

Black Duck SCA Help Center

Black Duck SCA 2026.7.2

Contents

1. Black Duck SCA Help Center 5

Welcome to Black Duck SCA 5
Getting started with Black Duck SCA 6
Viewing your dashboards 9
About the Summary dashboard 10
About the New Vulnerabilities dashboard 15
About the Watching and My Projects dashboards 16
About saved searches dashboards 19
Scanning Components 27
Understanding Component Scanning 27
About Black Duck - Binary Analysis 29
About Black Duck Detect Desktop 30
About Container Scanning 38
About Rapid Scanning 40
About Correlated Scanning 45
About SCA Scan Service 46
Signature Scanner 48
Understanding component matching in Black Duck SCA 98
Understanding projects in Black Duck 100
Creating a project 102
Deleting a project 105
Watching projects 105
Cloning projects 108
Updating project settings 110
Managing project team membership 112
Managing tags 114
Changing the project's SBOM alias 115
Viewing a project's or project version's activity 116
About Project Versions 117
About Long-Term support (LTS) projects 125
About project version BOMs 130
Viewing a project version's BOM 130
Project version BOM page reference 130
Reviewing the contents of a BOM 150
Editing a project version BOM 151
Managing comments in a BOM 181
Managing files associated with BOM components 185
Generating project version reports 190
Comparing BOMs 215
Printing a BOM 218
Viewing issues in a project 219
Viewing component versions with encryption 220
About Linux distributions in Black Duck 221
Match Review 221
About SCM read-only BOMs 225
Viewing risk in Black Duck 226
Viewing the health of your projects 229

Viewing overall risk ... 233
About security risk ... 236
Viewing the security vulnerabilities of your projects, project versions, and component versions ... 239
Viewing project version vulnerabilities ... 239
Viewing component version vulnerabilities ... 242
Analyzing the impact of a vulnerability ... 243
Viewing vulnerability details ... 246
Remediating security vulnerabilities ... 258
Finding data in Black Duck ... 264
Searching for projects ... 265
Searching for components ... 268
Searching for vulnerabilities ... 272
Searching the Black Duck KnowledgeBase ... 275
Saving and managing search results ... 277
Generating global reports ... 279
Vulnerability Remediation report ... 280
Vulnerability Status report ... 282
Vulnerability Update report ... 283
Vulnerability Exploitability EXchange (VEX) ... 286
Deleting reports ... 289
Managing Black Duck ... 290
Managing Components ... 290
Managing open source licenses ... 310
Managing policies ... 414
Managing Project Groups ... 434
Administering Black Duck ... 440
Creating system announcements ... 440
Viewing jobs ... 443
Downloading log files and heatmap data ... 448
Viewing heatmap data ... 463
Administering user accounts ... 464
Administering user groups ... 488
Access Tokens ... 499
Configuring Integrations ... 503
Configuring System Settings ... 515
Viewing project and project version audit information ... 549
Managing your code size limits ... 551
Working with notifications ... 553
About the Tools page ... 555
Hosted vs Air-gapped KnowledgeBase ... 556
Integrating Protex with Black Duck ... 559
Understanding the Protex BOM integration process ... 560
Requirements ... 561
Downloading the Protex BOM tool ... 561
Exporting a Protex BOM ... 561
Importing the Protex BOM file ... 565
Mapping or unmapping a Protex BOM ... 566
Black Duck C/CPP Tool ... 567
Black Duck integrations & APIs ... 567
Command line clients ... 567
Plug-in integrations ... 568
API reference ... 570
Legal Information ... 571

License Agreement..... 571
Privacy Policy.....571
Included Third Party Software..... 571
Customer support..... 571
Black Duck Community..... 571

1. Black Duck SCA Help Center

Welcome to Black Duck SCA

Resources

- Release Notes

  The contain information about the new and improved features, resolved issues, and known issues in the current release.

- What's New

  See what's new with Black Duck 2026.7.2 by logging into Black Duck and clicking  → What's New.

- Installation information

  contains information about installing and upgrading Black Duck using Docker Swarm.

  Instructions for installing Black Duck in a Kubernetes or OpenShift environment are:

  - Click here for instructions on using Helm to deploy Black Duck.
  -

- Getting Started Guide

  The provides first-time users with information on using Black Duck.

- Report Database

  The contains information on using the report database.

- Black Duck SDK

  contains overview information and a sample use case.

- Scanning Best Practices Guide

  The provides best practices for scanning.

Training

Black Duck Customer Education 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 Black Duck Community 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://blackduck.skilljar.com/page/black-duck or for help with Black Duck, select **Black Duck Tutorials** from the Help menu () in the Black Duck UI.

Customer Success Community

Black Duck Community is our primary online resource for customer support, solutions, and information. Black Duck 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 Black Duck 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 Black Duck 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 Black Duck 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 Black Duck experts and our Knowledgebase.
- Share – Collaborate and connect with Black Duck 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.

Black Duck Statement on Inclusivity and Diversity

Black Duck 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.

Getting started with Black Duck SCA

Black Duck 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, Black Duck 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.

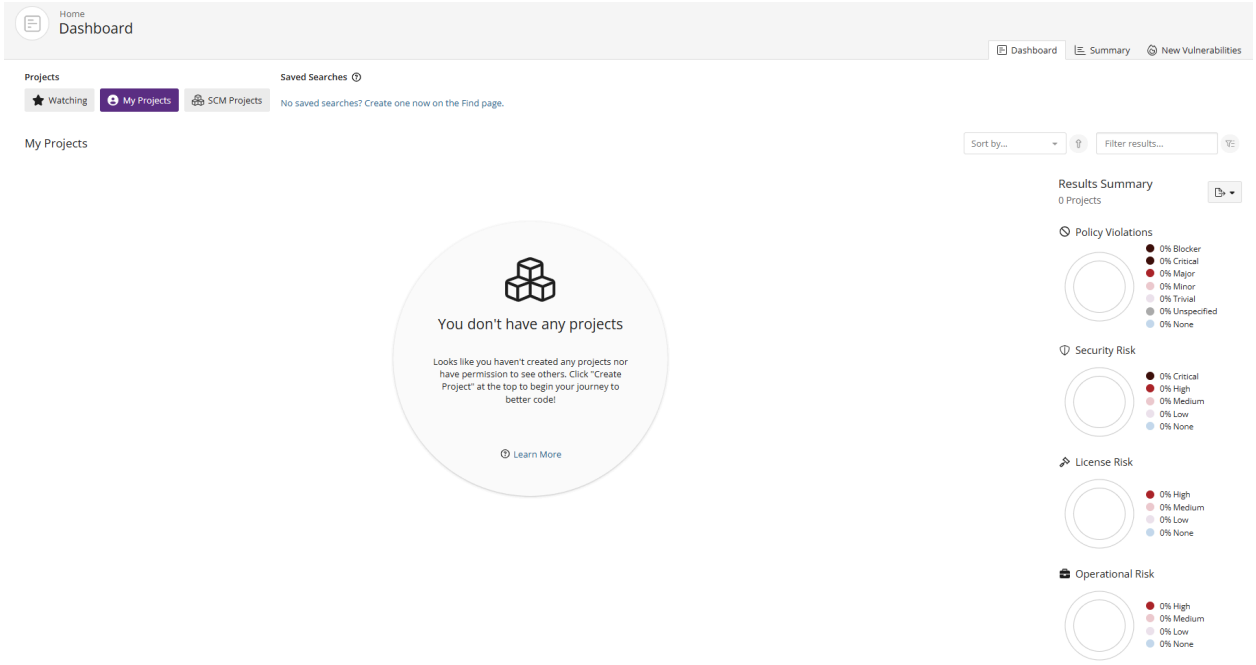
To log in to Black Duck:

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:

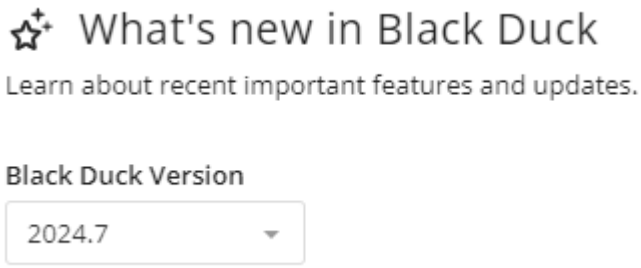
- navigation elements are visible to you on each page
- projects and project data you can view on each page
- actions you can perform inBlack Duck

Seeing What's New in Black Duck

Discover the latest features and enhancements introduced in the current Black Duck release with the new What's New window.

The What's New window will appear automatically after login, highlighting the most impactful updates in this version. Users can choose to disable this window for future logins, but it will reappear with the next Black Duck server upgrade. Even after being dismissed, the What's New content can be accessed under the Help menu.

From the What's New window, you can also select the desired Black Duck version to see its highlights.



Scanning your code and mapping scans to projects

Use these methods to scan your code:

- Detect Desktop which you can download from Black Duck's Tools page
- Plugins.

- Black Duck Detect. Use Black Duck 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.

Scanning C and C++ projects

C and C++ projects lack a standardized package manager or method for handling dependencies. As a result, generating an accurate bill of materials (or BOM) for these projects is more challenging. Use the C/CPP Tool to generate a BOM report for projects written in C/C++ by building the project, capturing the source and binary files involved, and then delivering a BDIO and signatures to Black Duck.

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.

Users have access to three distinct dashboards, each designed to provide valuable insights and facilitate effective vulnerability management:

- **Dashboard**: The main Dashboard serves as the central hub for users, offering a comprehensive view of your projects and their associated vulnerabilities. Here, you can quickly access key metrics, monitor overall security posture, and navigate to specific components or reports. The Main Dashboard provides a high-level overview, helping you stay informed about the status of your projects at a glance.
- **Summary**: The Summary dashboard provides a detailed breakdown of vulnerabilities and risks associated with your projects. This dashboard focuses on summarizing critical information, including the number of vulnerabilities, their severity levels, and remediation statuses. Users can leverage this data to prioritize their remediation efforts and track progress over time.

- **New Vulnerabilities**: The New Vulnerabilities dashboard is specifically designed to highlight recently identified vulnerabilities across your projects. This dashboard allows users to quickly identify and assess new risks, enabling prompt action to mitigate potential threats. It provides an updated view of vulnerabilities that have been added since the last scan, ensuring you are always aware of the most current security issues.

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, click 

.

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.

Exporting to CSV

You can export your Dashboard to CSV which converts the individual rows to tabular data. To do so, click the button and select CSV.

About the Summary dashboard

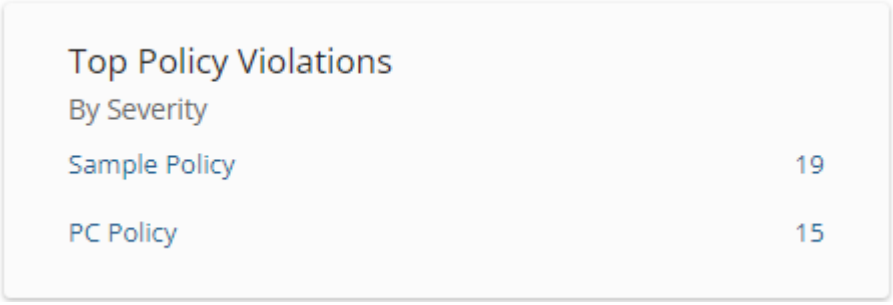
Use the **Summary** tab to view the overall health of your projects and identify areas of concern. The page consists of widgets that provide business critical information which you can use to quickly assess areas where you need to focus your attention.



Note: The **Summary** tab only displays information for the projects you have permission to view.

The following describes each widget shown on the **Summary** tab and, where available, how to view additional information. Note that the security risk values shown use CVSS v3.x or CVSS v4.x scores, depending on which security risk calculation you selected; by default CVSS v3.x scores are shown.

Top Policy Violations



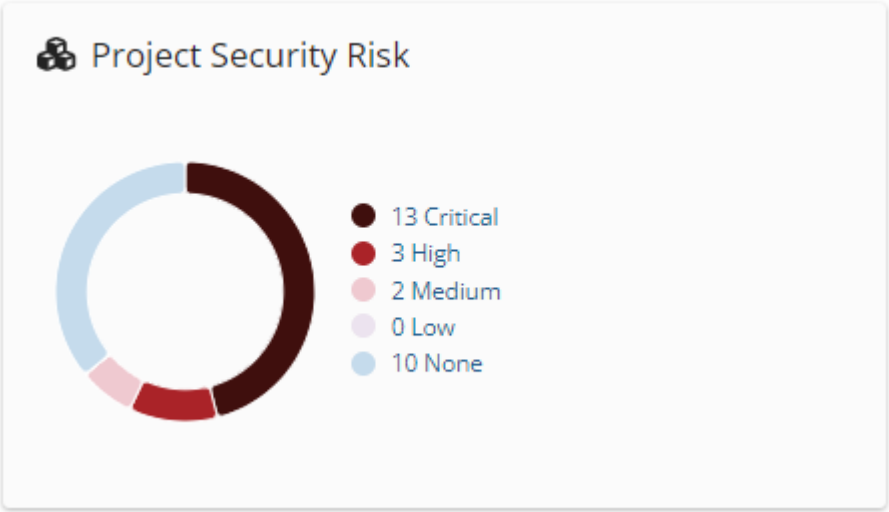
The **Top Policy Violations** widget displays up to the top five policy violations across all projects that you have permission to view.

Policy rules are listed by severity level and then by the number of policy violations, in descending order. If policy rules do not have severity levels assigned to them, the widget displays the top five policy violations, in descending order by the number of violations.

- 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.

Project Security Risk

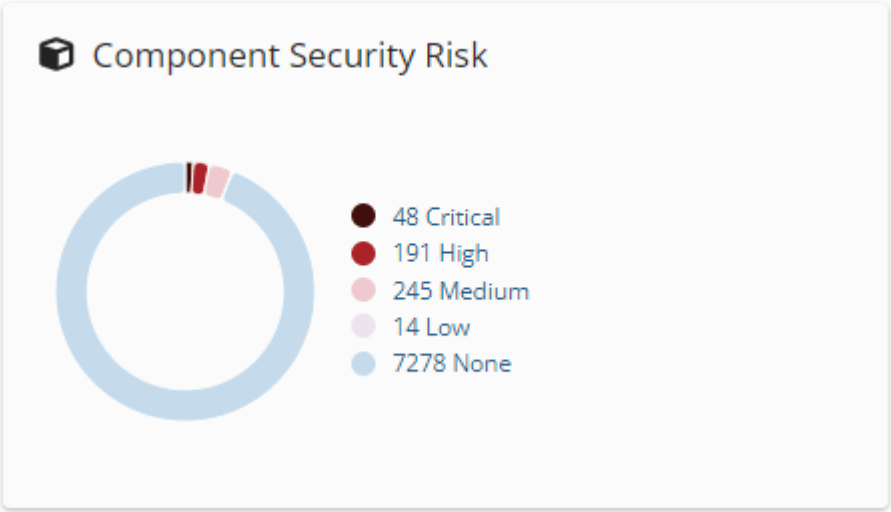


The **Project Security Risk** widget displays the number of projects you have permission to view for each level of security risk.

Note that this widget counts the highest security risk level for a project, not all security levels affecting a project. For example, if a project has medium and low security risks, it is counted as a project with medium security risk; it is not included as a project with low security risks.

Hover over the graph to view the number of projects with that level of security risk.

Component Security Risk

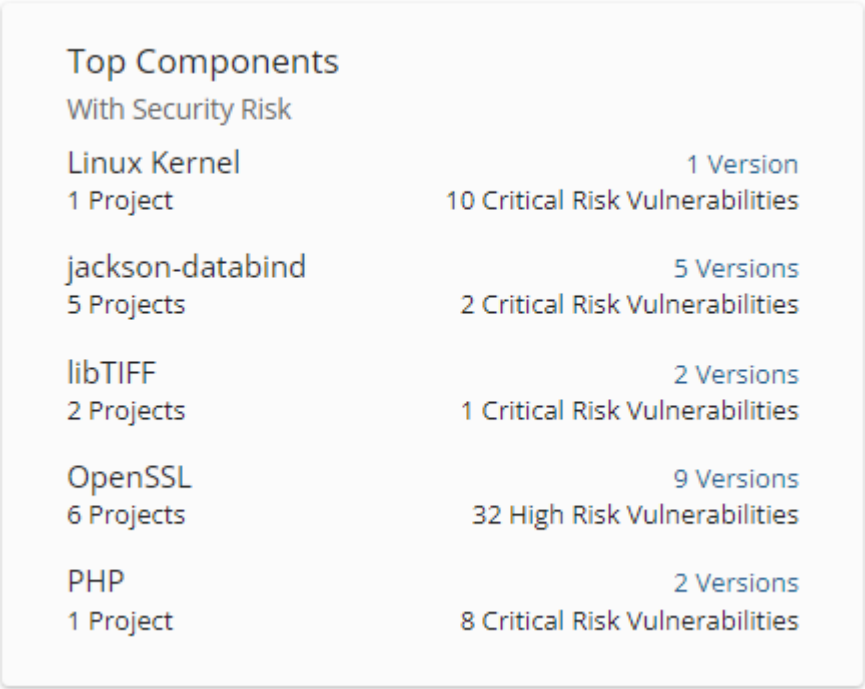


The **Component Security Risk** widget displays the number of components in projects you have permission to view for each security risk level.

Note that the widget counts only the highest security risk for a component. For example, if a component has medium and low security risks, it is counted as one component with a medium security risk.

Hover over the graph to view the number of components with that level of security risk.

Top Components



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:

- 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.

Components are organized by security risk, with those components with the highest risk listed first.

Select the specific version or number of versions to view the Component Version Details page.

Projects with critical/high vulnerabilities

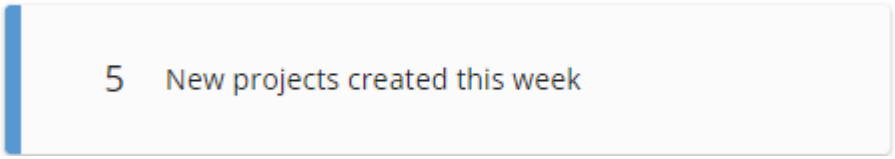


The **Projects have a critical/high vulnerability** widget displays the number of projects with versions that contain components with a critical and/or high security risk.

New vulnerable components this week

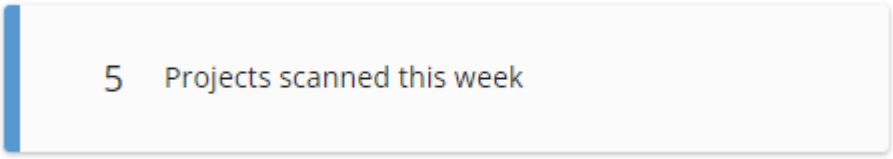
The **New vulnerable components this week** widget displays the number of components the Black Duck KB mapped a vulnerability to in the past seven days, including today.

New projects created this week



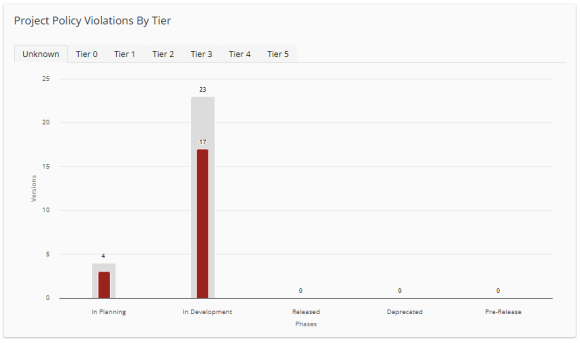
The **New projects created this week** widget displays the number of projects that you have permission to view that have been created in the past seven days, including today.

Projects scanned this week



The **Projects scanned this week** widget displays the number of projects with scans from the past seven days, including today.

Project Policy Violations by Tier

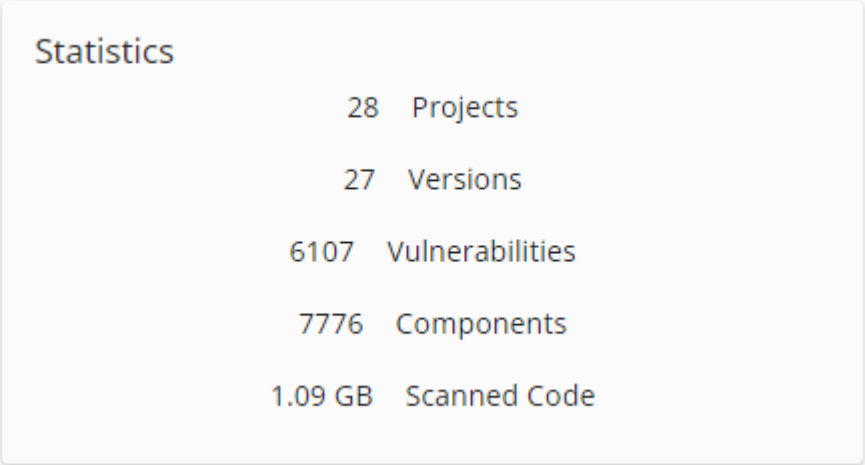


The **Project Policy Violations by Tier** widget displays the total number of projects by phase that have a policy violation, grouped by tiers.

- 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.

Statistics

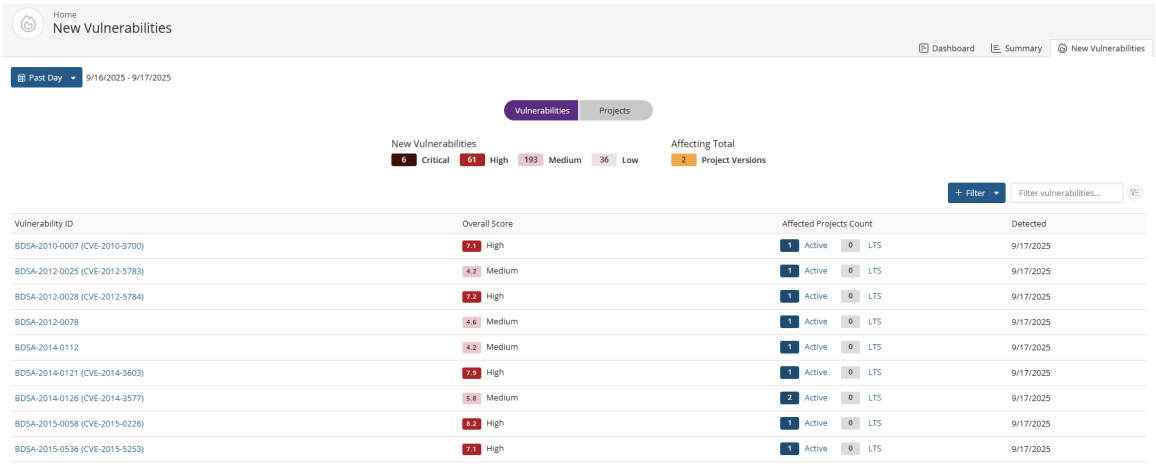


The **Statistics** widget displays the following information:

- **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, *including* ignored components.
- **Scanned Code** lists the number of GBs scanned for all scans.

About the New Vulnerabilities dashboard

The New Vulnerabilities dashboard is specifically designed to highlight recently identified vulnerabilities across your projects. This dashboard allows users to quickly identify and assess new risks, enabling prompt action to mitigate potential threats. It provides an updated view of vulnerabilities that have been added since the last scan, ensuring you are always aware of the most current security issues.



New Vulnerabilities Table

The main feature of the New Vulnerabilities dashboard is a table that displays a list of new vulnerabilities detected within the selected timeframe.

You have the option to click on the **Vulnerabilities** display to view detailed information about specific vulnerabilities or click on the **Projects** display to list all projects affected by the new vulnerabilities.

Graph Summary

A graphical summary will illustrate the count of all vulnerabilities by severity, including Critical, High, Medium, and Low.

Additionally, you can click on each severity category to view the specific vulnerabilities associated with that level.

Filtering Options

Users can filter the listed vulnerabilities by time using the following options:

- Past Day
- Past 3 Days
- Past 7 Days
- Custom range of days

Vulnerabilities view

The Vulnerability view provides you with a comprehensive overview of vulnerabilities identified within their projects. This feature allows users to easily access detailed information about each vulnerability, including its severity, affected components, and remediation status.

The table includes the following columns:

- **Vulnerability ID:** The unique identifier for each vulnerability. You can select any vulnerability in the list to navigate to the corresponding BDSA or CVE page for more detailed information.
- **Overall Score:** A score that reflects the severity and impact of the vulnerability.
- **Affected Projects Count:** The number of projects impacted by the listed vulnerability. You can select the **Active** or **LTS** link to see affected projects in either cycle while on the Vulnerabilities display.
- **Detected:** The date when the vulnerability was identified.

Projects view

The Projects view offers a detailed overview of all projects affected by newly identified vulnerabilities. This view features a table that displays key information for each project, including the **Project Version**, **Phase**, and the count of vulnerabilities categorized by severity.

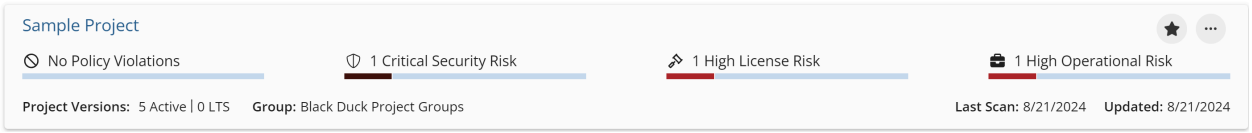
Configuring the retention setting

The **New Vulnerability** data retention setting lets administrators control how many days of this aggregated data the system keeps. Data older than the configured retention period is automatically purged.

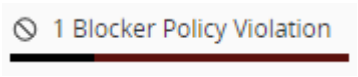
About the Watching and My Projects dashboards

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

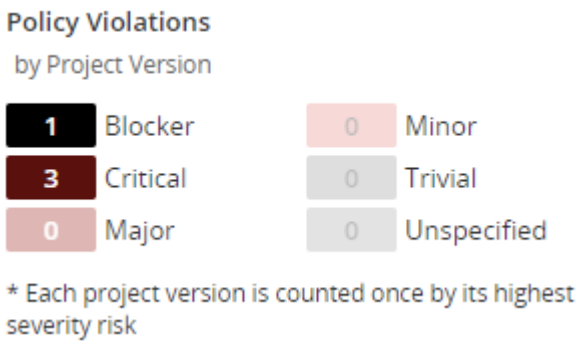
The following information is shown for each project:



- 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.

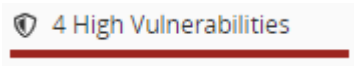
    

    Note: The text states the number of project versions with this highest policy severity level, not all policy severity levels affecting this project.
  - Hover over the bar to see the number of project versions with their highest severity level of policy violations:

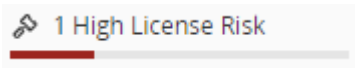
    

    In the above example, there are four project versions which have policy violations; one version has a policy violation which has Blocker as the highest severity level, the other three versions have Critical as the highest severity level. Note that this does not indicate the number of policy violations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

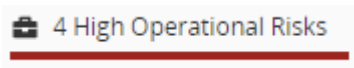
    Security risk:

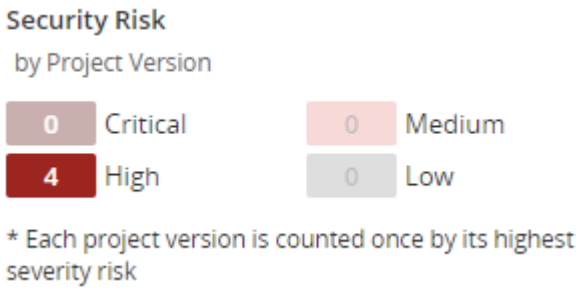
    License risk:

    Operational risk:

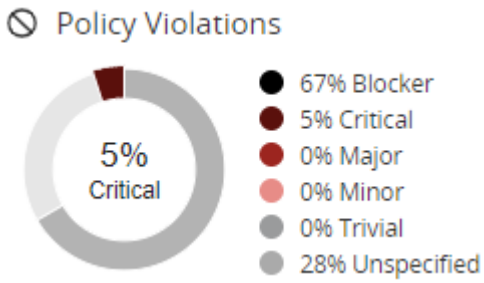
    

    Note: The text states the number of project versions with this highest risk level, not all risk levels affecting the versions.
  - Hover over a risk bar to see the number of versions of this project with their highest level of risk.

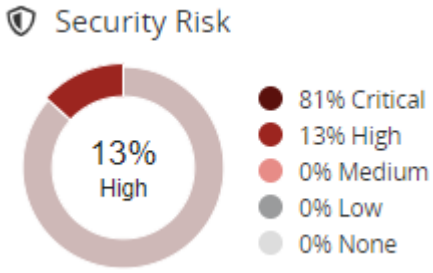


If a project version has risk, the version is only counted once and only its highest risk level is shown.

- 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:

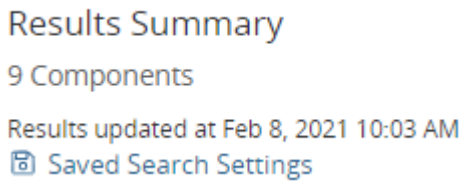
- Use the **Sort by** field to select an attribute to sort by and click an arrow to select the sort order (ascending) or (descending).
- Use the **Filter projects** field to filter the projects shown in either dashboard.

- Use the icons to manage your watched projects or delete a project.

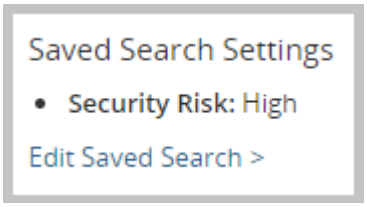
About saved searches dashboards

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.

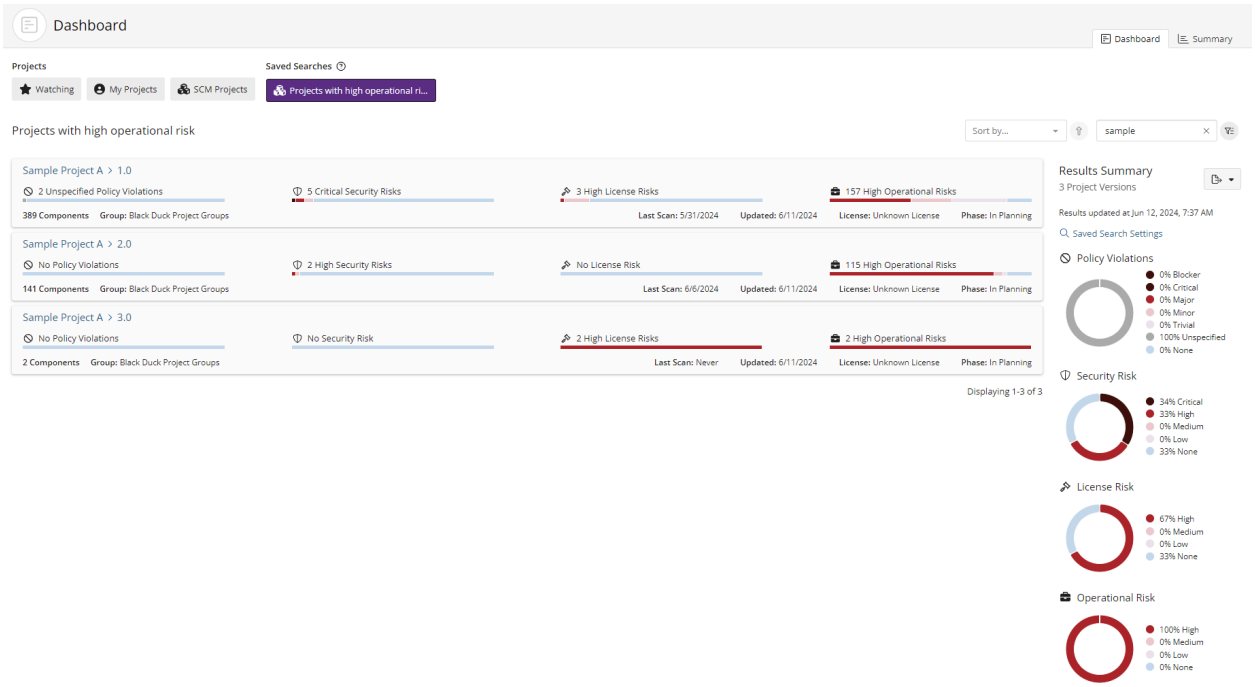


Select **Saved Search Settings** to view the filters for this saved search.

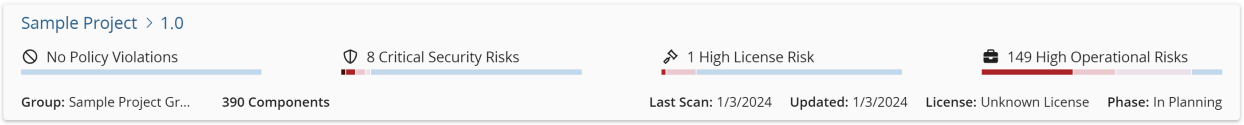


Select **Edit Saved Search** to open the Find page displaying your saved search. Use the page to edit and save this revised saved search.

Project version saved searches



The following information is shown for each project version:

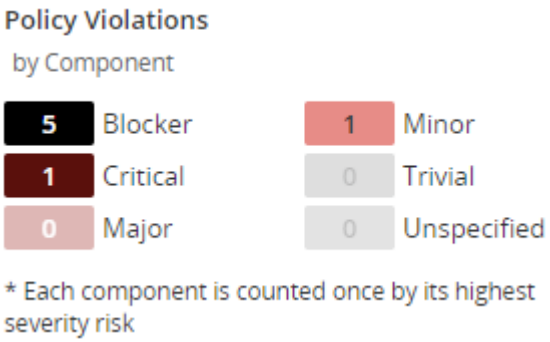


-  located in front of the saved search name indicates that this is a project saved search.
- To view policy violation information for a specific project version:
  - Use the bar to see the number of components with the highest policy severity level for this project version.

    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.

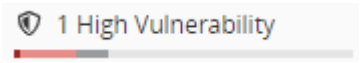
    Note: The text states the number of components with the highest policy severity level for this project version, not all policy severity levels affecting this project version.
  - Hover over the bar to see the number of components with policy violations by the highest policy severity level:



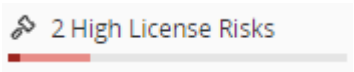
If a component has a policy violation, the component is only counted once and only its highest policy severity level is shown.

- 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.

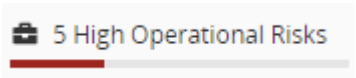
    Security risk:

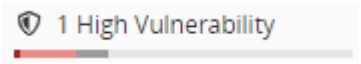
    License risk:

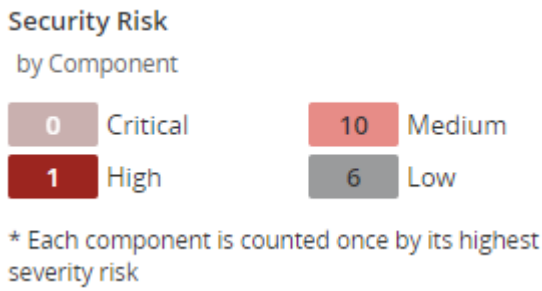
    Operational risk:

    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:

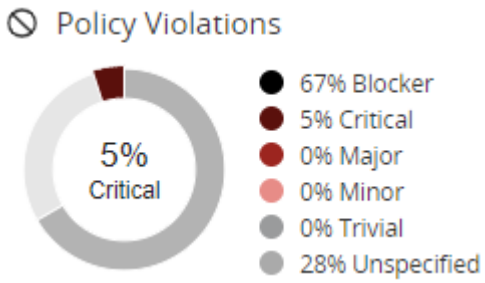
  - Hover over the bar to see the number of components for each risk category.

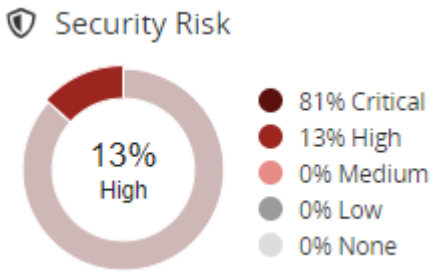
    In this example, there is one component that has a high risk level as its highest risk, 10 components that have medium risk as their highest risk level, and six components that have low risk as their highest risk level.

    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.

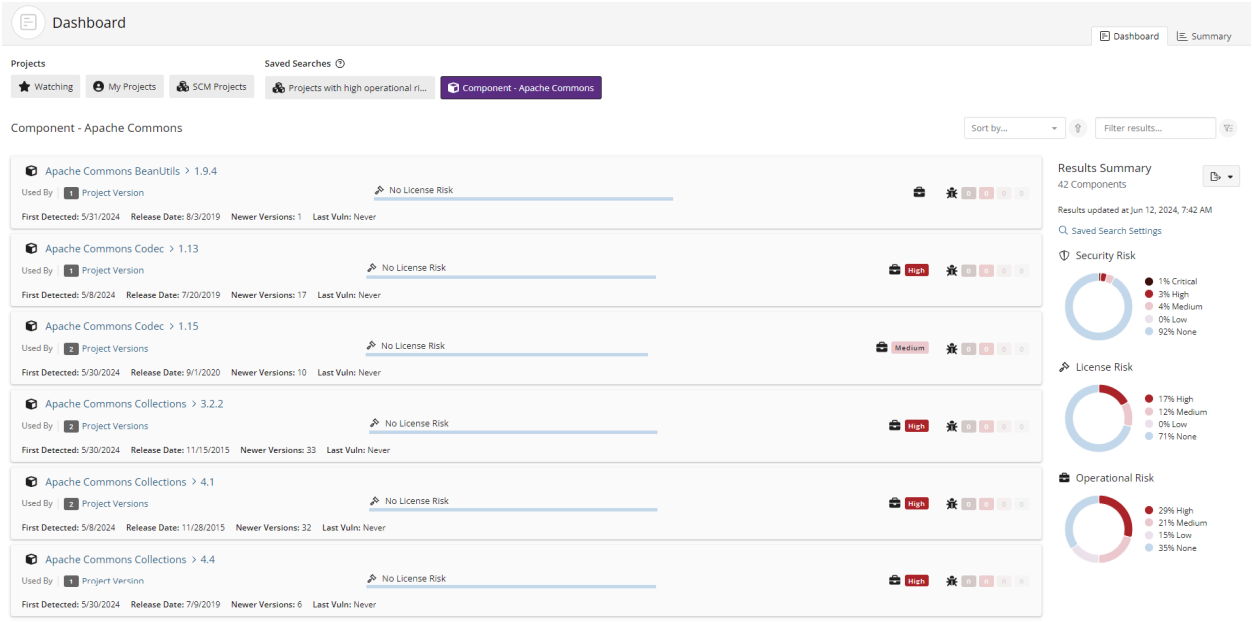


  - Hover over the graph to view the percentage of project versions in this dashboard for each risk level.



  - 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



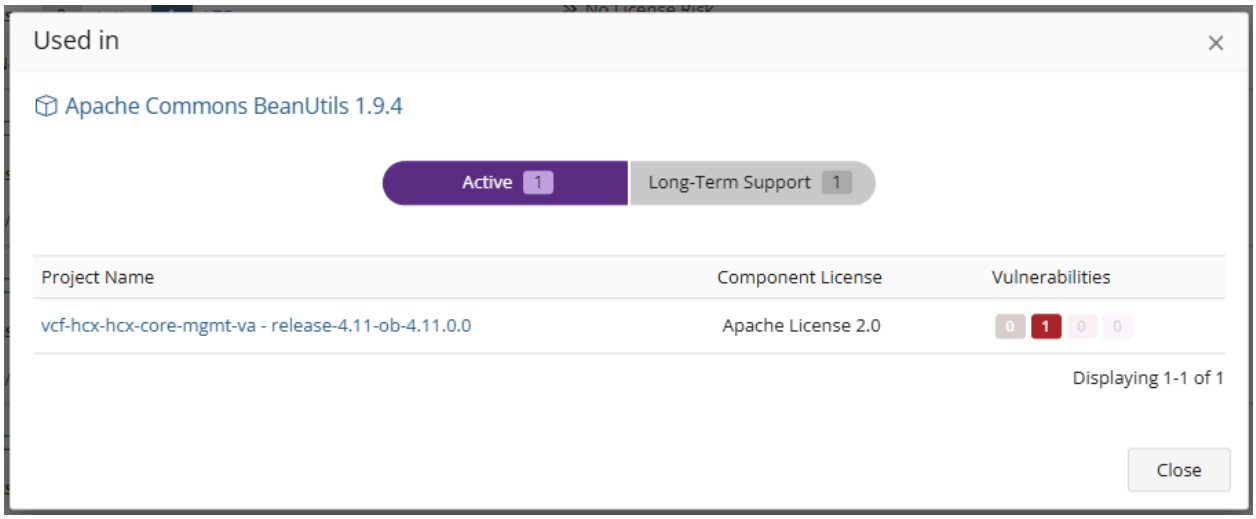
The following information is shown for each component.



-  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**.



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



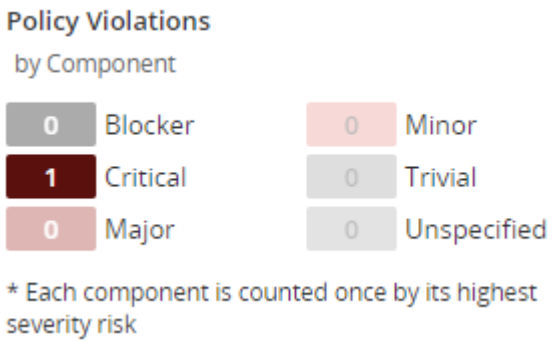
This dialog box shows the project versions that use this version of the component.

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	Lists the vulnerabilities for each severity level, from left to right: Critical, High, Medium, and Low.  Select a value to display the **Security** tab of the Black Duck KB*Component Name Version* page, which lists the vulnerabilities associated with this version of this component.

- 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:

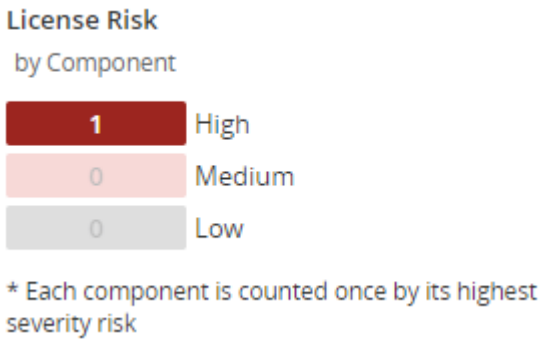


Note: A component is only counted once with the highest policy severity level, not all policy severity levels affecting this component.

- 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.

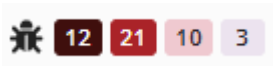


- View the operational risk for this component version:

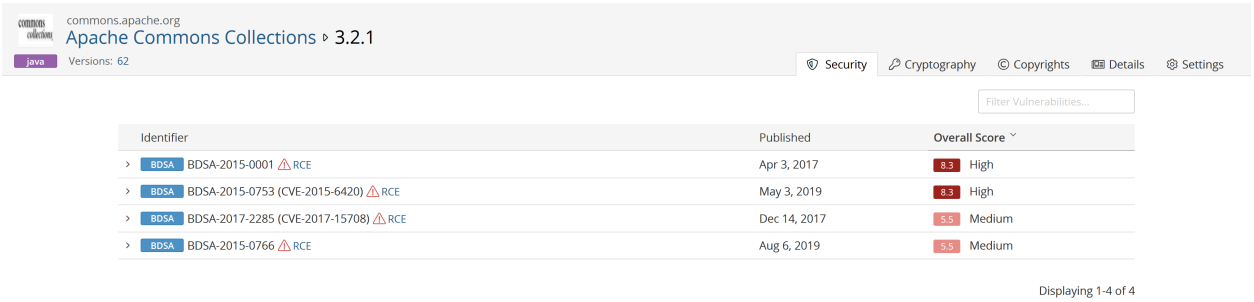
  

- View the number of vulnerabilities by severity associated with this component version for each severity level, from left to right: Critical, High, Medium, and Low.

  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).

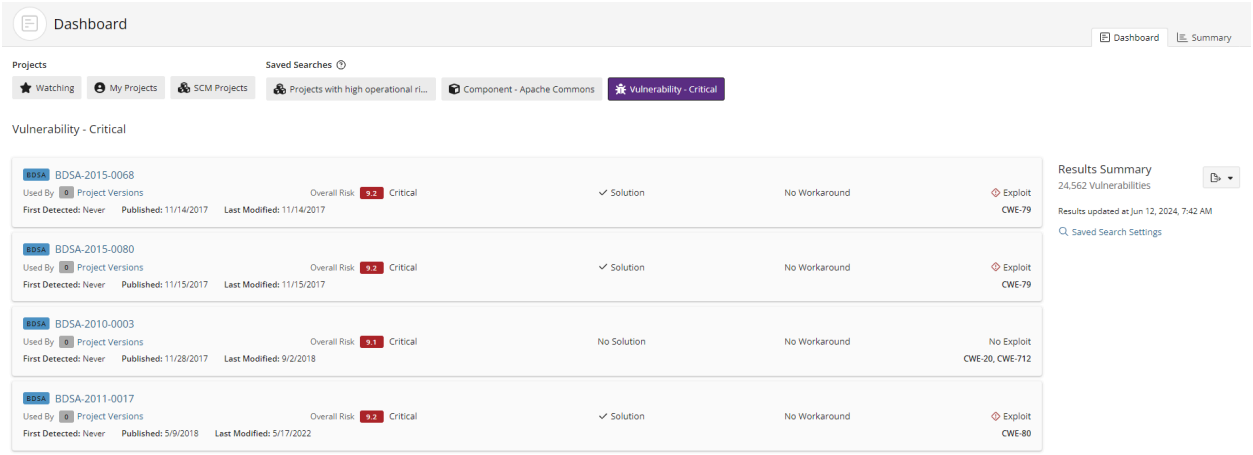
  

  Select a value to display the **Security** tab of the Black Duck KB*Component Name Version* page, which lists the vulnerabilities associated with this version of this component.

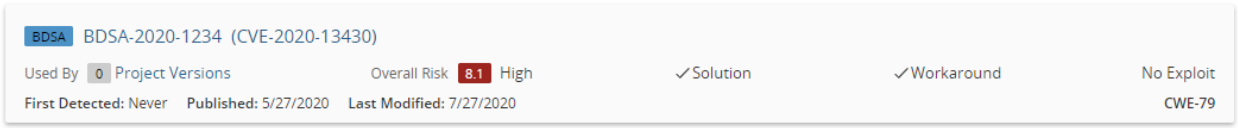
  

- For each component version, the search results also show:
  - Approval status. Status indicates whether this component version has been reviewed.
  - First detected date.
  - Date this component version was released.
  - Number of newer versions.
  - Date when a vulnerability for the component was last updated in Black Duck (by updates from 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



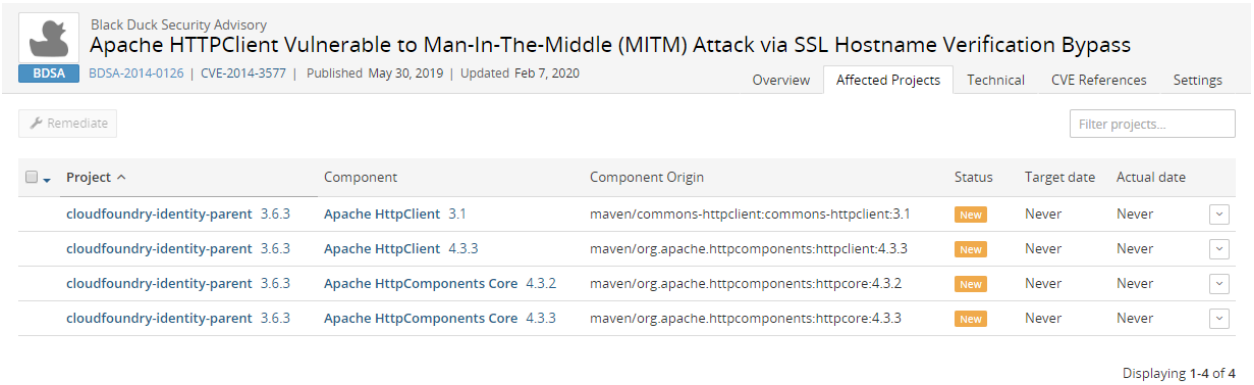
The following information is shown for each vulnerability:



- 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**.

