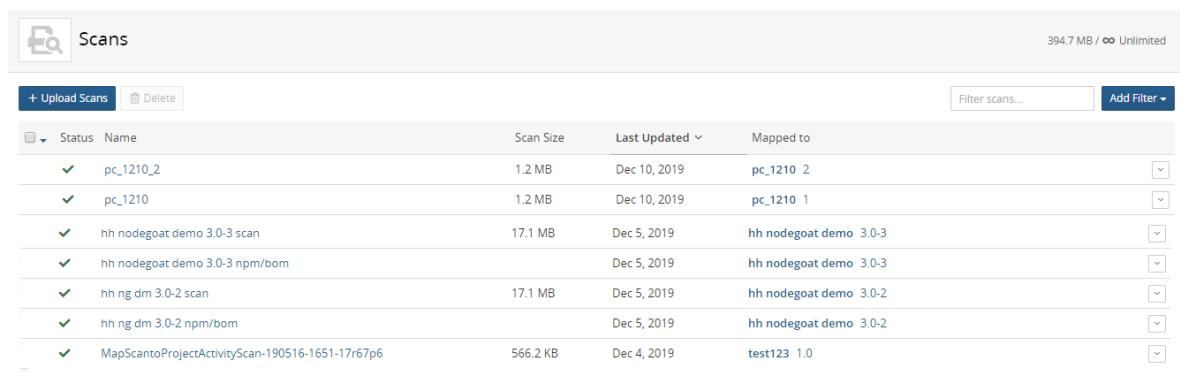
Importing the Protex BOM file

If you output the Protex BOM to a file, you need to import the file into Black Duck.

To import a Protex BOM file

1. Log in to Black Duck.

2. Click  Scans.



The screenshot shows the Black Duck Scans page. At the top, there is a search bar and a filter button labeled "Filter scans...". Below the header, there is a table with columns: Status, Name, Scan Size, Last Updated, and Mapped to. The table lists several BOM files, each with a dropdown arrow icon next to the "Mapped to" column. The table includes the following data:

Status	Name	Scan Size	Last Updated	Mapped to
✓	pc_1210_2	1.2 MB	Dec 10, 2019	pc_1210_2
✓	pc_1210	1.2 MB	Dec 10, 2019	pc_1210_1
✓	hh nodegoat demo 3.0-3 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh nodegoat demo 3.0-3 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh ng dm 3.0-2 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-2
✓	hh ng dm 3.0-2 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-2
✓	MapScantoProjectActivityScan-190516-1651-17r67p6	566.2 KB	Dec 4, 2019	test123 1.0

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.

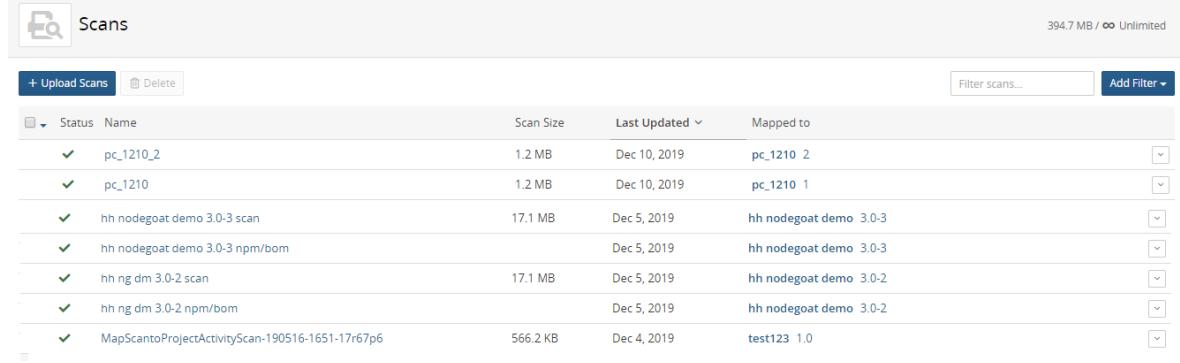
To map a Protex BOM to a project

1. Log in to Black Duck.



2. Click **Scans**.

The Scans page appears.



A screenshot of the Black Duck 'Scans' page. The page title is 'Scans'. It shows a table of scanned files with columns: Status, Name, Scan Size, Last Updated, and Mapped to. There are 9 rows listed. The last row is 'MapScantoProjectActivityScan-190516-1651-17r67p6' which has been mapped to 'test123 1.0'. A 'Filter scans...' search bar and an 'Add Filter' button are at the top right of the table.

Status	Name	Scan Size	Last Updated	Mapped to
✓	pc_1210_2	1.2 MB	Dec 10, 2019	pc_1210_2
✓	pc_1210	1.2 MB	Dec 10, 2019	pc_1210_1
✓	hh nodegoat demo 3.0-3 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh nodegoat demo 3.0-3 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-3
✓	hh ng dm 3.0-2 scan	17.1 MB	Dec 5, 2019	hh nodegoat demo 3.0-2
✓	hh ng dm 3.0-2 npm/bom		Dec 5, 2019	hh nodegoat demo 3.0-2
✓	MapScantoProjectActivityScan-190516-1651-17r67p6	566.2 KB	Dec 4, 2019	test123 1.0

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.

2. Click  Scans.

The Scans page appears.

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.