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

- 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.

Scanning Components

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 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 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 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 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 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, 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, 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 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 Black Duck Detect documentation.

Individual file matching

Individual file matching refers to identifying a component based on the checksum of a single file. However, enabling this feature does not guarantee that every file within an open-source library in the Knowledge Base (KB) will match to its corresponding component. Various factors, including code reuse and the organization of files within the repository, can influence the matching process. Therefore, while individual file matching is a powerful tool, the results may vary based on the specific files being scanned.

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, Black Duck 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 Black Duck 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 Black Duck Detect documentation for a list of supported package managers.

Scanning tools

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

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

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

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 Black Duck 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.

About Black Duck Detect Desktop

Detect Desktop provides an interface to make it easier to scan code.

With 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.
- View uploaded scans.

To get started with Detect Desktop:

1. Download Detect Desktop
2. Install Detect Desktop.
3. Configure Detect Desktop with your Black Duck SCA server settings and complete the installation process.

System Requirements

- **Operating Systems**
  - Windows 11
  - macOS 14
  - Linux (Ubuntu 20.04 or newer)
- **System Memory**
  - 8 GB RAM
- **Available hard disk space**
  - 150 MB+ free space
- **Other**
  - Java: OpenJDK 64-bit version 8, 11, 13, 14, 15, 16, 17, or 21. If using Java 11: 11.0.5 or higher is required.

Ensure that your system complies with the requirements for Black Duck Detect.

- For the latest system requirements, please refer to the Black Duck Detect system requirements page.

Downloading Detect Desktop

1. Log in to Black Duck SCA.
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 Black Duck Detect (Desktop)** section to download the executable from Google Cloud Storage.

Installing Detect Desktop

1. **Run the Executable**

   Run the executable downloaded above to install Black Duck Detect Desktop.

2. **Data Migration (if applicable)**

   If you are upgrading from a previous version of Black Duck Detect Desktop, an option will appear to migrate data from the earlier version.

Important Notes

- **Uninstallation of Previous Versions**: The installer will not automatically uninstall previous versions of Black Duck Detect Desktop or versions installed in non-default directories. You must manually uninstall all previous versions and fix or delete any shortcuts as necessary.
- **Sandbox Configuration Error**: If you encounter the following error after installation:

  "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/Black Duck Detect/chrome-sandbox is owned by root and has mode 4755." This indicates that your operating system does not support the Sandbox at the kernel layer. To run Black Duck Detect (Desktop) with the Sandbox disabled, enter the following command in the terminal:

  ```
  blackduck-detect --no-sandbox
  ```

Command line options for Windows

- **Unattended (Silent) Install**

  To perform a silent installation of Black Duck Detect, use the following command:

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

- **Install to a Specific Directory**

  To install Black Duck Detect to a custom directory, use:

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

Installing the Linux version of Black Duck Detect Desktop

1. **Download the Executable**

   Download the executable from your Black Duck SCA server, as described above.

2. **Install Black Duck Detect Desktop**:

   Open a terminal and navigate to your Downloads directory:

   ```
   cd Downloads
   ```

   - **For CentOS/RedHat**:

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

   - **For Ubuntu/Debian**:

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

3. **Change Permissions**:

   Change the permissions of the `chrome-sandbox` file:

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

4. **Run Black Duck Detect Desktop**:

   Finally, launch Black Duck Detect Desktop with the following command:

   ```
   ./blackduck-detect --no-sandbox
   ```

Downloading and Installing Detect Desktop

1. Log in to Black Duck SCA.
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 Black Duck Detect (Desktop)** section to download the executable from Google Cloud Storage.
4. Run the executable to install Black Duck Detect Desktop.

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

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

If the Black Duck 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/Black Duck
 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 Black Duck Detect Desktop with the Sandbox disabled, enter the following at the command line:

```
blackduck-detect --no-sandbox
```

Command line options for Windows

- Unattended (silent) install for Black Duck Detect

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

- Installing to a specific directory

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

Installing the Linux version of Black Duck Detect Desktop

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

   ```
   cd Downloads
   ```

   To install on CentOS/RedHat:

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

To install on Ubuntu/Debian:

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

3. Change the permission of chrome-sandbox:

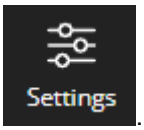
```
cd "/opt/Black Duck Detect"
sudo chmod 4755 chrome-sandbox
```

4. Run Black Duck Detect Desktop:

```
./blackduck-detect --no-sandbox
```

Configuring Detect Desktop

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

1. Click 

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

Black Duck server configuration

To add a server:

1. Select the **Server Configuration** tab and click **Add Server**.
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. Enter your API key (user access token).
4. Click **Submit**.

Black Duck Detect Desktop connects to the Black Duck server and displays the version of Black Duck you are connected to.

To remove an API key or configuration:

1. Select the **Server Configuration** tab.
2. Click in the row of the server and select:
   - **Remove API Token** to remove this API token.
   - **Delete Configuration** to remove both the server and API token.

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

Detect Desktop automatically retrieves and uses the appropriate version of Detect CLI based on the Black Duck instance it connects to. The CLI is only downloaded if a different version is required, ensuring compatibility without unnecessary updates. Additionally, Detect Detect respects and downgrade settings configured in your Black Duck environment, so you don't need to manage CLI versions manually.

Proxy Server

Accessing Black Duck Detect Desktop through a proxy is supported. Black Duck 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 Server** tab.
2. Select either **No Proxy Server**, **Use System Proxy Server Settings**, or **Manual Proxy Configuration**.
3. If you selected **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.

Black Duck Detect

Optionally, select **Black Duck Detect** and if necessary, define any Black Duck 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 Black Duck 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

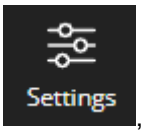
When connecting to Black Duck, you can ignore invalid or insecure SSL certificates.

1. Select the **Server Configuration** tab.
2. Check the **Ignore invalid or insecure SSL Certificates** checkbox.
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.

Alternatively, if you want to imported a self-signed certificate, this can be done following the standard keytool import process for your JRE.

Identify the location of the JRE being used by Black Duck Detect:

1. Click 

, located in the upper right corner display the Settings page.
2. Select the **Black Duck Detect** tab.
3. Select **paths** from the Properties menu. Alternatively, type **paths** in the Search Properties search field to narrow the options displayed.
4. If the **Java Executable** field has no value, Black Duck Detect will use the JRE installed under `$JAVA_HOME` set in your system environment variables.

Now that the location of the JRE that Black Duck Detect is using is known, the certificate should be imported to the relevant `cacerts` file (typically found in the `lib\security` folder).

1. Within a terminal session, run the following command (changing the paths to suit):

```
keytool -import -trustcacerts -keystore <path_to_keystore> -file <path_to_certificate> -alias
 <alias_for_cert>
```

2. You will be prompted for a password. Provide it and press enter.
3. You will be prompted whether or not to trust the certificate. Inspect the contents and accept as appropriate.

Note: It may be necessary to also import any intermediary certificates associated with a chain. If you encounter any issues with the import process, please contact your IT department.

Scanning with Detect Desktop

Black Duck Detect Desktop makes it easier to scan:

- Source directories
- Binaries or executables
- Containers
- 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.

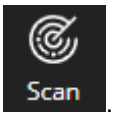
To specify project and/or version names:

1. Click  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 your code with Detect Desktop

Detect Desktop supports multiple scanning methods to help you analyze different types of code artifacts from your local environment. Whether you're working with source code, binaries, or container images, Detect Desktop simplifies the process by automatically detect and applying the the appropriate technique based on your input.

To perform a scan:

1. Click 

.
2. From the **What type of scan?** dropdown menu, select the desired scan type:
   - **Source Directory**. Scans source and build data for a project.
   - **Binary/Executable**. Scans a single binary or executable file.
   - **Container Scan**. Scan each layer of your container.
   - **Docker Inspector**. Scans an image name or distribution file (.tar) with Docker Inspector.
3. When selecting **Source Directory**, **Container Scan**, or **Docker Inspector** as the scan type, you must also specify a Scan Mode:
   - **Intelligent**: Performs a full analysis using multiple scanning techniques. Recommended for most use cases where accuracy is important.
   - **Rapid**. Prioritizes speed over depth by focusing on faster scanning methods. Ideal for quick assessments or when immediate feedback is needed.
   - **Stateless**. Runs the scan without storing results on the scanning machine and avoids using local scan state. Useful in automated or ephemeral environments where persistence is not required.
4. The way you provide input to Detect Desktop varies depending on the selected Scan Type. After choosing a scan type from the What type of scan downdown menu, you'll be prompted to select the appropriate file, direct, container image for analysis:
   - For **Source Directory**, **Binary/Executable**, and **Container Scan**: Click  and select the directory/file/container scan target you would like to scan.
   - For **Docker Inspector**, select either:
     - **Docker image name**: Provide a container image name (e.g., alpine:latest).
     - **Docker archive (.tar)**: Click  and select a Docker archive file (.tar) .
5. Click **Scan**.

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

You can optionally configure **Project Settings** or **Scan Settings** by clicking  next to each section. If your organization has a Snippet Scanning license and you wish to enable it, go to **Scan Settings**, select **Snippet Matching**, and toggle it on.

When the scan is complete, select the **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 **Black Duck** tab.

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.

Creating a scan file (offline mode)

Detect Desktop allows you to generate a scan file that can be uploaded to Black Duck later. This is especially useful for offline or air-gapped environments, or when your organization prefers to review scans before upload.

Note: Snippet scanning is not supported in offline mode, as it requires communication with the Black Duck server.

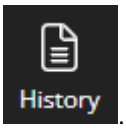
To create a scan file, toggle **Offline Mode** to the on position and perform a scan as described above.

Once the scan is complete, go to the **History** tab to view and manage the generated scan file. You can later upload this scan file using the command line, or by using the Black Duck UI.

Managing your Detect Desktop scans

The **History** tab lets you view and manage scans that were created on your local machine—particularly useful when working in offline mode or reviewing scans before you upload.

To open and use the **History** tab:

- Click 

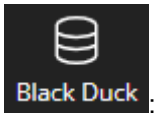
.

A list of previously saved scans appears in the left-hand column.

From this tab, you can:

- **View scan details**: Select a scan to review metadata such as scan status, operations performed during the scan, and the Detect parameters used for the scan.
- **Locate the scan file on your system**: Click **Show Files** to open the file location on your machine.
- **Delete a scan**: Hovering over a scan in the list and click . Click **Yes** to confirm.

Viewing uploaded Detect Desktop scans

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

.

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 Black Duck 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 Container Scanning

Container scanning improves the user experience for managing risk found in containers enabling you to manage the risk by container layer. When scanning a container using this feature, Black Duck SCA creates a new type of project that manages a new container scan. The container project displays the aggregated BOM and risk, but it also provides a way to view risk layer by layer, specifically adding support for components that are added or removed on a layer.

Note: In order to take advantage of this feature, you must have **Black Duck Secure Container (BDSC)** enabled on your product registration key.

Supported Container Image Formats

The following container image formats are supported under the Secure Container module:

- **Docker**

  Docker is one of the most widely used container image formats, allowing developers to package applications and their dependencies into standardized units. Black Duck SCA provides comprehensive scanning for Docker images, enabling users to identify vulnerabilities and manage risks effectively.

- **OCI (Open Container Initiative)**

  OCI is a set of open standards for container images and runtimes that promotes interoperability and compatibility across different container ecosystems. Black Duck SCA supports OCI images, ensuring that users can scan and analyze these containers for vulnerabilities and compliance with security policies.

- **Chainguard**

  Black Duck SCA Secure Container supports Chainguard container images, providing organizations with enhanced security for commercially maintained containers. This integration combines Chainguard's curated component inventory and vulnerability intelligence with Black Duck SCA's analysis capabilities, resulting in more accurate vulnerability detection, quicker access to security fixes, and reduced manual triage efforts for developers and security teams.

How do I perform a container scan?

Container scans are performed with Detect by using the following option during a scan:

```
--detect.tools=CONTAINER_SCAN
```

See Container Scanning in the Detect documentation for more information on the container scanning process.

What is a container layer?

When Docker builds an image, the image is composed of layers which represent a specific modification to the container. These modifications can include commands such as `RUN`, `COPY`, `FROM`, etc.

What is a container project?

The output of a container scan automatically creates a container project in Black Duck and associates the scan to the project as a project version. This project version cannot have any other type of scan type. Multiple container scans can be mapped to a single project version.

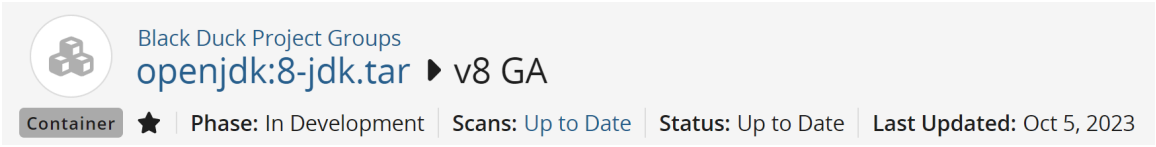
The following are valid combinations of scans (code locations) that can be mapped to single project version:

- Any combination of non-container scans mapped to project version.

- One or many container scans mapped to project version.
- One or many container scans along with one or many IaC/Malware scans mapped to the same project version.

All other combinations of mapped code locations are invalid and the scan process will fail if the mapping of corresponding code location will result in invalid combination.

When you open the resulting BOM, the project's header will indicate that it is a container project:



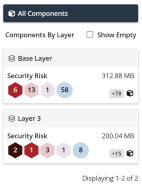
To view the containers mapped to this project version, click the  button. This will open a search model from which you can either type the name of the container you want to view or you can select it from the presented list.



What is in a container project BOM?

The BOM created during a container scan is, in many ways, the same as a regular BOM generated by other scanning processes with a few differences:

- The left side of the container scan BOM displays a list of components found by layer.



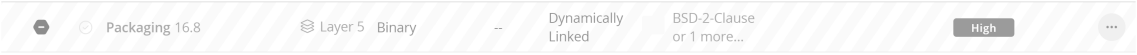
  Here you can find more details on security risks found throughout the container. By defaut, only the layers with security risks will be displayed but you can check the **Show Empty** checkbox for a complete list of all layers in the container.

  Layers containing security risks display the following information:

  - The layer where the security risks are found.
  - The count of components with critical, high, medium, low, or no security risks.
  - The file size of the layer.
  - The number of components added or removed from the container.

  Clicking the individual layers updates the table in the right-hand pane which displays specific details on the components and security risks found on that layer. On layers where components have been

removed, the removed components will be grayed out indicating that no action is required to remedy this security risk.



- The container scan BOM does not have a Source tab which would allow you to manage the files associated with BOM components.

What can I do with a container scan BOM?

As mentioned above, a container scan BOM is much like a regular BOM. For more information on how to manage the information in a BOM, please see Understanding the information in a project version's BOM.

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 Black Duck 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 Black Duck 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 Black Duck 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 Black Duck 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 policy conditions are:

  Component 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
- Published Date
- Vendor Fix Date
- Workaround
- Solution
- CWE IDs

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
    - RCE (Remote Code Execution)

    All other policy conditions are silently ignored.

    Upgrade guidance information is not used as policy conditions, however it is returned as part of Rapid Scan results separately to the policy violations output.

    Note that the severity of the policy rule determines how Black Duck Detect reacts to violations. Black Duck Detect treats violations of Blocker and Critical policies as errors; if any such violations are found, Black Duck 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, Black Duck Detect treats violations as warnings; Black Duck Detect will print warning messages, but they alone will not cause Black Duck Detect to terminate with a non-zero exit code.

- Use Black Duck Detect 8.0 and later.
- Enable Rapid Scanning in Black Duck Detect by including these properties:
    - **--detect.blackduck.scan.mode="RAPID"**

    Refer to the Black Duck 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 Black Duck 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 Black Duck 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

Scan Custom Config File

The policy overrides config will be ingested by Black Duck through a custom config YAML file that a developer could check-in to SCM alongside other build config files. Detect or CodeSight would then archive that file and scan header file and upload through the "Initialize Scan With Custom Config" endpoint when creating a scan.

```yaml
---
version: 1.0
policy:
  overrides:
  - policyName: policyA
    components:
    - name: component1
      version: version1
    - name: component2
  - policyName: policyB
    components:
    - name: component3
      version: version3
```

Each policy override must apply to a list of specific components, on a specific version (e.g. component1 + version1) or on all versions (e.g. component2).

The endpoint to initialize a rapid scan with a custom config is:

`POST /api/developer-scans`

`Consumes: application/vnd.blackducksoftware.developer-scan-1-ld-2-yaml-1+zip`

This endpoint's request body is a zipped archive of the custom config YAML file and the bdio header file.

BOM Comparison Modes

By using the `X-BD-RAPID-SCAN-MODE` header or detect option `detect.blackduck.rapid.compare.mode` and providing it one of the values below, you can change the Rapid Scan mode.

Set the compare mode of rapid scan:

- `ALL`: The default operation. It will evaluate policy rules that are `RAPID` or (`RAPID` and `FULL`). When the header is absent, this is the default behaviour.
- `BOM_COMPARE`: Will evaluate policy rules like the `ALL` option, but will now evaluate differently based on the type of policy rule modes. When the policy rule is (`RAPID` and `FULL`) it will behave like `BOM_COMPARE_STRICT` but if the policy rule is only (`RAPID`) it will evaluate the result of the rapid scan against the policy ignoring results in the BOM.
- `BOM_COMPARE_STRICT`: Will only evaluate policy rules that are (`RAPID` and `FULL`). Policy violations are compared to the existing project version BOM. If the policy violation was already known and visible in the BOM (active or overridden) it is not part of the rapid scan positive result, it will still be part of the full result following existing restrictions.

In order to run either of the `BOM_COMPARE` modes there must be an existing project version in HUB.

Enabling full results data for BOM support

You can configure Rapid Scan to provide a full results format to include data points for BOM support. To do so, set the following environment variable: `BLACKDUCK_RAPID_SCAN_EXTENDED_DATA=true`. The additional data points include:

- componentDescription
- Homepage link
- OpenHub link
- Declared License definition
- Release date
- Meta links to component id component version id and component version origin URIs
- External namespace
- Package URL
- SPDX license ID
- Source (NVD or BDSA)
- Match type

About Correlated Scanning

Correlated Scanning is a scanning method which allows different matching technologies to scan the same application target and correlate results together to perform more accurate component and component version matching.

Black Duck SCA currently supports correlation between single signature scans and one/many package manager scan results only. Correlated scans will continue to each get their own unique ID, but will share a UUID called a correlation ID.

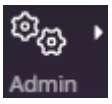
Prerequisites for correlated scanning

- **Detect:** Version 10.0.0 or newer
- **SCA Scan Service**: Ensure this service is available and properly configured.
- **Black Duck:** Version 2024.10.0 or newer
- **Match as a Service (MaaS):** Must be enabled on your account

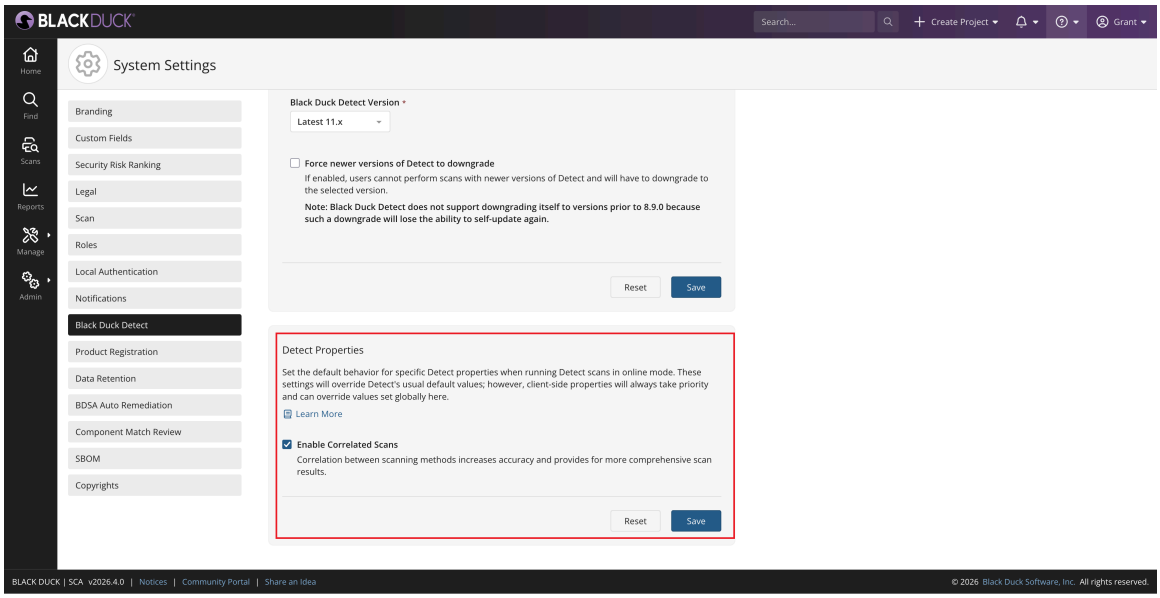
Note: Match correlation is currently not supported for air-gapped KB installations.

Enabling correlated scans

To enable correlated scans:

1. Log into Black Duck SCA as a System Administrator user.
2. Select 

→ **System Settings**.
3. Select **Black Duck Detect** in the **Scanning** section.
4. Navigate to the **Detect Properties** section.



5. Check the **Enable Correlated Scans** box.

Performing a correlated scan

Correlated scans are executed with Detect with the additional flag:

```
--detect.blackduck.correlated.scanning.enabled=true
```

Once the command is performed, Detect will execute one signature scan and one package manager scan. This will result in two code locations (one for each scan) mapped to the desired project version. BOM results in this project version will be presented in the same way as for non correlated scans (signature and package manager scans mapped to the same project version).

Please note, snippet scanning or using the following option in Detect is currently not supported for correlated scanning:

```
--detect.blackduck.signature.scanner.snippet.matching=SNIPPET_MATCHING
```

⚠ **Warning:** The correlated scan flag is only supported for single Signature and one/many Package Manager scan results only. Using it with other scan types is not recommended.

About SCA Scan Service

SCA Scan Service (SCASS) is a scalable solution for performing software composition analysis scans outside of the traditional Black Duck SCA environment. SCASS supports Package Manager, Signature, Binary, and Container scans, making it a versatile choice for various scanning needs.

This services provides significant benefits for both on-premise and hosted customers:

- On-premise customers: Resource requirements for non-specialized scanning are greatly reduced, steamlining infrastructure needs.
- Hosted customers: Cloud Infrastructure efficiency improves with dynamic scaling based on overall scan demand, enhancing performance and reducing operational overhead.

Key functions of SCA Scan Service

SCA Scan Service enables the following functionality:

- Scanning of provided code artifacts using one or more scan type methodologies.
  - Build artifacts, repositories, binary, and Docker images.
  - Signature, package manager, binary, and Docker container scan type methodologies.
- Matching of evidence to OSS components.
  - Signatures, packages, and Docker layer digests.
  - Singular and correlative matching techniques.
- Results delivery.
  - Unopinionated results of identified components, their declared licenses, and their associated vulnerabilities.
  - Intermediate evidence data (BDIO) in support of replay and serviceability.

Performance Improvements

We have conducted extensive performance and scalability testing with the SCA Scan Service. As a result, scan performance has improved across the board, along with a reduction in resource requirements.

We strongly recommend that customers utilize at least `Gen05/120sph` hardware scale for production environments, as `10sph` is not suitable or supported for production use. The Gen05 hardware configuration demonstrates a significant reduction in resource requirements, utilizing approximately 25% less hardware for the same scan volume compared to Gen04. This improvement is largely attributed to enhancements and optimizations made in the SCA Scan Service (SCASS).

However, please note that individual scan performance will vary based on specific scan configurations, hardware setups, and network bandwidth.

For more information on hardware resource reductions, please refer to our announcement in the 2025.1.0 Release Notes and the Black Duck Hardware Scaling Guidelines.

Security considerations

As part of the SCA Scan Service (SCASS) implementation, it is crucial to ensure the security of communication sessions with the service hosted at **scass.blackduck.com**. While there have been no changes to the data transmitted, internal processing enhancements have been made to improve efficiency. Customers are encouraged to engage their security teams, review relevant policies, and stay informed about the rollout progress, especially concerning binary and container scanning features. This proactive approach helps maintain robust security practices while utilizing SCASS.

Other features

SCASS also enables Correlated Scanning, a new feature that leverages SCASS to enhance match results by combining insights from multiple scanning techniques.

Additionally, SCASS delivers faster delivery of scanning bug fixes, independent of Black Duck release cycles. While scan results are stored transactionally, this streamlined service enhances the flexibility and scalability of scanning across platforms.

Please note, this feature must be enabled on your product registration key to take advantage of the service. With SCASS, you can simplify resource management and enjoy a more scalable scanning experience. Contact your Black Duck representative to learn more.

Signature Scanner

Using the Signature Scanner

The Signature Scanner is the default method for scanning your code. However, you may also use Black Duck Detect or Black Duck Detect (Desktop) 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 Signature Scanner. Client systems must have a minimum of 6 GB of RAM.

Downloading and installing the Signature Scanner CLI

Ensure your client meets the requirements and then download and install 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 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 Signature Scanner.

To download the Signature Scanner CLI from the Black Duck user interface:

1. Log in to Black Duck SCA.
2. Navigate to the drop-down menu under your username, and select **Tools**.
3. On the Tools page under **Legacy Downloads**, click the expand arrow 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:



Installing the Signature Scanner CLI on Alpine Linux or ARM64

Install a supported version of Java. Then set the `BDS_JAVA_HOME` environment variable to the same path at the `JAVA_HOME` variable for that Java installation. This tells Detect to use the correct Java version for the Signature Scanner.

For example:

```
export BDS_JAVA_HOME=$JAVA_HOME
```

Or:

```
export BDS_JAVA_HOME=/your/jre/root
```

Defining your version of JRE for 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, 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.

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.

How to run a component scan with the Signature Scanner command line

This section provides instructions for executing a component scan using the Signature Scanner command line tool. A component scan allows you to identify all components within a specified archive or directory of files.

Attention: By default, the Signature Scanner does not include components declared in supported package management files. Use Black Duck Detect to discover declared dependencies.

Command Line Parameters

This section lists all available parameters for the Signature Scanner command line tool, along with their descriptions. Each parameter modifies the behavior of the scan, allowing for customization based on your specific requirements. Review the parameters carefully to ensure you use the correct options for your scanning needs.

Command syntax:

```
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.
--bdio2	Use BDIO 2's JSON-LD format for storing and transmitting scan evidence to the Black Duck server.
--binaryAllowedList <*file extensions*> **--sourceAllowedList** <*file extensions*>	Use to create approved signature lists. • **--binaryAllowedList** *x, y, x* 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.
--cloneFrom <*version*>	Specifies the name of an existing project version to use as a clone. To clone a project version, use the: • **--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. For example, to clone version 1.0 of project SampleProject to a new version called 2.0, you would include these parameters: `--project SampleProject --release 2.0 --cloneFrom 1.0`
--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.
--correlationId <*value*>	Used by the Black Duck system to provide the ability to correlate results from different types of scan (package manager scan and signature scan) for the same code location to improve the reliability of results.

Parameter	Description
--debug	Shows debug output.
--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 Signature Scanner outputs a BDIO file with the original file metadata used for scanning. The Signature Scanner does not connect to or post the scan to the Black Duck server. Note that the `data` directory is created inside the specified directory.
--exclude <*pattern*> **--exclude-from** <*Filename*>	Excludes specified directories from scanning. For detailed guidelines and automatic exclusions, see Exclusion Parameter Guidelines.
--fs-wait-time <*number of minutes*>	Number of minutes the scan client waits for the File System scan (signature scan) to be in either the completed or error status before starting the snippet or string search scan. Must be a positive integer number.
--host <*host*>	Server hosting the Black Duck installation.
--individualFileMatching <*option*>	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: • **source**. Performs individual file matching only on files with this extension: `.js`, `c`, `h`, `c+`, `cc`, `cpp`, `cxx`, `hh`, `hpp`, `hxx`, `h+`, `cs`, `idl`, `rc` • **binary**. Performs individual file matching on files with these extensions: `.apklib`, `.bin`, `.dll`, `.exe`, `.o`, and `.so`. • **all**. Performs individual file matching on all files with extensions matching "*". • **both**. Performs individual file matching on file extensions only using both approved signature lists. Any other value will be ignored and individual file matching will remain disabled. Click here for more information on using this parameter with approved signature lists.
--insecure	Ignores TLS validation errors, allowing the Signature Scanner to connect to the Black Duck server.
--license-search	Enables searching for embedded licenses.
--logDir <*log directory*>	Location of the `log` directory which contains all Signature Scanner log files. You must specify the **--logDir** parameter for log files to be saved.
--matchConfidenceThreshold <*value*>	Specify the match confidence threshold as a percentage value between 1 - 100. A low value returns more matches; whereas a high value returns fewer matches.
--max-request-body-size <*size*> **--max-update-size** <*size*>	Controls how scan data is streamed (buffered) from the Signature Scanner to Black Duck. 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

Parameter	Description
	your network or increasing the default values if your network is highly stable. • **--max-request-body-size**. Size of the main request that uploads the scan data for scanned paths. Specify a value, in bytes. The default is 20000000 bytes. The recommended minimum value is 2000000 bytes; the recommended maximum value is 2000000000 bytes. • **--max-update-size** Buffers an update request to inform Black Duck SCA when Signature Scanner has completed uploading the data of individual URIs (scanned paths). 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.
--min-scan-interval=<*time in hours*>	Minimum scan interval setting (in hours), which may be used to limit daily rate of the Signature Scanner 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*>	Unique name identifying this scan. This name is displayed on the Scans page. Click here for more information. **Note:** The **--name** parameter is not supported when specifying multiple scan paths in a single command line.
--no-prompt	Non-interactive mode. Instead of the **--no-prompt** parameter, you can set the BD_HUB_NO_PROMPT environment variable to enable non-interactive mode.
--no-signature-generation	Use the traditional Signature Scanner instead of the Enhanced Signature Generation.
--password <*password*>	Provides the password for the Black Duck SCA server account.
--port <*port*>	Port on which the Black Duck server instance is listening.
--project <*project*>	Name of the project to which you want to map the scan results. If you specify a project, you must specify a version. • If the project and project version exist, the Signature Scanner maps or remaps the scan results. • If the project exists, but the version does not, the Signature Scanner creates the version and maps the scan results.
--project-group <*project group name*>	Assigns the project to the designated project group. If the project does not already exist, it is created in the corresponding project group. This parameter has no effect if the project already exists or if the specified project group does not exist.
--release<*release*>	Name of the project version to which you want to map the scan results. If you specify a version, you must specify a project. • If the project and project version exist, the Signature Scanner maps or remaps the scan results.

Parameter	Description
	• If the project exists, but the version does not, the Signature Scanner creates the version and maps the scan results.
--retain-unmatched-files **--discard-unmatched-files**	Used to retain or discard, respectively, any unmatched files discovered by this scan and this scan only. If either option is supplied, project and global retention settings are ignored; otherwise, retention is determined by project or global settings as described in "Settings". Specifying both options with a single scan is an error.
--scheme <*scheme*>	Protocol to use to connect to the server hosting the Black Duck SCA 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 Signature Scanner outputs a JSON file which contains the complete scan status information.
--selfTest	Performs a self-test; will not connect to or post the scan to the Black Duck SCA server.
--snippet-matching **--snippet-matching-only** **--snippet-matching-all-source** **--full-snippet-scan**	These parameters control snippet matching behavior during scans.

--tlscertpass	Forces the Signature Scanner to prompt you for the password for the client certificate.
	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.
	The result of specifying the **--tlscertpass** parameter depends on whether the key is encrypted.
	• If the key *is* encrypted, the scan will fail if you do not set the BD_HUB_CLIENTCERT_PASS environment variable *or* specify the **--tlscertpass** parameter.
	  • If you set the environment variable *and* specify the **--tlscertpass** parameter, the Signature Scanner prompts you for the password; it does not check the password value against the value specified in the environment variable.
	• If the key *is not* encrypted, regardless of whether the BD_HUB_CLIENTCERT_PASS environment variable is set:
	  • Specifying the **--tlscertpass** parameter forces the Signature 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*>	Black Duck client certificate private key file. Automatically sets **--scheme** to https. **Note:** This parameter is optional as the key and certificate can be in included in the key store file specified with **--tlscert**.

Parameter	Description
--tlscert <*certFile*>	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.
--upload-source	Uploads the source file, an optional feature for snippet matching, embedded license search, and copyright text search. This parameter is optional with the **--snippet-matching** or **--snippet-matching-only** parameters or with the **--license-search** and/or **--copyright-search** parameters.
--upload-csv	Generates a CSV file during the scan and uploads it to Black Duck. The CSV file can be downloaded from the Scans page.
--username <*username*>	Black Duck user account with the code scanner role. Instead of specifying a username and password, use the **BD_HUB_TOKEN** environment variable to specify a Black Duck API token.
-V, **--version**	Shows the version information of this tool.
-v, **--verbose**	Sets the logging level to verbose.
Other environment variable: • **BD_HUB_TOKEN**	Used to specify the Black Duck API token which is the preferred authentication method over username and password. Use the Profile page in the Black Duck UI or the api-token-rest-server API to manage API tokens.

Command Line Examples

These examples demonstrate common scanning scenarios using the Signature Scanner command line.

- Scanning and sending scan data toBlack Duck
- Scanning and mapping the scan data

Important:

- All examples require a user account with the code scanner role.
- The examples show only required parameters. Add optional parameters as needed.

Basic Scan

This example shows how to scan files and send the results to the Black Duck server.

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

   For example:

   Linux/MAC OSX:

   `/opt/blackduck/hub/scan.cli-2026.7.2/scan.cli-2026.7.2/bin`

   Windows:

   `C:\scan.cli-2026.7.2\scan.cli-2026.7.2\bin`
3. Run the following command to configure and initiate the scan.

   For example:

Linux/Mac OSX:

```
./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 , which adds the identified components to the project BOM.

HTTPS Scan with Project Mapping

This example demonstrates scanning over HTTPS and automatically mapping the results to a specific project version.

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

   Linux/MAC OSX:

   `/opt/blackduck/hub/scan.cli-2026.7.2/scan.cli-2026.7.2/bin`

   Windows:

   `C:\scan.cli-2026.7.2\scan.cli-2026.7.2\bin`
3. Run the following command to configure and initiate the scan.

   Linux/Mac OSX:

   ```
   ./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.

Command Line Status Codes

When a signature scan completes, it returns one of the following status codes to indicate the result.

Success Codes

Code	Name	Description
0	SUCCESS	The scan completed successfully.

Execution Status Codes

Code	Name	Description
1	FAILURE	Generic failure.
2	NOT_EXECUTED	Scan was not executed because the minimum scan interval was

Code	Name	Description
		not exceeded. See the `--min-scan-interval` parameter.

Command Usage Codes

Code	Name	Description
64	USAGE	Incorrect command usage (wrong arguments or syntax).
65	DATA_ERROR	Input data was incorrect.
78	CONFIGURATION	System found in an unconfigured or misconfigured state.

Resource Access Codes

Code	Name	Description
66	NO_INPUT	Input file does not exist or is not readable.
67	NO_USER	Specified user does not exist.
68	NO_HOST	Specified host does not exist.
73	CANNOT_CREATE	Cannot create output file.
77	NO_PERMISSION	Insufficient permission to perform the operation.

System Error Codes

Code	Name	Description
69	UNAVAILABLE	A service is unavailable.
70	SOFTWARE	Internal software error detected.
71	OS_ERROR	Operating system error detected.
72	OS_FILE	System file error (missing, cannot open, or has syntax error).
74	IO_ERROR	Input/output error on a file.
75	TEMPORARY_FAILURE	Temporary failure.
76	PROTOCOL	Remote system returned an unexpected response during protocol exchange.
79	NO_REGISTRATION	Registration to Black Duck is not valid.

You can also find more information about these exit codes here.

Exclusion Parameter Guidelines

The **--exclude** and **--exclude-from** parameters allow you to specify directories to be excluded from signature scanning. When using these parameters, follow these guidelines:

- **Single Directory Exclusion:** Use `--exclude <pattern>` to exclude a specific directory.
- **Multiple Directory Exclusions:** Use `--exclude-from <Filename>` to specify a file that lists multiple directories to exclude.
- **Automatically Excluded Items:** The scanner automatically excludes certain directories and files.

Excluded Directories	Excluded Files
• `CVS` • `.svn` • `.git` • `.hg` • `.bzr` • `__MACOSX`	• `.cvsignore` • `.git` • `.gitignore` • `.gitattributes` • `.gitmodules` • `.hgignore` • `.hgsub` • `.hgsubstate` • `.hgtags` • `.bzrignore` • `vssver.scc` • `.DS_Store`

- **Exclusion Guidelines:** Follow specific rules regarding leading slashes, directory names, and file formats.
  - Leading and trailing forward slashes are required.

    For example, if you enter `exclude /directory`, a warning message will appear and the directory will not be excluded. If you enter `/directory` in the file, the directory will not be excluded.
  - 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.
- **Additional Methods:** There are two additional methods you can use to exclude directories from scanning:
  - Create an `ignore` file located in the `$HOME/config/blackduck` directory.

    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.

    Lines in the `ignore` file must have the path from source root to the ignored directory, may have multiple subdirectories, and must have leading and trailing forward slashes (/).

    Create one of the following environment variables, as shown here, configured for Linux or Mac OS X:

- ```
  export JAVA_TOOL_OPTIONS=" -Duser.home=<path>"
  ```

  The `.ignore` file must be located here: `<user.home>/.config/blackduck/ignore`

- ```
  export XDG_CONFIG_HOME=<path>
  ```

  The `.ignore` file must be located here: `$XDG_CONFIG_HOME/blackduck/ignore`

- ```
  export JAVA_TOOL_OPTIONS="-Dblackduck.scan.excludesFile=<path>/<file>"
  ```

  The `.ignore` file can be in any location and have any name.

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

- Create individual `.bdignore` files which can be located in any directory.

  Use this file to list the excluded subdirectories in the directory where the `.bdignore` file is located. You must create a `.bdignore` file in each directory that has subdirectories you want to exclude.

You must also follow the exclusion guidelines as described above when using either of these methods.

Tip: Use the **--debug** parameter when excluding directories to ensure that the scanner visited and excluded the directory.

Password Parameter Guidelines

Forces the signature scanner to prompt for the password associated with the user account that has the code scanner role. Here's how to use it:

- **Without a Value**: Specifying the `--password` parameter without a password value prompts the user for the password.
- **With a Value**: If you provide a password value, a warning will appear indicating that passing the password on the command line will not be supported in future versions, and the scan will proceed.

It's recommended to set the `BD_HUB_PASSWORD` environment variable with the server password instead of using the `--password` parameter:

- **If `BD_HUB_PASSWORD` is set**:
  - **With `--password`**: The scanner prompts for the password and ignores the environment variable value.
  - **Without `--password`**: The scanner uses the environment variable without prompting.
- **If `BD_HUB_PASSWORD` is not set**: The scanner will prompt for the password regardless of the `--password` parameter.

Important:

If the `--password` parameter is the last parameter before `<scan_path>`, include `--` to indicate the end of parameters:

```
--password -- <scan_path>
```

Otherwise, the scanner may interpret `<scan_path>` as the password.

Snippet Matching Parameter Guidelines

The following command line parameters control how snippet matching is performed during signature scans:
```

--snippet-matching

Enables a two-phase approach to scanning. First, a component scan is completed for files that have changed since the previous scan. After that, a snippet scan runs on those newly scanned files only.

Black Duck Software recommends using this parameter for snippet scanning.

--snippet-matching-only

Runs a snippet scan only on files that have changed; a component scan is not performed. Ensure you have completed a full file scan prior to using this parameter.

--snippet-matching-all-source

Runs a snippet scan for all files with supported extensions, regardless of whether they belong to unmatched directories or archives.

--full-snippet-scan

Forces the snippet scan to search the KnowledgeBase regardless of locally cached matches from previous scans. This parameter must be used with one of the snippet matching parameters:

- **--snippet-matching**: Performs a signature scan first, then a snippet scan on unmatched files.
- **--snippet-matching-only**: Performs a snippet scan on unmatched files without executing a signature scan.
- **--snippet-matching-all-source**: Performs a snippet scan on all files after a signature scan.

To upload source files, you must use the **--upload-source** parameter, as described in the command section above.

Reducing the number of parameters entered on the command line for 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 Signature Scanner and specifies those parameters that will not change.

1. Open a terminal window and optionally, go to the directory where 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 Signature Scanner.

1. Open a command prompt and optionally go to the directory where 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

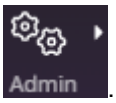
This setting allows users to change the maximum hourly frequency of which signature scans can be performed for a given code location when using the enhanced signature scanning. This will allow customers to reduce the load on their servers, thus making scans running faster and with less errors which result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

The default setting is set to 0, or no maximum 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: For users of Detect 8 and 9, Detect will only finish with a success message in this scenario if the `detect.force.success.on.skip` value has been changed to true. (Default is false). Please see Detect's Configuration Property Details page for more information.

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 

.
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**.

Running an offline component scan using Signature Scanner

If a client does not have access to Black Duck, you can use the command line for 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 SCA.
2. Click 

.
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 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 Signature Scanner is installed.
   Linux/MAC OS X:

   ```
   /opt/blackduck/hub/scan.cli-2026.7.2/scan.cli-2026.7.2/bin
   ```

   Windows:

   ```
   C:\scan.cli-2026.7.2\scan.cli-2026.7.2\bin
   ```

3. 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 Signature Scanner is installed.
   Linux/MAC OS X:

   ```
   /opt/blackduck/hub/scan.cli-2026.7.2/scan.cli-2026.7.2/bin
   ```

   Windows:

   ```
   C:\scan.cli-2026.7.2\scan.cli-2026.7.2\bin
   ```

3. 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>
   ```

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.

About Reduced Persistence Signature Scanning

In the source view, many users rarely look at unmatched files from signature scanning, but all of that data is by far the biggest consumer of space in the database. Performance testing also shows that inserting the file data into the database is a significant fraction of scan time.

Reduced Persistence Signature Scanning aims to decrease database size and growth and also increase scan performance by no longer saving data for files that are not matched by signature scanning. However, if your use case requires that unmatched file data be saved, there are settings to retain that data available at the global, project, and/or scan levels.

There is also the option to purge existing unmatched files to further reduce database disk usage.

Settings

Unmatched files are no longer retained by default. However, retention policy can be changed.

- **Global setting**. In System Settings under Data Retention, there is an option to enable retention of unmatched files. When it is not enabled (the default), there is an option to purge all existing unmatched files from all projects. The option to purge is not presented if retention is enabled. Note that the global setting only applies to projects and scans that do not explicitly specify their own setting; similarly, changing the global setting does not affect projects or scans that do specify their own setting.
- **Project setting**. The Project Settings tab has similar options to enable retention and purge existing unmatched files, except that they only apply to files and scans done under the project. Other than scope, the primary difference from the global retention setting is that the project setting has three possibilities: (1) disabled, (2) enabled, and (3) use the global setting (the default). The global setting only affects scans under a project when option (3) is selected.
- **Per-scan setting**. Each individual signature scan can specify that unmatched files discovered by that scan be retained or not retained; see "Scan Options" below. If no retention option is provided by a scan, retention is determined by the project or global setting as described above.

Note that the determination of whether unmatched files resulting from a scan need to be retained is made at the beginning of a scan and cannot be changed afterwards.

⚠ **Warning:** Once unmatched files are purged, they cannot be recovered except by restoring from backup.

Custom signature matching

When custom signatures are enabled for a project, unmatched files in that project must be retained for the feature to work so that scans of other projects can match against them. In such cases, retention is enabled by default and cannot be disabled unless custom signatures are disabled first.

Note that if a project's custom signature setting is changed from disabled to enabled, unmatched file retention will automatically be enabled as well. However, if retention was previously disabled, unmatched files within that project will be missing, and custom signature matching will not work as expected. In such cases, all versions in that project will need to be rescanned so that all files can be retained.

Scan options

The scan CLI has two new options, `--retain-unmatched-files` and `--discard-unmatched-files`, which will retain or discard, respectively, any unmatched files discovered by this scan and this scan only. If either option is supplied, project and global retention settings are ignored; otherwise, retention is determined by project or global settings as described in "Settings". Specifying both options with a single scan is an error.

If scanning with Detect, use one of the following arguments as appropriate:

```
--detect.blackduck.signature.scanner.arguments='--retain-unmatched-files'
```

```
--detect.blackduck.signature.scanner.arguments='--discard-unmatched-files'
```

Use cases requiring retention

Retention must be manually enabled to support two potential use cases:

1. Snippet-only scans operating on files previously discovered by a signature scan. If unmatched files from that signature scan were not retained, the subsequent snippet-only scan will be unable to scan them.

   Note: Snippet-only scans require unmatched file retention to be enabled.

2. Workflows requiring unmatched files be examined.

Temporary retention

Some signature scan options such as snippet matching, license search, and copyright search will require unmatched files to be retained so that those features can operate. However, if unmatched file retention was not enabled for such a scan, the unmatched files will be purged within a short time after the scan is complete.

Note that these temporarily retained unmatched files may be briefly visible in the source tree view until they are purged.

About Stateless Scanning

Stateless Scan is a scan mode that does not create or use any permanent storage within Black Duck, thus there is no bill of material (BOM) stored. It is used to quickly find policy violations within the designated scan target. In order to use the Stateless Signature Scan, you must have the following:

- Black Duck Detect 8.2.0 or later
- Black Duck 2023.1.0 or later
- Hosted KnowledgeBase
- Match as a Service must be enabled

Enabling Stateless Scan mode

Enable this feature by adding --detect.blackduck.scan.mode=STATELESS to a run of Detect.

Restrictions and Limitations

Stateless Scan Mode has a unique set of restrictions, mode of configuration and set of results. It is similar to Rapid Scan Mode however it differs in that it supports usage of the SIGNATURE_SCAN tool:

1. A limited subset of Tools can be run.
   - The currently supported tools are: DETECTOR, BAZEL, SIGNATURE_SCAN and DOCKER.
   - The Stateless Scan will not persist on Black Duck.
   - All other tools are disabled when running in Stateless Scan mode.
2. Stateless Scan and non-persistent SIGNATURE_SCAN
   - To perform a non-persistent Signature Scan in Stateless mode, SIGNATURE_SCAN must be included within `--detect.tools`.

- Permitted tools omitted from the detect.tools list will not be run.

3. Stateless Scan requires Black Duck policies.
   - Stateless Scan only reports components that violate policies.
   - If no policies are violated or there are no defined policies, then no components are returned.
4. Stateless Scan does not support `detect.policy.check.fail.on.severities`
   - Black Duck Detect will fail with `FAILURE_POLICY_VIOLATION` if any component violates Black Duck polices with a CRITICAL or BLOCKER severity.
   - Stateless Scan supports the same policy conditions as Rapid Scan. Click here for a list of policy conditions that are supported by Stateless Scan.
5. Stateless Scan does not support `detect.policy.check.fail.on.names`
6. Stateless Scan will not create a Project or Version in Black Duck.
7. Stateless Scan when running `SIGNATURE_SCAN` requires communication with Black Duck.

How to invoke a stateless scan

To invoke Stateless scan only:

- `--detect.tools=SIGNATURE_SCAN --detect.blackduck.scan.mode=STATELESS`

To invoke Stateless package manager scans:

- `--detect.tools=DETECTOR --detect.blackduck.scan.mode=STATELESS`
- `--detect.tools=BAZEL --detect.blackduck.scan.mode=STATELESS`
- `--detect.tools=DOCKER --detect.blackduck.scan.mode=STATELESS`
- `--detect.target.type=IMAGE --detect.blackduck.scan.mode=STATELESS`

Stateless scan results

Unlike persistent scans, no data is stored on Black Duck and all scans are done transiently. These scans are primarily intended to be fast, although the `SIGNATURE_SCAN` can take some time as communication with Black Duck is a requirement.

The results are saved to a json file named 'name_version_BlackDuck_DeveloperMode_Result.json' in the Scan Output directory, where name and version are the project's name and version.

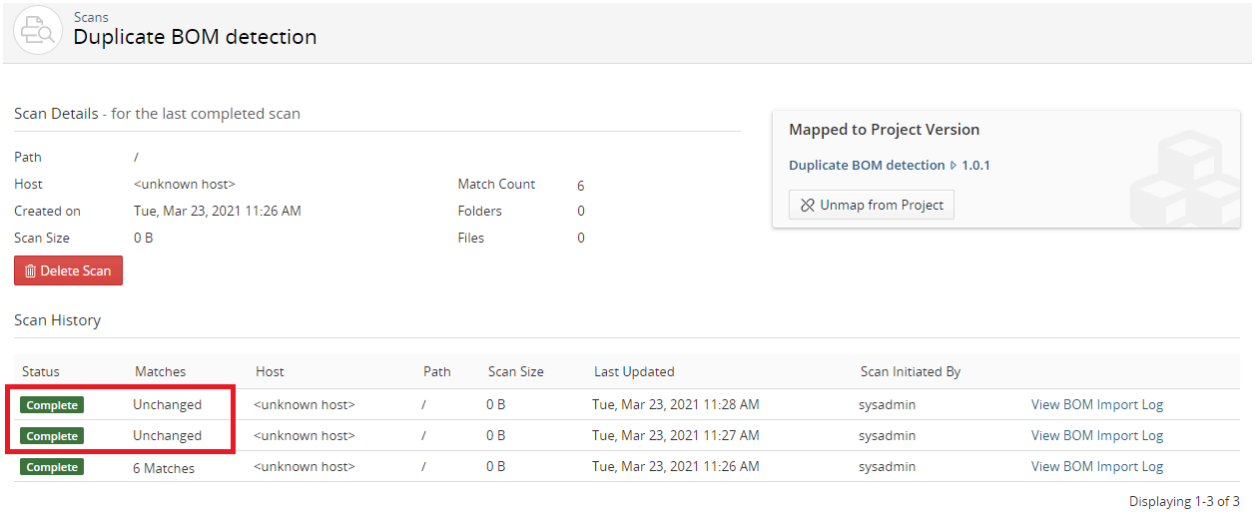
```
2021-07-20 13:25:18 EDT INFO  [main] --- Stateless Scan Result: (for more detail look in the log
 for Stateless Scan Result Details)
2021-07-20 13:25:18 EDT INFO  [main] ---
2021-07-20 13:25:18 EDT INFO  [main] ---   Critical and blocking policy violations for
2021-07-20 13:25:18 EDT INFO  [main] ---    * Components: 0
2021-07-20 13:25:18 EDT INFO  [main] ---    * Security: 5
2021-07-20 13:25:18 EDT INFO  [main] ---    * License: 0
2021-07-20 13:25:18 EDT INFO  [main] ---
2021-07-20 13:25:18 EDT INFO  [main] ---   Other policy violations
2021-07-20 13:25:18 EDT INFO  [main] ---    * Components: 101
2021-07-20 13:25:18 EDT INFO  [main] ---    * Security: 0
2021-07-20 13:25:18 EDT INFO  [main] ---    * License: 0
2021-07-20 13:25:18 EDT INFO  [main] ---
2021-07-20 13:25:18 EDT INFO  [main] ---   Policies Violated:
2021-07-20 13:25:18 EDT INFO  [main] ---    Security Vulnerabilities Great Than Or Equal to High
2021-07-20 13:25:18 EDT INFO  [main] ---    Warn on Low Security Vulnerabilities
2021-07-20 13:25:18 EDT INFO  [main] ---    Warn on Medium Security Vulnerabilities
2021-07-20 13:25:18 EDT INFO  [main] ---
2021-07-20 13:25:18 EDT INFO  [main] ---   Components with Policy Violations:
2021-07-20 13:25:18 EDT INFO  [main] ---    Apache PDFBox 2.0.12
 (maven:org.apache.pdfbox:pdfbox:2.0.12)
```

```
2021-07-20 13:25:18 EDT INFO  [main] ---    Handlebars.js 4.0.11 (npmjs:handlebars/4.0.11)
2021-07-20 13:25:18 EDT INFO  [main] ---
2021-07-20 13:25:18 EDT INFO  [main] ---   Components with Policy Violation Warnings:
2021-07-20 13:25:18 EDT INFO  [main] ---    Acorn 5.5.3 (npmjs:acorn/5.5.3)
```

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":



Note the following:

- Duplicate BOM Detection is currently for *package manager scans* only and works with any version of Black Duck Detect. No additional Black Duck 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.

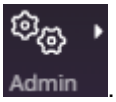
Viewing component dependency duplication

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 

.
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**.

Maximum limit for component matches

Black Duck uses a system property to control the maximum number of nodes (matches) per component added to resulting dependency tree in package manager scan:

`blackduck.match.limit.per.component`

This system property needs to be set for match engine container, therefore the following must be added to the `MATCHENGINE_SERVICE_OPTS` environment variable:

```
-Dblackduck.match.limit.per.component=<value>
```

The default value of this system property is 10, thus the number of duplicated components in the tree can not exceed the `blackduck.match.limit.per.component` value (match limit per component). The allowable range of values for this property is 1 to 100 inclusively. If the setting falls outside of that range, it will automatically be set to default value (with corresponding warning in the log).

Component Dependency Duplication Sensitivity still applies: match limit per component restricts number of duplicates above Component Dependency Duplication Sensitivity level, but if the node level is below Component Dependency Duplication Sensitivity, the match is dropped regardless (under condition that it is already added to the tree). In other words, `blackduck.match.limit.per.component` sets maximum number of duplicates that can be added to the dependency tree above Component Dependency Duplication Sensitivity level (below that level, duplicates are dropped).

For example, say a component comes as transitive dependency from many other components, so that there are 100 of these components in the dependency tree.

Black Duck parameters are set to:

- `blackduck.match.limit.per.component = 10`, and;
- Component Dependency Duplication Sensitivity = 5.

In this case, if there are 20 of said components on the level above 5, component dependency tree will have 10 of these components.

If there are 7 components on the level above 5, resulting tree will have 7 components (since remaining 93 components below level 5 will be dropped even when match limit per component is not reached).

Troubleshooting

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 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.

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 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
- Black Duck Detect Desktop
- Black Duck 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 analyzes only the first 1 to 4 MB of data for codeprints. The maximum snippet size can be configured within this range, with the default setting set to 1 MB.

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

If the **Snippets** feature is registered on your product registration, you gain access to the following parameters:

- **--snippet-matching**. Using this parameter enables a two-phase approach to scanning. First, a signature scan is completed. Once the signature scan is completed, a snippet scan is performed on unmatched files or files belonging to unmatched directories/archives.

  Black Duck recommends using this parameter for snippet scanning.

- **--snippet-matching-only**. When using this parameter, a snippet scan is performed on unmatched files or files belonging to unmatched directories/archives; a signature scan is not executed, but it must already exist. 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.

  Note: Snippet-only scans require unmatched file retention to be enabled.

If you have the **Full Snippet Matching** registered on your product registration, you gain additional snippet analysis capabilities, allowing for more comprehensive detection of code matches. These parameters become available:

- **--snippet-matching-all-source**. First, a signature scan is completed. Once that scan is completed, a snippet scan is performed for all files with supported extensions (whether they belong to unmatched directories/archives or not).

- **--full-snippet-scan**. Selecting this parameter performs a snippet scan on *all applicable* 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.

  Using this option may cause significant performance and scalability issues, and should only be used in extreme situations.. If you are interested in enabling this feature on your registration key, please contact Black Duck Support for assistance.

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 Black Duck 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 Black Duck Detect

Use the **--detect.blackduck.signature.scanner.snippet.matching** property to enable snippet scanning in Black Duck Detect. With this property enabled, Black Duck 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

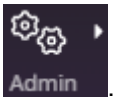
Black Duck does not include any vulnerabilities related to components/versions that are identified through snippet matching *only*: vulnerabilities are not counted when showing the total number of vulnerabilities for a project/project version and are also excluded from vulnerability reports. Black Duck 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 1MB of data for snippet codeprints.

You can modify this default value and select a value from 1MB to 4MB.

To modify the default maximum snippet file size:

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

.
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 4 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

Extension	Language
.adb	Ada
.ads	Ada
.aidl	Interface Definition Language (IDL)
.as	ActionScript
.as8	Assembly
.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

Extension	Language
.com	DCL
.cpp	C++
.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)
.glsl	Fortran (free-format)

Extension	Language
.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
.jar	Java archive
.java	Java
.js	JavaScript
.jsp	Java, JavaScript
.jws	Java
.kt	Kotlin
.l	C
.lhs	Haskell
.lisp	Lisp
.lsp	Lisp
.lua	Lua
.m	Text
.m2	Modula-2
.m3	Modula-3
.m4	Text
.ml	OCaml
.mli	OCaml

Extension	Language
.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
.rs	Rust
.ru	Ruby
.s	Assembly
.sc	Scala
.scala	Scala
.scm	Scheme
.sh	Shell
.sqb	SQL
.sql	SQL
.ss	Scheme

Extension	Language
.st	Smalltalk
.swift	Swift
.tcl	Tcl
.tk	Text
.ts	Typescript
.v	Coq
.vb	Visual Basic
.vba	Visual Basic
.vbe	VBScript
.vbs	VBScript
.vert	OpenGL Shading Language (GLSL)
.vhd	VHDL
.vhdl	VHDL
.vim	Vimscript
.y	Text
.z80	Assembly
.zip	ZIP archive

About custom scan signatures

Your software projects may contain a mix of open source, third-party, and proprietary software components. While 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 where internal components are used in your applications. The internal component would need to be scanned by Black Duck with custom scan signatures enabled with Retain Unmatched File Data enabled.

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

Please note, there may be performance issues when using this feature.

Identifying custom code signatures in your code

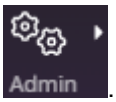
As the scan client scans the code, it generates “signatures” of the files and directories it is scanning. After the scan completes, these signatures are initially sent to the Black Duck KnowledgeBase (KB) web service where the match service uses the signatures to identify the open source components/versions 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. Global Project Administrators and 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:

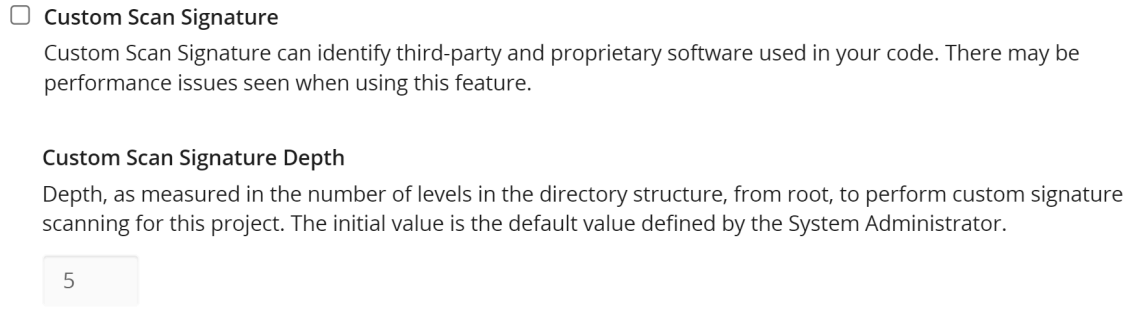
1. Log in to Black Duck with the System Administrator role.
2. Click 

.
3. Select **System Settings**.
4. Select **Scan** from the lefthand menu.
5. In the **Custom Scan Signature Level** section, enter an integer for the number of levels to perform custom signature scanning. You cannot enter 0.
6. Click **Save**. To indicate that the default value has changed, the button changes to **Saved**.

Custom scan signatures on the project level

To enable custom scan signatures for a project:

1. Log in to Black Duck SCA.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the **Settings** tab.
4. Check the **Custom Scan Signature** checkbox. You can also set the custom scan signature depth.



5. Click **Save**.

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.

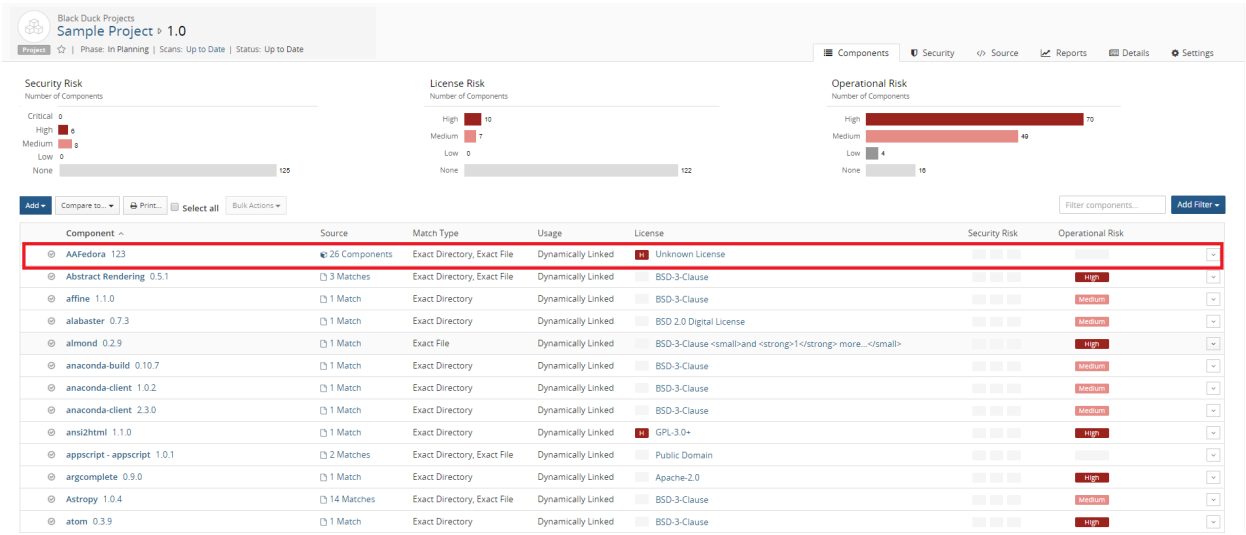
Note: A project needs to be created before and custom signatures must be enabled for that project before taking advantage of this feature. Also, ensure purging unmatched data is disabled **before** running a scan.

To create custom scan signatures:

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

   Use the **--detect.blackduck.signature.scanner.individual.file.matching** property set to **ALL** in Black Duck Detect.

2. Map the scan to a project version.
3. Identify this project as a custom scan signature in the project's **Settings** tab:
   a. Enable the feature
   b. 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.
   c. Click **Save**.
4. Scan your code. The custom scan signature appears in your BOM as a subproject:



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

Note the following:

- If a project contains several versions of a custom scan signature project, the BOM will display only one match to one version of the custom code signature project.
- If the custom scan signature project contains open source components, values for security and operational risk may also appear in the BOM.
- Although you may have selected only one custom code signature project, if you have scanned several projects, you will experience performance issues.
- Policy violations within the subproject will not appear in the BOM. However, a policy violation will appear in the BOM for the subproject if a policy rule is violated at the project level.

- Users who do not have permission to the subproject will not be able to drill down to view additional data about that project version.
- A Custom Scan Signature filter has been added to the Project dashboard and the BOM page to help you find custom scan signature projects.

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:
   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.

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, x* 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 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
  - Matches with all file extensions.
- **both** option
  - *Only* uses the file extensions from your SHA-1 and clean SHA-1 signature list (**binaryAllowedList** and **sourceAllowedList**).

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 SCA.
2. Click 

.
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 contains 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 calender selector allowing you to choose between two dates.

Removing filters

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

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.

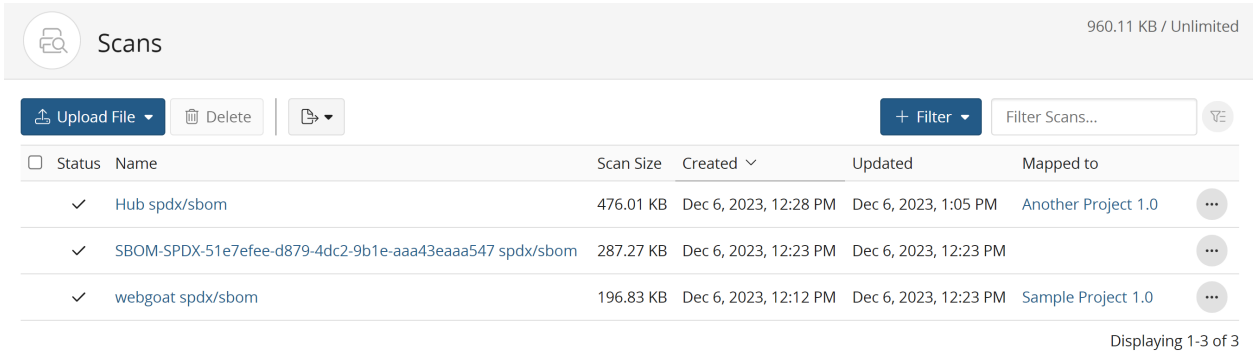
Uploading a scan file in Black Duck SCA

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

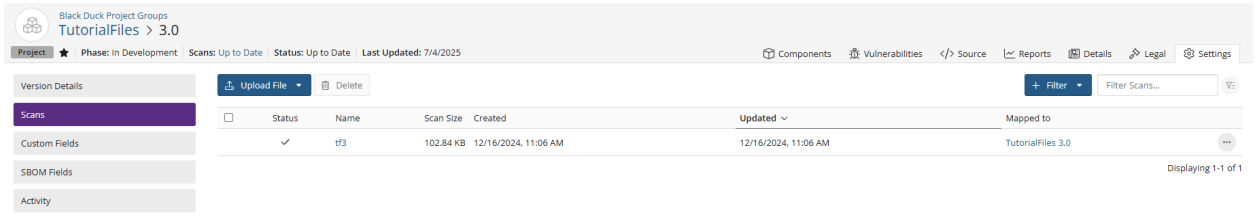
To upload a file:

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

.



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



3. Click **Upload File** and select a file format:
   - BDIO Scan: Supported file types: .json, .bdio, .bdmu.
   - SBOM-SPDX: Supported file types and formats: .json, .yaml, .rdf, .spdx
   - SBOM-CycloneDX: Supported file types and formats: .json
4. Upload the desired file(s) in **Upload *file format*** dialog box:
   - Click **Browse Computer...** or anywhere inside the dotted line box and navigate to the desired report file.
   - Drag the report file into the dialog box.

   You can multiple files by repeating the step above. The selected report files will be listed as Queued, ready to be uploaded.

   You can remove unwanted report files from the list by clicking the ⊗.
5. Optionally, you can enable the **Unmatched Component Auto-Creation** checkbox to automatically create custom components from SBOM unmatched origin IDs.
6. Click **Scan** in the Upload dialog box after uploading the file.

   The **Upload *file format*** dialog box will remain open after the scan(s) have completed, should you want to add additional report files to scan.

7. Click **Close** to dismiss the **Upload *file format*** dialog box.

Warning: When uploading a BDIO or SBOM file, it will override an existing BDIO or SBOM file if they share the same name. This will update the BOM of the project version to which it is mapped.

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, if the scan is unmapped, use Black Duck to map the file to a project.

Error code references

If the scan upload fails, it will display an error code and a message.

Table 1: External communication errors

Error code	Message
ERR01_1001	ScanCLI REST communication error
ERR01_1002	Host name could not be resolved
ERR01_1003	ScanCLI Host connection error

Table 2: Internal communication errors

Error code	Message
ERR02_1001	Unknown connection error
ERR02_1002	Scan processing timed out

Table 3: Resource allocation errors

Error code	Message
ERR03_1001	Failed to establish the validity of the server's certificate
ERR03_1002	Failed to establish a secure connection to the server
ERR03_1003	Failed to verify the server certificate for host
ERR03_1004	Unable to secure the connection to the host
ERR03_1005	File not found

Table 4: Registration errors

Error code	Message
ERR04_1001	Registration is invalid

Table 5: Internal error codes

Error code	Message
ERR05_1001	Unable to update scan status
ERR05_1002	Unable to persist results to scan database

Error code	Message
ERR05_1017	Unrecognized error
ERR05_1018	Failed to run the scan
ERR05_1019	An error occurred while ingesting chunk
ERR05_1020	BDIO archive upload processing failed
ERR05_1021	Unable to process KnowledgeBase component matching
ERR05_1022	Unable to process KnowledgeBase signature matching
ERR05_1023	Unable to process KnowledgeBase signature polling request
ERR05_1024	kbMatchData is required for IP scans
ERR05_1025	Scan processing failed in Match Engine
ERR05_1026	Knowledge Base lookup failed in Match Engine
ERR05_1027	Signature lookup failed in Match Engine
ERR05_1028	Failed to create or update code location
ERR05_1029	Scan match creation failed
ERR05_1030	Failed to run scan auto BOM job
ERR05_1031	Snippet scan auto BOM job failed
ERR05_1032	Failed to run BOM job
ERR05_1033	Failed to transfer BDIO data
ERR05_1034	An error occurred while ingesting chunk
ERR05_1035	FS scan waiting period was surpassed
ERR05_1036	Exception thrown when trying to poll match results
ERR05_1037	Scan data was not present in the database
ERR05_1038	Error processing KB request message
ERR05_1039	Error processing KB request message
ERR05_1040	Failed saving document data for document
ERR05_1041	Failed to complete post work for scan
ERR05_1042	Scan failed while adding chunk data

Table 6: External error codes

Error code	Message
ERR06_1003	Cannot create output directory
ERR06_1004	Username missing
ERR06_1005	Authorization failure
ERR06_1006	Bad exclude file

Error code	Message
ERR06_1007	Provided bdio archive has wrong format
ERR06_1008	Failed exclude file
ERR06_1009	Failed global exclude file
ERR06_1010	Invalid key store
ERR06_1011	Failed key store
ERR06_1012	Certificate password missing
ERR06_1013	Invalid certificate password
ERR06_1014	Failed key pair
ERR06_1015	Invalid dry run file
ERR06_1016	Server scan was not found

Table 7: Stopped by user codes

Error code	Message
ERR06_1002	Scan Stopped by user

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.

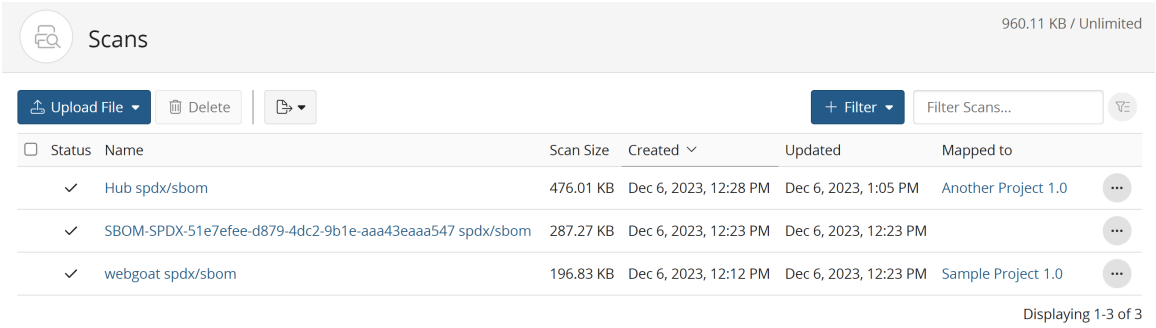
To browse component scans:

1. Log in to Black Duck SCA.
2. Click 

.

Scans page

The Scans page displays a list of all scans available in Black Duck. These can be created through Detect (as BDIO files) or by importing SBOM files.



The Scans page is composed of the following elements:

- The header bar contains the options to upload new scans, delete scans, export the scan list to CSV, and filter the scan list.

- The table contains the list of scans available in Black Duck:
  - The **Status** column displays whether or not the scan was a success with a ✔ or a failure with a ⚠.
  - The **Name** column displays the scan's name. If the scan or upload failed, an error will be displayed in this column under the name of the scan.
  - The **Scan Size** column displays the file size of the scan.
  - The **Created** column displays when the scan was added to Black Duck. Note that this timestamp may not necessarily reflect when the scan was created. To find the creation date of the scan, click the scan and see the **Created On** timestamp in the **Scan Details** section.
  - The **Updated** column displays the date when the scan was last modified.
  - The **Mapped To** column displays the project name to which this scan is mapped.
  - Additional options by clicking ••• for the desired scan:
    - **Map to Project**
    - **Download Scan Archive**
    - **Download Scan CSV Data**
    - **Delete**

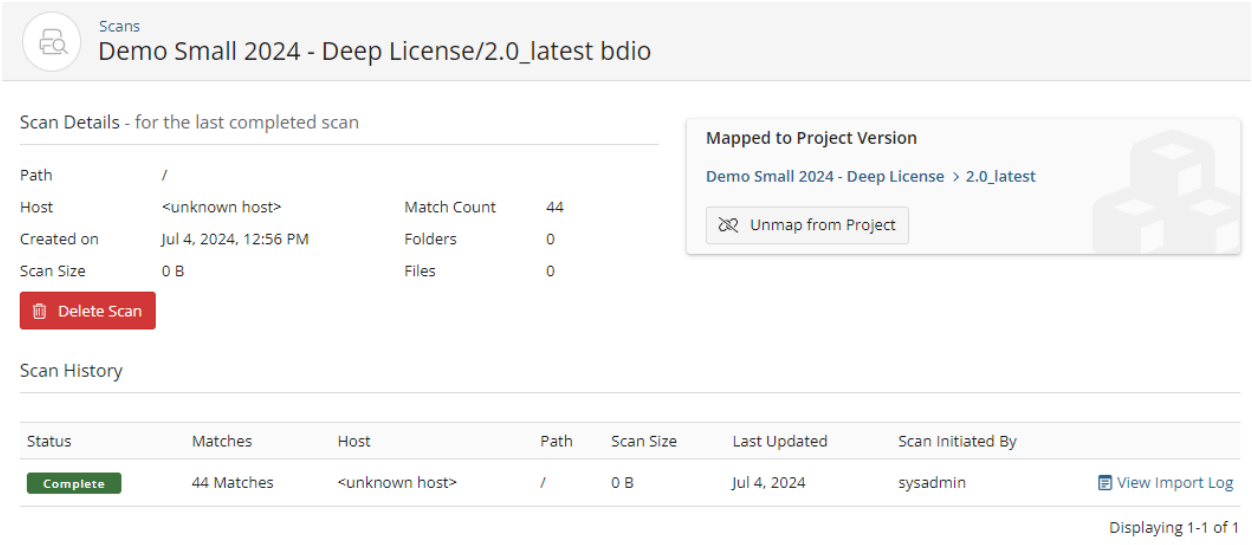
Exporting to CSV

You can export your search results to CSV which converts the individual rows to tabular data. To do so, click the button and select CSV.

Scan name page

To view the details of a particular scan:

1. Click the name of the scan in the **Name** column to open the *Scan Name* page.



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

- Path: Path to the code.

- Host: Name of the machine where the latest scan was performed.
- Created on: When the scan was created. This is the specific timestamp when the scan was completed. This may not necessarily reflect the time displayed in the Scan table.
- Scan Size: File size of the scan.
- Match count: The total number of folders and files matched.
- Folders: Number of folders found in the scanned code.
- Files: Number of files in the scanned code.

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:

- State of a scan. Possible values are:
  - **PROCESSING**: Scanning is in progress. This is a running state. A reason will also be added to further explain the current state. These include:
    - **TOOL_SUBMISSION**
    - **USER_UPLOAD**
    - **SCAN_INGESTED**
  - **COMPLETE**: The scanner has completed the scan successfully. This is a terminal state. It will be accompanied with a transition reason explaining further how the scan was successfully completed. These include:
    - **COMPLETE**: The scan and matching process is complete and that BOM computation may proceed. Note that this status also appears if Black Duck has determined that the scan was a duplicate.
    - **CLONED**: Black Duck is cloning the project version.
    - **SKIPPED**: The scan has been skipped.
  - **ERROR**: The scanner was not able to complete the scan successfully. This is a terminal state, meaning that it will be accompanied with a transition reason explaining further how the scan failed. These include:
    - **CANCELLED**: A user cancelled the scan before it was completed.
    - **ERROR_TOOL**: "Scan Submitted and Errored". The scan was submitted but an error or timeout occurred in the tool that submitted the scan and the tool is failing the scan.
    - **ERROR_SCANNING**: "Scan Error". The scan could not be completed by scanner.
    - **ERROR_SAVING_SCAN_DATA**: "Saving Scan Data Error". An error occurred when attempting to save scan data.
    - **ERROR_MATCHING**: "Matching Error". An error occurred during the matching process.
    - **ERROR_BUILDING_BOM**: "Building BOM Error". An error occurred when attempting to build the BOM. This is for migration and backward compatibility only.
    - **ERROR**: A schema error has occurred.
- 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.
- View Import Log: The import log is a collection of audit records that detail information on KB component matching successes and failures for external namespaces and identifiers. An example of use would be to help identify what components were "not found" during the scan and subsequently not added to the BOM report as it may not be immediately obvious from looking at the BOM.

  This only applies for the following scan types:

  - Package manager scans
  - Binary analysis scans
  - Docker inspector scans
  - Protex BOM import scans

  Signature/snippet scans do not have this functionality and that is intended. The Source tab should be used when reviewing signature/snippet matches.

Viewing an audit log for a BOM file

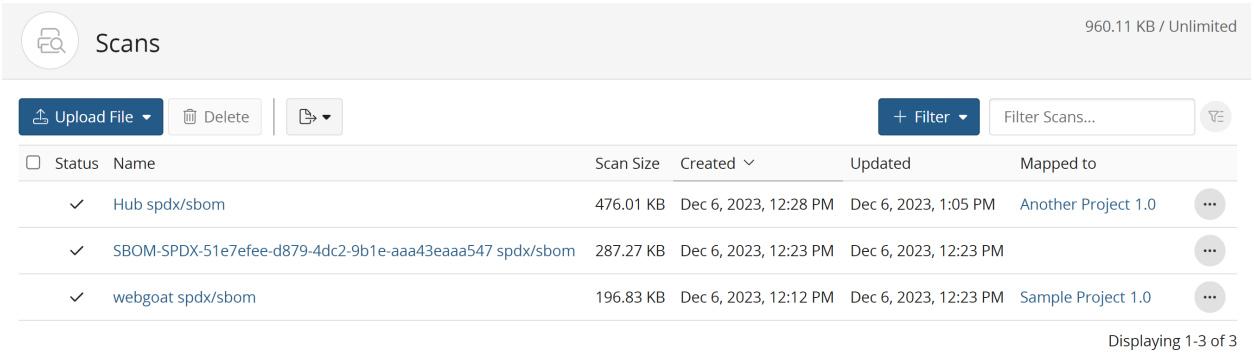
Use the View Import Log to view your results of importing a BOM file. This log lists the components and licenses that were mapped to Black Duck. It also provides details for any items that were unable to be mapped.

To view an audit log:

1. Log in to Black Duck SCA.
2. Click 

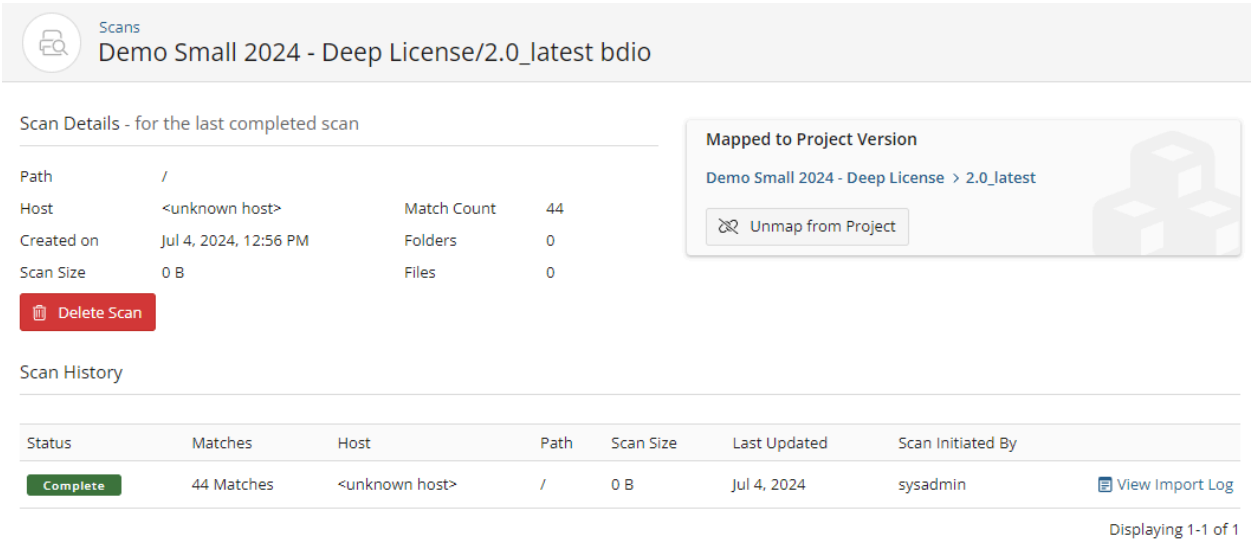
.

   The Scans page appears.



3. Select the scan you wish to view.

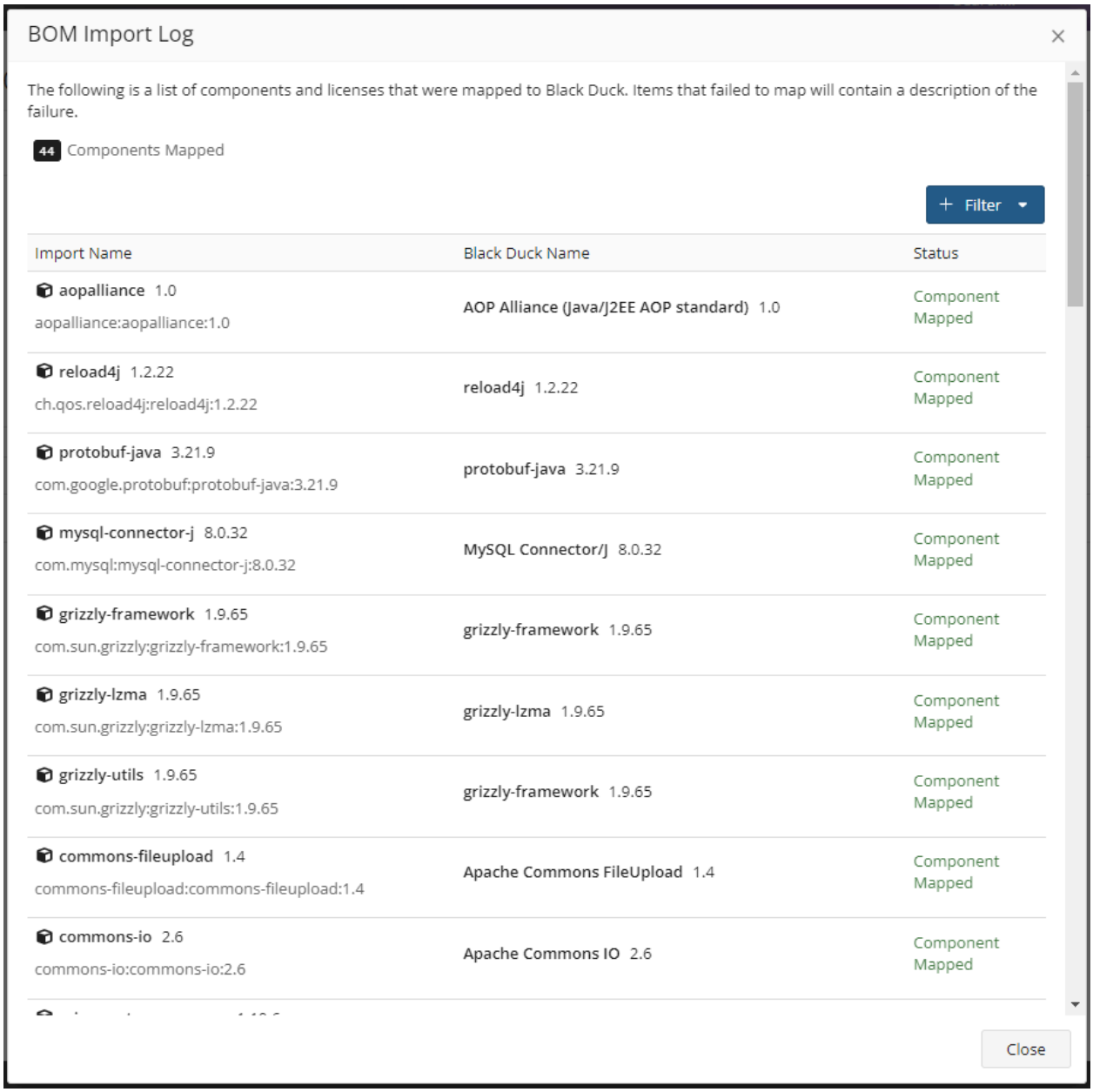
   The *Scan Name* page appears. The host, path, and scan details (status of the scan, date, or time (if the date is today) the scan completed, and the username of the user that ran the scan) appear at the top of the page.



4. Select **View Import Log** to view the log. If this code location has been scanned multiple times, only the latest BOM Import Log will be available to view.

   The BOM Import Log displays, if applicable, the number of components mapped and components not found above the table.

   Note: The BOM Import Log will not display licenses missing, mapped, or not found for scans imported after 2024.7.0. However, they will continue to be displayed for scans imported prior to 2024.7.0.



Component/License Import Events & Statuses

The Status column will display whether or not the component or license was successfully imported. If a component or license was not successfully imported, an error message will explain the reason for the failure.

Tip: The Import BOM Log table can also be filtered by these statuses.

The possible statuses are as follows:

- **Component mapped**. This status indicates that the component version was successfully mapped to the Black Duck KnowledgeBase.

- **Component not found**. This status indicates that the scanned component version was not successfully mapped to the Black Duck project version because no mapping is present for the given external identifier.
- **License mapped**. This status indicates that the license was successfully mapped to the Black Duck KnowledgeBase.
- **License not found**. This status indicates that the scanned license was not successfully mapped because no mapping is present for the given external identifier in the Black Duck KnowledgeBase .
- **License missing**. This status indicates that the scanned license was not found.

Using the `GET/api/bom-import/<graphId>/component-import-events` API request

When using the `GET /api/bom-import/<graphId>/component-import-events` API request, the statuses will be displayed in the following responses. If the import was unsuccessful, the `failureReason` value will provide more information. See below for response examples.

A successful component import:

```
    {
      "event": "COMPONENT_MAPPING_SUCCEEDED",
      "importComponentName": "imported component name",
      "importComponentVersionName": "imported component version",
      "componentName": "full component name",
      "componentVersionName": "full component version",
      "externalId": "external identification"
    },
```

An unsuccessful component import:

```
    {
      "event": "COMPONENT_MAPPING_FAILED",
      "importComponentName": "imported component name",
      "importComponentVersionName": "imported component version",
      "externalId": "external identification",
      "failureReason": "Unable to map scanned component version to Black Duck project
 version because no mapping is present for the given external identifier"
    }
```

A successful license import:

```
    {
        "event" : "LICENSE_MAPPING_SUCCEEDED",
        "protexLicenseName" : "license name",
        "externalId" : "external identification",
    }
```

An unsuccessful license import may fail in one of the following reasons:

- Unable to map scanned license to Black Duck license because no external identifier is present
- Unable to map scanned license to Black Duck license because no mapping is present for the given external identifier
- Unsupported license mapping

Regardless of the failure, the API response will be formed as follows:

```
    {
      "event": "LICENSE_MAPPING_FAILED",
      "protexLicenseId": "license ID",
      "protexLicenseName": "license name",
      "externalId": "external identification",
      "failureReason": "see reasons above"
    }
```

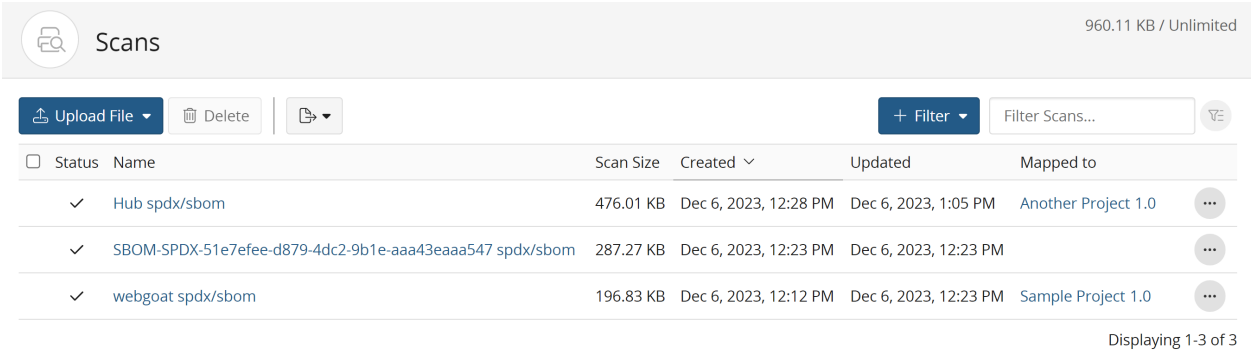
Mapping a scan to a project

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

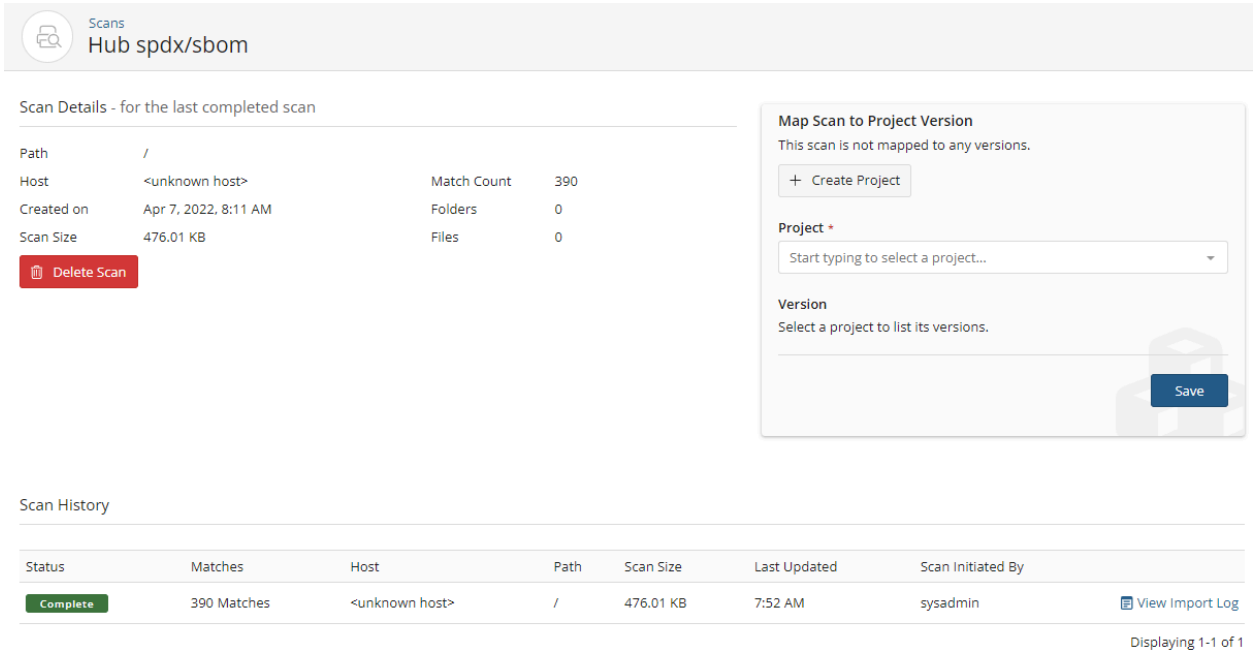
Note: You can scan a Docker image or file directory location or archive more than once, but you only have to map it to a project version once. The host and path may be changed, but as long as code location name is the same, Black Duck automatically updates the BOM of the project with any new information discovered during subsequent scans.

To map a scan to a project:

1. Log in to Black Duck SCA.
2. Click Scans.



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.



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**.

   Black Duck displays the name and version of the project to which you mapped the component scan. Select the link to open the BOM page.

   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.

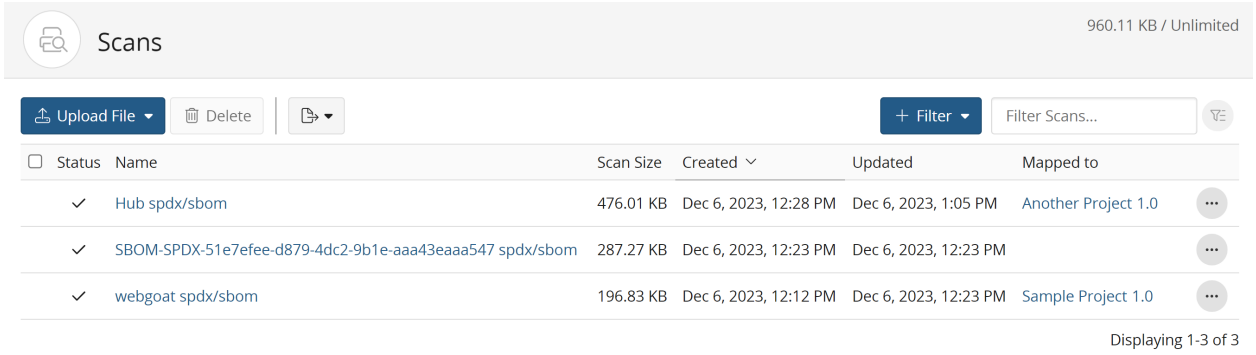
Removing a scan from a project

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

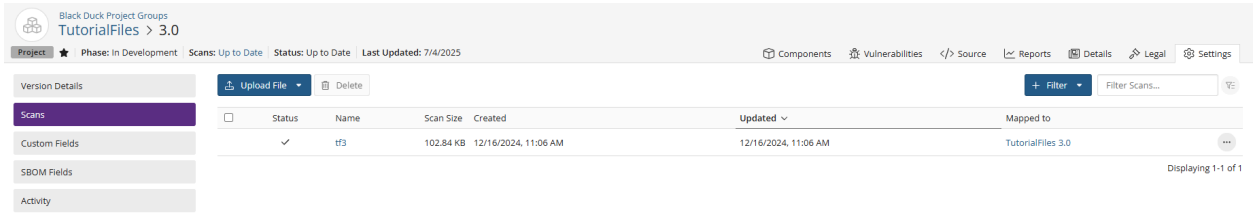
To remove a mapping:

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

   - Click Scans.



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



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

Un-mapping and re-mapping scan files

Please be aware that remapping scan files to a project does not initiate the matching process. After you remap the files, the BOM computation will continue to utilize the Knowledge Base (KB) IDs that were identified during the initial matching phase. This means that any changes made during the remapping will not affect the existing mappings or the resulting BOM until the matching process is run again.

Deleting a scan

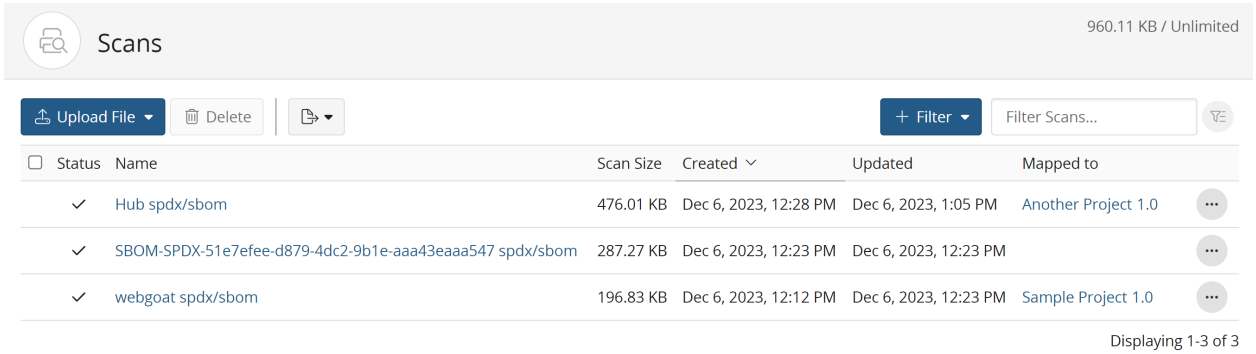
If you have scanned an incorrect path or Docker image, no longer require the scan, or want to free up space, you can delete the scan(s).

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

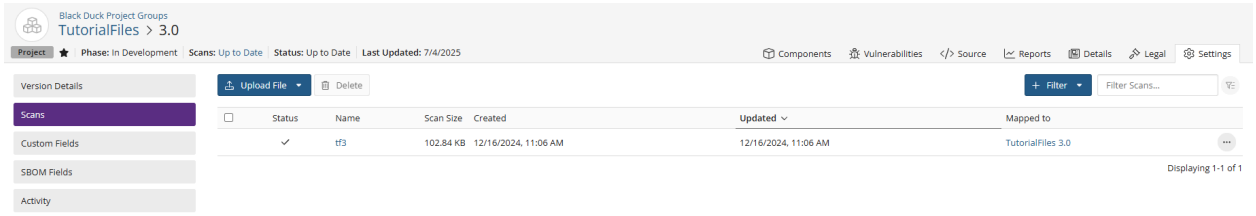
To delete a scan:

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

.



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



3. Select the scan(s) you want to delete by using the checkbox(es) and click **Delete**.

   You can also click and select **Delete** in the row of the scan that you want to delete.
4. In the Delete Scan dialog box, confirm that you have selected the correct scan(s), and click **Delete**.

Black Duck removes the scan.

Downloading 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.

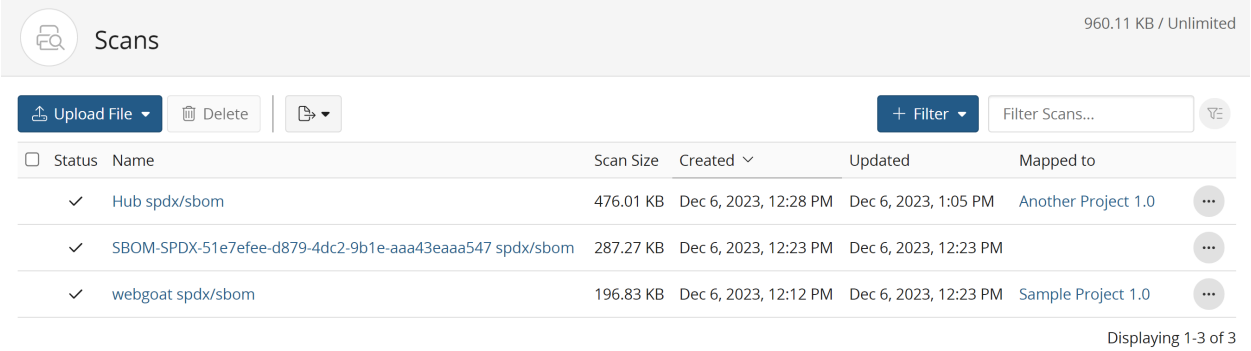
Note: This feature is not available if you initially scanned using Black Duck version 5.x or earlier. If the option does not appear, delete the code location and re-scan.

Downloading Scan Archive

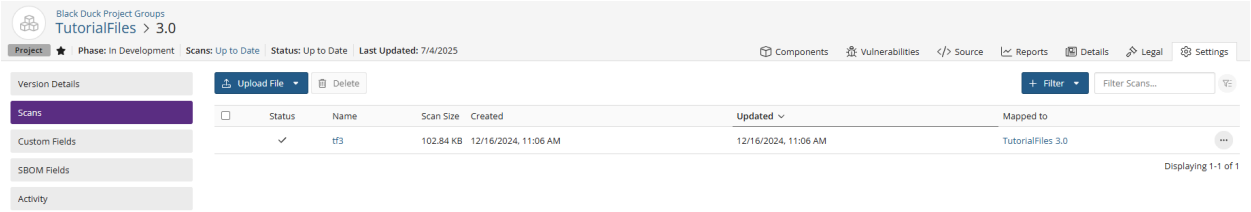
To download a scan archive:

1. Log in to Black Duck SCA.

2. Do one of the following:
   - For unmapped scans, click Scans.



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



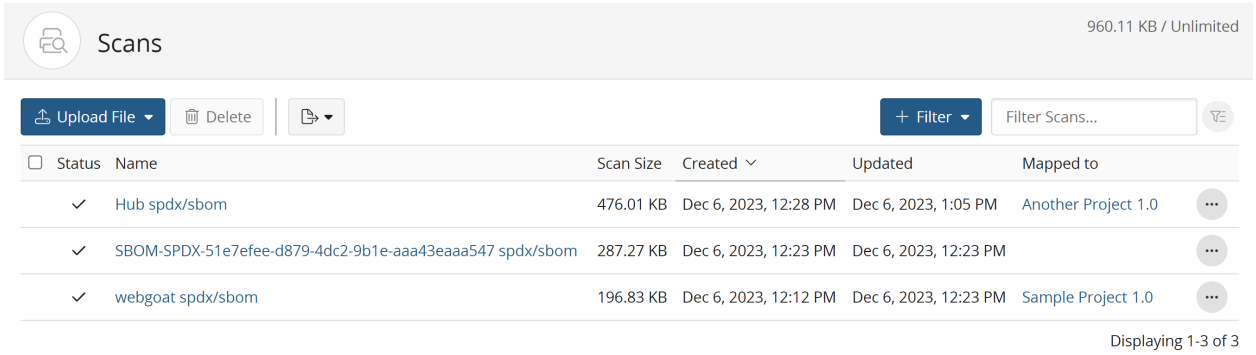
3. Click ••• and select **Download Scan Archive** in the row of the scan that you want to obtain a scan file.

   The file is downloaded with a `.bdio` extension and is a compressed zip file. It contains the original scan data, without any modifications made after the initial scan.

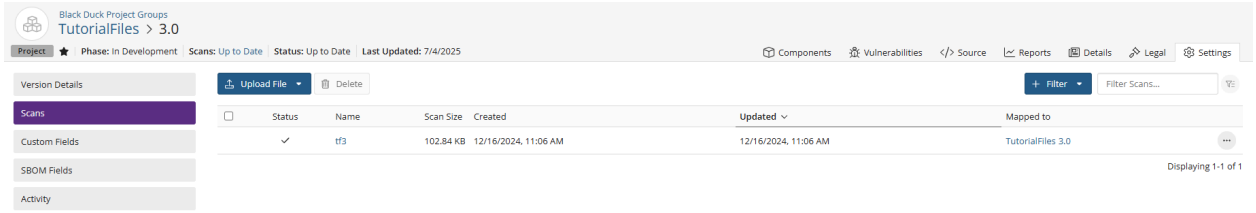
Downloading Scan CSV Data

In order to download scan CSV data, the original scan must have been performed with the `--upload-csv` scan CLI option. To download a scan CSV data:

1. Log in to Black Duck SCA.
2. Do one of the following:
   - For unmapped scans, click Scans.



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



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

   The file is downloaded with a `.csv` extension and is a compressed zip file. It contains the original scan data, without any modifications made after the initial scan.

Understanding component matching in Black Duck SCA

Black Duck SCA uses sophisticated matching techniques to identify components in your codebase. As customers increasingly use package manager-produced SBOMs and other scanning methods, understanding how Black Duck identifies and matches components becomes essential. This document explains our fuzzy matching and match correlation capabilities, providing clarity on how components are identified during package manager and SBOM import scans.

Component Identification Process

Black Duck SCA employs a multi-layered approach to component identification:

1. **Exact Matching**: First, we attempt to find an exact match for components based on their package URL (PURL) or other identifiers.
2. **Fuzzy Matching**: When exact matches aren't found, we employ fuzzy matching techniques to identify similar components.
3. **Match Correlation**: For scans using multiple techniques, we use match correlation to consolidate and improve results.

Package URL (PURL) Matching Elements

Package URLs (PURLs) contain several elements that Black Duck uses for matching. Below is a table showing which elements we consider mandatory for accurate matching:

PURL Element	Required for Matching	Notes
Type	✓	Package manager type (npm, maven, etc.)
Namespace	✓	Organization or grouping
Name	✓	Component name
Version	✓	Component version
Architecture	✗	Optional but considered when available
Epoch	✗	Optional but considered when available
Qualifiers	✗	Optional but considered when available
Subpaths	✗	Optional but considered when available

For detailed PURL specifications, refer to the official PURL specification.

Fuzzy Matching Process

When exact matches aren't found during package manager or SBOM import scans, Black Duck employs fuzzy matching through the following process:

1. **Component Name and Version Analysis**: We analyze variations of component names and versions.
2. **Namespace Flexibility**: We consider potential variations in namespaces.
3. **Version Normalization**: Version strings are normalized to account for different versioning formats.
4. **Knowledge Base Comparison**: The processed information is compared against our Knowledge Base to find the closest match.

For example, if an exact match for "org.example:library:1.0.0" isn't found, our fuzzy matching might identify "org.example:library:1.0" or "org.example.library:1.0.0" as potential matches.

Match Correlation vs. Fuzzy Matching

It's important to understand the difference between these two techniques:

- **Fuzzy Matching**: Applies to a single scan type, looking for approximate matches when exact matches aren't found.
- **Match Correlation**: Combines results from multiple scan techniques (e.g., signature scan and package manager scan) to provide more accurate identification.

To enable match correlation when using Detect, use the `--detect.blackduck.correlated.scanning.enabled=true` option.

Architectural Components Involved

The matching process involves several architectural components:

- **Knowledge Base (KB)**: Contains the comprehensive component information used for matching.

- **Match as a Service (MaaS)**: Handles the complex matching algorithms.
- **SCA Scan Service (SCASS)**: Coordinates scanning and initial processing.

Note: Fuzzy matching and match correlation features are not currently supported for air-gapped (KB on-premise) installations.

Ongoing Improvements

Component matching is an area of active development at Black Duck. We continuously refine our algorithms and processes to improve match accuracy. We welcome customer feedback on matching results.

If you have questions about specific fuzzy or correlated matches, please submit a support case through the standard process. Your feedback helps us improve our matching capabilities.

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.

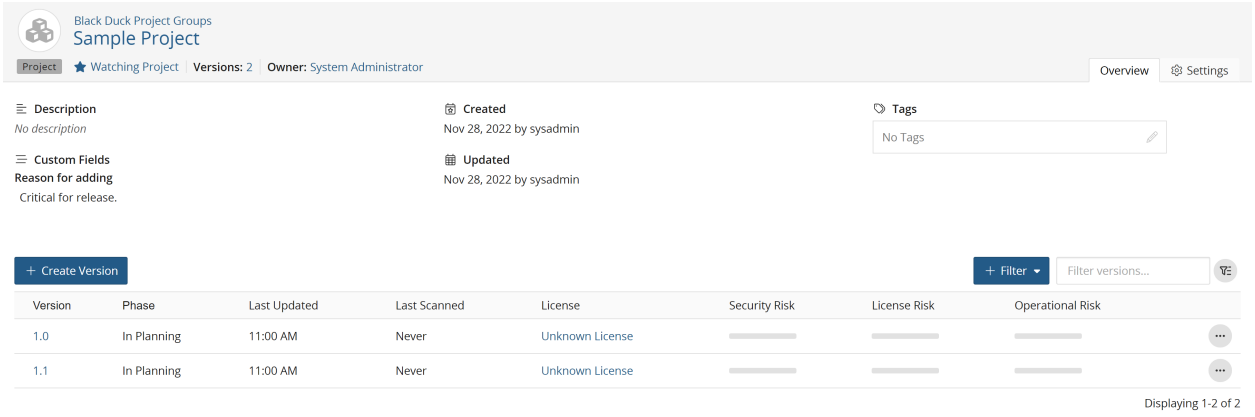
At the project level, team members can:

- 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.



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

Column	Description
N/A	Icons shown to the left of the version name:

Column	Description
	• Policy violation. Select the icon to view information on the policy violation. • Policy violation has been overridden. Select the icon to view information on the policy violation.
Version	Name of the project version.
Phase	The development phase of this version. The possible values are: • In Planning • In Development • Pre-release • Released • Deprecated • Archived 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.
Last Updated	When this project version was last updated. Hover over the value to see: • 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	Bars show the critical, high, medium, and low security risk levels for the OSS components in this version of the project. Select the bar to view the number of affected components.
License Risk	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. Select the bar to view the number of affected components.
Operational Risk	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. Select the bar to view the number of affected components.

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

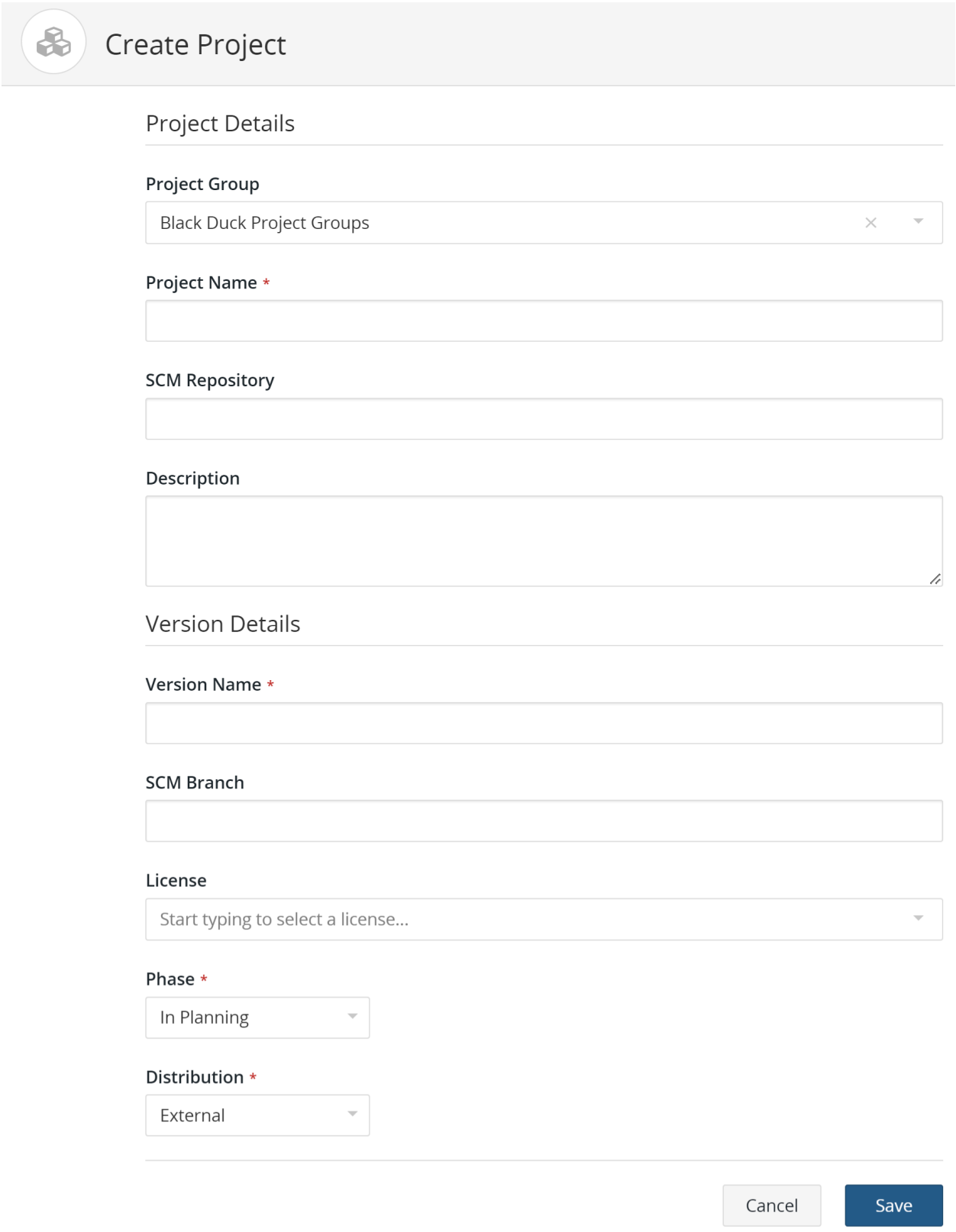
A project is the base unit in Black Duck. A project can be both a stand-alone development project and part of another project. For example, Apache Tomcat is a project in its own right but it may also be part of other, larger projects. You must create the projects that you want to make available for search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

Projects or applications are limited to 10GB of Managed Code base.

Note: If SCM Integration is enabled in your environment and you want to create a SCM project, see Creating a SCM project.

To create a project:

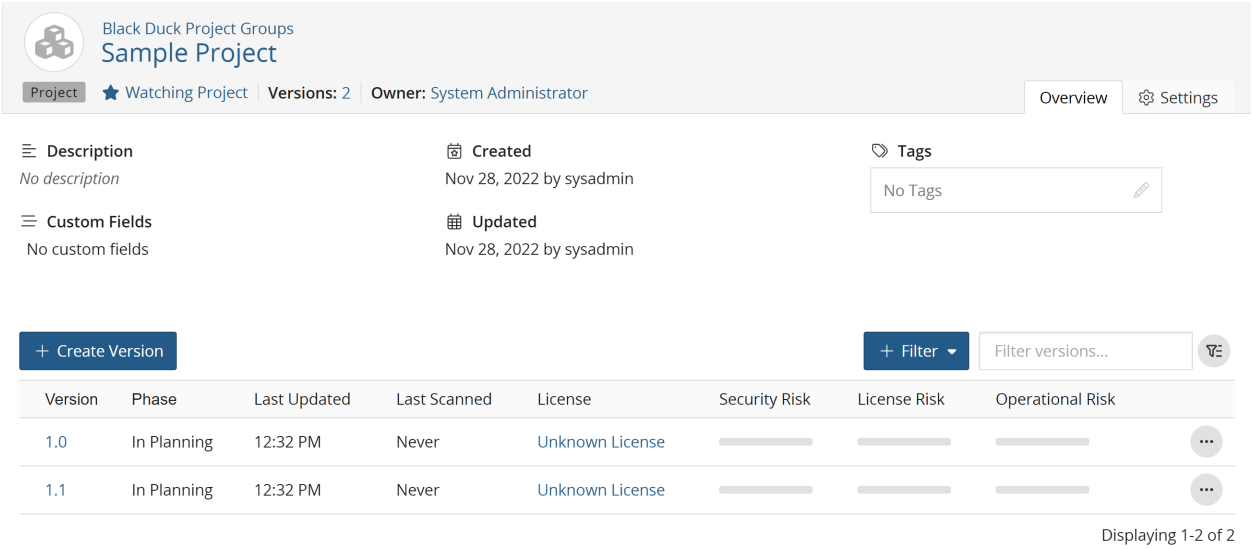
1. Log in to Black Duck SCA.
2. Click **+ Create Project** at the top of any page. If SCM Integration is enabled in your environment, select **Standard Project** from the menu. The **Project Details** page will display.



3. 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 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, enter additional information such as:
   - **SCM Repository**: The URL of the source code management (SCM) repository where your code resides. This field is visible only if this feature is enabled in your environment. It can be manually edited or automatically populated by Detect after completing a package manager scan. Manually changing the SCM repository URL could break an existing scan if the URLs don't match. Note that this feature is available with Detect 8.x or later.
   - **Description**: 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.
5. Type the version for this project in the **Version Name** field.
6. Click **Save**.
   Black Duck displays the *Project Name* page.



Creating a SCM project

If you have SCM integration enabled, you can create projects based from the repositories found in your SCM providers. The process to create a project adds an additional step to select the SCM repository from where the project originates. The servers displayed will depend on what SCM providers your organization has configured for use.

Important: Users with only the **Global Project Viewer** role cannot view SCM projects. To create, view, and manage SCM projects, users must have either the **Lite Global Project Manager** role or the **Integration Manager** role.

Creating projects from a SCM Provider

To create a SCM project:

1. Log into Black Duck SCA as a Global Code Scanner user.
2. Click **+ Create Project** at the top of any page.
3. Select **SCM Project** from the menu.
4. Select the SCM provider that applies to your project. You must be authenticated to use the SCM provider.
5. Select any number of repositories from the **Repository** list presented. Repositories maked with the Mapped tag have already been
6. Click the **Create and Scan** button.

A project will be created for each repository scanned and the default branch will be used as the project version. You will be given the opportunity to scan other branches afterwards. The result of the scan creates a read-only bill of materials (BOM) which is lighter than the usual BOM.

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

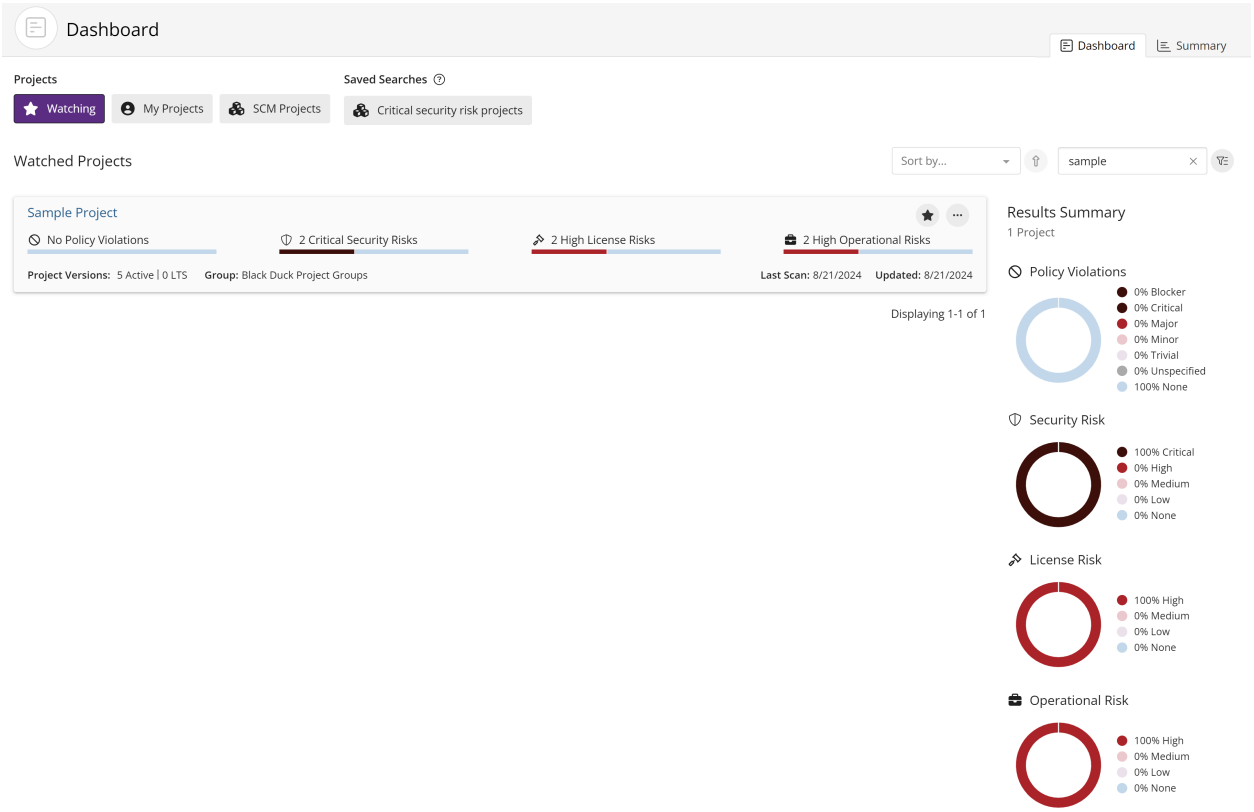
The **Watching** dashboard lists your watched projects. The list of your watched projects also appears on the My Profile page.

To view a list of your projects using the Watching dashboard:

1. Click 

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

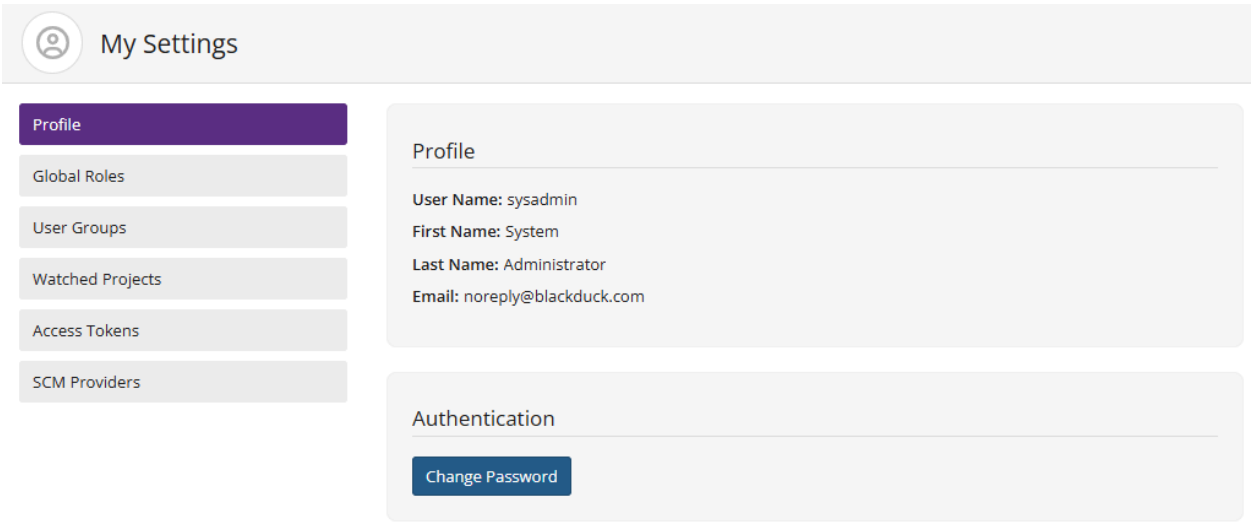
The dashboard of your watched projects appears.



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

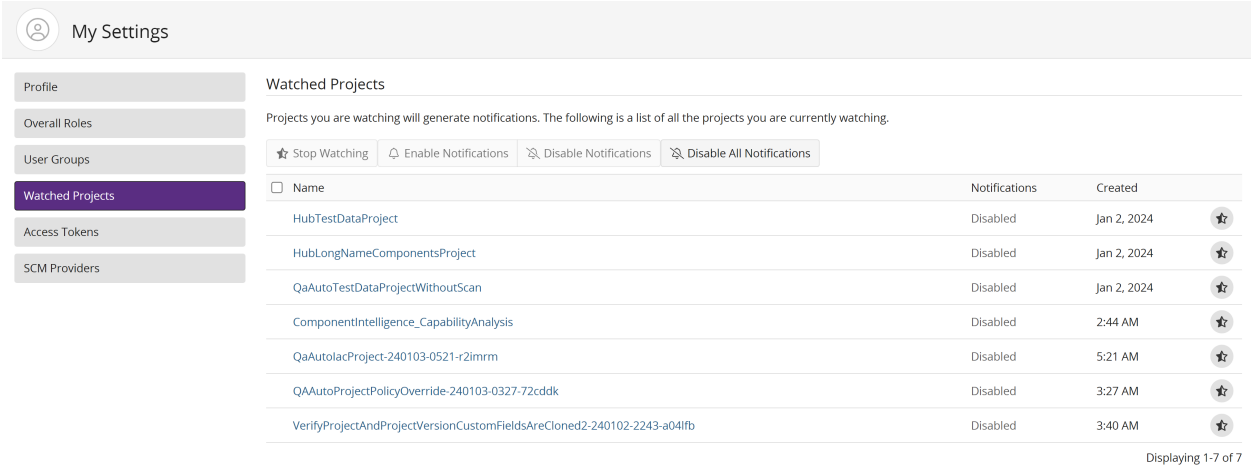
1. From the user menu located on the top navigation bar, select **My Profile**.

   The My Profile page appears.



2. Select **Watched Projects**.

   The table with your watched projects appears.



The table lists each project name; select the project name to view the *Project Name* **Overview** tab.

The **Notifications** column displays whether or not you will receive notifications for this project. You can enable or disable notifications for any project listed in this table by checking the checkbox next to the desired project or the topmost checkbox to select all projects, and then clicking either the **Enable Notifications** or **Disable Notifications**. You can also click the **Disable All Notifications** to stop receiving notifications from all projects in the list.

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.



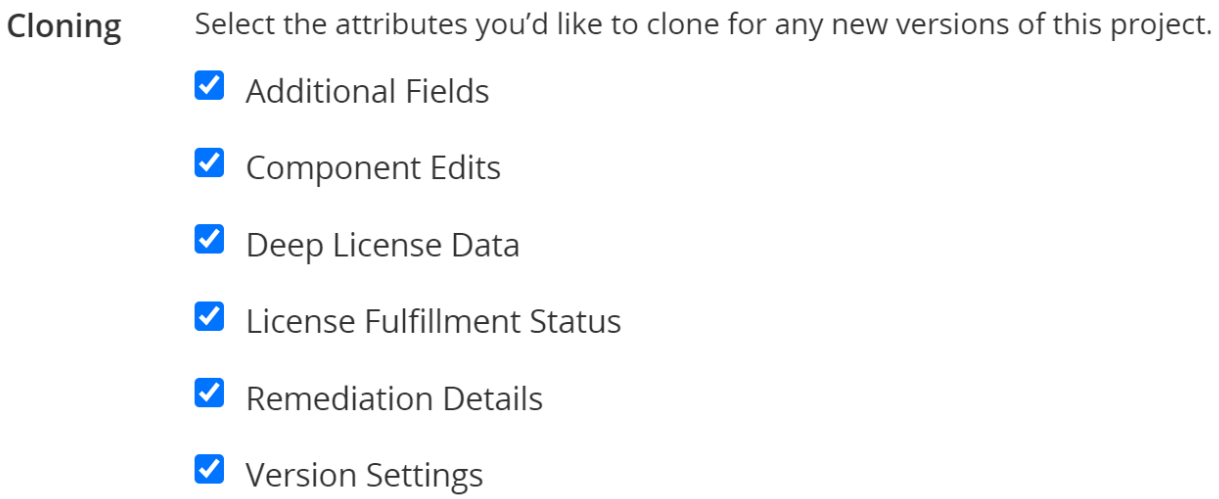
The heading now indicates you are watching the project:



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:



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.

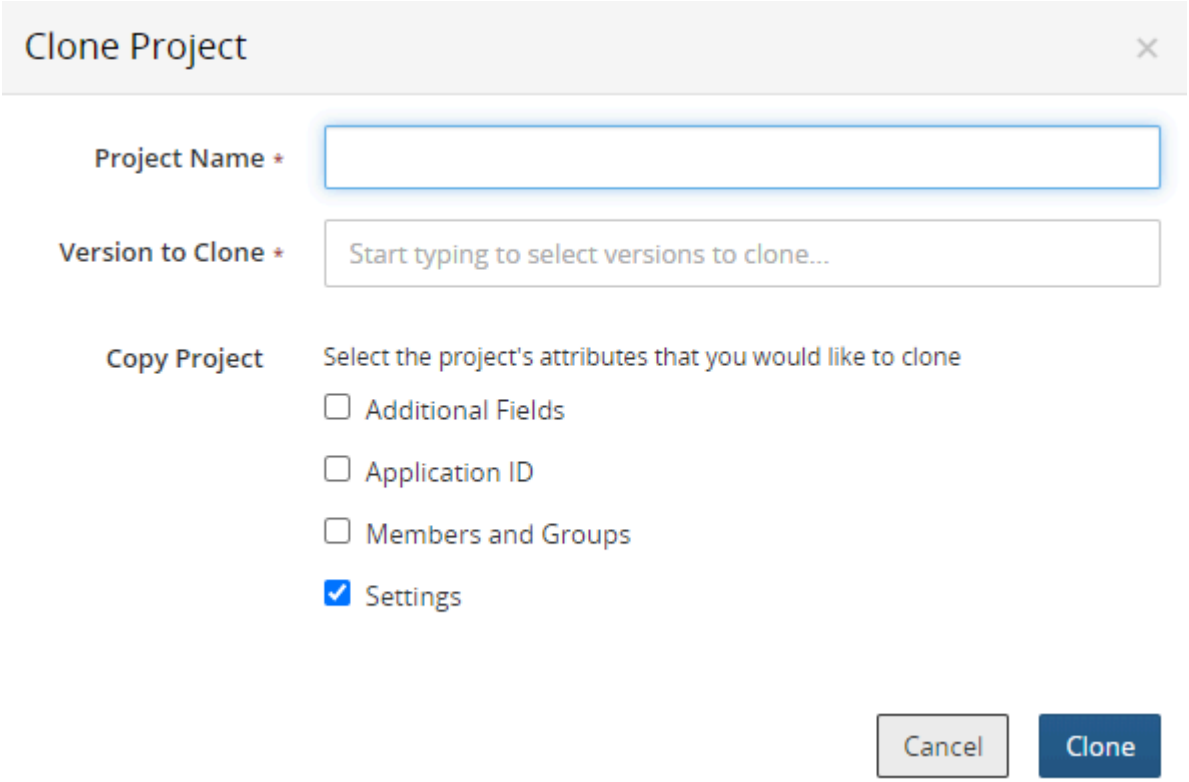
Cloned information appears in the cloned project versions for components that are the *same* as in the original project version for this project. A scan will need to be performed to replicate the components of the base project to the cloned project. If a component is not included in the newly scanned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

Project team members can view and update project settings, such as:

- Project Details
- SCM Repository. This field is visible only if this feature is enabled in your environment.
- Users
- Groups
- Custom Fields
- SBOM Fields

- Activity

To configure a project's settings:

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

Updating project details

From the project **Overview** tab, you can perform the following tasks:

- Update the Project Name, Description, Owner, URL, or Tier.

  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.
- Configure the ability to apply edits to all versions of a project.
- Configure snippet adjustments.
- Update project version cloning settings.
- Configure custom scan signatures.
- Configure your project's unmatched files data retention.
- Set how deep license data applies to your BOMs or snippet component matches.
- Set an Application ID. This field is used to store an external mapping ID for the project to an external system, such as an asset management system or application catalog.
- Clone this project.
- Delete this project.
- Enable license conflicts to apply license conflicts data to your components

Setting a project's SCM Repository

Projects can be associated with a single SCM Repository. Once mapped, then each version can be associated with a branch in the repository. Select an SCM repository, or manually enter a repository URL.

To update the project's SCM repository, click and select from the following options:

- Select New Repository. Choose from the list of configured SCM providers and repositories.
- Manually Enter Repository. Enter the desired repository URL.
- Clear Repository. Remove the project's linked SCM repository.

Updating project members

Manage the users and groups associated to this project.

Updating a project's custom fields

If there are custom fields created for projects, you can provide the requested details here.

Updating a project's SBOM fields

These are additional fields that can be included in the SBOM report. These field values will propagate when this project is used as subproject, you can override them at the BOM level. See SBOM Project fields for more information.

Viewing a project's activity audit trail

The Activity Audit Trail retains the activity audit records of user actions and key events, such as project version, component, and vulnerability records, that affect a project or project version.

Note: Audit records are only available for activities that occur while audit tracking is enabled. If audit tracking is enabled after a project has already been created or modified, earlier activities are not backfilled and will not appear in the audit trail.

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

Adding users to a project team

To add users to the project team:

1. Log in to Black Duck SCA.
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 **Users** to view the list of members for this project.
4. Click **+ Add User**.
5. In the **Add User** dialog box, you can:
   - Type the name of the user that you want to add in the **Users** field. 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.
   - Click the **Users** field to see a list of available users.
6. Select the username to add this user to the project team. Optionally, you can add multiple users by typing or selecting the name of additional users.
7. Select the roles for this user for this project.
8. Click **Add**.
   The user(s) are added to the project team.

Removing users from a project team

To remove a member from the project team:

1. Log in to Black Duck SCA.
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 **Users** to view the list of members for this project.
4. Click in the row of the user you want to remove from the project team and select **Delete Direct Access**.
5. In the Remove Member dialog box, click **Delete**.
   The user is removed from the project team.

Adding a group to a 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 SCA.
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.
5. Click **+ Add User 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 SCA.
2. Click 

→ **Groups** to display the **Users & Groups** page.
3. Click the *name* of the group.
4. Click the **Projects** tab.
5. Click the **+ Add Project** button.
6. In the **Add Project** dialog box, you can:
   - Type the name of the project that you want to add in the **Projects** field. The list is type-ahead enabled, so you can see a list of available projects that contain the text you have typed and whether those projects are active.
   - Click the **Projects** field to see a list of available users.
7. Select the desired project(s). Optionally, you can add multiple projects by typing or selecting the name of additional projects.

8. Select the project roles for this group and click **Add**.

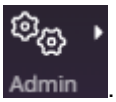
Removing a group from a project team

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 SCA.
2. Locate the project using the **Projects** tab on the Dashboard.
3. Select the name of the project to go to the *Project Name* page.
4. Select the **Settings** tab and then select **Groups** to view the list of groups for this project.
5. Click in the row of the group that you want to remove from the project team and select **Delete Direct Access**.
6. In the Remove Group dialog box, click **Delete** to confirm.

From the *Group Name* page:

1. Log in to Black Duck SCA.
2. Click 

.
3. Select **Groups** to display the **Users & Groups** page.
4. Select the name of the group you want to remove.
5. Click the **Projects** tab.
6. Click in the row of the group you want to remove and select **Delete Direct Access**.
7. Click **Delete** 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 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 and custom components:

- 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 SCA.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Type the tag in the Tags field 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 

> **Component Management**.

   The Component Management page appears.
3. Select the name of the custom component to go to the *Custom Component Name* page.
4. Type the tag in the Tags and press **Enter**.

   The tag is added to the custom component.

To edit a tag:

1. Click 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**.

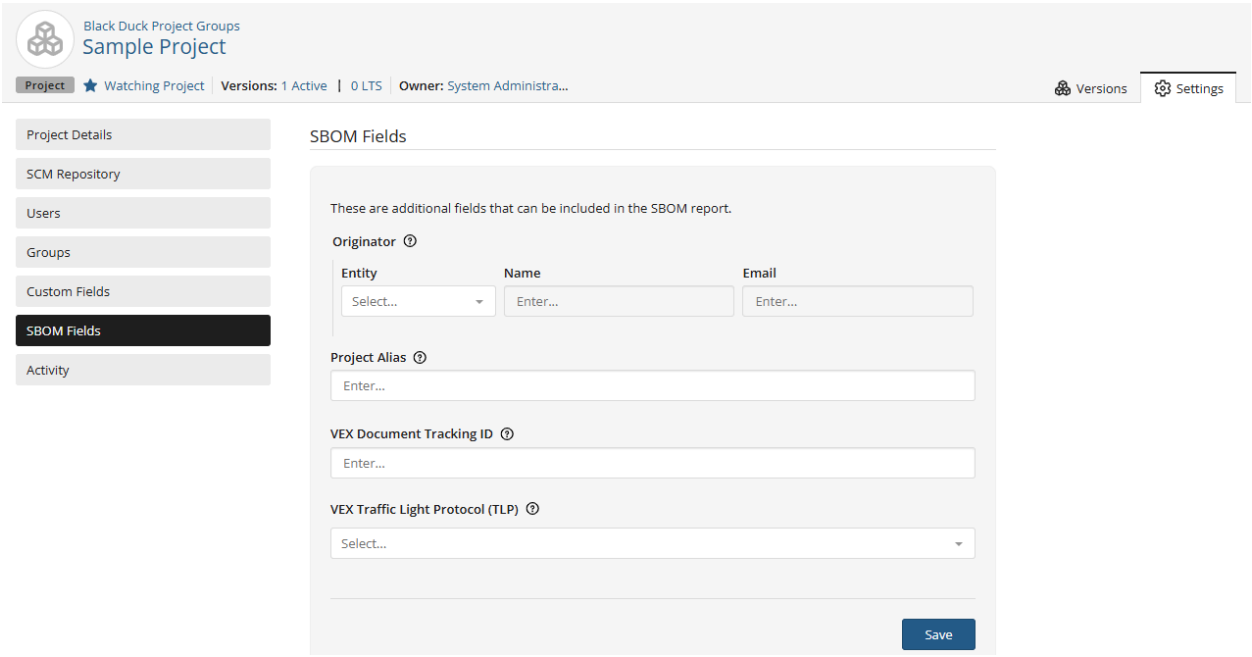
To remove a tag:

1. Click the **Tags** field.
2. Select **X** next to the tag.

Changing the project's SBOM alias

You can override project name and version info field at the project level.

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



5. Enter a new project name in the **Name** field.

Viewing a project's or project version's activity

The **Activity** tab displays the records of user actions and key events affecting this project or project version.

Note: Activity records are available only for actions and events that occur while audit tracking is enabled. If audit tracking is enabled after a project or project version has already been created or modified, earlier activities are not backfilled and will not appear in the Activity tab. This behavior applies to existing projects and project versions.

The events table

The Activity page contains an events table that lists activities recorded for the project or project version.

The table includes the following information:

Field	Description
Object	The object type and name associated with the activity, such as a project, component, vulnerability, source file, or user.
Event	The action or event that generated the activity record.
Cause	The entity that triggered the action, such as a user, policy, or scan.
Date and Time	The date and time when the event occurred.

Select an event to expand the row and view additional details about the activity.

Filtering the events table

Use filters to narrow the displayed activity records:

- Click**+ Filter**.
- Select one or more of the following filters:

Filter	Description
Cause Names	Filter events by the name of the user or entity that triggered the activity.
Date	Display events that occurred within a specified date range.
Events	Filter the table by event type.
Object Names	Display events related to a specific object name.
Object Types	Display events associated with a selected object type.

After applying filters, the events table updates to display only matching activity records.

About Project Versions

A project version represents a specific release or iteration of a project in Black Duck SCA. Each version is scanned and analyzed independently, allowing you to track the open source and third-party components used in a particular release of your software and manage the associated security, license, and operational risks.

Project versions are created within a parent project, and multiple versions of the same project can exist simultaneously. This allows you to manage risk across different stages of your software's lifecycle — for example, maintaining an active production release while continuing development on a new version.

Each project version maintains its own:

- **Bill of Materials (BOM)**. The complete list of open source and third-party components, AI models, and subprojects identified through scanning, along with their associated risk data.
- **Scan history.** A record of all scans mapped to this project version.
- **Risk profile.** Security, license, and operational risk data specific to the components in this version.
- **Settings and metadata.** Configuration options, lifecycle phase, distribution type, and other version-specific properties.

A project version can also be added as a subproject of another project version, allowing you to represent shared or reused components across projects.

Managing project versions

Depending on your role and permissions, you can perform the following actions:

- Creating a new version of a project
- Updating project version information
- Cloning project versions
- Deleting a project version

To view a project version:

1. Select the project name using the **Watching** or **My Projects** dashboard.
2. Select the desired version name.

Creating a new version of a project

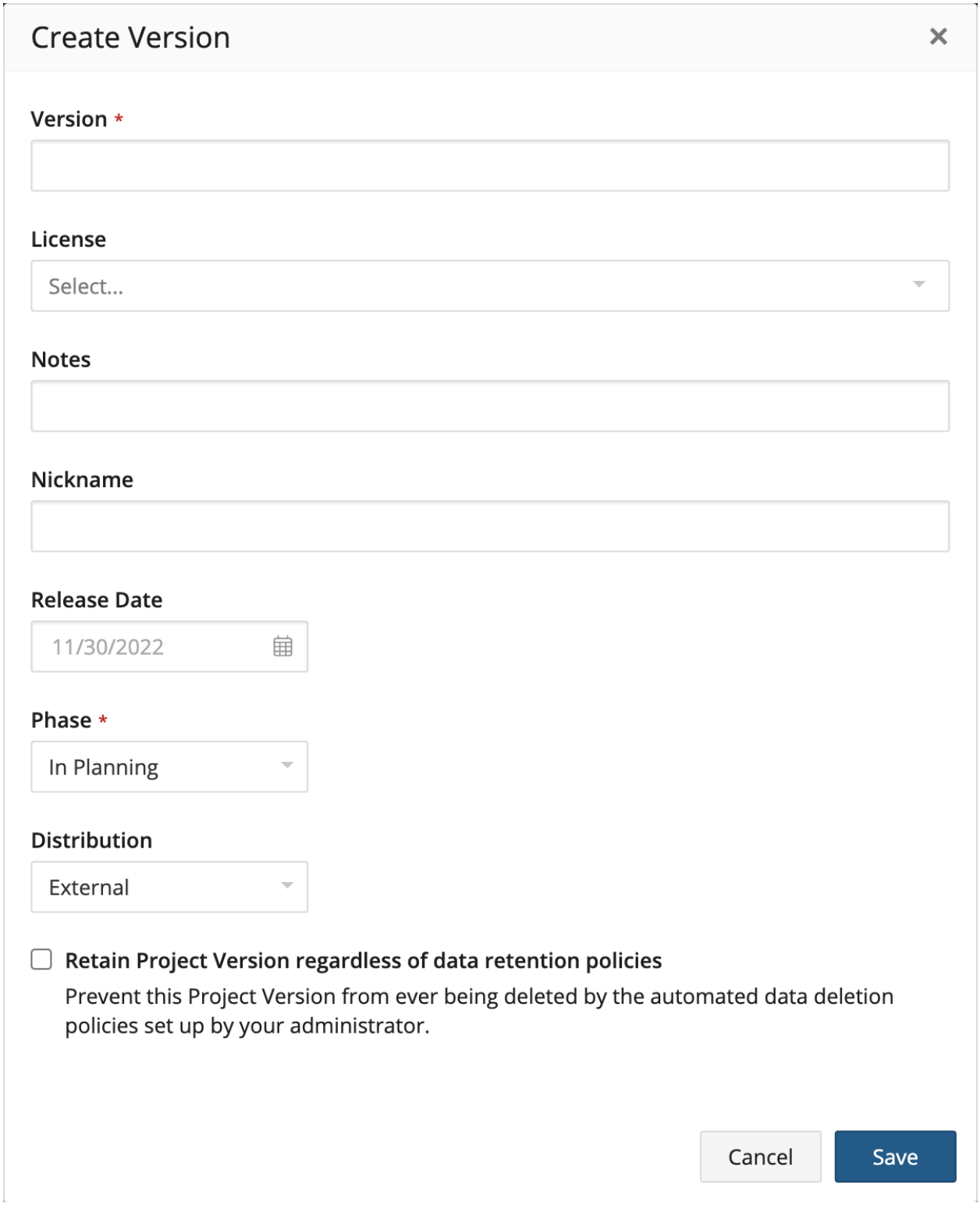
When you create a project, it has one version. You can create more project versions as needed.

To add a new project version:

1. Log in to Black Duck SCA.
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.



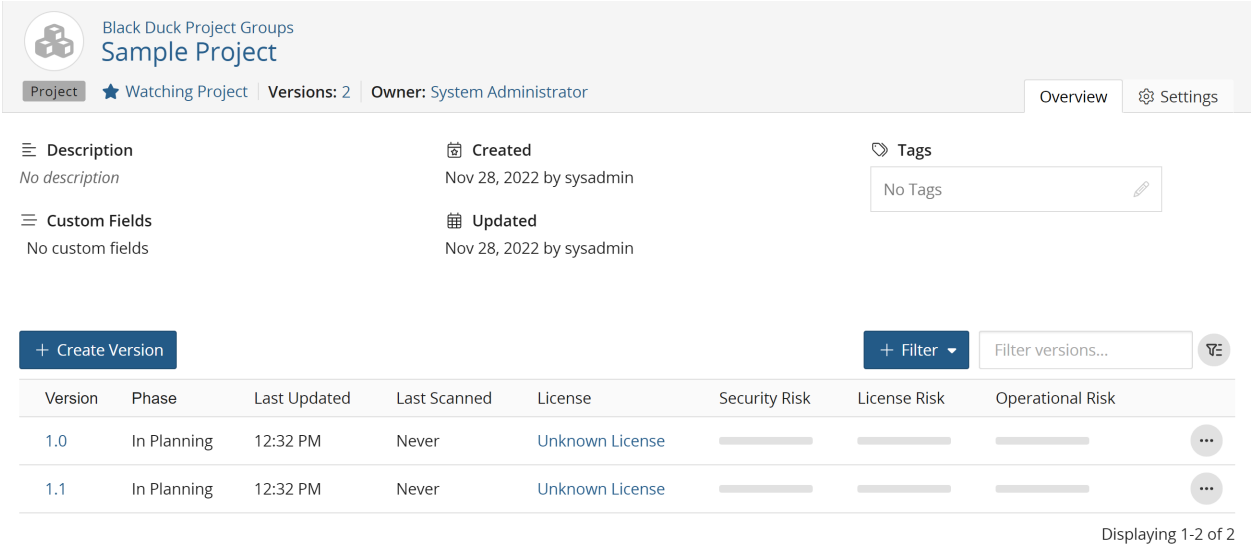
4. Type a name for this version of the project. This name can be a numerical release number, a text description of the version, or any combination of both.
5. From the drop-down list in the **License** field, select the license for this project version. This value is used, for example, for the license 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): An application/product that is sold to customer's externally for use. The source of the application/product is not available to the users but the product is provided externally.
    - SaaS (Software as a Service): An application/product that is shipped as a SaaS service (hosted only).
    - Internal: An application/product not sold and stays within the company. Essentially never leaves the company and gets into customer's hands.
    - Open Source: An application/product shared, essentially full open source (shared on github, etc. with full source code).

    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. Enable or disable **Data Retention**. Enabling this checkbox will prevent this Project Version from ever being deleted by the automated data deletion policies setup by your administrator.

    Once the project has been created, the project version on the project's page will display a lock (🔒) icon at the end of its row to indicate that this project version is protected from automated data deletion.



12. Click **Save**.

    Black Duck saves the project version.

Updating project version information

You can rename a project version and update the following information:

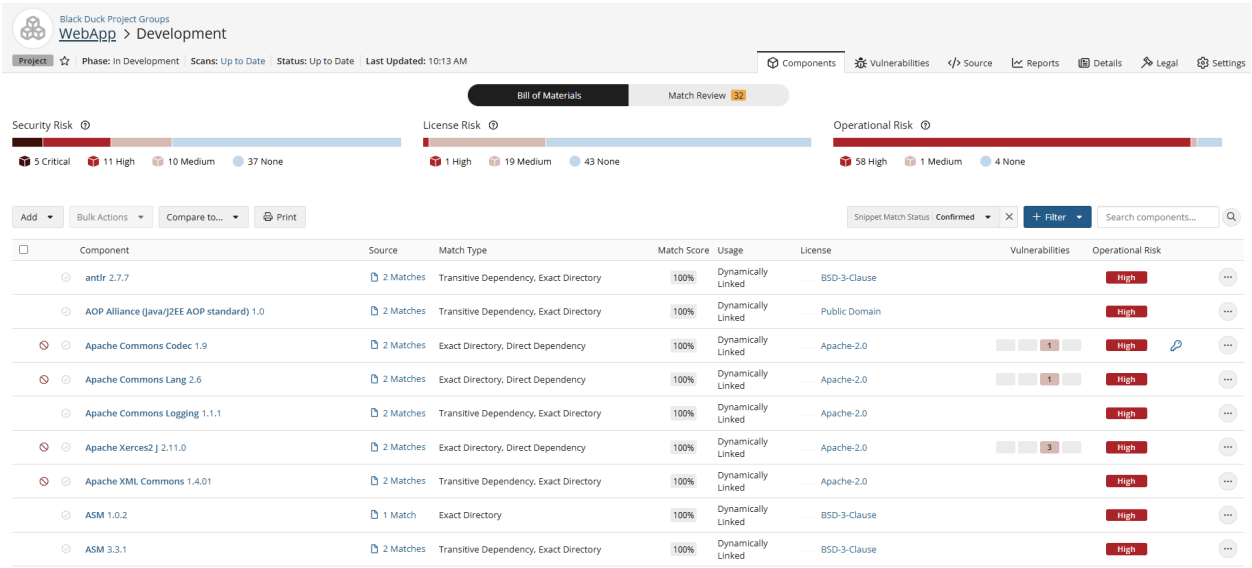
- Version
- License
- SCM Repository Branch

  Note: The SCM Repository Branch field is visible only if this feature is enabled in your environment. Manually changing the SCM branch name could break existing scans.
- Notes
- Nickname
- Release Date
- Phase
- Distribution
- Scan Retention
- Data Retention

  Note: This option is displayed only if the respective flag is set in your environment.

To update project version information:

1. Log in to Black Duck SCA.
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 SCA 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
- Deep License Data
- 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

Note: Cloning a project version in the **Archived** phase will not retain the **Archived** status. The cloned version will default to the **In Development** phase.

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 SCA.
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: The *In Planning* phase is where the project solution is further developed in as much detail as possible and the steps necessary to meet the project's objective are planned.
- In Development: The *In Development* phase is when the project plan is put into motion and the work of the project is performed.
- Pre-release: The *Pre-release* phase is when the project development has been completed and testing is underway.
- Released: The *Released* phase is when the project has officially progressed to general availability and the software product is ready to be delivered or has been delivered.
- Deprecated: The *Deprecated* phase is when the software has reached end-of-life and is no longer sold or supported. In this phase, the software might still be used by customers.
- Archived: The *Archived* phase is similar to the Deprecated phase, however the software is no longer available and customers are no longer utilizing it, having moved on to more recent releases.

Note: The definitions above serve only as examples. You can use the phases in any way to suit your organization's software development life cycle steps.

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 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, modifying component usage and 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 Black Duck KnowledgeBase regarding security vulnerabilities *are* applied to archived project versions.

  Other updates from Black Duck KB, such as updates to license information, *are not* applied to archived project versions.
- New policy rules and updated expressions are not evaluated in archived project versions.
- Disabled and deleted policy rules violations will be removed from archived project versions.

About Long-Term support (LTS) projects

Long-Term Support (LTS) project versions enable tracking of vulnerability data for released product versions. LTS projects are intended for software already in use by end users or customers. Designed with scalability in mind, LTS projects can support extremely high volumes of project versions.

LTS project versions retain minimal data from Active project versions, focusing on tracking newly discovered vulnerabilities in the components within the LTS Bill of Materials (BOM). When converting an Active version to LTS, a Software Bill of Materials (SBOM) is automatically created to facilitate sharing with required third parties. While LTS version do not currently support notifications, they will receive new vulnerability data as it is published to the Black Duck KnowledgeBase.

Note: Converting a project version to LTS is a one-way process and cannot be reverted once started.

- Converting a project version to LTS

Converting a project version to LTS

Long-Term Support (LTS) project versions are designed to track new vulnerabilities for released software artifacts. They are not intended for use by developers or workflows related to application development or compliance workflows.

LTS versions omit project and scan data. This includes:

- Source file information
- Scans and all information related to match types
- Snippets
- IaC results, unmatched component data, and malware information
- Comments

- License conflicts, copyrights, and deep license data
- Vulnerability triage information
- BOM custom fields

Important considerations before converting your project versions to LTS

- Converting from an LTS version back to Active is not currently supported.
- AI models are not retained.
- Custom component vulnerability associations are preserved during LTS conversion. If a custom component version is associated with vulnerabilities through a valid CPE 2.3 identifier, those vulnerabilities remain available in the LTS project version.
- Vulnerability records with multiple remediation origins and different statuses will be merged during conversion. The most severe remediation status among the merged records will be retained.
- Comments from all merged remediation records will be concatenated into a single consolidated comment. This normalization is intentional and ensures consistency in the LTS model, which does not support remediation information at the origin level.
- This behavior reflects the transition from origin-based remediation tracking to the LTS model's component-version-based approach and applies when converting active projects to LTS.

How to convert a project version to LTS

To convert a project version to LTS status:

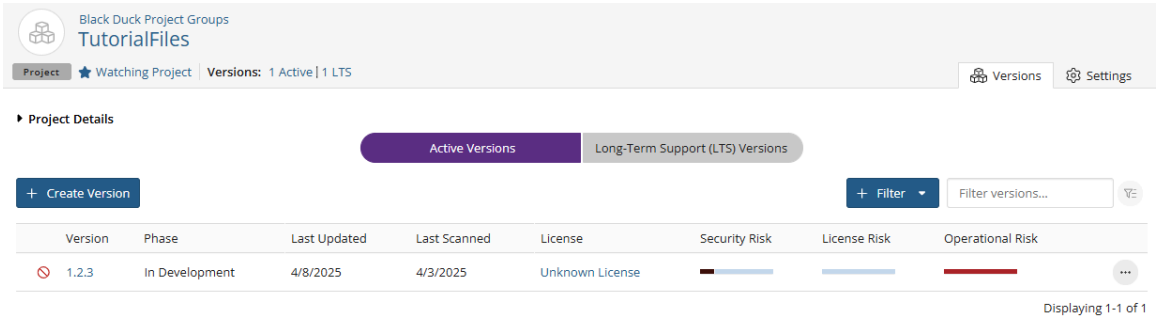
- Log in to Black Duck SCA.
- Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
- Click for the desired project version and select **Convert to LTS**.
- Configure the SBOM report that will be generated during the conversion process:
  - Select a SBOM Template. You can expand the **Template Details** section to view the enabled fields which will appear in the SBOM report.
  - Select a SBOM specification and report format.
  - Click **Convert**.

Once the conversion is complete, the project version will be moved from the **Active Versions** tab of project versions to the **Long-Term Support (LTS) Versions** tab.

Viewing LTS project versions

To view the list of LTS project versions:

- Log in to Black Duck SCA.
- Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
- Click the **Long-Term Support (LTS) Versions** tab.

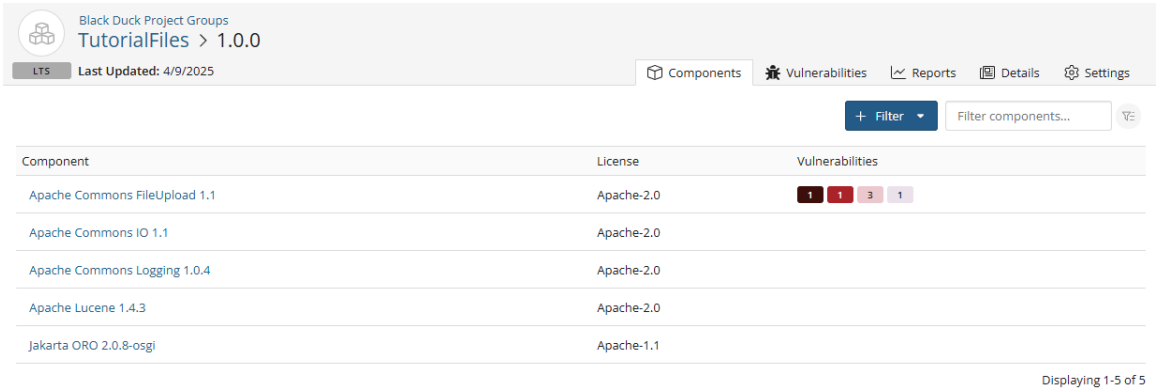


What can I do on the Long-Term Support (LTS) Versions tab?

From the **Long-Term Support (LTS) Versions** tab, you can:

- Filter the table to help find desired versions. You can filter by:
  - Distribution
  - License
  - License Risk
  - Operational Risk
  - Release Phase
  - Security Risk
- View specific LTS project versions BOMs. Click the desired project version in the table brings you to its **Components** tab to view its BOM.
- Delete a LTS project version. Click and select **Delete**.

Viewing a LTS project version BOM

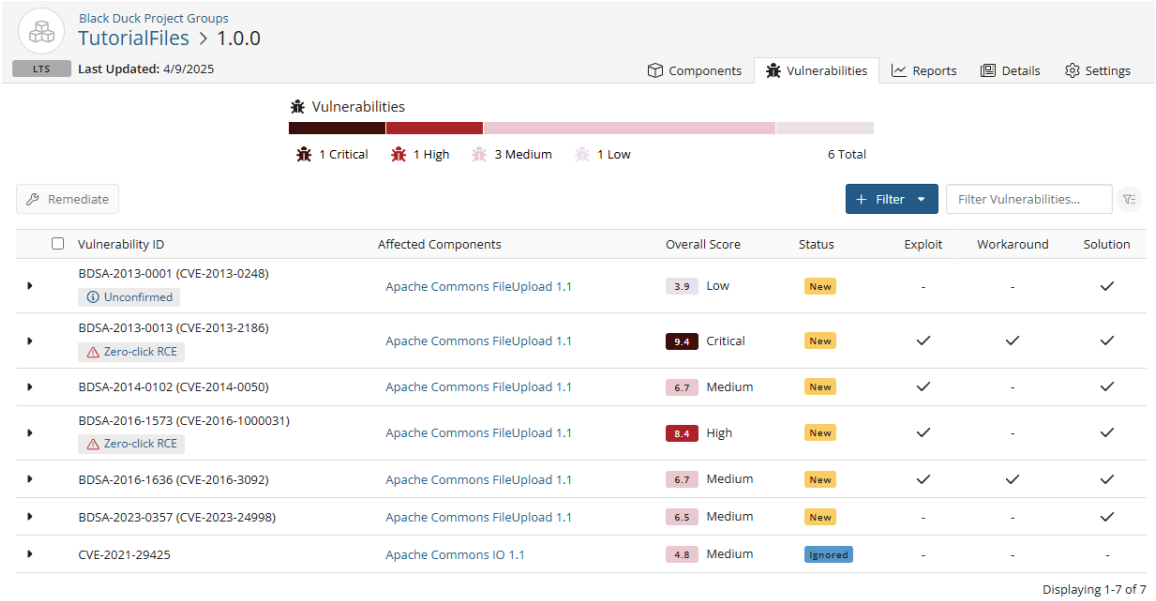


When you click an LTS project version from the list, you'll be taken directly to the Components tab for that version, where you can review the associated bill of materials (BOM):

- Component. The component's name and version.
- License. The component's license.
- Vulnerabilities. Displays a count of the component's known vulnerabilities (Critical, High, Medium, Low). This section also displays upgrade recommendations to mitigate risk.

- SBOM Fields. The SBOM values for the component's PURL, CPE, Originator, Supplier, License Comment, Package Valid Until Date, and Download Location.

Vulnerabilities tab



The Vulnerabilities tab provides a centralized view of all known vulnerabilities assocated with the selected LTS project version. This view is designed to help users quickly assess vulnerability exposure, prioritize remediation efforts, and filter relevant information.

The tab includes the following:

- **Risk profile bar**: A visual summary showing the total number of vulnerabilities, broken down by severity: Critical, High, Medium, and Low.
- **Filter and Search options**: Allows the user to narrow the list of vulnerabilities by specific criteria or search for particular vulnerability IDs or keywords.
- **Vulnerability table**: Displays detailed information for each vulnerability, including:
  - Vulnerability ID: The unique identifier assigned to the vulnerability. This may be a CVE or a BDSA ID.
  - Affected Components: The component(s) within the project version that are impacted by the vulnerability.
  - Overall Score: The highest severity score for the vulnerability across supported scoring systems (e.g., CVSS 3.1, CVSS 4.0, BDSA). Used to prioritize risk.
  - Status: Indicates the current remediation status of the vulnerability.
  - Exploit: A checkmark appears if there is known exploit code publicly available for this vulnerability.
  - Workaround: A checkmark appears if there is a known workaround available that may mitigate the vulnerability without an official fix.
  - Solution: A checkmark appears if a documented solution or patch is available to address the vulnerability.

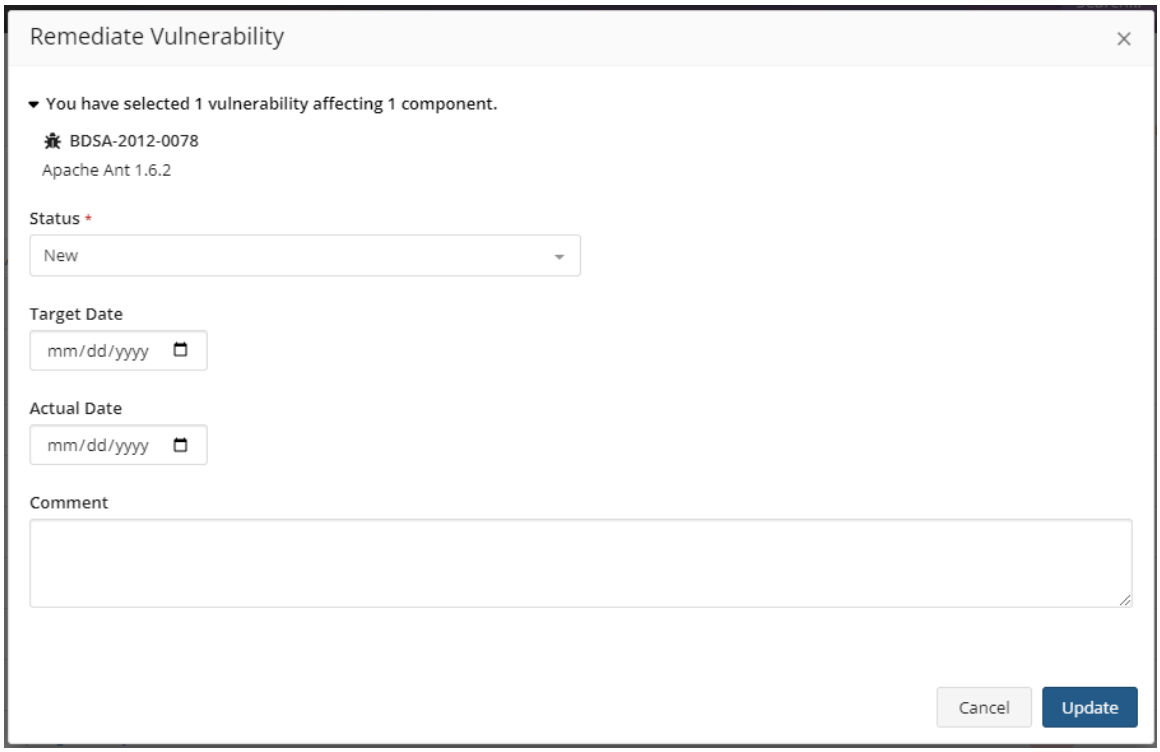
  Note: The Exploit, Workaround, and Solution columns display a checkmark only when applicable information is available to address the vulnerability.

In addition to viewing vulnerability details, you can set the remediation status for each vulnerability. This status helps track whether a vulnerability has been addressed or remains unresolved.

To set the status of a vulnerability:

1. Select the desired vulnerability or vulnerabilities by:
   - Checking the checkbox next to the desired vulnerability and then clicking Remediate. You can select multiple vulnerabilities to remediate using this method.
   - Expanding the vulnerability by clicking the ▸ and either:
     - Check the checkbox next to the desired vulnerability and then click Remediate.
     - Hover your mouse cursor over the **Component** or **Status** areas and then click the 🔧.

   This opens the **Remediate Vulnerability** modal. Clicking the ▸ displays the selected vulnerability and the affected component version.



2. Choose a status from the **Status** drop down menu. Click here for more information on remediation statuses.
3. Select a **Target Date** or **Actual Date** if updating the vulnerability remediation. Click here for more information on these two fields.
4. Add any comments in the **Comments** field.
5. Click **Update**.

Note: Remediation is applied at the component version level. If a component version has multiple origins, all origins will be remediated simultaneously. It is not possible to remediate individual origins separately.

Reports tab

The Reports tab contains the SBOM report created when the project version was converted to LTS.

Details tab

The Details tab contains the same information as displayed when the project was Active.

Settings tab

You can delete the project version from the Settings tab. Once you delete a version, you can't restore it and you will lose all information related to it.

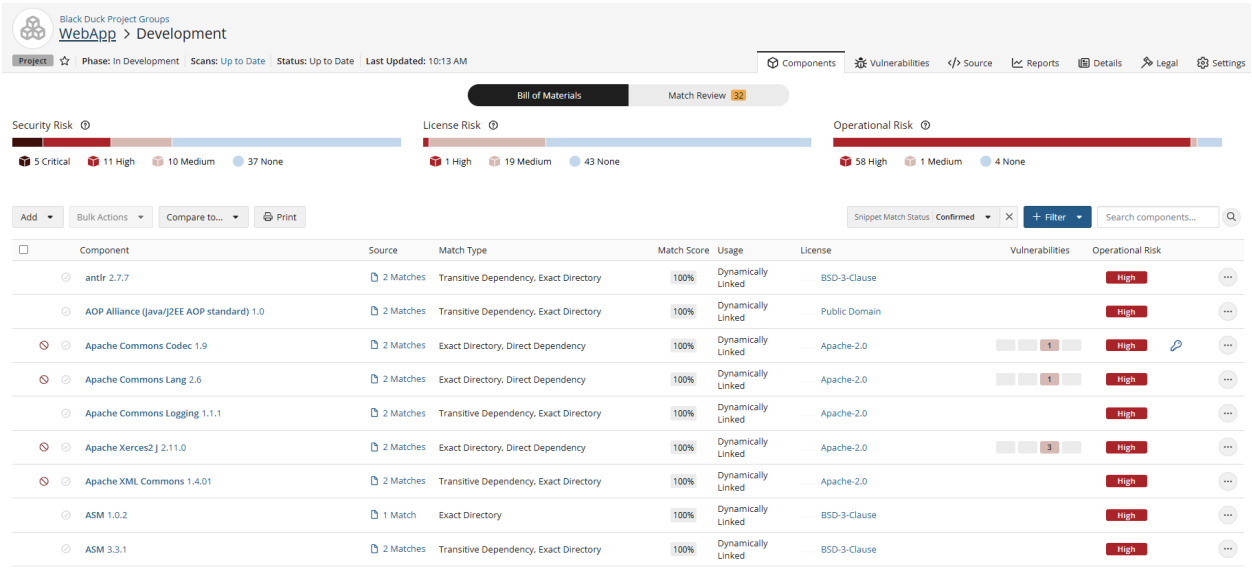
About project version BOMs

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 SCA online help system for information on how users with the BOM Manager and Project Manager role can modify the project version's BOM to reflect how you are actually using the OSS components in the project.

Project version BOM page reference

The Bill of Materials (BOM) provides a comprehensive inventory of all open-source components, third-party libraries, AI models, and subprojects identified within a specific project version. Black Duck SCA uses

the BOM to analyze security vulnerabilities, license compliance risks, and operational risks across your codebase.

To view a project version BOM:

1. Select the project name using the **Watching** or **My Projects** dashboard.
2. Select the desired version name.

Header information

The header of the project version page displays the following information:

- **Owner.** The user assigned as the owner of this project version.
- **Tier.** The tier level assigned to this project.
- **Phase.** The current development phase of this project version, such as In Development or Released.
- **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.** Displays the current processing status of the BOM:
  - Processing. The Black Duck SCA 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 SCA system is currently not processing any events and is up-to-date, however an error has occurred.

  For **Processing** or **Error** statuses, select the indicator to open the BOM Processing Status dialog box.

  The BOM Processing Status dialog box lists each BOM event along with the submitter, submission date and time, start time, elapsed time, and current status. Use this dialog box to identify events that are pending, taking a long time to complete, or have failed.

  - Select **>** next to a failed event to view its error message.
  - Select the delete icon to dismiss individual errors, or select **Dismiss all** to clear all errors at once.

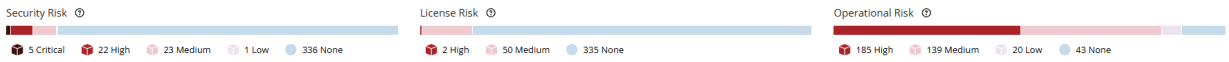
  The dialog box refreshes the event table automatically every 30 seconds while open. To retrieve an immediate update, close and reopen the dialog box.

  Note: Refer to the installation guide for information on configuring the frequency of the BOM event cleanup job (VersionBomEventCleanupJob), which clears BOM events that may become stuck due to processing errors or topology changes.
- **Last Updated.** The date and time the project version was last updated.

Risk summary bars

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

- 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.

Unconfirmed snippets and unmatched components

If there are any unconfirmed snippets, you will see them displayed on the right side of the Risk graphs. Clicking this link will take you to Source view in order to take action on these items.

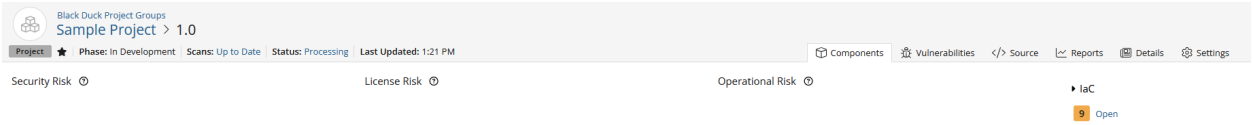


Note: Unmatched components are found on the Match Review page for the project version.

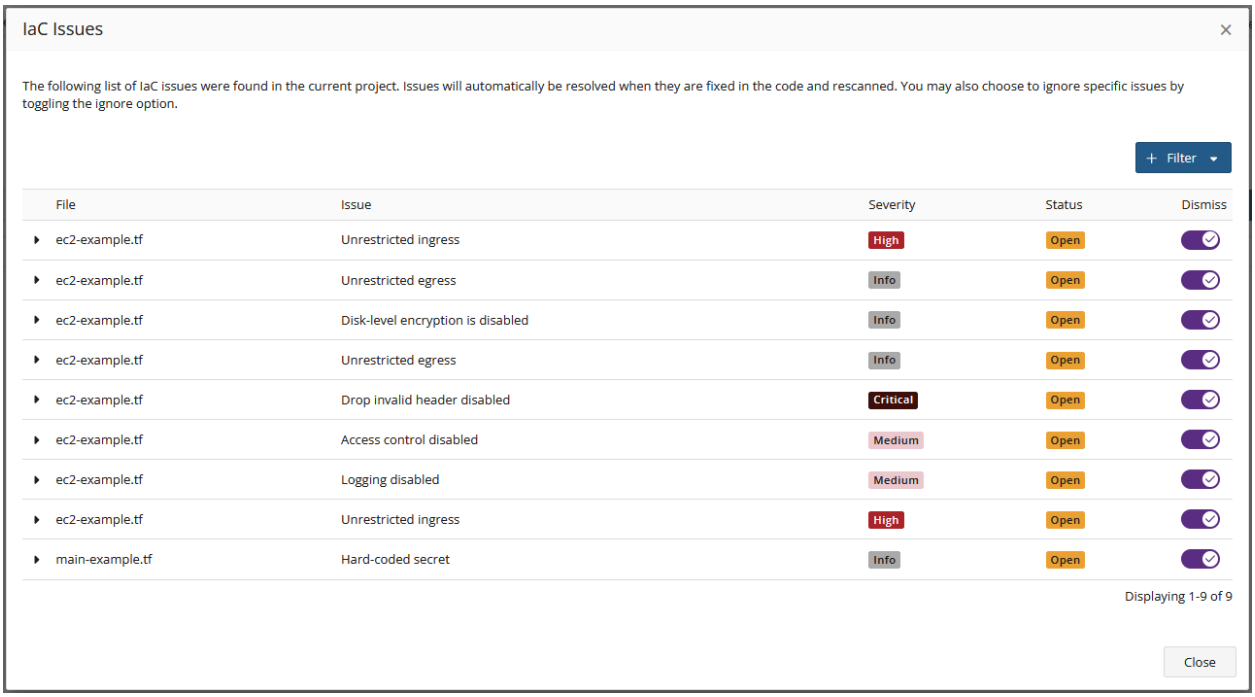
Infrastructure as Code

Infrastructure as Code (IaC) is the management and provisioning of infrastructure (networks, virtual machines, load balancers, and connection topology) through code or configuration files instead of through manual processes.

If your scan included Infrastructure as Code, you will see the current amount of Open issues displayed on the right side of the Risk graphs. You can then expand the IaC link to view the Total amount of IaC issues and Dismissed issues.



You can then take action on the Infrastructure as Code issues discovered in your BOM by clicking either the Open link or the amount to the left of the Open link. This will open a dialogue box displaying all the IaC issues.



From here, you can:

- Expand a row to see specific details of the nature of the issue. This information includes a description of the issue, the severity, the suggested remediation, and the code location of the issue.
- Dismiss an issue. By toggling the slider in the Dismiss column, you can mark an issue as dismissed. This means that you have either remediated the issue or have chosen to ignore it.
- Filter the list. Click the Filter button to add filters to the list. Options include Issue, Severity, and Status.

For more information regarding Infrastructure as Code scanning, please refer to the Sigma User Guide.

Bill of materials table

The Bill of Materials (BOM) table contains information about the components and subprojects in this version of the project.

In the Bill of Materials view, select the options icon located in the far-right column of a row to modify, ignore, or (for manually added components) delete components or subprojects from the BOM.

When you edit a component — using either the BOM or the Source tab — an ⓘ icon appears in the table row to indicate that a manual adjustment was made to that component. Select the ⓘ icon to open the Component Details dialog box, which displays the edits made to the component.

The BOM table includes the following columns. Select a column name below for detailed information about columns that require further explanation.

Column	Description
First column	Displays status icons to the left of the component or subproject name, indicating policy violations and review status.
Component	Lists the names and versions of all components, AI models, and subprojects identified in the project version. Select an entry to open

Column	Description
	the slide-out panel or navigate to the full details page. See Component column for more information.
Source	Displays the number of matching files or archives for components, or the number of components for subprojects. Select the value to navigate to the relevant view.
Match Type	Indicates how the component was matched to a version in the Black Duck KnowledgeBase. See Match Type column for more information.
Match Score	Indicates the confidence level that the matched component is the correct component and version. See Match Score column for more information.
Usage	Indicates how the component is intended to be included in the project when this version is released. See Usage column for more information.
License	Displays the declared license of the component or subproject and its associated license risk level.
Vulnerabilities	Displays the number of critical/high, medium, and low risk vulnerabilities associated with the component or subproject.
Cryptography	Indicates that this component version contains encryption algorithms.
Operational Risk	Displays the operational risk level — High, Medium, or Low — for the component or subproject. See Operational Risk column for more information.

First column

The first column displays icons to the left of the component or subproject name to indicate its status:

- Policy violation.
- Policy violation has been overridden.
- Component or subproject has not been reviewed.
- Component or subproject has been reviewed.

Component column

The Component column lists the names and versions of all open source and third-party components and AI models identified in the selected project version, as well as any subprojects that have been added as components. Each entry represents a specific component version or subproject and serves as a link to more detailed information, including license, security, and usage data. Components shown include both top-level (parent) and subcomponents (children).

AI models are marked with a distinctive icon at the beginning of the row, preceding the model name.

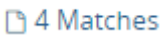
- Selecting a component opens a slide-out panel for quick review, or allows navigation to the full component details page.
- Select the version number to open the Black Duck KnowledgeBase component version page, which displays a list of the projects and project versions in which this version of the component is used.
- Select **?**, which indicates an unknown version, to open the KnowledgeBase component page, which provides general information about the component.

- Hover over the component to view its origin and origin ID. If the component version has multiple origin IDs, they are listed in the popup.

Note: If a component has more than one origin for a version, the table displays the highest risk values.

If the component version's origin is not specified, the popup will notify you that the license risks for this component are estimated and that you should manually specify a version for a more accurate result. If the component's version cannot be identified, the popup will indicate as such.

Source column

For components: Number of archives or files that match. For example: 

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.

For subprojects: Number of components in the subprojects. For example: 

Select the value to open the BOM for this project version. The BOM only appears if you have permission to view the project.

Match Type column

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

Match Type	Description
Binary	Binary match from Black Duck Binary Analysis.
Direct Dependency	Direct dependency identified via package manager scanning.
Direct Dependency Binary	Direct dependency identified from Black Duck Binary Analysis.
Exact Directory	Exact directory match from Signature scanning or Binary Analysis.
Exact File	Exact file match from Signature scanning or Binary Analysis.
File Dependency	Deprecated and no longer used.
Files Added/Deleted	A fuzzy signature match to a directory where some of the OSS component's files were added, deleted, or modified in the scanned archive. This may be a match to a previous or subsequent version of the component.
Files Modified	A fuzzy signature match to a directory where some of the archive files were modified. This may be a match to a previous or subsequent version of the component.
Manually Added	Component manually added to the BOM in Black Duck SCA.
Manually Identified	Manually identified BOM component related to signature match files.
Manually Identified Package	Manually identified BOM component related to an identified package manager package.
Partial	Deprecated and no longer used.
Snippet	Snippet scanning identified a portion of code in your file that matches code in one or more KnowledgeBase files.
SBOM	Imported from an SBOM.

Match Type	Description
Transitive Dependency	Transitive dependency identified via package manager scanning.
Transitive Dependency Binary	Transitive dependency identified from Black Duck Binary Analysis.

When viewing components across the various views, precedence of Direct Dependency over Transitive Dependency is applied. If the source hierarchy has a component as both a direct and transitive dependency, the Match Type field will always show that component as a Direct Dependency, even when viewing the transitive dependency in the Source Tree.

The following match types apply to automatic matches from an imported Protex BOM: Exact, Partial, and File Dependency.

The match type for subprojects is **Manually Added**.

Match Score column

Indicates the level of confidence that a particular matched component is in fact the component and version displayed.

The overall match score is calculated based on two factors:

1. **Degree of ambiguity.** The number of possible matches for this component, including the one selected in Black Duck SCA.
2. **Percentage of KnowledgeBase artifact matched.** The percentage of a KnowledgeBase download from the BOM entry's data that matched the scanned data. This is based on the download with the highest percentage match to the scanned files.

The match score value does not change after a BOM component resulting from a signature scan is edited or modified.

Note: Manually added components and components imported from SBOMs will always display 100% match confidence.

Important: When SCASS is enabled in Black Duck SCA, match scores are returned for package manager scans.

The match score appears in one of two colors:

- Yellow, if the score is within the warning threshold.
- Gray, otherwise.

Selecting a match score displays a popover that reveals additional details, including the values described above.

Usage column

For components, indicates how the 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 can indicate in the BOM that they will not be included in the released version of the project.

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.

Usage Value	Description
Dynamically Linked	A moderately integrated component that is dynamically linked in, such as with DLLs or .jar files. This is the default value.

Usage Value	Description
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. Typically used when packaging a component's sources with the build, a binary, or distribution.
Separate Work	Intended for loosely integrated components where your work is not derived from the component. Your application has its own executables with no linking between the component and your application.
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.
Implementation of Standard	Intended for cases where you implemented according to a standard, such as 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, such as components used for building, development, or testing.
Unspecified	The usage for this component has not yet been determined and requires further investigation.

For subprojects, usage defaults to **Dynamically Linked**.

License column

Displays the declared license of the component or subproject in use in this version of your project. A colored icon indicates the license risk level:

- High license risk.
- Medium license risk.
- Low license risk.
- No license risk.

For known licenses, select the license name to view license details and license text.

If the license text indicates that there is more than one license for a component version — for example, "Apache 2.0 and 3 more..." — hover over the license name to view the names of all licenses.

Vulnerabilities column

Displays the number of critical/high (fully red), medium (partially red), and low (gray) risk vulnerabilities associated with this version of the component or subproject.

Select a value to open the project version Vulnerabilities tab, which displays the vulnerabilities for that component or subproject. If the component has an unknown version, a modal will appear detailing the estimated security risk for the component.

For subprojects, the value shown is the total number of vulnerabilities for all components. Note that these values may not match the values shown on the subproject version's BOM page, which lists the number of components with a vulnerability.

Note: If you do not have permission to view the project, you will not be able to access this page.

Note: Risk counts do not include items that are in match review.

Cryptography column

Indicates that this component version has encryption algorithms.

Operational Risk column

Displays the operational risk level — High, Medium, or Low — for the component or subproject in use in this version of your project.

The operational risk level is calculated using a combination of the following factors:

- **Version status.** Based on the version of the component used compared to the number of newer versions released and the time since the newest version was released. Using older versions when newer versions are available is considered risky.
- **Activity status.** Based on the commit activity trend for the component over the last 12 months. Increasing or stable commit activity is considered less risky than decreasing commit activity.

The final operational risk value is the higher of these two calculations.

For components, hover over the value to view the factors that determined the risk level shown.

For subprojects, hover over the value to see the number of components in this project version at each operational risk level. Note that these values 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.

Component slide-out panel

When you select a component, subproject, or AI model from the Components column, a slide-out panel opens from the right side of the page. This panel provides a quick summary of important information without navigating away from the Bill of Materials view.

The slide-out panel for component and subproject includes:

- **Vulnerabilities**: The number of known security vulnerabilities associated with the component.
- **License**: The license associated to the component version.
- **Upgrade Recommendation**: Suggested versions, if any, to remediate known vulnerabilties or policy violations. Short-term typically reflects upgrading a component's minor or patch version. Long-term will focus on upgrading a component's major version.
- **Source**: The origin of the component.
- **Fields**: Values such as CPE, PURL, and any populated custom fields.
- **More Details**: Additional information such as component links, description, tags, and approval status.

The slide-out panel for AI models includes:

- **Source**: The origin of the AI model.
- **More Details**: Additional information such as model description, tags, and approval status.

Clicking the link at the top of the slide-out panel opens the corresponding page for the component, component version, subproject, or AI model depending on the type of item selected.

Understanding direct and transitive dependencies

When analyzing your project's bill of materials (BOM), it's important to understand the difference between direct and transitive dependencies:

- **Direct dependency**: A component that your project explicitly declares and includes in its code or configuration files. These are dependencies you directly add to your project.

  For example, if your project's configuration file includes `library-A`, then `library-A` is a direct dependency.

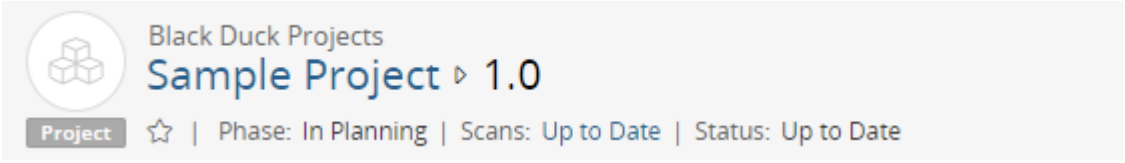
- **Transitive dependency**: A component that is not directly declared by your project, but is required by one of your direct dependencies. These are automatically included through the dependency chain.

  For example, if `library-A` (your direct dependency) includes `library-B`, then `library-B` is a transitive dependency in your project.

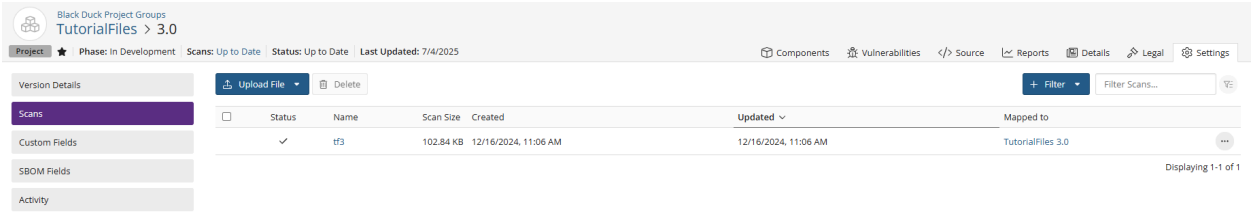
Understanding this distinction helps you make informed decisions about risk, remediation, and license compliance when managing open source components in your project.

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.



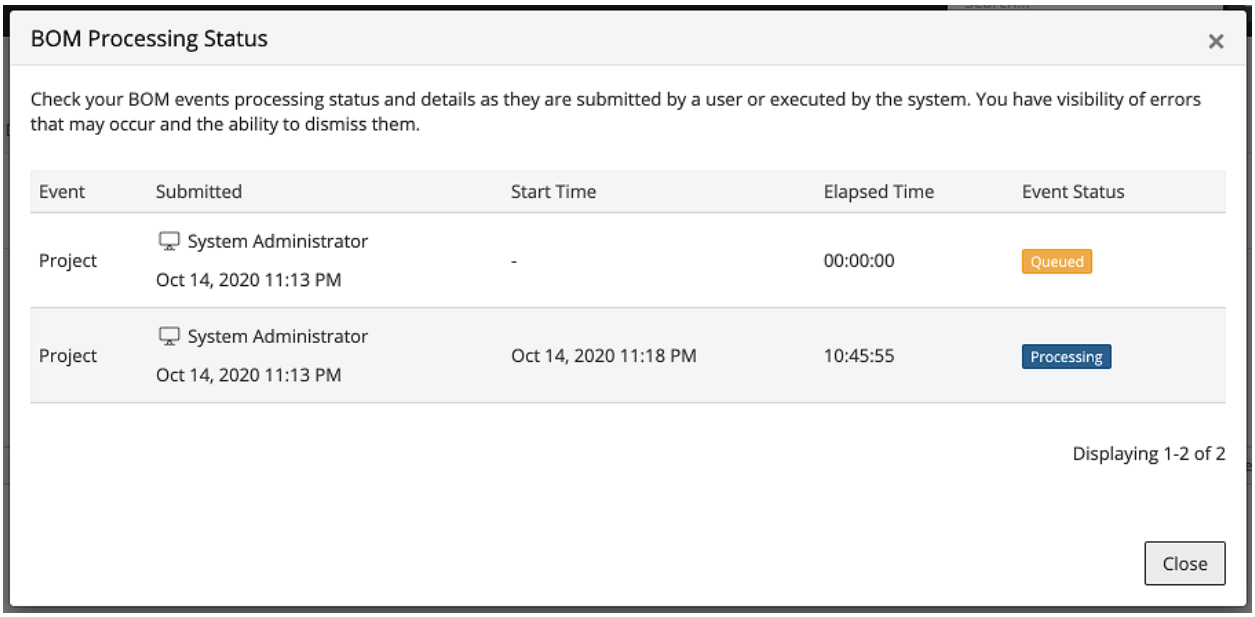
Scans provides the status of the scans being processed for this BOM. Once the scan completes successfully, an Up to Date status appears. Select the link to view the **Scans** tab of the *Project Name Version Name* **Settings** tab. Use this page to manage the scans for this project version.



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

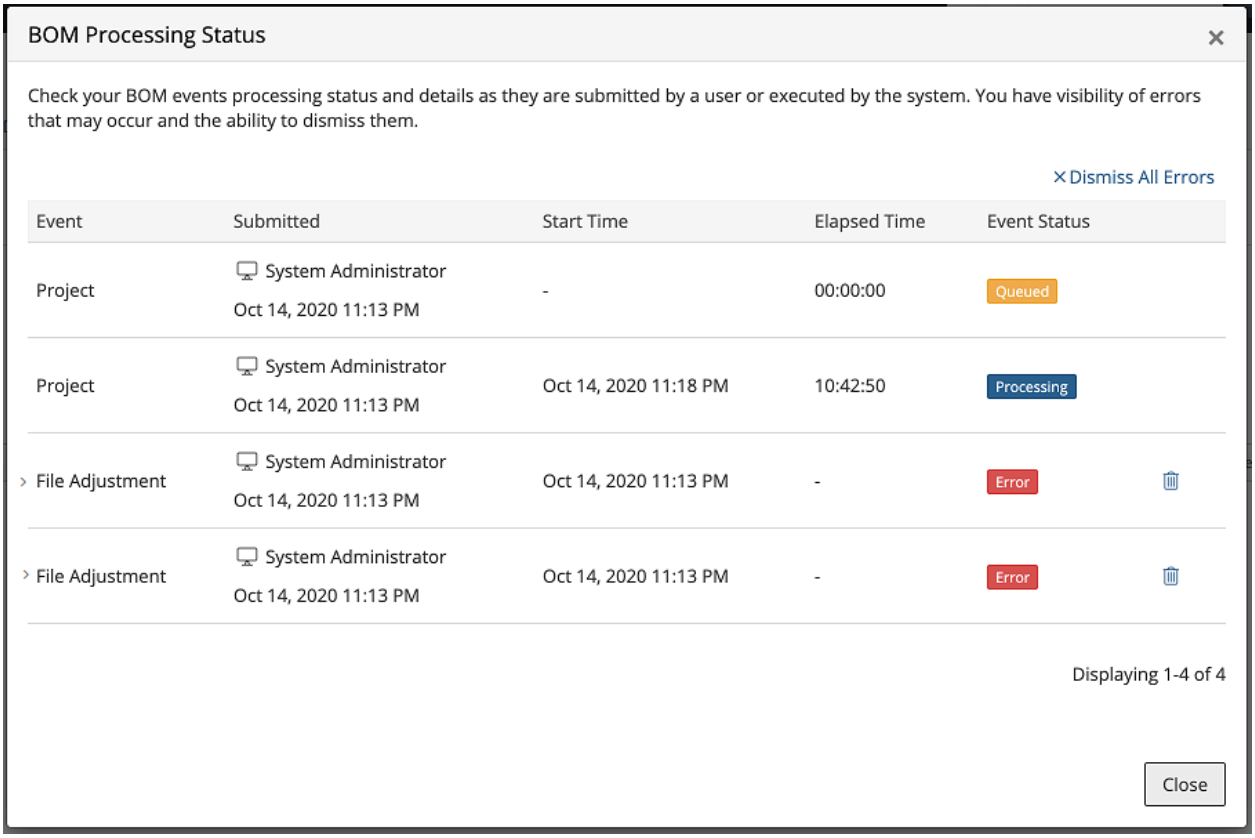
- Processing. The Black Duck system is processing events to create or update the BOM.
- Up to Date. The BOM is up-to-date; there are no errors.
- Error. An error has occurred while processing an event or the Black Duck system is currently not processing any events and is up-to-date, however an error has occurred.

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



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.



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

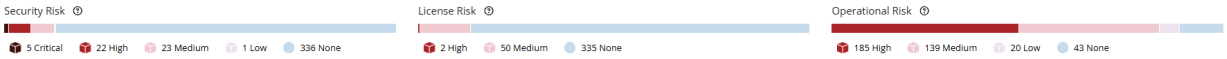
If left open, the dialog box updates the information shown in the table every 30 seconds, otherwise close and reopen the BOM Processing Status dialog box for a fresh update on the status of events.

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.

Unconfirmed snippets and unmatched components

If there are any unconfirmed snippets, you will see them displayed on the right side of the Risk graphs. Clicking this link will take you to Source view in order to take action on these items.

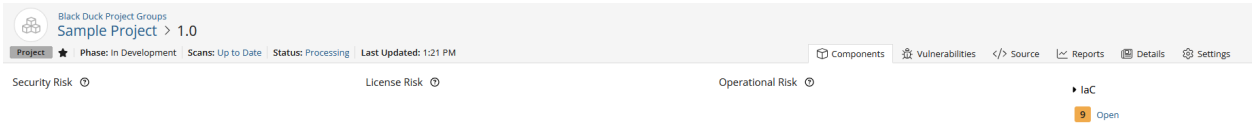


Note: Unmatched components are found on the Match Review page for the project version.

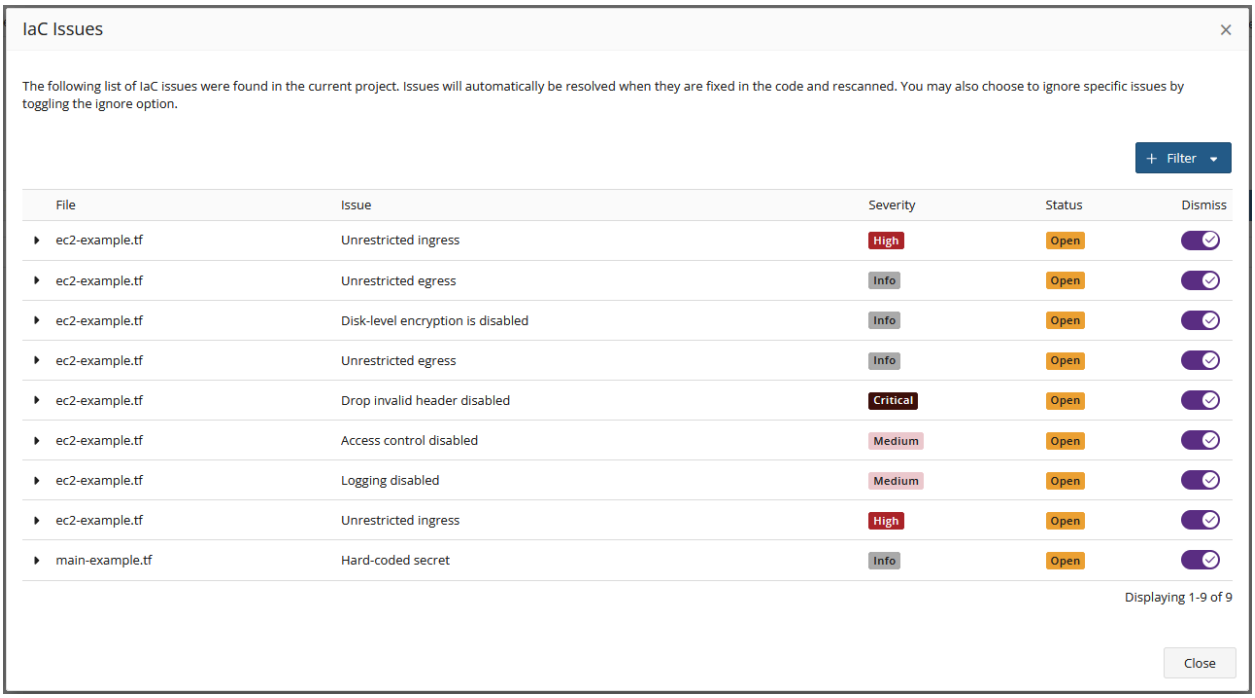
Infrastructure as Code

Infrastructure as Code (IaC) is the management and provisioning of infrastructure (networks, virtual machines, load balancers, and connection topology) through code or configuration files instead of through manual processes.

If your scan included Infrastructure as Code, you will see the current amount of Open issues displayed on the right side of the Risk graphs. You can then expand the IaC link to view the Total amount of IaC issues and Dismissed issues.



You can then take action on the Infrastructure as Code issues discovered in your BOM by clicking either the Open link or the amount to the left of the Open link. This will open a dialogue box displaying all the IaC issues.



From here, you can:

- Expand a row to see specific details of the nature of the issue. This information includes a description of the issue, the severity, the suggested remediation, and the code location of the issue.
- Dismiss an issue. By toggling the slider in the Dismiss column, you can mark an issue as dismissed. This means that you have either remediated the issue or have chosen to ignore it.
- Filter the list. Click the Filter button to add filters to the list. Options include Issue, Severity, and Status.

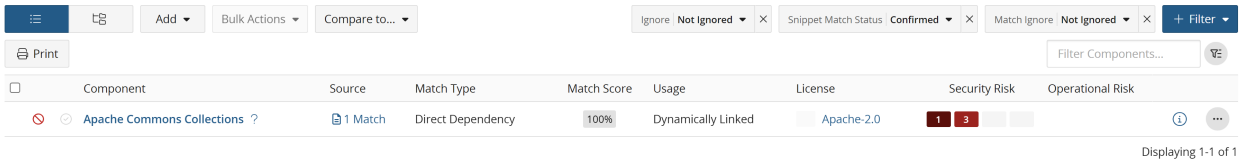
For more information regarding Infrastructure as Code scanning, please refer to the Sigma User Guide.

Bill of materials data table

The Bill of materials (BOM) 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:



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

The data table has the following columns:

- First column
- Component
- Source
- Match Type
- Match Score
- Usage
- License
- Vulnerabilities
- Cryptography
- Operational Risk

First column (N/A)

Icons shown to the left of the component or subproject name:

- ⦸ Policy violation.
- ⓘ Policy violation has been overridden.
- ✓ Component or subproject has not been reviewed.
- ✓ Component or subproject has been reviewed.

Component column

The Component column lists the names and versions of all open source and third-party components and AI models identified in the selected project version, as well as any subprojects that have been added as components. Each entry represents a specific component version or subproject and serves as a link to more detailed information, including licence, security, and usage data.

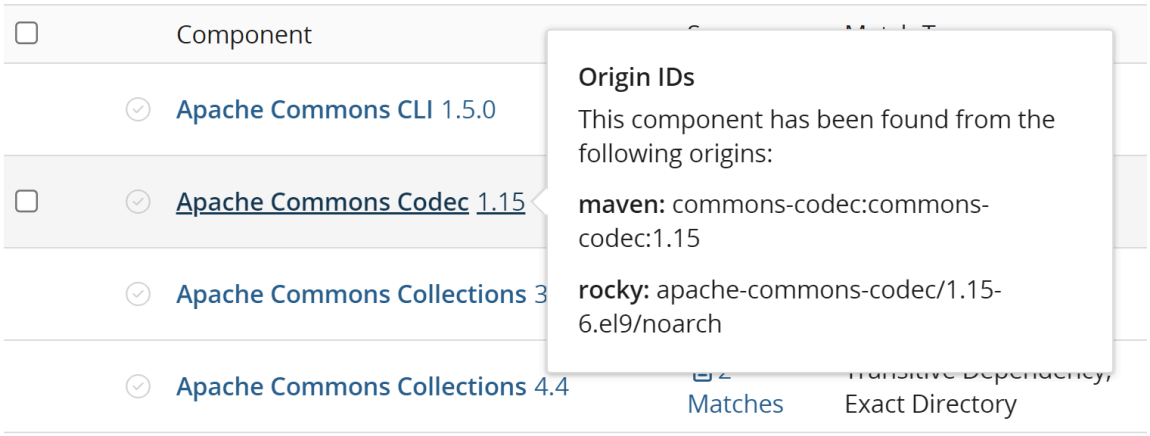
Components shown are top-level (parent) and subcomponents (children).

AI models are marked with a distinctive icon (✦) in the component list. This visual indicator appears at the beginning of the row, preceding the model name.

- Selecting a component opens a slide-out panel for quick review or allows navigation to the full component details page.
- Select the version number to open 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.

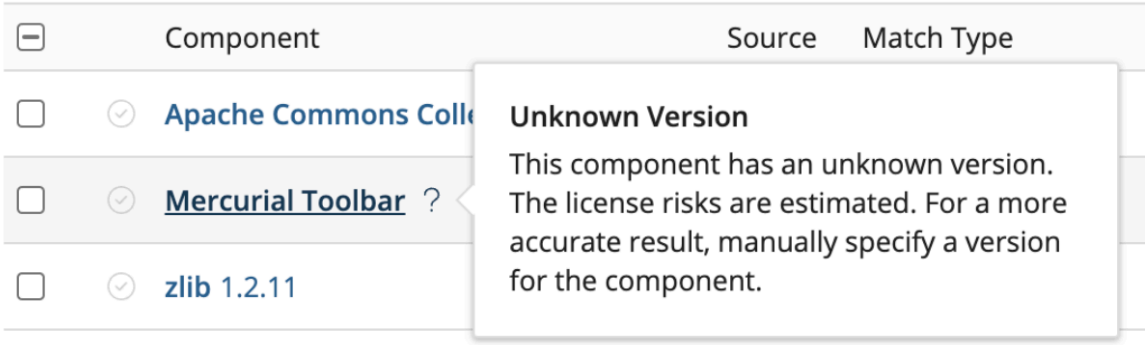
- Mouse over the component to view the origin and origin ID.

  If the component version has multiple origin IDs, they will be listed in the popup.

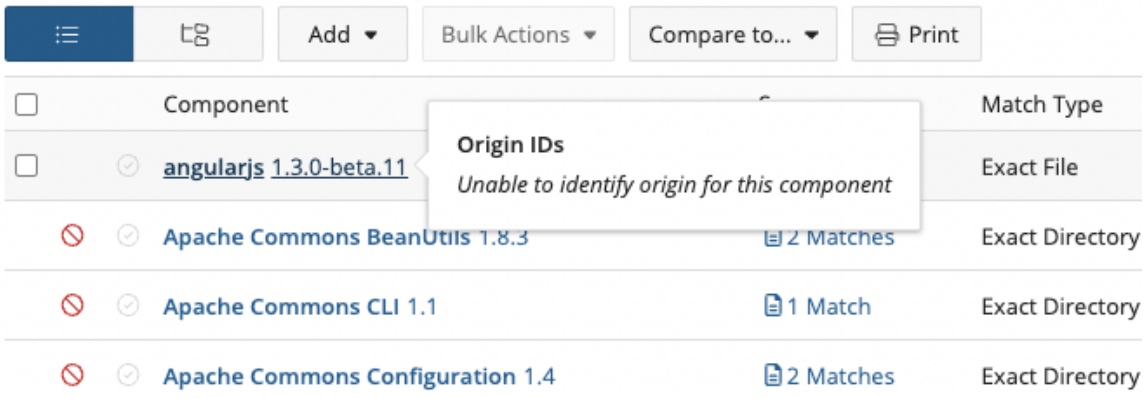


Note: If a component has more than one origin for a version, the table displays the highest risk values.

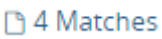
If the component version's origin is not specified, the popup will notify you that the license risks for this component are estimated and that you should manually specify a version for a more accurate result.



If the component's version cannot be identified, the popup will indicate as such.



Source column

For components: Number of archives or files that match. For example: 

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.

For subprojects: Number of components in the subprojects. For example: 

Select the value to open the BOM for this project version. The BOM only appears if you have permission to view the project.

Match type column

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

Possible values are:

- **Binary**. Binary match from Black Duck Binary Analysis.
- **Direct Dependency**. Direct dependency identified via package manager scanning.
- **Direct Dependency Binary**. Direct dependency identified from the Black Duck Binary Analysis.
- **Exact Directory**. Exact directory match from Signature scanning or Binary Analysis.
- **Exact File**. Exact file match from Signature scanning or Binary Analysis.
- **File Dependency**. Deprecated and no longer used.
- **Files Added/Deleted**. A fuzzy signature match to a directory where some of the OSS component's files were added, deleted, or modified in the scanned archive. This may be a match to a previous or subsequent version of the component, which might have been missing from Black Duck KB at the time that the match was made.
- **Files Modified**. A fuzzy signature match to a directory where some of the archive files were modified. This may be a match to a previous or subsequent version of the component, which might have been missing from Black Duck KB at the time that the match was made.
- **Manually Added**. Component manually added to BOM in Black Duck.
- **Manually Identified**. Manually identified BOM component related to signature match files.
- **Manually Identified Package**. Manually identified BOM component related to a identified package manager package.
- **Partial**. Deprecated no longer used.
- **Snippet**. Snippet scanning identified a portion of code in your file that matches code in one or more KnowledgeBase files. More details about snippets can be found here.
- **SBOM**. Imported from a SBOM.
- **Transitive Dependency**. Transitive dependency identified via package manager scanning.
- **Transitive Dependency Binary**. Transitive dependency identified from the Black Duck Binary Analysis.

When viewing Components in the various views (Component Tab, Source Tab, etc), precedence of "Direct Dependency" over "Transitive Dependency" is given. If the source hierarchy has a component as both a Direct and Transitive Dependency, the "Match Type" field will always show that component as a Direct Dependency even when viewing the transitive dependency in the Source Tree.

The following are automatic matches from an imported Protex BOM:

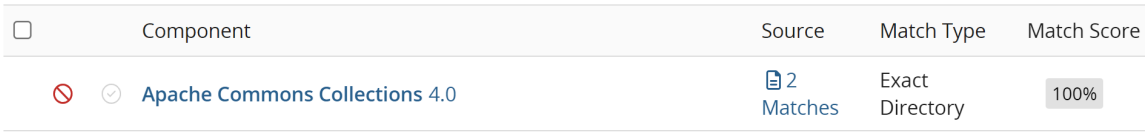
- **Exact**
- **Partial**
- **File Dependency**

Click here for more information.

The match type for subprojects is **Manually Added**.

Match score column

Indicates the level of confidence that a particular matched component is in fact the component and version displayed.



The overall match score is calculated based on two factors:

1. Degree of ambiguity: The number of possible matches for this component (including the one selected in Black Duck).
2. Percentage of KnowledgeBase artifact matched: This value represents the percent of a KnowledgeBase Download from the BOM Entry's data which matched the customer's scanned data. The KnowledgeBase's representation of the BOM Entry may have many downloads. The % of KnowledgeBase Artifact Matched is based on the Download with the highest percent match to the customer's scanned files.

The match score value does not change after a BOM component (resulting from a signature scan) is edited or modified.

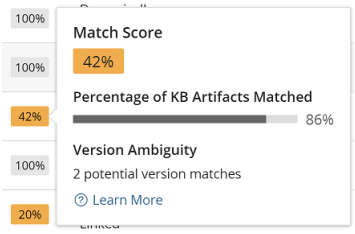
Note: Manually added components and components imported from SBOMs will always display 100% match confidence.

Important: When SCASS is enabled in Black Duck SCA, match scores are returned for package manager scans.

The match score will appear as one of two colors:

- Yellow if the score is within the "warning" threshold
- Gray otherwise

Clicking on a match score displays a small popover that reveals the additional details about the match score (including the other values discussed above):



Usage column

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 released version of the project.

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.

The possible usage statuses are:

- **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.

For subprojects, usage defaults to **Dynamically Linked**, as described above.

License column

Declared license of the component or subproject in use in this version of your project.

- H indicates that the component/subproject has a high license risk.
- M indicates that the component/subproject has a medium license risk.
- L indicates that the component/subproject has a low license risk.
- (white box) indicates that there is no license risk.

For known licenses, select the license name to view license details and license text.

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.

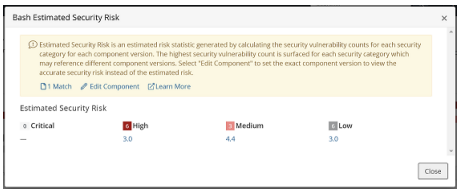
Click here for more information on how license risk for a component is determined.

Vulnerabilities column

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:



Select a value to open the project version page Vulnerabilities tab which displays the vulnerabilities for that component or subproject. If the component has an unknown version, a modal will be displayed detailing the estimated security risk for the component.



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.

Note: If you do not have permission to view the project, you will not be able to access this page.

Attention: Risk counts do not include items that are in match review.

Cryptography

Indicates that this component version has encryption algorithms.

Operational risk column

Operational risk level for the component or subproject in use in this version of your project:

- High risk
- Medium risk
- Low risk

The operational risk level in this version of your project is calculated using a combination of:

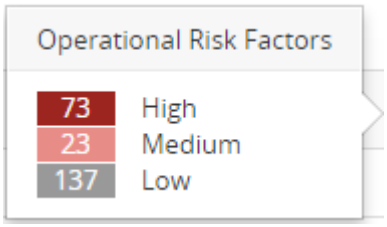
- Version status. Part of the component's operational risk calculation is based on the version of the component used compared to the number of newer versions that have been released and the time since the newest version was released. Using older versions of a component is considered risky when newer 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.

The final operational risk will be the higher of these two risk calculations.

In the component list view, for components, hover over the value to view the factors that determined the value shown:



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:



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.

Component slide-out panel

When you select a component, subproject, or AI model from the Components column, a slide-out panel opens from the right side of the page. This panel provides a quick summary of important information without navigating away from the Bill of Materials view.

The slide-out panel for component and subproject includes:

- **Vulnerabilities**: The number of known security vulnerabilities associated with the component.
- **License**: The license associated to the component version.
- **Upgrade Recommendation**: Suggested versions, if any, to remediate known vulnerabilties or policy violations. Short-term typically reflects upgrading a component's minor or patch version. Long-term will focus on upgrading a component's major version.
- **Source**: The origin of the component.
- **Fields**: Values such as CPE, PURL, and any populated custom fields.
- **More Details**: Additional information such as component links, description, tags, and approval status.

The slide-out panel for AI models includes:

- **Source**: The origin of the AI model.
- **More Details**: Additional information such as model description, tags, and approval status.

Clicking the link at the top of the slide-out panel opens the corresponding page for the component, component version, subproject, or AI model depending on the type of item selected.

Understanding direct and transitive dependencies

When analyzing your project's bill of materials (BOM), it's important to understand the difference between direct and transitive dependencies:

- **Direct dependency**: A component that your project explicitly declares and includes in its code or configuration files. These are dependencies you directly add to your project.

  For example, if your project's configuration file includes `library-A`, then `library-A` is a direct dependency.

- **Transitive dependency**: A component that is not directly declared by your project, but is required by one of your direct dependencies. These are automatically included through the dependency chain.

  For example, if `library-A` (your direct dependency) includes `library-B`, then `library-B` is a transitive dependency in your project.

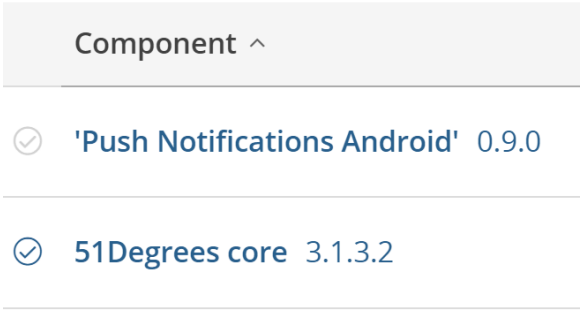
Understanding this distinction helps you make informed decisions about risk, remediation, and license compliance when managing open source components in your project.

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.
   - **Mark as unreviewed** 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 Tree View of the BOM.

Editing a project version BOM

Applying adjustments to all versions of a project

You can select whether adjustments to a component apply to a specific version of a project or if adjustments apply to all versions of a project.

What is the Component Adjustments setting?

Component Adjustments in the Project settings ensures that adjustments made to components are consistently applied across all versions of a project. However, it is important to understand how this setting interacts with manually added components.

Functionality of Component Adjustments

When you make adjustments to components that are shared across multiple project versions, those adjustments will be applied to all versions where the component exists. This means that if you mark a component as ignored in one project version, it will also be ignored in all other versions of that project where the component is present.

If you enable Component Adjustments then adjustments 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.

Manual Addition of Components

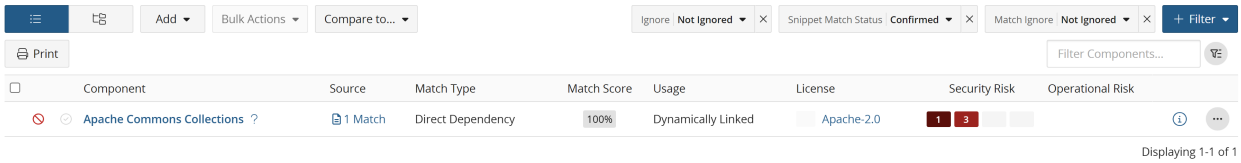
If a component is manually added to a specific project version, it does not automatically create a corresponding entry in other project versions. Each project version needs to have the component manually added if it is not already present. However, once a component is added to multiple versions, any further adjustments will apply universally to all instances of that component.

Note: There are instances when adjustments may not propagate to all versions. See Component Adjustment examples below.

Component Adjustments are enabled by default when you create a project.

CAUTION: Please exercise caution when changing this option in order to avoid the Component Adjustment examples below. 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 Source page), a ⓘ appears in the table row to indicate that a manual adjustment was made to this component:



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.

Component Adjustments examples

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

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

Table 8: Scenario 1

Steps	Example	Final Result
Component Adjustments are enabled.		
An adjustment is made to an item in a component in one version of the project.	The license for Component A is changed in Version 1 of the project.	The adjustment is propagated to all versions of the project.
Component Adjustments are then disabled.		

Steps	Example	Final Result
An adjustment is made to the *same item* in Component A in a version of the project.	The license for Component A is changed in Version 1 (or Version 2) of the project.	Although Component Adjustments are disabled, the edit is propagated to *all versions* of the project as the original adjustment was made when Component Adjustments were enabled.

Table 9: Scenario 2

Steps	Example	Final Result
Component Adjustments are disabled.		
An adjustment is made to an item in a component in one version of the project.	The license for Component A is changed in Version 1 of the project.	The adjustment appears in only Version 1 of the project.
Component Adjustments are then enabled.		
An adjustment is made to the same item in the same component in the same version.	The license for Component A is changed again in Version 1 of the project.	The adjustment is applied to only that version of the project (Version 1 in our example). The adjustment does not propagate to other versions of the project as the original adjustment was made when Component Adjustments were disabled.

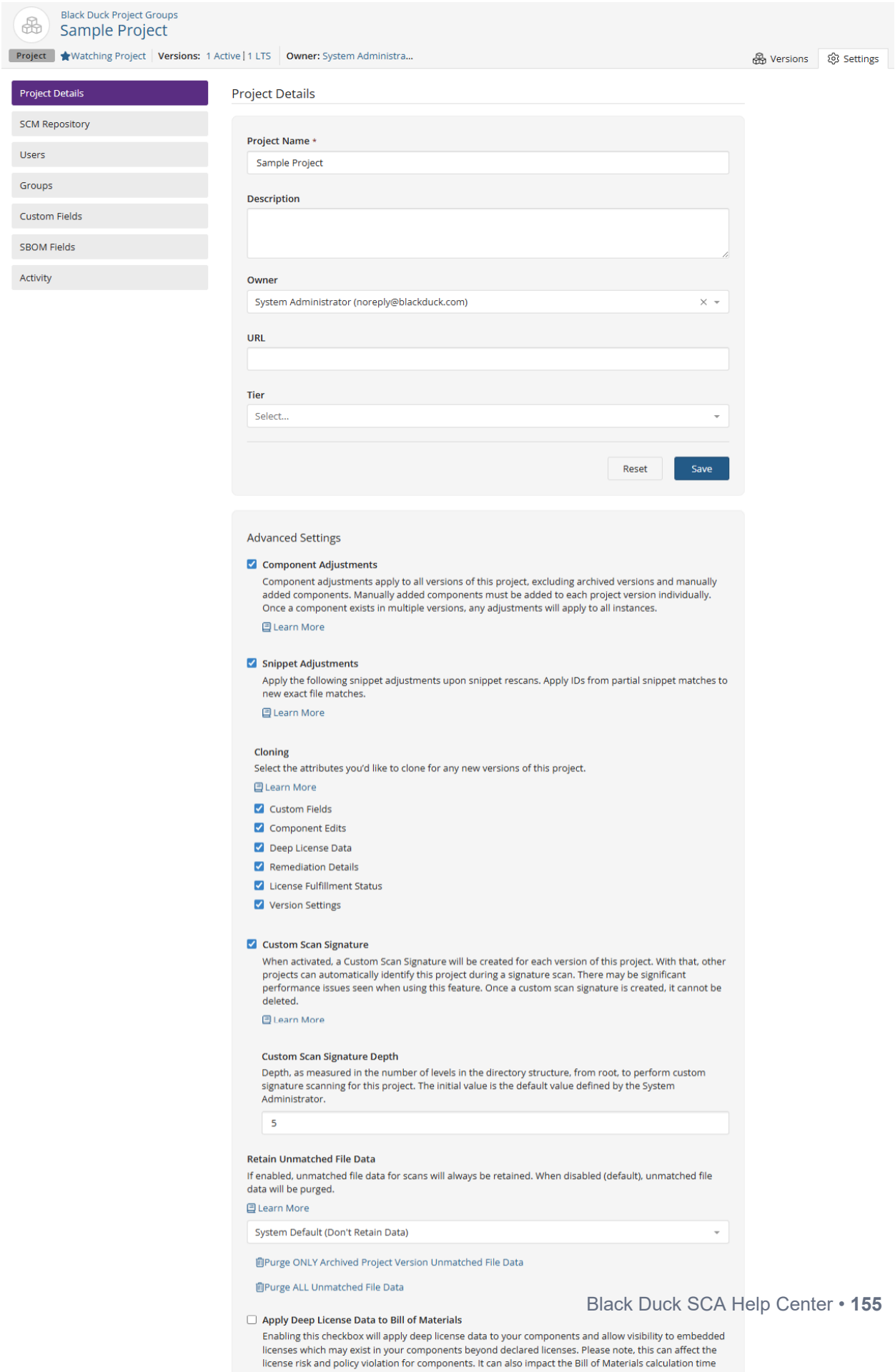
Table 10: Scenario 3

Steps	Example	Final Result
Component Adjustments are disabled.		
An adjustment is made to an item in a component in one version of the project.	The license for Component A is changed in Version 1 of the project.	The adjustment appears in only Version 1 of the project.
Component Adjustments are then enabled.		
An adjustment is made to the same item in a component in a different version of the project.	The license for Component A is changed in Version 2 of the project.	The adjustment is propagated to all versions *except* Version 1.

Enabling or disabling Component Adjustments for a project

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

3. Select the **Settings** tab.


Black Duck Project Groups
Sample Project
Project
Watching Project
Versions: 1 Active | 1 LTS
Owner: System Administra...
Versions
Settings
Project Details
SCM Repository
Users
Groups
Custom Fields
SBOM Fields
Activity
Project Details
Project Name *
Sample Project
Description
Owner
System Administrator (noreply@blackduck.com)
URL
Tier
Select...
Reset
Save
Advanced Settings
Component Adjustments
Component adjustments apply to all versions of this project, excluding archived versions and manually added components. Manually added components must be added to each project version individually. Once a component exists in multiple versions, any adjustments will apply to all instances.
Learn More
Snippet Adjustments
Apply the following snippet adjustments upon snippet rescans. Apply IDs from partial snippet matches to new exact file matches.
Learn More
Cloning
Select the attributes you'd like to clone for any new versions of this project.
Learn More
Custom Fields
Component Edits
Deep License Data
Remediation Details
License Fulfillment Status
Version Settings
Custom Scan Signature
When activated, a Custom Scan Signature will be created for each version of this project. With that, other projects can automatically identify this project during a signature scan. There may be significant performance issues seen when using this feature. Once a custom scan signature is created, it cannot be deleted.
Learn More
Custom Scan Signature Depth
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.
5
Retain Unmatched File Data
If enabled, unmatched file data for scans will always be retained. When disabled (default), unmatched file data will be purged.
Learn More
System Default (Don't Retain Data)
Purge ONLY Archived Project Version Unmatched File Data
Purge ALL Unmatched File Data
Apply Deep License Data to Bill of Materials
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

4. Check or uncheck the **Component Adjustments** check box to enable or disable persistent edits. Archived project versions and manually added components are excluded.
5. Click **Save**.

Manually adding a component to a BOM

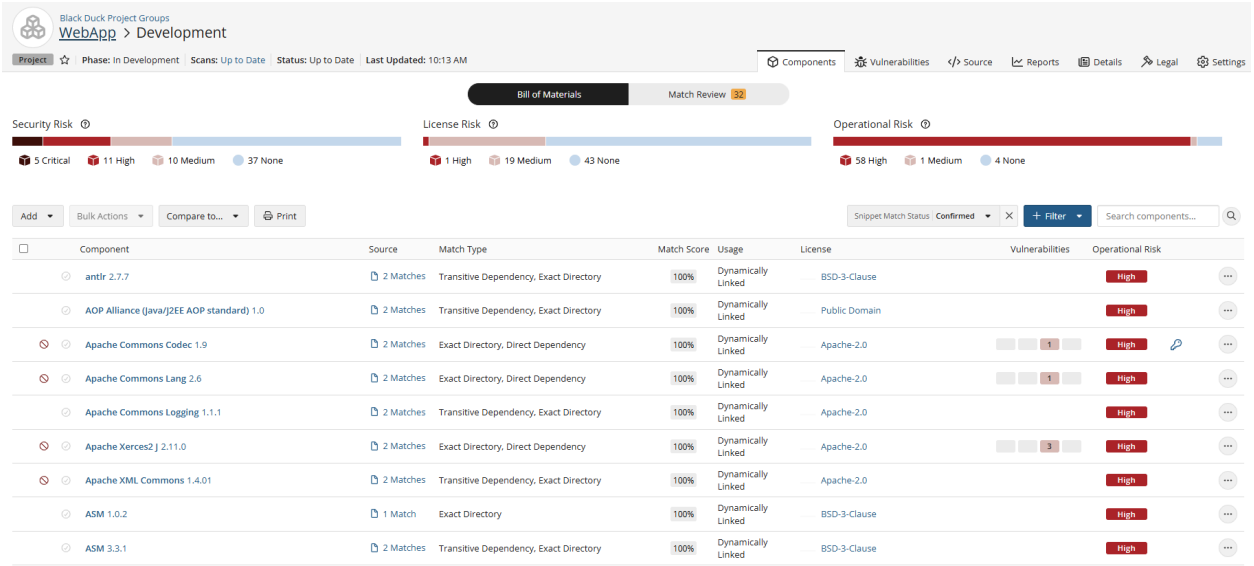
Once you have mapped a component scan to a project version, the scan results automatically populate the project version's BOM with the discovered components. Although the BOM contains all the components discovered in the mapped scan, there may be other components that you are using in that version of 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:

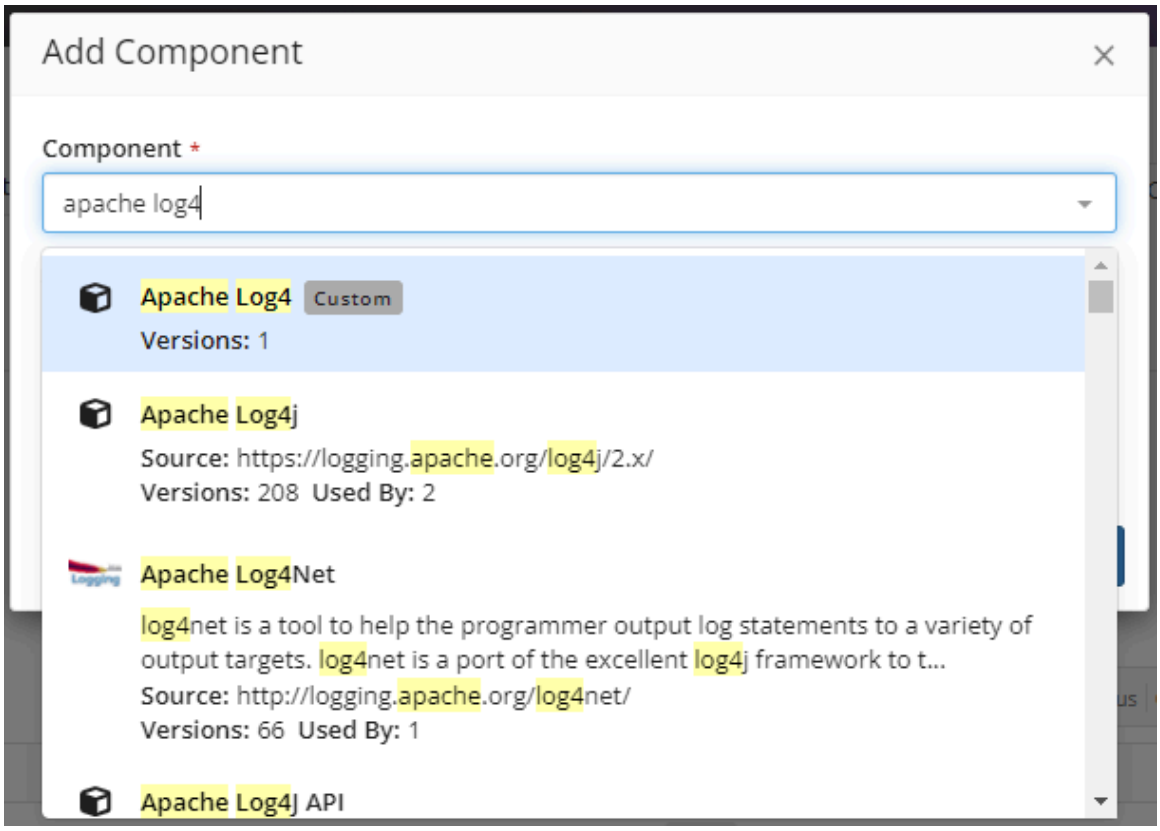
1. Log in to Black Duck SCA.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab.



4. Click **Add** and select **Component** to open the Add Component dialog box.
5. Enter the name of the component that you want to add. The search results will attempt to return possible matches based on this priority ranking:
   - Exact string matches regardless of component origin i.e. "apache log4j"
   - Custom components with fuzzy match
   - Fuzzy KnowledgeBase component match used in this or other projects

- Other fuzzy match KnowledgeBase components

Note below, when searching for "apache log4", the “Apache Log4” component appears first as this is a exact string match and is a custom component. It is then followed by Apache Log4j, the KnowledgeBase component match.



6. Optionally, enter or select a version and an origin ID.
7. Optionally, select **Advanced Attributes** and do the following:
   - Enter the purpose for adding this component.
   - Select **Modification** if you modified this component and optionally, enter information regarding the modification.
8. 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 Black Duck KB and used to update the security, license, and operational risks for this version of your project.

Manually adding an AI model to a BOM

Black Duck supports the ability to add AI models to your Bill of Materials (BOM), allowing for comprehensive tracking and management of AI components within your projects. This feature enables organizations to maintain visibility into the AI models they use, monitor associated risks, and ensure compliance with relevant policies.

This section will walk you through the process of adding AI models to your BOM, configuring their properties, and managing them effectively within your project ecosystem.

Prerequisites

Before adding AI models to your BOM, ensure the following requirements are met:

- **Feature Activation**

  The AI Model Scanning feature must be enabled on your product registration key. If you do not see this functionality in your Black Duck instance, please contact your Black Duck representative or support team to have this feature activated for your organization.

- **User Permissions**

  To add and manage AI models in a BOM, you must have appropriate permissions, including:

  - Project access with editing capabilities
  - BOM Management permissions

- **Scan Configuration**

  You must have performed a Detect scan using the following parameter:

  ```
  --detect.blackduck.signature.scanner.individual.file.matching=ALL
  ```

Adding an AI model to a BOM

Adding AI models to your Bill of Materials follows a similar workflow to manually adding components. This streamlined process ensures consistency in your BOM management practices.

1. **Navigate to the Project Version**
   - Access your project list from the main dashboard.
   - Select the specific project and version where you want to add the AI model.
2. **Access the BOM**
   - Click on the **Components** tab to view the current Bill of Materials.
3. **Initiate the Add Process**
   - Locate and click the **Add** button near the top of the BOM view.
   - From the dropdown menu, select **AI Model**.
4. **Enter AI Model Details**
   - In the dialog that appears, provide the following information:
     - **AI Model**: Enter the name of the AI model.
     - **Commit**: Specify the commit version of the AI model.
   - Optionally, provide additional details by expanding the **Advanced Attributes** section:
     - Enter the **Purpose** for adding this component.

- Select **Modification** if you modified this component and optionally, enter information regarding the modification.

5. **Confirm and Save**
   - Review the entered information
   - Click "Add" or "Save" to add the AI model to your BOM

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.

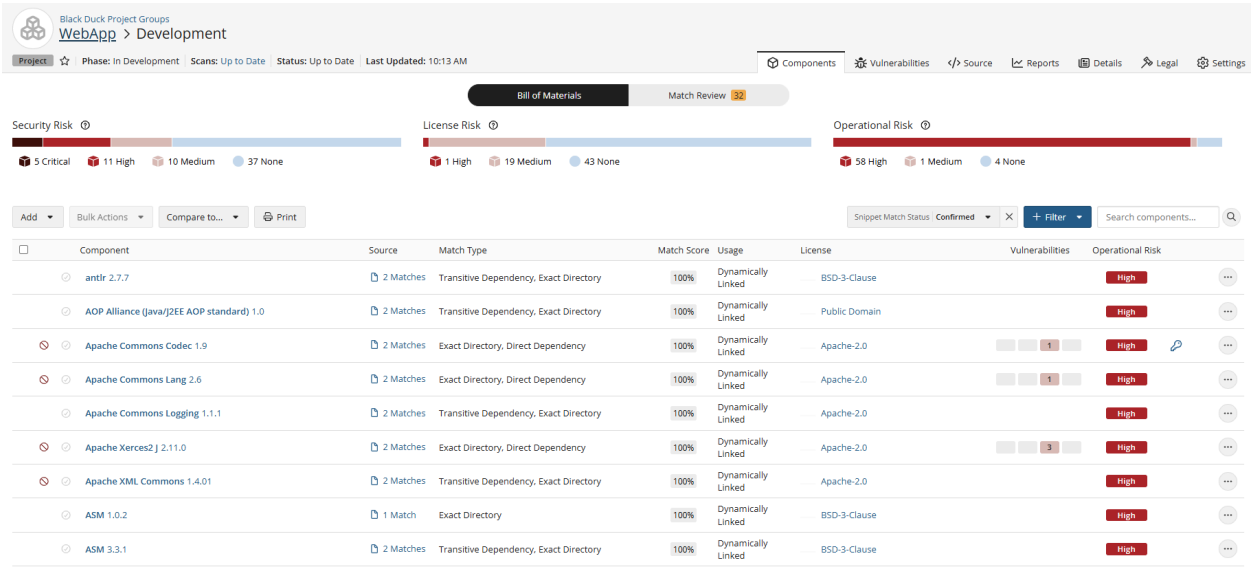
You can change a component's usage to indicate that it is not included in the project version's BOM because it is not actually being distributed with the released project version. For example, if scanning identified development tools in scanned code or a Docker image mapped to the project version, but 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 SCA.
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:

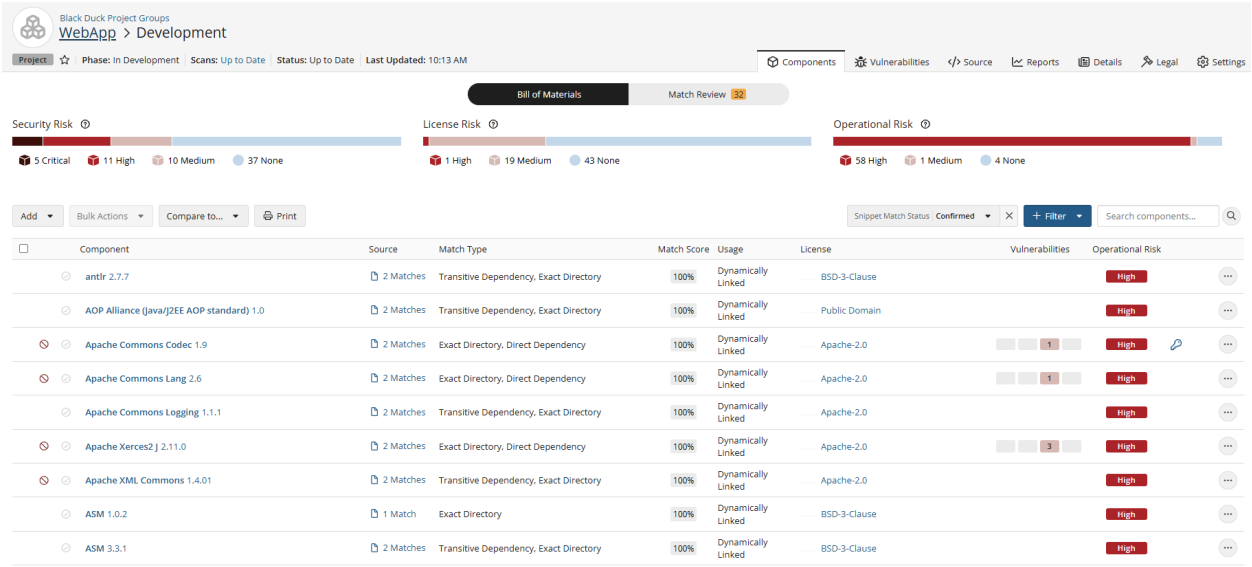
- The same component was discovered in a later component scan and automatically added to the BOM.
- The component version that you selected when you added it was not the correct version.
- You are no longer using component in that project version.

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 SCA.
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.

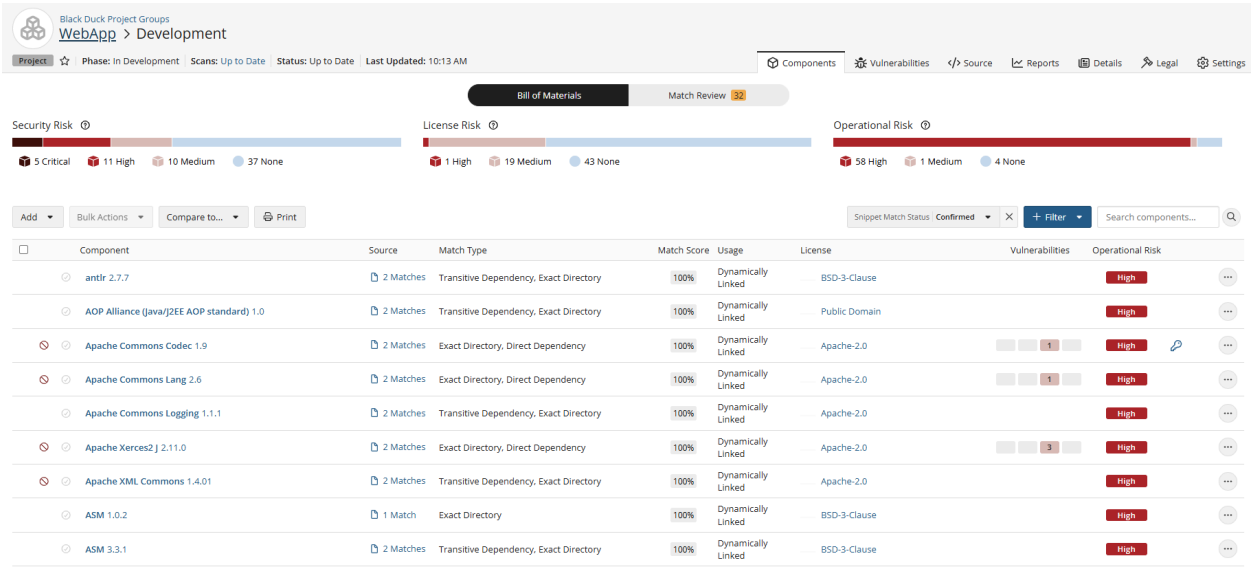
Removing components from a BOM

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.

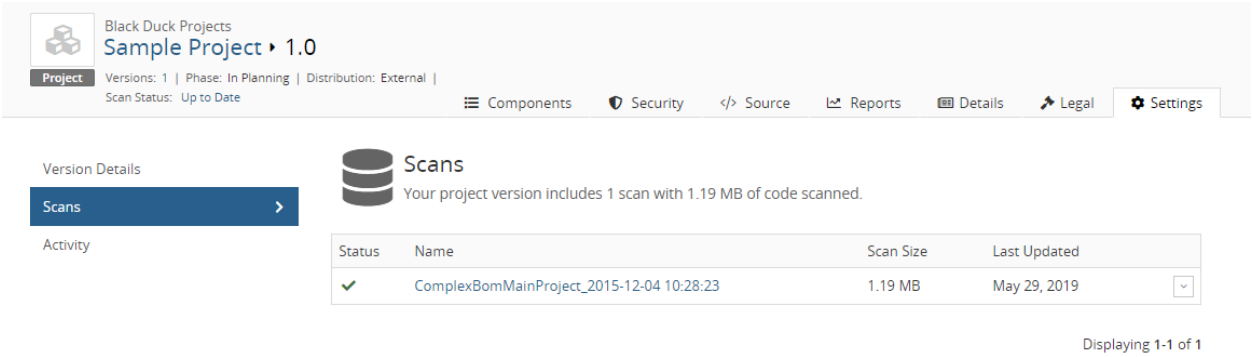
To remove a scan from a project version to update the BOM:

1. Log in to Black Duck SCA.
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 **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.



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 can 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 choose to apply persistent edits, ignoring a component will apply to all versions of the project.

When you ignore an automatically-added OSS component from a project version BOM, it will be moved to the Match Review page for further action. This ensures that users can review and address these components later. Notably, ignored components will continue to be excluded from risk calculations even if the code where they were discovered is rescanned to update the BOM.

Note: You cannot ignore manually added components.

How to ignore one or more components in a BOM

To ignore a single component in a BOM:

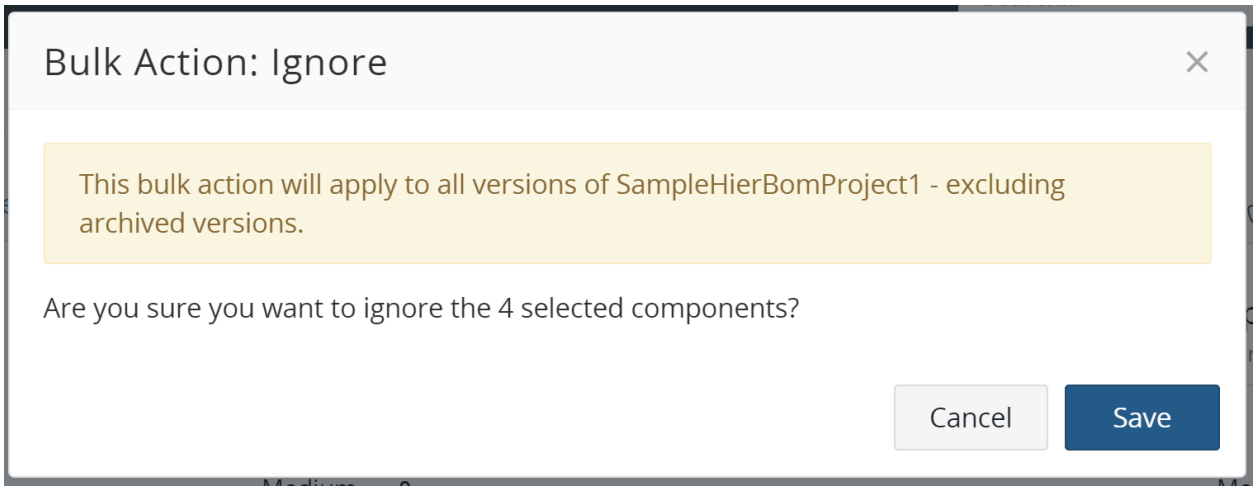
1. Log in to Black Duck SCA.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab.
4. Click and select **Ignore** to open the Ignore Component dialog box.
5. Click **Ignore**.

   The component is relocated to the Match Review page, will not be included in the risk calculations for the project version, and will not be displayed in the BOM.

To ignore multiple components in a project version BOM:

1. Display the project version BOM. Ensure you are in the Components view.
2. Check the box next to any number of components.
3. Click the **Bulk Actions** button.
4. Select **Ignore**.

   The **Bulk Action: Ignore** dialog box appears.



5. Click the **Save** button to perform the action or the **Cancel** button to exit the dialog box.

How to unignore one or more components

To unignore a single component in a BOM:

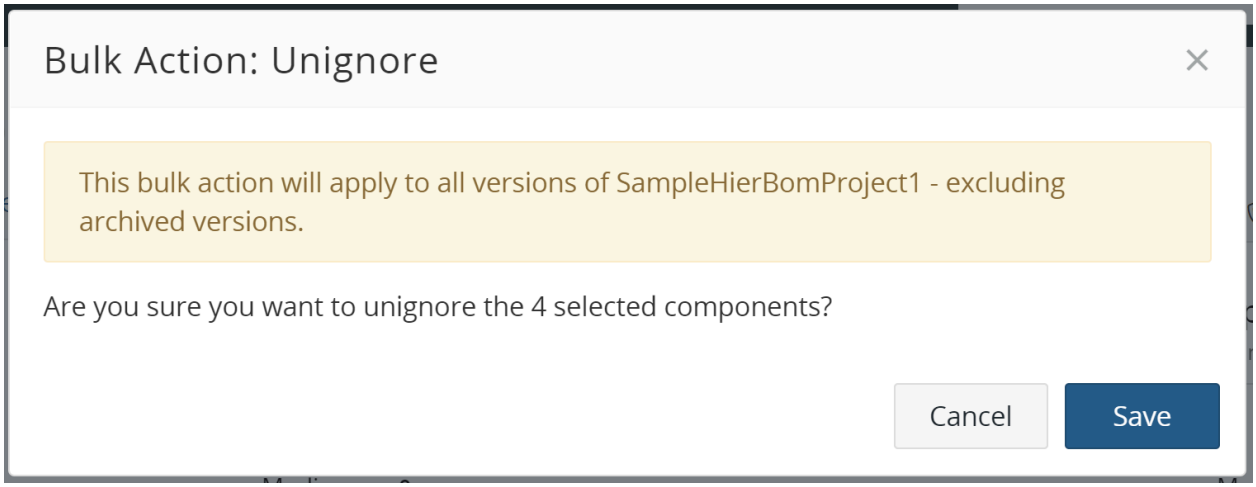
1. Log in to Black Duck SCA.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to open the **Components** tab.
4. Click **Match Review**.

5. Click ••• and select **Ignore** to open the Ignore Component dialog box.
6. Click **Ignore**.
   The component is relocated to the **Bill of Materials** page.

To unignore multiple components in a project version BOM:

1. Display the project version BOM. Ensure you are in the Components view.
2. Click **Match Review**.
3. Check the box next to any number of components.
4. Click the **Bulk Actions** button.
5. Select **Unignore**.

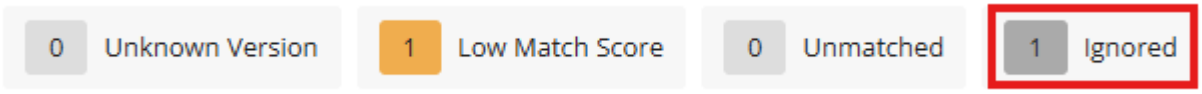
   The **Bulk Action: Unignore** dialog box appears.



How to view ignored components

To view ignored components:

1. While in the **Components** tab of the project version BOM, click **Match Review**.
2. Select **Ignored** from the filters list.



   The table displays all ignored components.

Adjusting the component and/or component version in a BOM

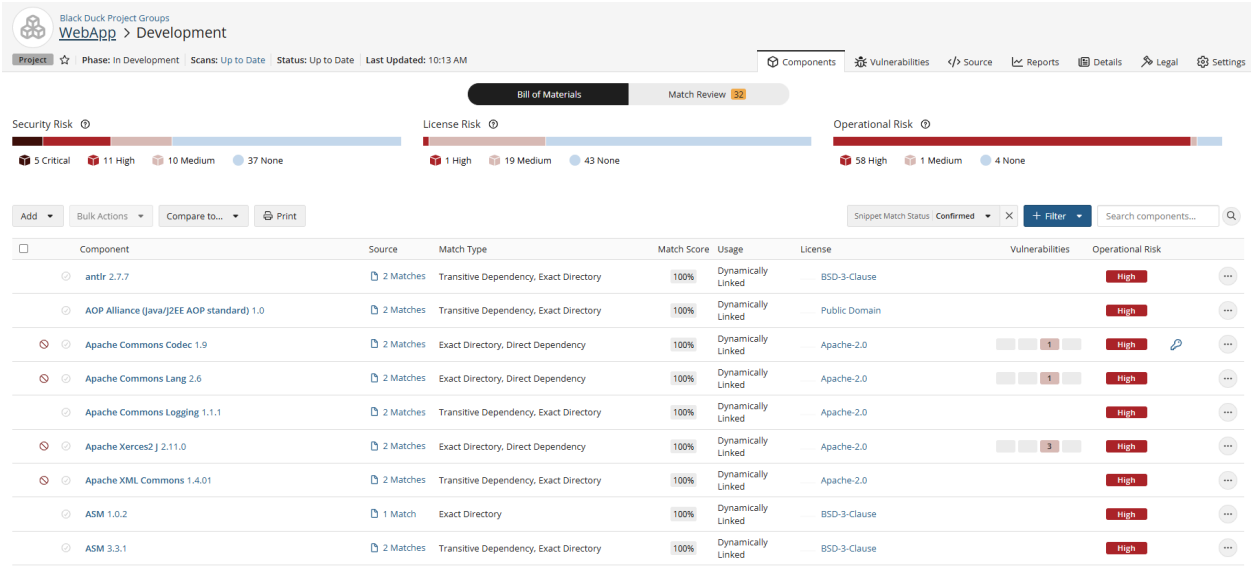
Once you have mapped a component scan to a project version, the scan results automatically create the project version's BOM. Although component scanning automatically discovers the open source component and component version from most archive files by comparing them to components in Black Duck KB, you may be using a version of the component that is not available in 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 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 Black Duck KB, users with the Component Manager role can create a custom version and add it to the BOM.

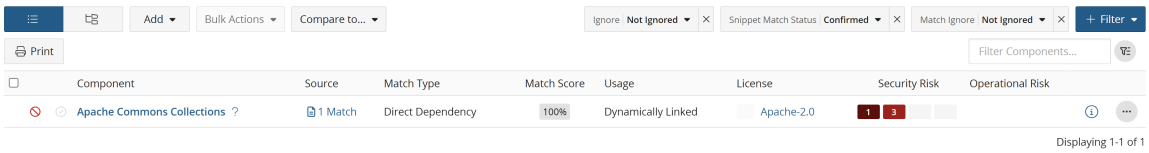
To select an alternate component and/or version match for a component in a BOM:

1. Log in to Black Duck SCA.
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 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:

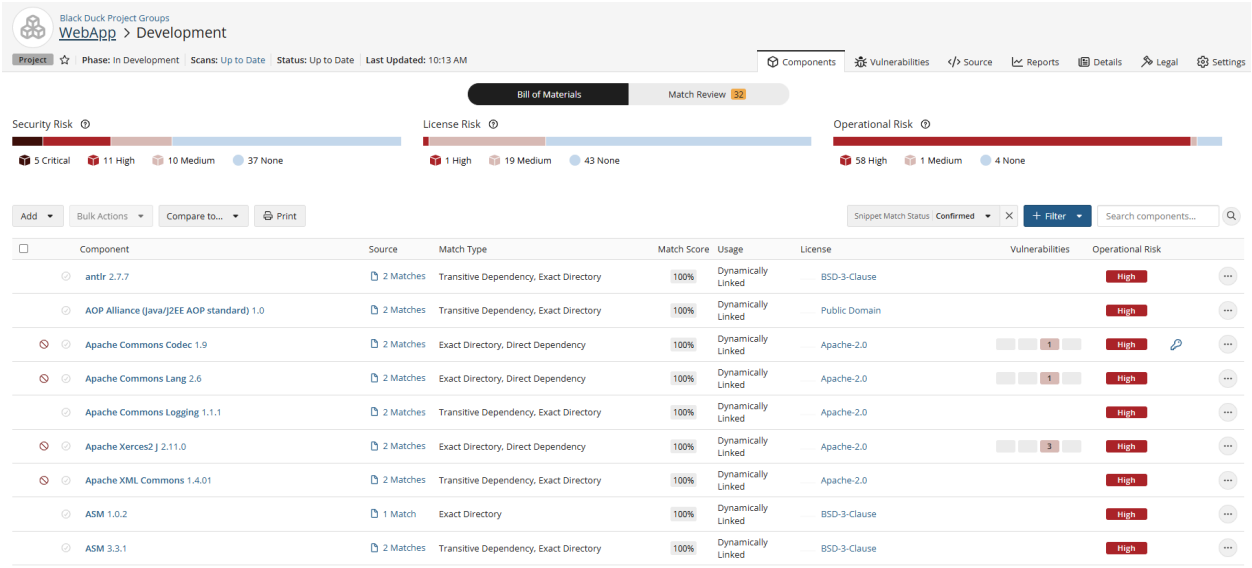


Editing an origin or origin ID

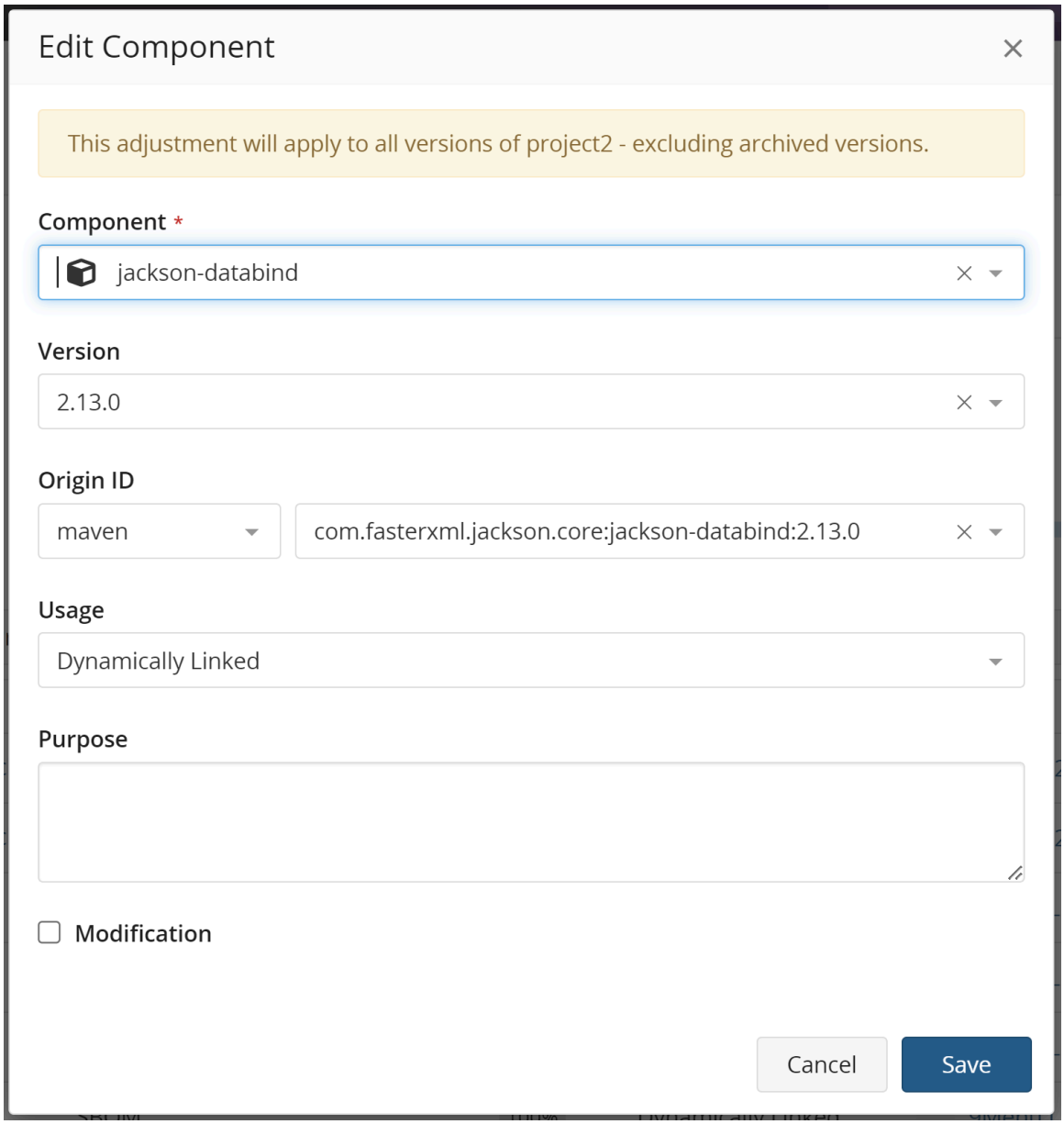
You can select a different origin or origin ID shown for a Linux distribution and used in a project version's BOM.

To select a different origin or origin ID:

1. Log in to Black Duck SCA.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Select the version name to display 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. 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.
6. 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.
7. Optionally, enter a purpose for this adjustment and/or select the **Modification** checkbox and enter information regarding this modification in the field.
8. 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.

Editing the component usage type

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

The usage statuses are:

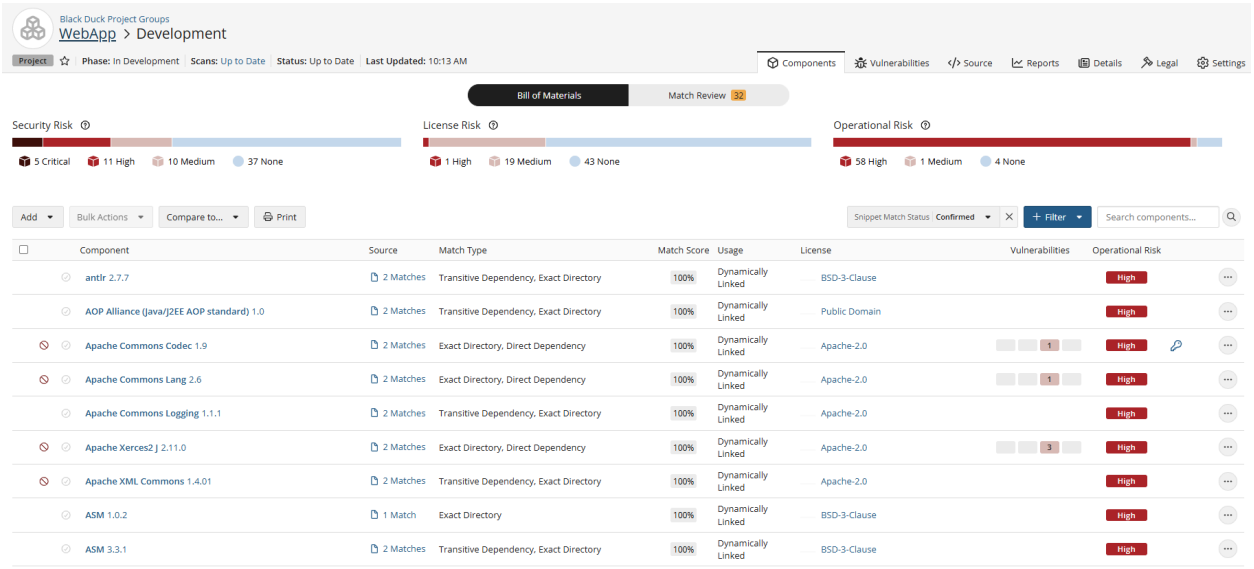
- Dynamically Linked
- Statically Linked
- Source Code
- Separate Work
- Implementation of Standard
- Merely Aggregated
- Prerequisite
- Dev. Tool / Excluded
- Unspecified

Click here for more information on usage.

Note: It is not possible to edit the usage type for binaries and snippets through the Source view. Usage can only be edited via the BOM component.

To change a component's usage type:

1. Log into 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.

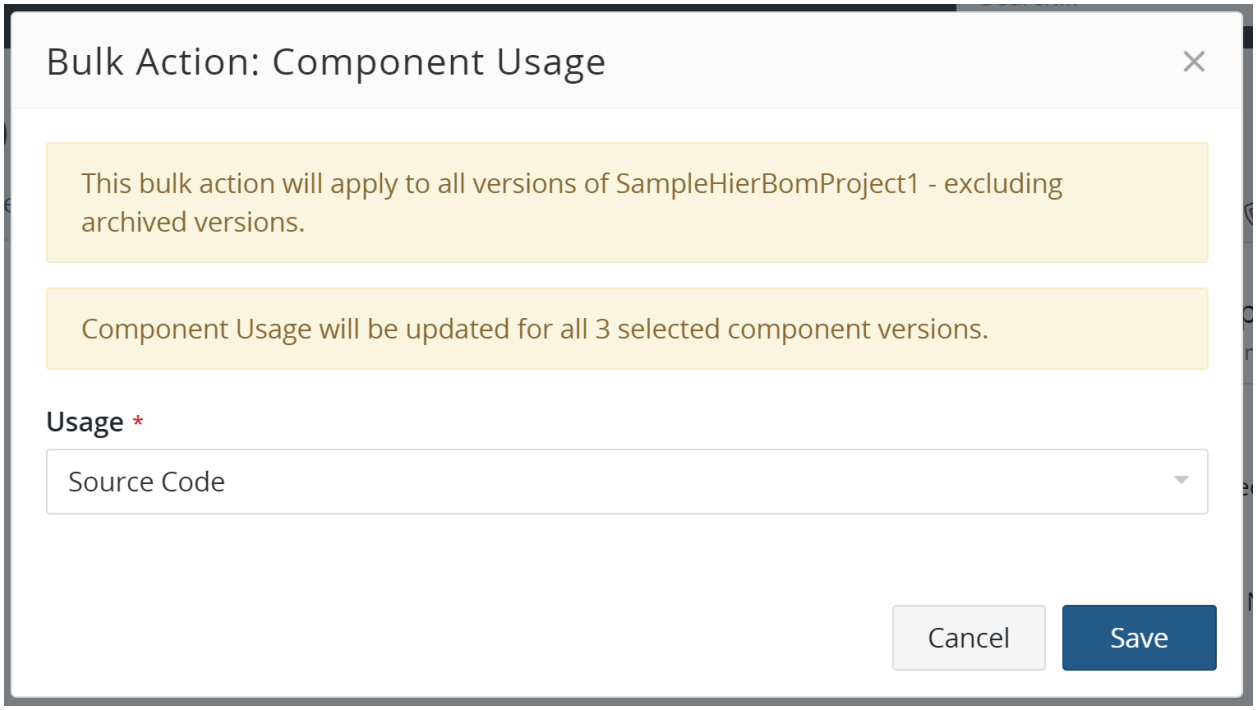


4. In the component list view of the BOM, click and select **Edit** to open the Edit Component dialog box.
5. Select any of the options from the **Usage** list.
6. Click **Save**.

To change multiple component's usage type:

1. Display the project version BOM. Ensure you are in the Components view.
2. Check the box next to any number of components.
3. Click the **Bulk Actions** button.
4. Select **Component Usage**.

   The **Bulk Action: Component Usage** dialog box appears.



5. Select any of the options from the **Usage** list.
6. Click **Save**.

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 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

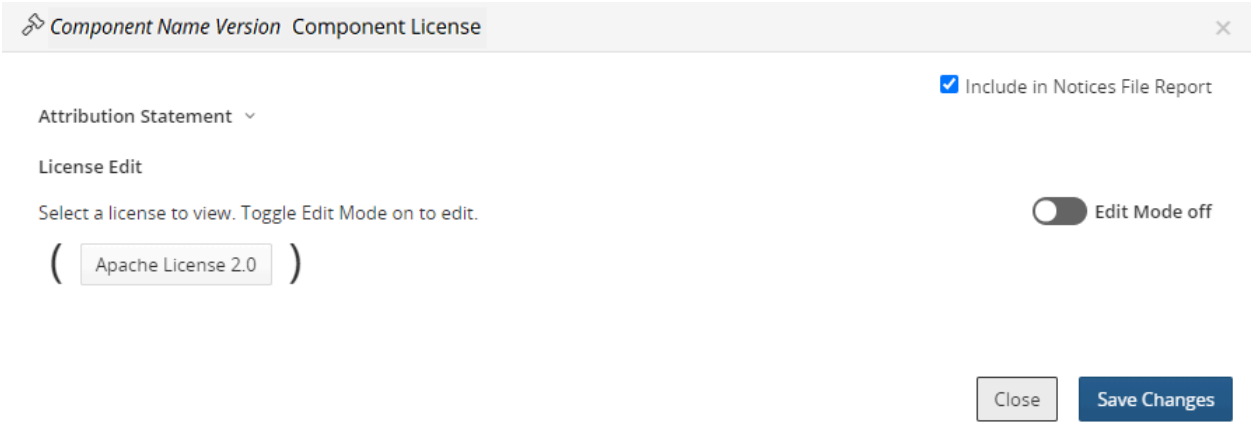
If you have modified a license, you can select to revert it back to the license as defined by Black Duck KnowledgeBase.

Note: License adjustments can be made at both the project and project version levels. This behavior is controlled by the **Component Adjustments** project setting, which also governs other component adjustments.

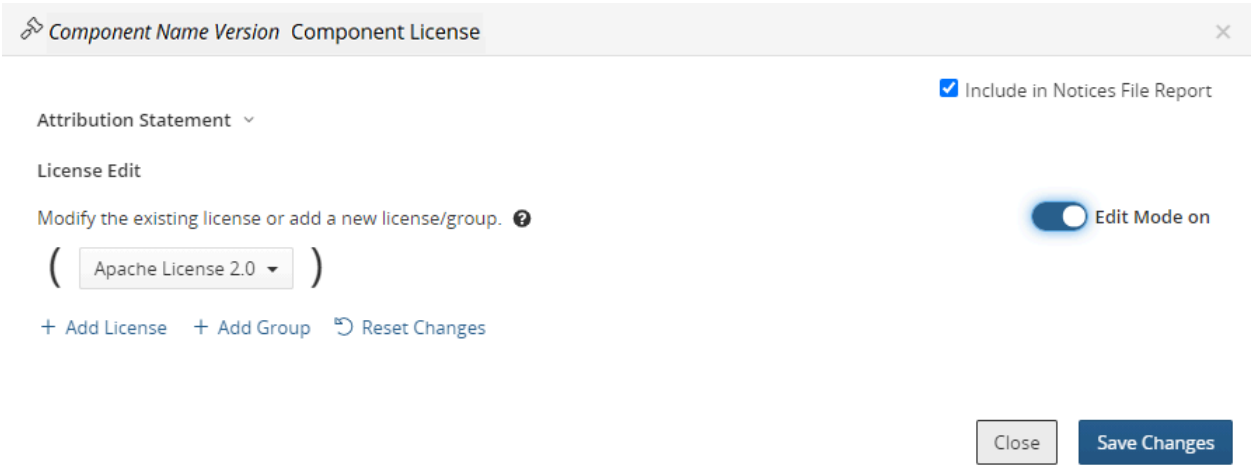
To modify licenses:

1. Log in to Black Duck SCA.

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.

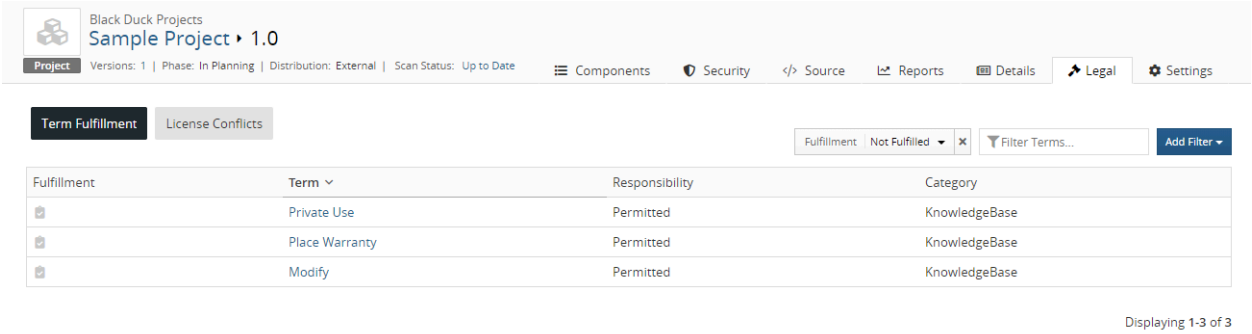
Selecting the license term fulfillment status

Once a License Manager user defines the required license terms, and a System Administrator enables the Term Fulfillment setting, BOM Manager users and other authorized users can indicate the fulfillment status of 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::

1. From a project version BOM, select the **Legal** tab, and if necessary, the **Term Fullfilment** tab, to view a list of license terms that require fulfillment.



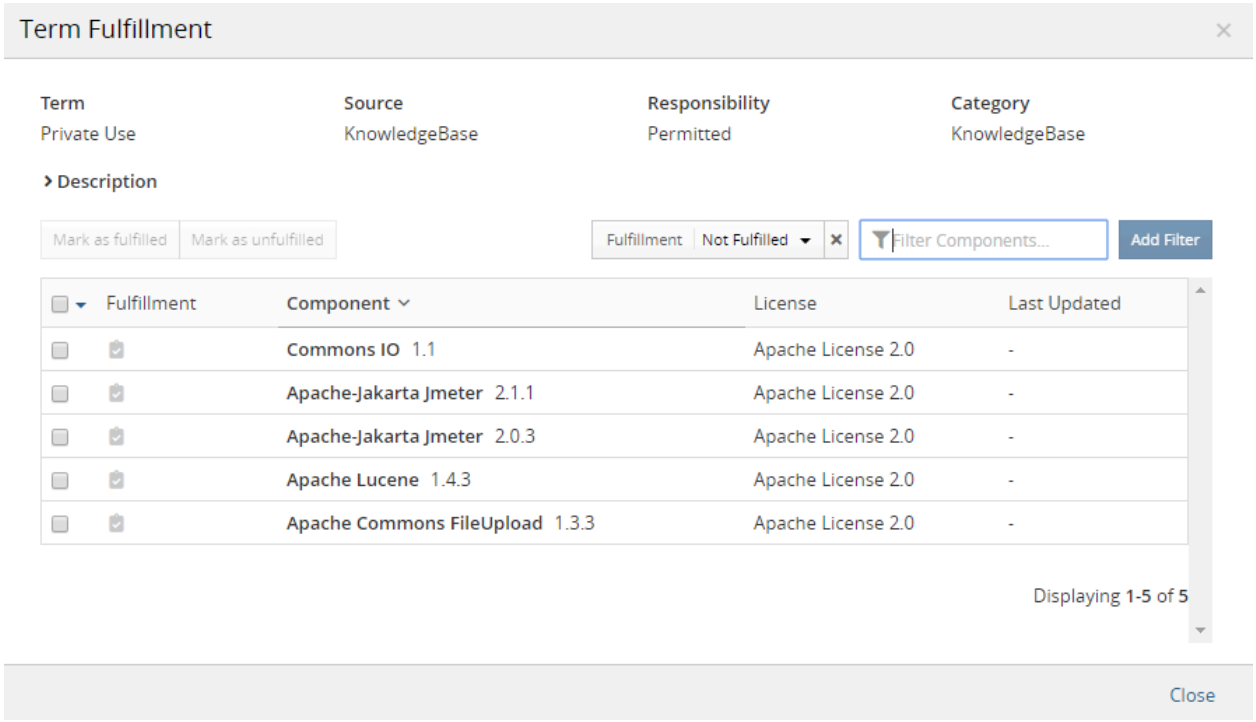
By default, the **Legal** tab is filtered to show all license terms that are not fulfilled.

The tab displays the following information:

Column	Description
Fulfillment	Indicates fulfillment status: • 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.

2. 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**.

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.

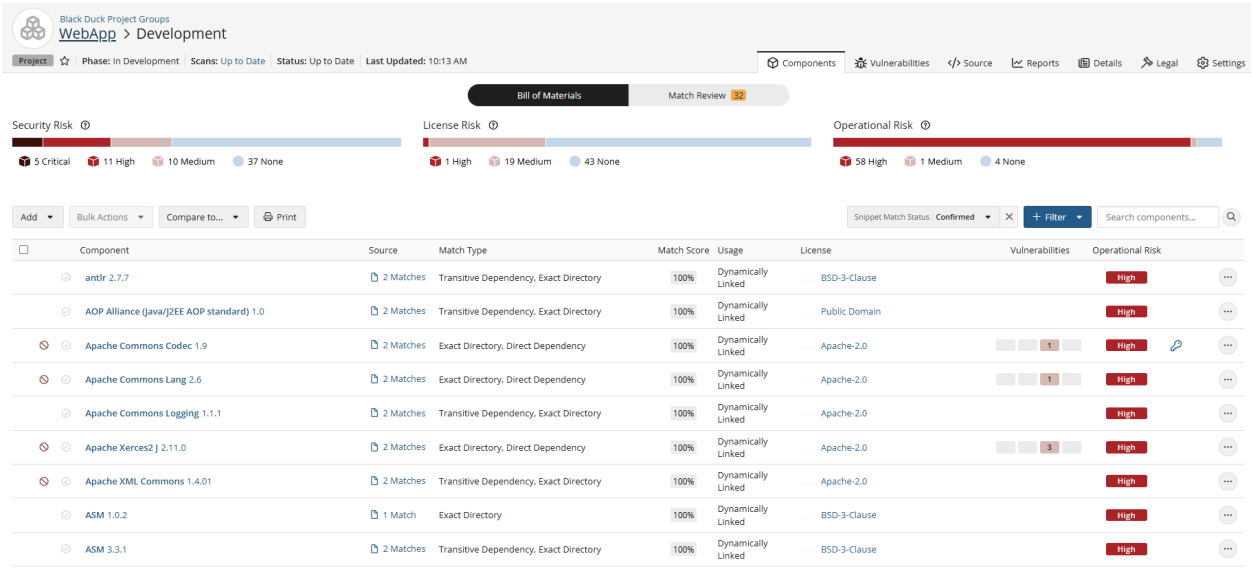
  An exception to this is the subproject version license: edits made to the project version license may or may not propagate to the license shown for the subproject in the BOM:

  - 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.
- Subproject security risks are added to the sum in the parent project.
- You cannot add a Project Version to itself. However, you can add a Project Version to another version of the same project, provided that this addition does not create a dependency cycle.

To add a subproject:

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. The license selected for the subproject will determine its license risk.
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:

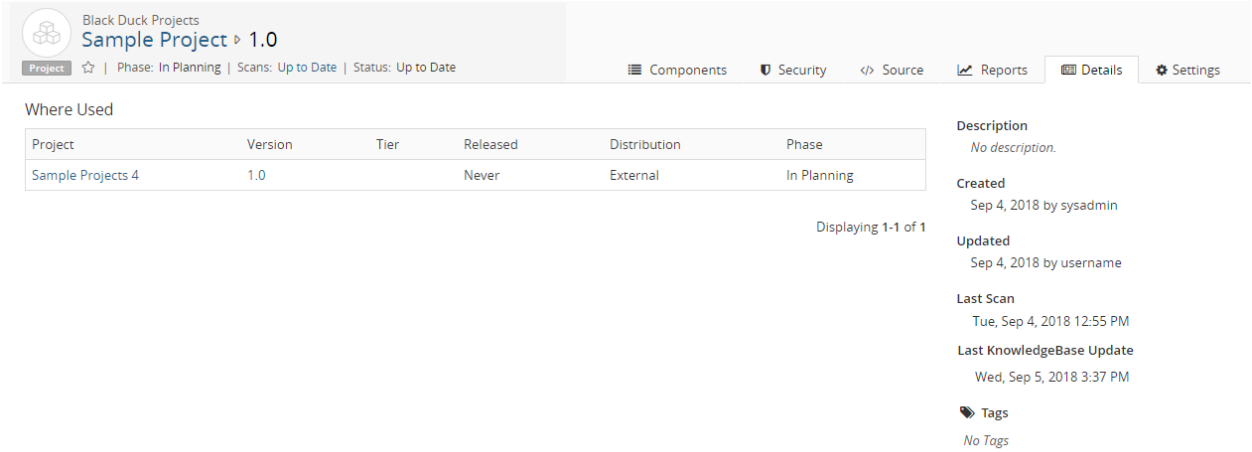
1. Select the BOM as described in the previous section.
2. Click and select **Delete** to open the Delete Component dialog box.
3. Click **Delete**.

   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 **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.

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:

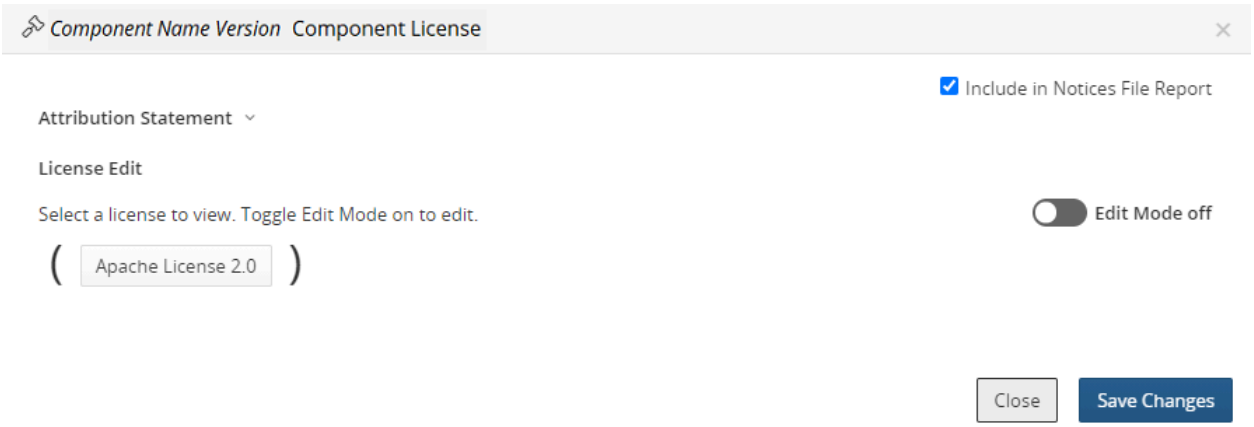
- Edits to license text only apply to the license text for that component version: edits do not apply to other components with the same license.
- If you selected to make edits persistent then edits to license text apply to all existing versions of a project and will also be carried forward as additional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

  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 or Project Manager has edited the license,

To edit license text:

1. Log in to Black Duck SCA.
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.



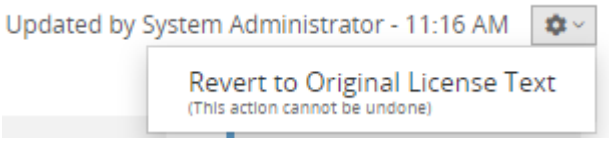
5. Select the license you wish to edit.

   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**.



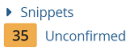
Reviewing snippet matches

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

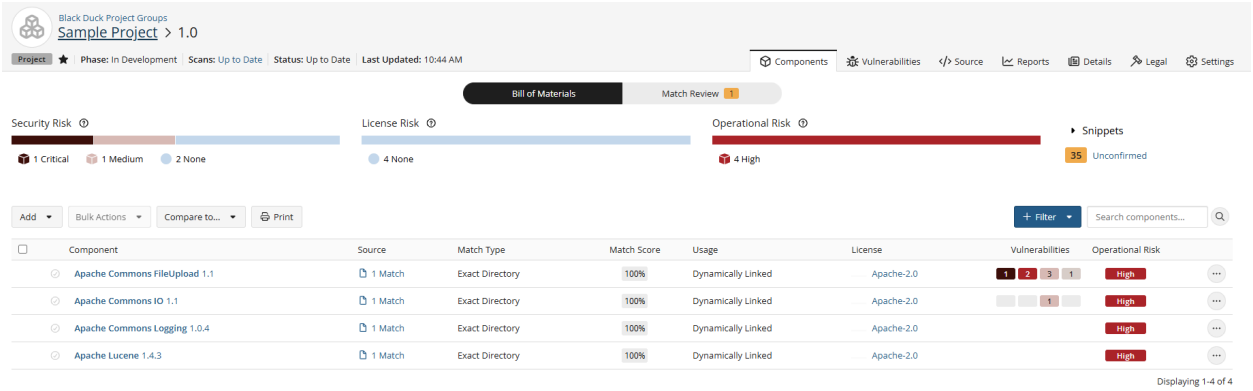
Click here for more information on using the **Source** tab.

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.



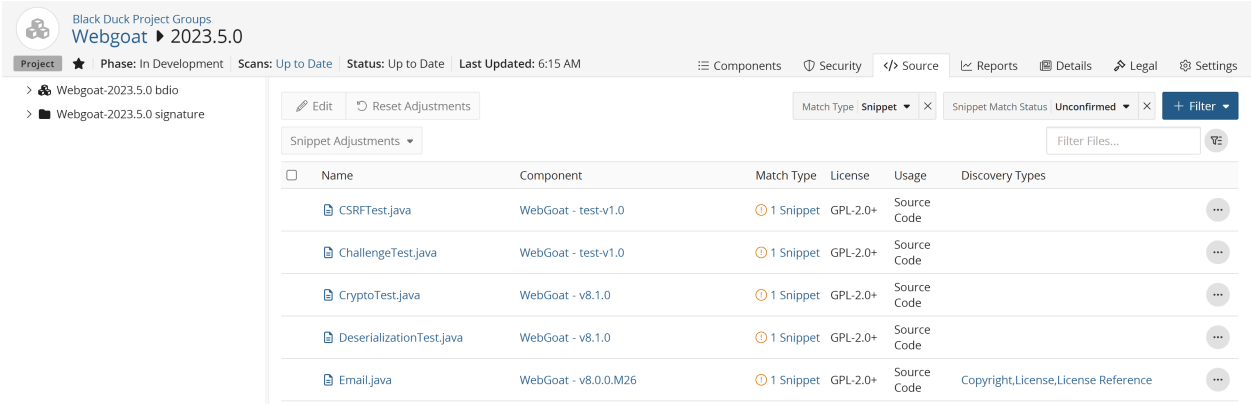
Retaining partial snippet identifications

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**.

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.

- 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. Clicking the link opens the Snippet View.

- 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.



Confirming, ignoring, and editing snippet matches

To confirm/unconfirm a snippet match:

1. Check the box next to the file.
2. Click Snippet Adjustments.

3. Select **Confirm Match** or **Undo Confirmation**.

To ignore/unignore a snippet match:

1. Check the box next to the file.
2. Click Snippet Adjustments.
3. Select **Ignore Match** or **Unignore Match**.

To edit a snippet match:

1. Check the box next to the file and then click Edit, or;

   Click the ••• button at the end of the file's row and select Edit.
2. Use this dialog box to modify the component, version, or origin ID.
3. Select **Adjust Snippets and Confirm** which adjusts and automatically confirms the snippet match.
4. Click **Update**.

Bulk confirming, ignoring, and editing snippet matches

Bulk confirming or ignoring snippet matches works similarly to process of confirming or ignoring individual snippet matches described above.

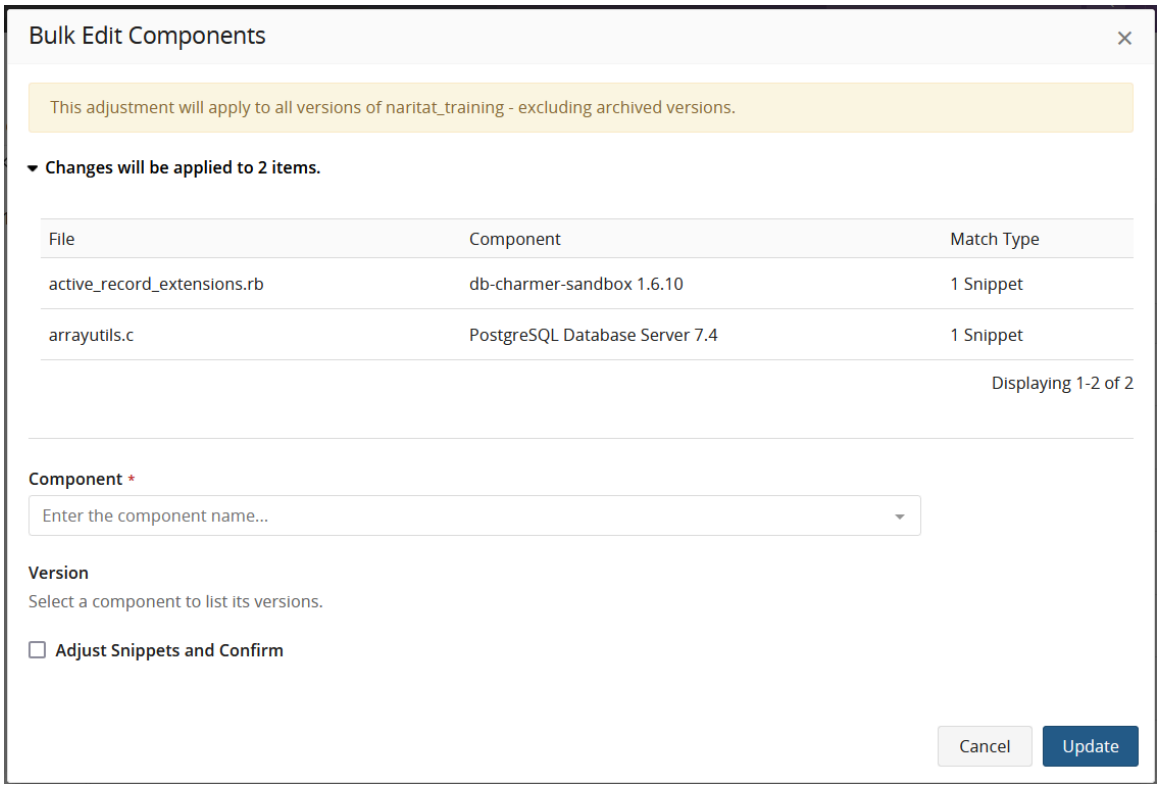
To bulk confirm or ignore multiple snippet matches:

1. Check the box next to the desired snippet matches, or the box next to the **Name** column header to select all snippet matches.
2. Click Snippet Adjustments.
3. Select **Confirm Match**, **Undo Confirmation**, **Ignore Match**, or **Unignore Match**, based on your intended action.

To bulk edit snippet matches:

1. Check the box next to the desired snippet matches, or the box next to the **Name** column header to select all snippet matches.
2. Click Edit.

   The Bulk Edit Components dialog box appears.

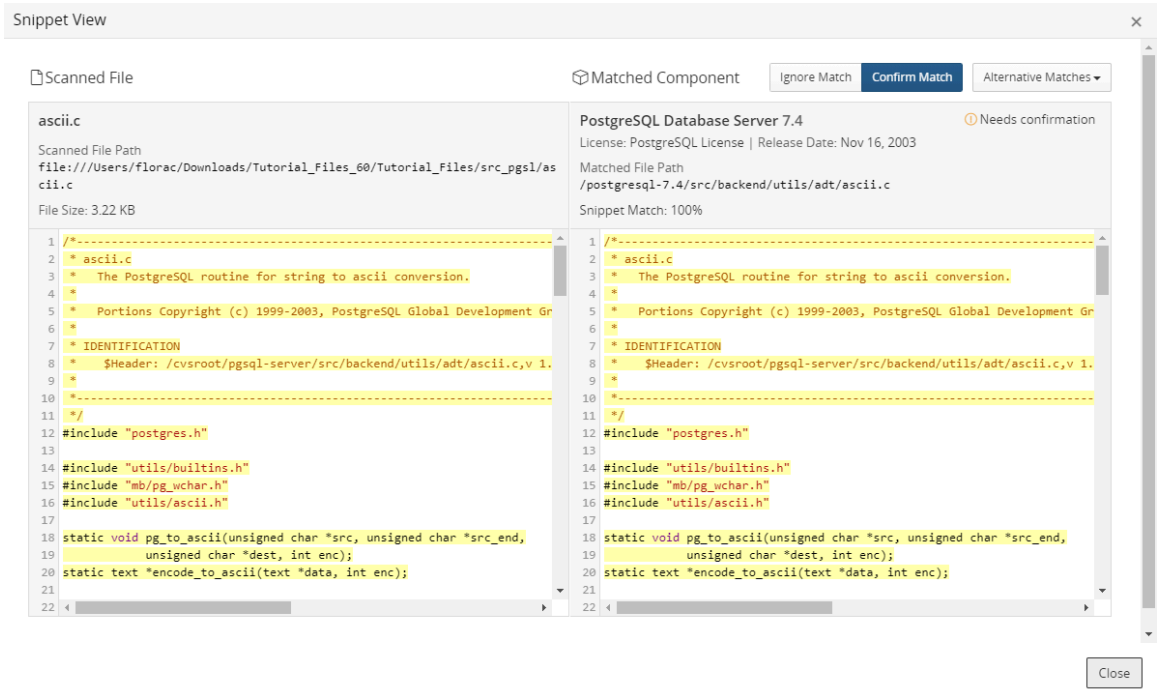


3. Use this dialog box to modify the component or version.
4. Select **Adjust Snippets and Confirm** which adjusts and automatically confirms the snippet match.
5. Click **Update**.

Using the Snippet View

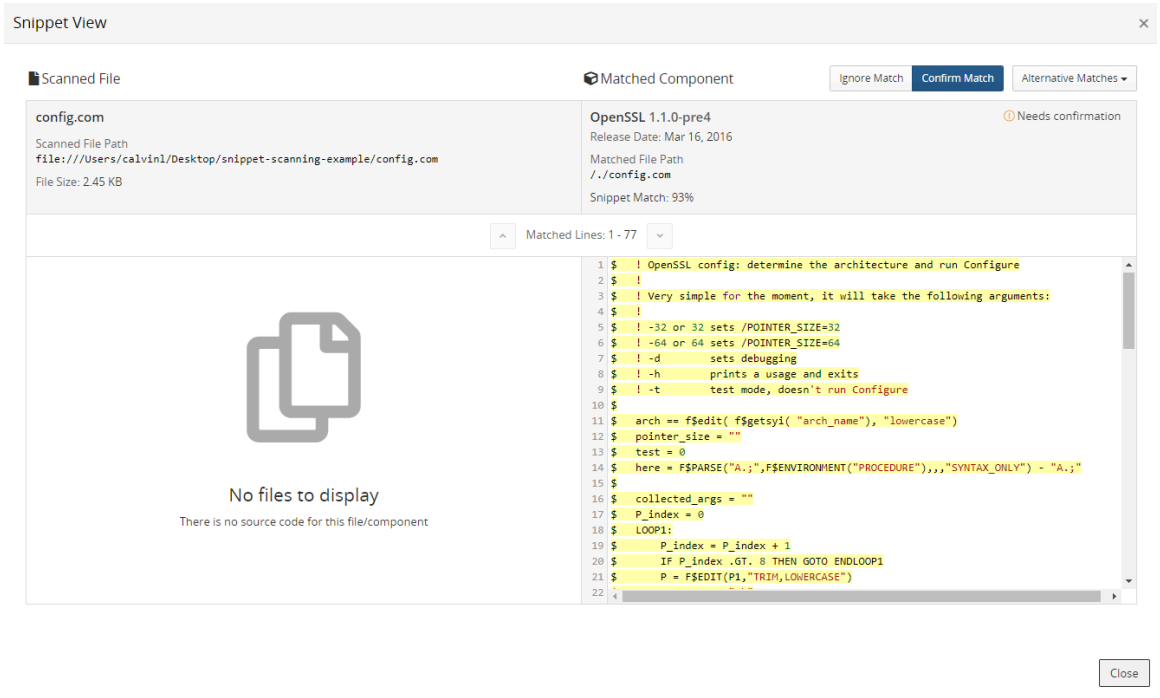
Clicking **# snippet** displays the Snippet View. The information shown here depends on whether you uploaded source files during the snippet scan.

- If you uploaded source files, the Snippet View displays the source file on the left pane and the matched component on the right pane:



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:



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**
  - **Undo Confirmation**
  - **Ignore Match**
  - **Unignore Match**

Reviewing a snippet match in the Snippet View

1. In the **Source** tab, select **# snippet** in the **Match Type** column for the snippet match you wish to review.

   Select one of these options:
   - **Confirm**. If the snippet match has not been confirmed.
   - **Undo Confirmation**. If the snippet match has been confirmed and you want to unconfirm it.
   - **Ignore Match**. If the snippet match has not been ignored.
   - **Unignore Match**. If the snippet match has been ignored and you want to unignore it.
2. 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**.

Managing comments in a BOM

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 can be persisted to a cloned project if the **Component Edits** option is enabled in the Cloning section of the project **Settings** tab.
- 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.

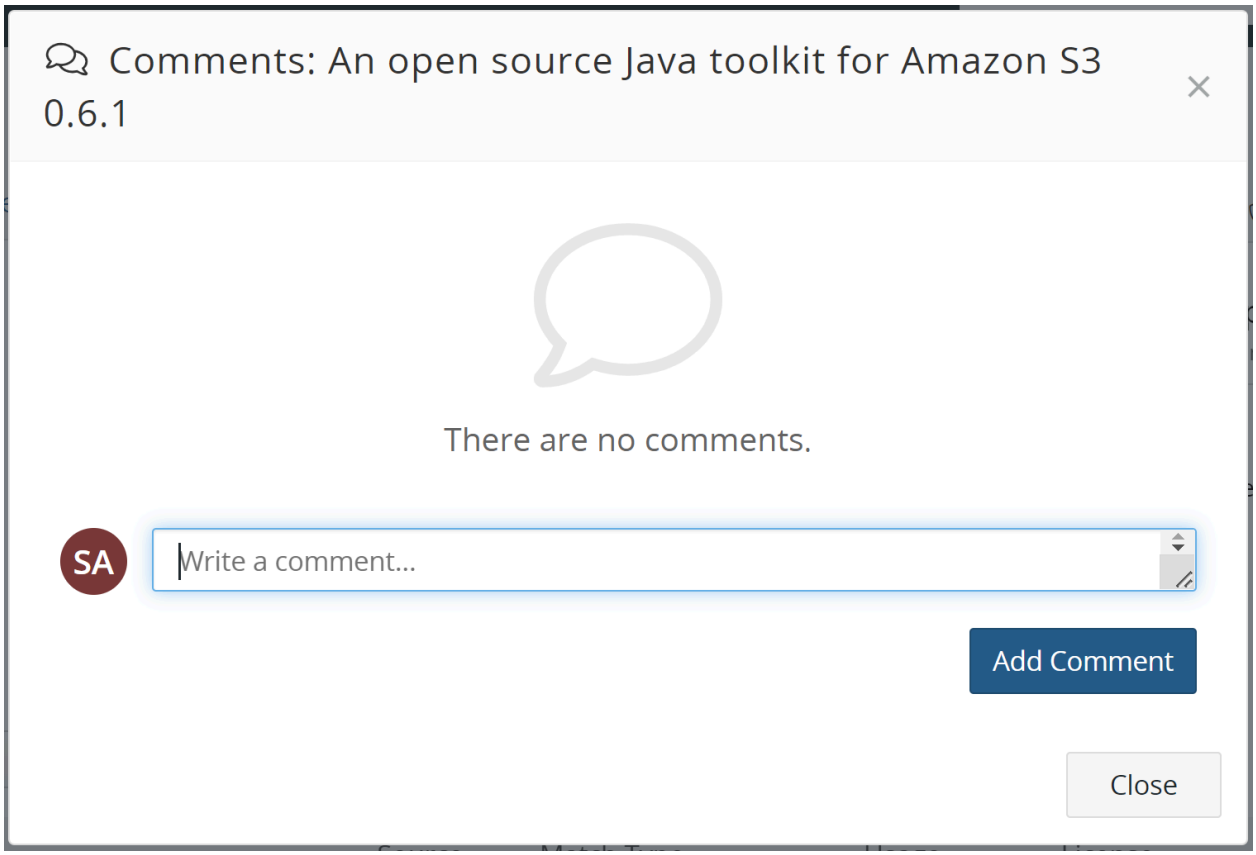
Adding a comment in the Components view

Adding comments to components makes your feedback more clear, giving everyone the ability to discuss any actions to be taken on a particular component. You can add a comment to a single comment, or you can comment on a number components by using the Bulk Actions button.

To comment on a single component:

1. Display the project version BOM. Ensure you are in the Components view.
2. Click in the row where you want to add a comment and select **Comments**.

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

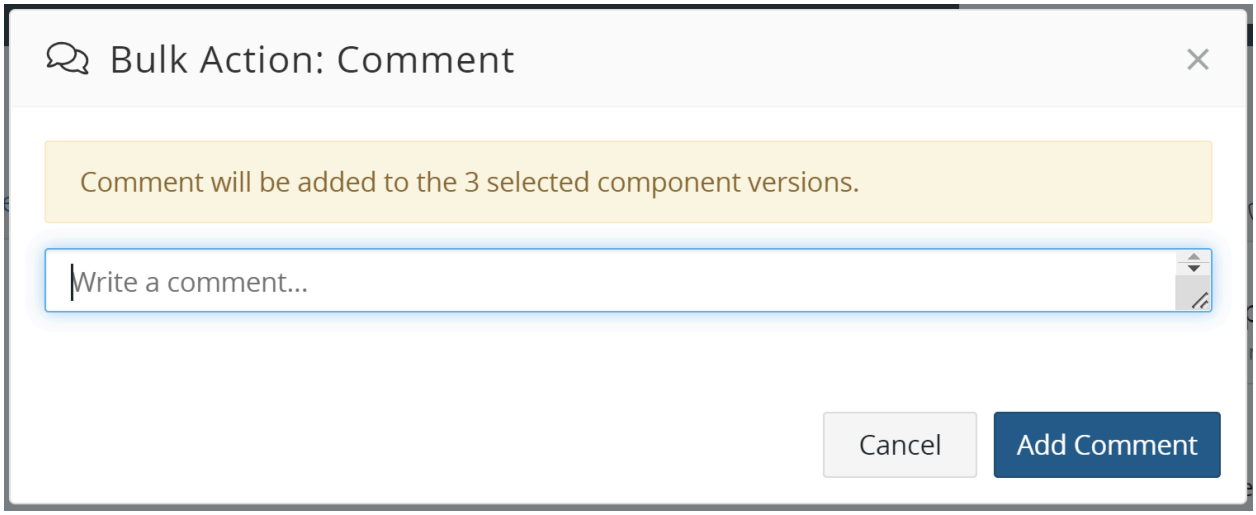


3. Enter the comment and click **Add Comment**.

To comment on multiple components simultaneously:

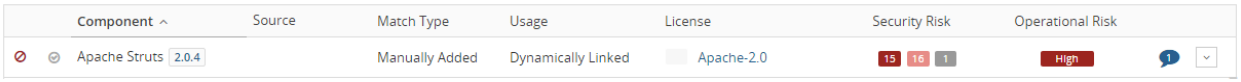
1. Display the project version BOM. Ensure you are in the Components view.
2. Check the box next to any number of components.
3. Click the **Bulk Actions** button.
4. Select **Comment**.

   The **Bulk Action: Comment** dialog box appears.



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

After either of the actions above, a comment icon (2) 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.



Viewing a comment

Click 2 in the row where you want to view a comment.

Editing a comment

Only the original writer can edit their comment.

1. Click ⌄ or 2 (if there are already comments) 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

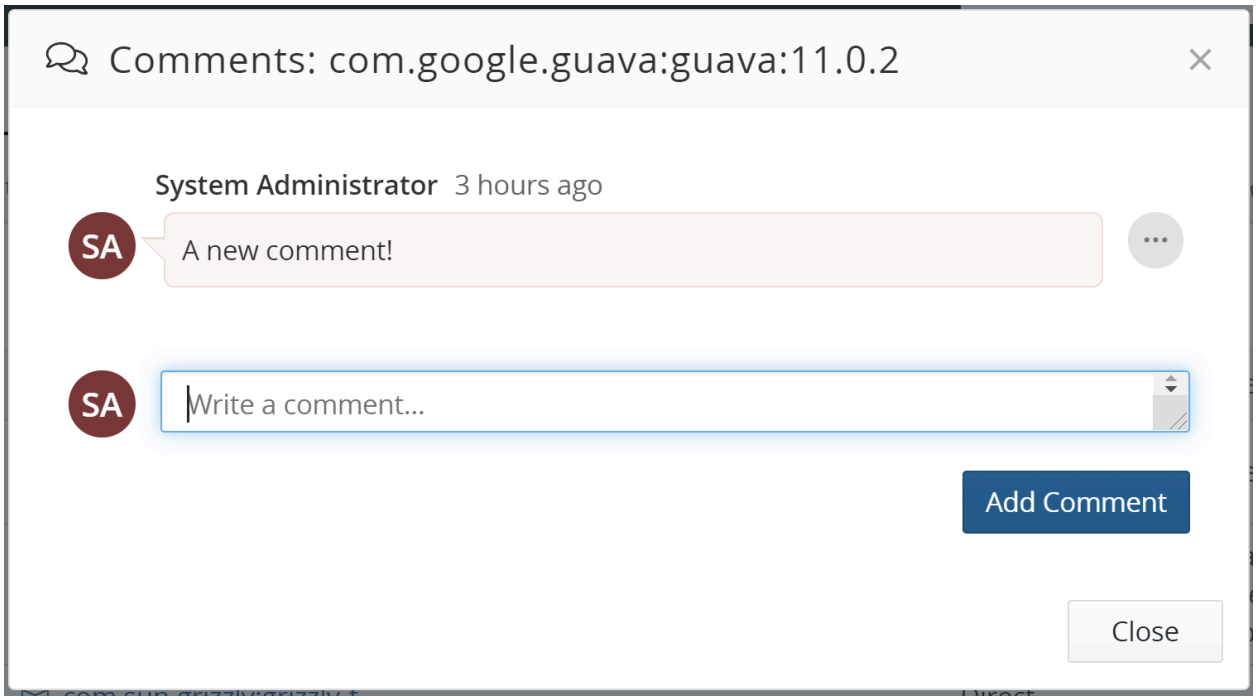
Only the original writer of the comment or Project Administrator can delete a comment.

1. Click or (if there are already comments) 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**.

Adding a comment in the Source view

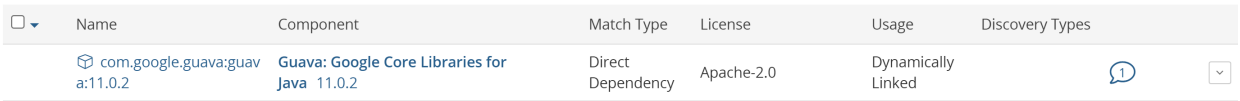
1. Display the project version BOM. Ensure you are in the Source view.
2. Click the desired item in the file tree.
3. Click in the row where you want to add a comment and select **Comments**, or click if there are already comments present.

   The Comments dialog box appears.



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

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



Viewing a comment

Click in the row where you want to view a comment and select **Comments**, or click if there are already comments present.

Editing a comment

Only the original writer can edit their comment.

1. Click or  (if there are already comments) in the row where you want to edit a comment and select **Comments**.
2. Click next to the comment you want to edit and select **Edit**.
3. Edit the comment, click **Save**, and then select **Close**.

Deleting a comment

Only the original writer of the comment or Project Administrator can delete a comment.

1. Click or  (if there are already comments) in the row where you want to edit a comment and select **Comments**.
2. Click next to the comment you want to delete and select **Delete**.

Managing files associated with BOM components

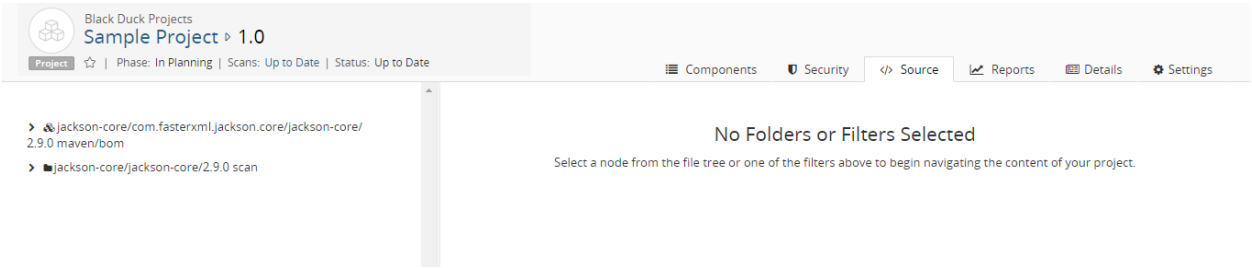
Use the **Source** tab to manage the files associated with BOM components. Common cases include:

- 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

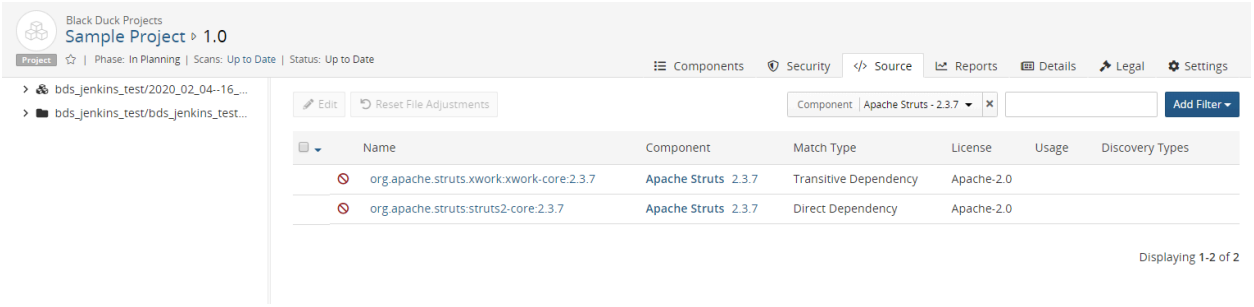
You can access the **Source** tab to view all files in a project or automatically filtered to view specific matches.

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.



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.



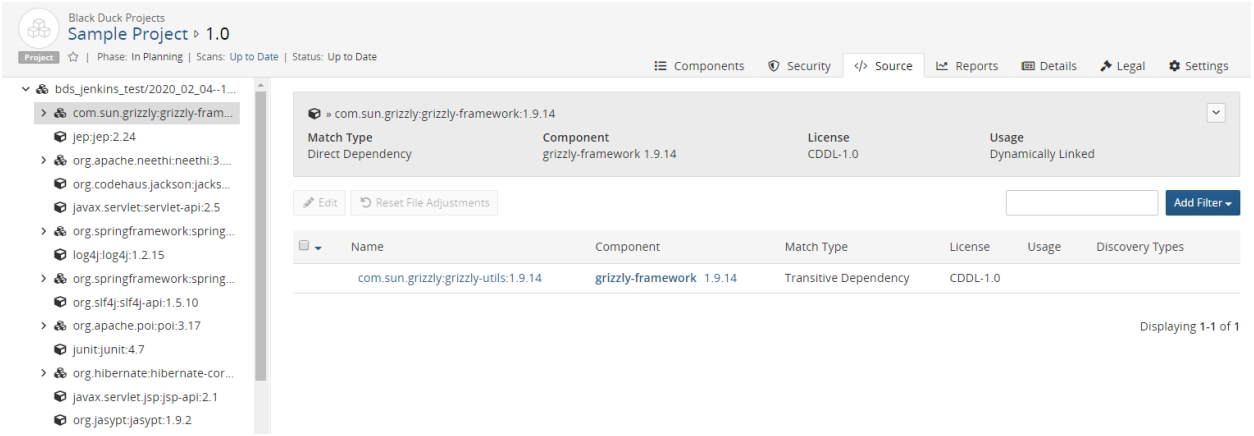
About the Source tab

The **Source** tab consists of:

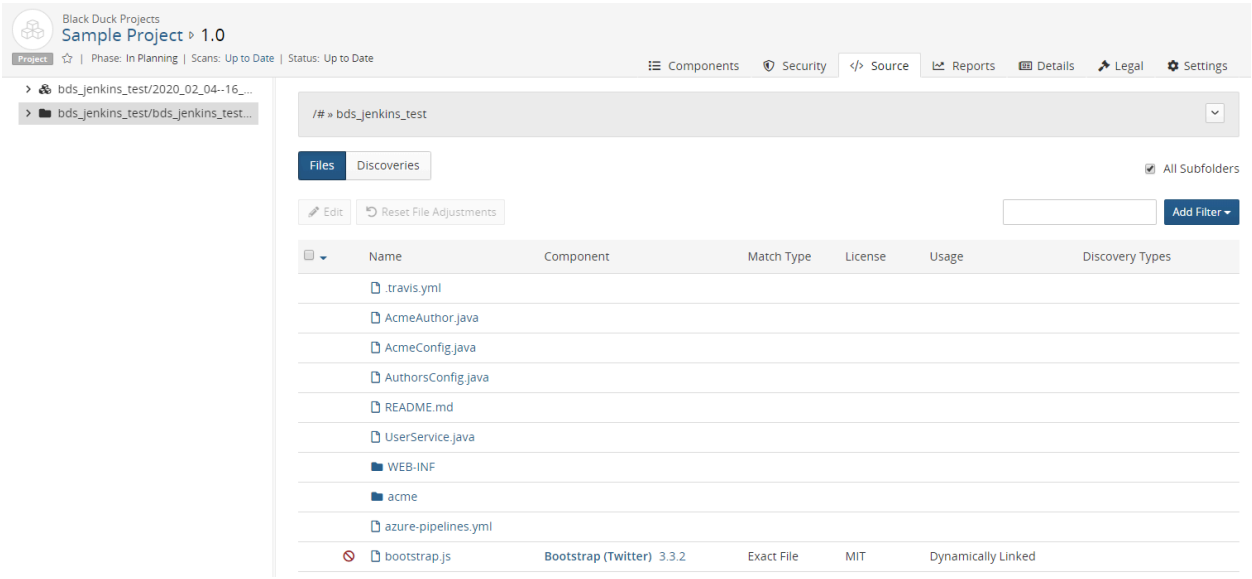
- A left pane which shows the tree structure of the files. Use this pane to navigate and select the information shown in the table.

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

  Selecting to view an archive:



  Selecting to view a folder:



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 right pane which displays the following information:
  - A header banner containing relevant information for the selected component such as the file name or namespace for the component. In the case of imported SBOM files, the banner will contain information such as the SBOM specification (SPDX or CycloneDX), when the SBOM was imported, who supplied the SBOM, and the version of the tool used to create the SBOM.
  - 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 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 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 Edit located above the table.
3. In the Edit Component (if you selected one item) or Bulk edit (if you selected multiple items) dialog box, modify the component, version, origin ID, and/or usage.

   Click here for more information about modifying snippet matches.
4. Click **Update**.

Identifying unmatched components

An unmatched component means that it was not possible to match the stated external identifier to a component in the Black Duck KnowledgeBase. The external identifiers in the KnowledgeBase are taken from the public Forges, like Maven Central, etc. Unmatched components found in the BOM Import log can also be seen in the Source view of a scan. You can triage these components and either identify them to a KnowledgeBase component or create a custom component and associate it with the components.

1. Open the **Source** tab as described above.
2. Click the root folder of the scan in the left-hand panel of the Source view.
3. Select one or more entries and click Edit. The Edit Component dialog box (if you selected one item) or Bulk Edit dialog box (if you selected multiple items) appears.
   - If the component already exists, enter the name in the **Component** field and specify a version.
   - If the component does not exist, create a custom component first before completing this step.
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 Package**.

Identifying unmatched files

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

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

Validating matched files

1. Open the **Source** tab as described above.
2. Click Add filter and select **Match Type > Type of match(es)** and click **OK**.
3. Select one or more entries and click Edit. The Edit Component dialog box (if you selected one item) or Bulk Edit dialog box (if you selected multiple items) appears.
   - If the file was incorrectly matched to a component during the scan, enter the new name in the **Component** field and specify a version in the **Version** field.
   - If the file was incorrectly matched to an origin or origin ID, specify a different value using the **Origin** and **Origin ID** fields.
   - If the file should not be included in the project, select **Dev. Tool / Excluded** from the **Usage** list.
4. Click **Update**.

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

Resetting files and components

You can revert manually adjusted files and components 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 or components and click **Reset Adjustments**.

   If you select multiple files or components, 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.

Generating project version reports

Use project version reports to export and share the content of a single project version. You can run a Version Details report, Vulnerability report, Notices File report, or a Software Bill of Materials (SBOM) report.

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.
- Export and share the information of a single project version.

Note: Reports include subproject information *if* you have permission to the subproject.

Version Details report

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 well as sub-projects.

  As Black Duck caches this data, the information shown in this report may lag 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 New Report** and select **Version Details**.
5. Check or uncheck the **Include Subprojects** checkbox.
6. 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
7. 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.
8. Download the report and extract the zip locally.

Vulnerability report

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 New 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-remedation-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 (_).

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:

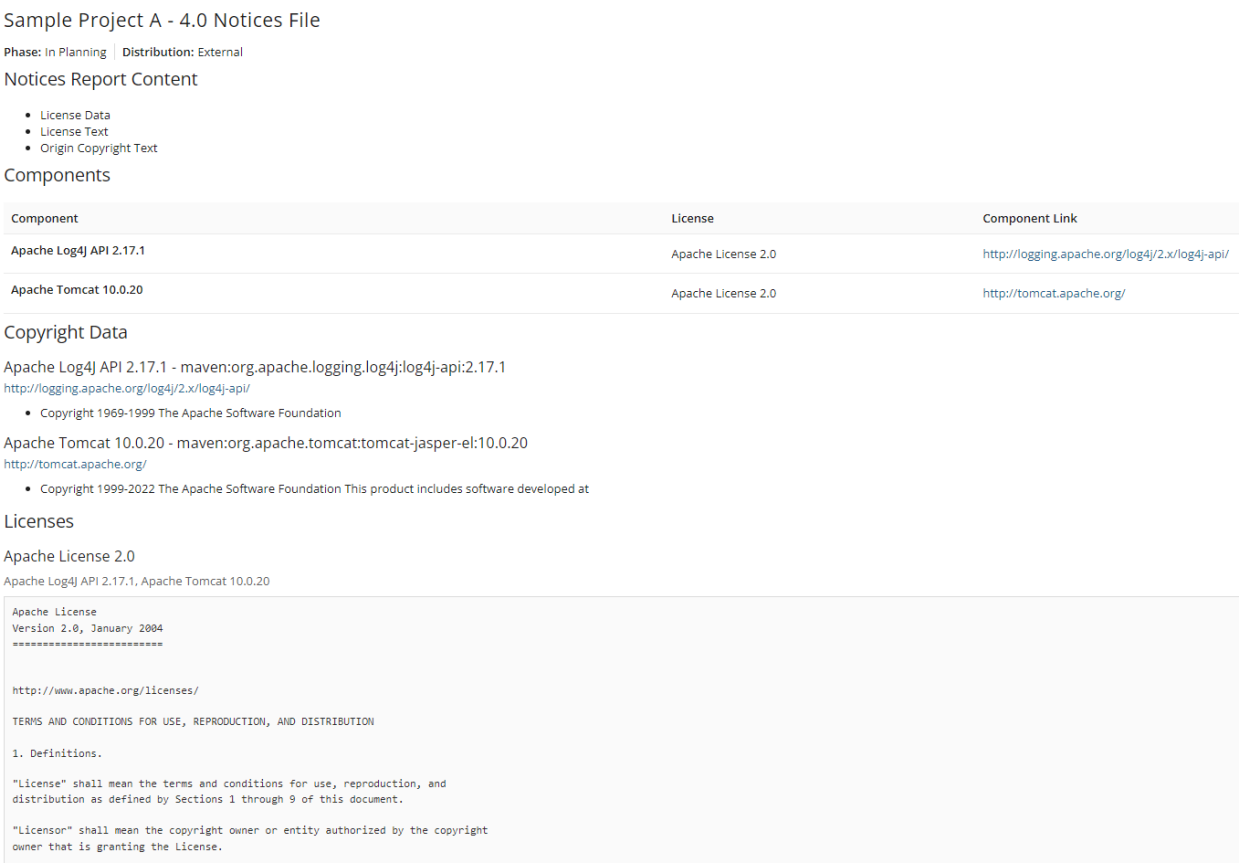
- Header information. Lists the project name, version, phase, and distribution.
- Components. Lists all components, component versions, subprojects, subproject versions, and associated licenses, including deep license data.

  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.
- Other options:
  - Deep License Data: Adds deep licenses discovered via component origin to the list of components. Only available if deep licenses are enabled for the project.
  - File Copyright Text: Provides a Copyright Text section that contains copyright statements discovered in file matches. Only available if file matches are present.
  - File License Data: Licenses discovered in file matches. Only available if file matches are present.
  - License Data. Choose to include the licenses of components in the project.
  - License Text. Choose to include the text of licenses in the project.
  - Origin Copyright Text. 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.
  - Unmatched File Discoveries: Provides an Unmatched File Data section that includes file discoveries unassociated with components in the project. Only available if unmatched files are present in the project.

The following is an example of a portion of the HTML version of the report:



Sample Project A - 4.0 Notices File

Phase: In Planning | **Distribution:** External

Notices Report Content

- License Data
- License Text
- Origin Copyright Text

Components

Component	License	Component Link
Apache Log4J API 2.17.1	Apache License 2.0	http://logging.apache.org/log4j/2.x/log4j-api/
Apache Tomcat 10.0.20	Apache License 2.0	http://tomcat.apache.org/

Copyright Data

Apache Log4J API 2.17.1 - maven:org.apache.logging.log4j:log4j-api:2.17.1
http://logging.apache.org/log4j/2.x/log4j-api/

- Copyright 1969-1999 The Apache Software Foundation

Apache Tomcat 10.0.20 - maven:org.apache.tomcat:tomcat-jasper-el:10.0.20
http://tomcat.apache.org/

- Copyright 1999-2022 The Apache Software Foundation This product includes software developed at

Licenses

Apache License 2.0
Apache Log4J API 2.17.1, Apache Tomcat 10.0.20

```
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 Black Duck 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 New Report**.
5. Select **Notices File** and then select the format for the report:
   - Text
   - HTML
6. Check or uncheck the **Include Subprojects** checkbox.
7. Optionally, select one or more of the following options:
   - Deep License Data
   - File License Data
   - File Copyright Text

- License Data
- License Text
- Origin Copyright Text
- Unmatched File Discoveries

The Notices File report may take more time to run if any of these options are selected.

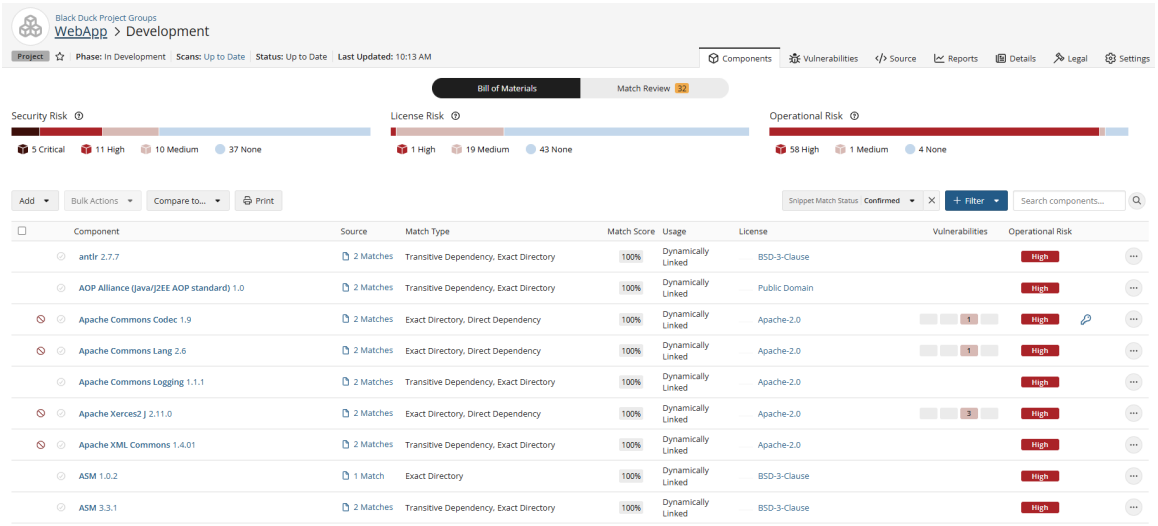
8. Click **Create** to run the report.
9. A link that includes the project, version name, and date appears when the report completes. Any user who is a member of the project can access the link.
    - If you selected the text format, download the report and extract the zip file locally.
    - If you 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

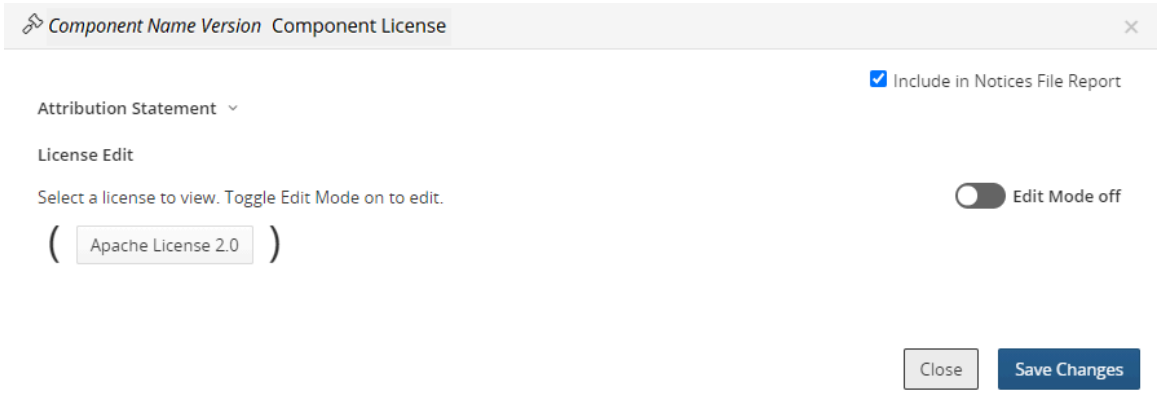
By default, all components and subprojects are included in the Notices File report.

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.

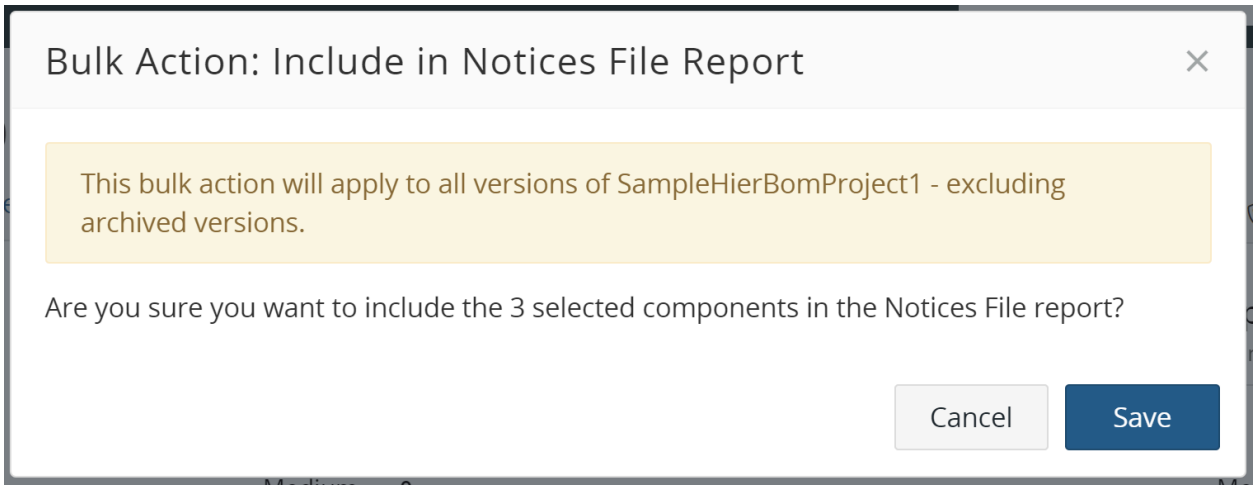
   Select **Include in Notices File Report** to include the component or subproject in the report.

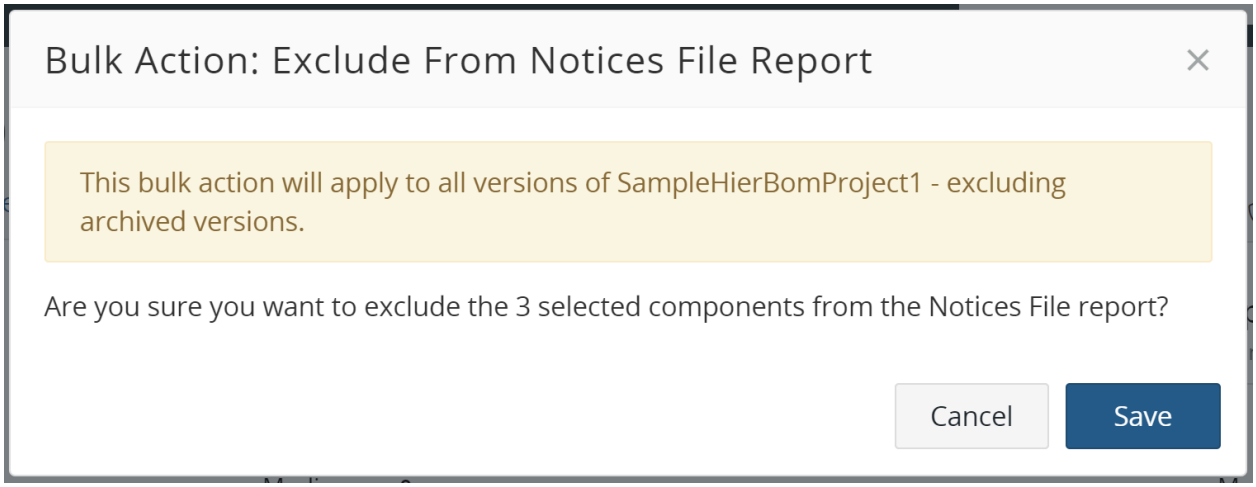
5. Click **Save Changes**.

To include/exclude multiple components in the Notices File report:

1. Display the project version BOM. Ensure you are in the Components view.
2. Check the box next to any number of components.
3. Click the **Bulk Actions** button.
4. Select **Include in Notices File Report** or **Exclude from Notices File Report**.

   The **Bulk Action: Include in Notices File Report** or **Bulk Action: Exclude From Notices File Report** dialog box appears.





5. Click **Save**.

Software Bill of Materials (SBOM) report

A software Bill of Materials (SBOM) is a list of all the open source and third-party components present in a codebase. An SBOM also lists the licenses that govern those components, the versions of the components used in the codebase, and their patch status, which allows security teams to quickly identify any associated security or license risks. See the individual SPDX and CycloneDX mapping entries for additional details on fields found in their SBOM reports.

You can export your SBOM report for a specific project version. SBOM reports can also be used to import project information into Black Duck.

To run a Software Bill of Materials 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 New Report** and select **Software Bill of Materials (SBOM)**.
5. Select a SBOM template from the **Template** dropdown menu. The default SBOM template will automatically be selected, but can be changed if desired.
6. Select the desired SBOM specification:
   - SPDX v2.2
   - SPDX v2.3
   - SPDX v3.0
   - CycloneDX v1.3
   - CycloneDX v1.4
   - CycloneDX v1.5
   - CycloneDX v1.6
7. Select the desired Report Format:
   - JSON (CycloneDX SBOM reports only support this format)

- YAML
- RDF
- tag:value

8. Optionally, you can expand the **Template Details** to see the fields included in the selected SBOM template.
9. Click **Create** to run the report.
10. Click the link to download and view the report.

Note: If the **Don't generate SBOM reports for projects with policy violations** option has been enabled for this project's group and the project has policy violations, the option to generation a SBOM report will be disabled.

Match Review items in SBOM and VEX reports

When generating SBOM and VEX reports, components and vulnerabilities on the **Match Review** page behave as follows:

- **Components pending Match Review:** Components currently listed under **Match Review** are excluded from the exported SBOM component list.
- **VEX reports and vulnerabilities:** Vulnerabilities associated with components in **Match Review** appear on the project version's **Vulnerabilities** tab to ensure potential risks remain visible. The inclusion and reported disposition of these vulnerabilities in exported VEX reports depend on their current vulnerability remediation status.
- **Ignored Match Review components:** Components explicitly ignored on the **Match Review** page do not contribute vulnerabilities to the **Vulnerabilities** tab and are excluded from both SBOM and VEX exports.

What fields are imported from SBOMs

When importing Software Bill of Materials (SBOMs), not all fields are processed by Black Duck SCA. Understanding which specific fields are considered is essential for users to effectively utilize SBOM functionality and ensure comprehensive vulnerability management. This section outlines the fields that Black Duck SCA evaluates during the SBOM import process, providing greater transparency into the detailed SPDX functionality and how it can be leveraged in your projects.

Table 11: SBOM Fields Imported by Black Duck

Field	CycloneDX	SPDX 2.x	SPDX 3.x	Notes
Component/ Package Name	`Component.name` Reference	`Package.name` Reference	`Software.Package.name` Reference 1 Reference 2	**Mandatory**, the whole import fails if even a single Component does not have the name field (an empty value in the name field causes no failure) Always exported
Component/ Package	`Property` with names `BlackDuck-Component`,	`ExternalRef` with types `BlackDuck-Component`,	`ExternalRef` with types `BlackDuck-Component`,	**Optional**, used for matching

Field	CycloneDX	SPDX 2.x	SPDX 3.x	Notes
BlackDuck IDs	`BlackDuck-ComponentVersion`, `BlackDuck-ComponentOrigin` Reference	`BlackDuck-ComponentVersion`, `BlackDuck-ComponentOrigin` Reference	`BlackDuck-ComponentVersion`, `BlackDuck-ComponentOrigin` Reference	(*OriginID* > *VersionID* > *ComponentID*) Always exported
Component/ Package URL	`Component.purl` Reference	`ExternalRef` with type `purl` Reference	`ExternalIdentifier` with type `purl` Reference	**Optional**, used for matching Export controlled by template
Component/ Package Version	`Component.version` Reference	`Package.versionInfo` Reference	`Software.Package.packageVersion` Reference 1 Reference 2	**Optional** Always exported
Component/ Package Supplier	`Component.supplier` Reference	`Package.supplier` Reference	`Software.Package.supplier` Reference 1 Reference 2	**Optional** Only persisted for matched *Components* Export controlled by template
Component/ Package Originators	`Component.authors` Reference	`Package.originator` Reference	`Software.Package.originatedBy` Reference 1 Reference 2	**Optional**, **Export only**, not persisted during import Export controlled by template
Component/ Package CPE	`Component.cpe` Reference	`ExternalRef` with type `cpe22Type` or `cpe23Type` Reference	`ExternalIdentifier` with type `cpe` Reference	**Optional** Only persisted for matched *Components* Export controlled by template
Component/ Package Hash	`Component.hashes[0]` Reference	`Package.checksums[0].value` Reference	`Software.Package.verifiedUsing` Reference 1 Reference 2 Reference 3 Reference 4	**Optional** Only first hash persisted in import Export controlled by template
Component/ Package Hash Algorithm	`Component.hashes[0].alg` Reference	`Package.checksums[0].algorithm` Reference	`Software.Package.verifiedUsing` Reference 1 Reference 2 Reference 3 Reference 4	**Optional** Only the algorithm of the first hash persisted in import Export controlled by template
Component/ Package Comment	N/A	`Package.comment` Reference	`Software.Package.comment` Reference 1 Reference 2	**Optional** Only persisted for matched *Components* Export controlled by template

Field	CycloneDX	SPDX 2.x	SPDX 3.x	Notes
Component/ Package Valid Until Date	N/A	`Package.validUntilDate` Reference	`Software.Package.validUntilTime` Reference 1 Reference 2	**Optional** Only persisted for matched *Components* Export controlled by template
Component/ Package Download Location	`ExternalReference` with type `distribution` Reference	`Package.downloadLocation` Reference	`Software.Package.downloadLocation` Reference 1 Reference 2	**Optional** Only persisted for matched *Components* Export controlled by template
Component/ Package Usage	N/A CycloneDX has no `RelationshipTypes`, so we always assign `HAS_PREREQUISITE`	`RelationshipType` Reference	`Core.relationshipType` Reference	**Optional** Only persisted for matched *Components* Always exported
Component/ Package Declared License	N/A	`Package.licenseDeclared` Reference	`Relationship with hasDeclaredLicense` Reference 1 Reference 2	**Optional**, during import used only during *Component* auto-creation functionality, not persisted (used for finding out the *License* for the *Custom Component*)
Component/ Package Concluded License	`Component.licenses` Reference	`Package.licenseConcluded` Reference	`Relationship` with `hasConcludedLicense` Reference 1 Reference 2	**Optional**, during import used only during *Component* auto-creation functionality, not persisted (used for finding out the *License* for the *Custom Component*)
Component/ Package Declared License Comment	N/A	`Package.licenseComments` Reference	`Relationship` with `comment` (e.g., `hasDeclaredLicense`) Reference 1 Reference 2	**Optional** Only persisted for matched *Components* Export controlled by template
SBOM Type	`Metadata.lifecycles.phase` Reference	N/A	`Software.sbomType` Reference 1 Reference 2	**Optional** Export controlled by template

Ignored SBOM fields during import

CycloneDX

- Additional *Component/Package* checksums/hashes & algorithms after first one
- *Relationship*s and *Relationship* comments
- Originators beyond author parsing
- All properties except the *BlackDuck-Component,BlackDuck-ComponentVersion* and *BlackDuck-ComponentOrigin*
- Non-*DISTRIBUTION* external references (they're parsed but only *distribution* used for the *downloadLocation*)
- Additional supplier contacts after the the first email

SPDX SBOM

- Additional *Component/Package* checksums/hashes & algorithms after first one
- *Relationship*s and *Relationship* comments beyond *RelationshipType* (*SPDX 2.x* and *SPDX 3.x*), and *License*s (*SPDX 3.x*)
- Originators beyond author parsing

SPDX 2.2 and 2.3 data fields

Field	Description
`SPDXID`	This field contains the identify of the current SPDX document which may be referenced in relationships by other files, packages internally and documents externally.
`spdxVersion`	The version of SPDX used to generate this report.
`creationInfo`	`comment`: An optional field for creators of the SPDX document to provide general comments about the creation of the SPDX document or any other relevant comment not included in the other fields. (SPDX 2.2 and 2.3): If SBOM Type was enabled in the SBOM template when the report was generated, it will be included here as `SBOM Type = <value>`. `created`: The date and time (timestamp) when the document was created. `creators`: The company or organization that created the SPDX document. If the SPDX document was created by an individual, the person's name will be indicated. If the SPDX document was created using a software tool, the name and version for that tool will be indicated. `licenseListVersion`: The version of the SPDX License List used when the SPDX file was created.
`name`	The BOM project name.
`dataLicense`	The licensing under which the creator of this SPDX document allows related data to be reproduced. The only valid value for this property is http://spdx.org/licenses/CC0-1.0.
`documentNamespace`	URL to Black Duck's license and readme page on Github.
`documentDescribes`	This field contains the parent ID for the package. Displayed as `SPDXRef-package-[BOM project version UUID]`.

`packages`	This section contains both the exported project version (which is described by the `documentDescribes`) and also the project version BOM component(s). Each component will have the items listed below: `SPDXID`: The unique ID for the specified entry; project version or project version BOM components. `comment`: General comments about the package being described. `copyrightText`: The copyright text for the exported project version or its BOM component(s). `description`: This field is a short description of the package. `downloadLocation`: The URL to download the project version or its BOM component(s). `externalRefs`: This section lists outside sources of information, metadata enumerations, asset identifiers, package manager URLs, or content relevant to the Package, such as a structured naming scheme identifying Packages with known security vulnerabilities. `homepage`: The URL of the exported BOM project version or its project version BOM component(s). `licenseConcluded`: The modified license(s) for the exported BOM project version or its project version BOM component(s). `licenseDeclared`: The license(s) from the KnowledgeBase. `name`: The name of the exported BOM project version or its project version BOM component(s). `originator`: If the package identified in the SPDX file originated from a different person or organization than identified as Package Supplier, this field identifies from where or whom the package originally came. `supplier`: The origin namespace ID. `validUntilDate`: The end of the support period for a package from the supplier. `versionInfo`: The version information of the exported BOM project version or its project version BOM component(s). **Note:** Some of the values above may not have any information and will display a value of `NOASSERTION`.
components section	This section contains the same fields above, detailing the information for each of the components found in the project with the following exceptions: • The `externalRefs` section will list the URLs for the component and component version pages in Black Duck.
`files`	Does not contain any data, displaying `[]` only.
`relationships`	`relationshipType`: Represents a relationship between two `SpdxElements`. Can be either `DEPENDS_ON` or `CONTAINS`. `spdxElementId`: Contains the parent package ID, as displayed in the `documentDescribes` section above. `relatedSpdxElement`: Contains the child or related package ID for the component that is part of the relationship.

`hasExtractedLicensingInfos`	Contains all the licenses that do not have SPDX ID in the KnowledgeBase. They are added with a document-unique license ID. `name`: Name of the license. `licenseID`: Contains the license's ID. `extractedText`: Provide a copy of the actual text of the license reference extracted from the package or file that is associated with the License Identifier Assigned to aid in future analysis.

SPDX 3.0 Data Fields

SPDX 3.0 is the latest version of the System Package Data Exchange specification, which standardizes the way software bill of materials (SBOM) information is communicated. Black Duck SCA supports SBOMs in SPDX 3.0 format, providing detailed metadata for software packages, components, and associated security data.

More information on the data fields can be found on the SPDX specification page.

Enhanced vulnerability information

The SPDX 3.0 SBOM format has been enhanced with improved vulnerability information. Your reports now include clear vulnerability identifiers (using CVE, EUVD, and/or BDSA IDs as configured in your SBOM export template), detailed scoring information through various CVSS assessment relationships, and explicit status indicators for each vulnerability's current remediation state. These enhancements enable you to provide stakeholders with comprehensive security insights while ensuring the reports remain structured and easy to interpret.

What's included in your SBOM reports

Your SBOM reports now automatically filter vulnerabilities to show the most relevant information:

- Vulnerabilities marked as "Duplicate," "Ignored," or "New" are not included in the reports
- Only after changing a vulnerability's status from "New" to another status will it appear in your SBOM reports

Your SPDX 3 reports include:

- Vulnerability identifiers (CVE, EUVD, and/or BDSA, depending on your SBOM export template settings)
- Detailed scoring information
- Clear status indicators showing if vulnerabilities are affected, fixed, not affected, or under investigation

Vulnerability Class

Black Duck SCA includes the Vulnerability class in SPDX 3 reports. The vulnerability identifier types included in your reports are controlled by your SBOM export template settings, where you can enable or disable CVE, EUVD, and BDSA identifiers. All three are enabled by default.

Primary identifier selection

When multiple identifier types are enabled, a single primary identifier is selected using the priority order CVE > EUVD > BDSA. The highest-priority enabled identifier that exists for the vulnerability is used as the primary `spdxId` and drives the scoring data in the associated CVSS assessment relationships.

- CVE ID: Used as the primary identifier when available. Appears with `externalIdentifierType` of `cve`.

- EUVD ID: Used as the primary identifier when no CVE exists. Appears with `externalIdentifierType` of `securityOther`.
- BDSA ID: Used as the primary identifier when no CVE or EUVD exists. Appears with `externalIdentifierType` of `securityOther`.

Additional identifiers

When a vulnerability has identifiers beyond the primary, all enabled identifiers appear together in the `externalIdentifier` array of the Vulnerability object. For example, a vulnerability with both a CVE and an EUVD identifier would include both in the array:

- CVE identifier with `externalIdentifierType` = `cve`
- EUVD identifier with `externalIdentifierType` = `securityOther`

Note: SPDX 3.0 does not define dedicated `externalIdentifierType` values for EUVD or BDSA. Both use the generic `securityOther` type. You can distinguish them by the identifier string itself — EUVD identifiers begin with "EUVD-" and BDSA identifiers begin with "BDSA-".

CVSS Assessment Relationships

To provide detailed scoring information, the following relationships have been added:

- **CvssV2VulnAssessmentRelationship**: Contains CVSS v2 scoring information
- **CvssV3VulnAssessmentRelationship**: Contains CVSS v3 scoring information
- **CvssV4VulnAssessmentRelationship**: Contains CVSS v4 scoring information

These relationships include the complete CVSS vector strings and scores, similar to the ratings section in CycloneDX.

When multiple vulnerability identifier types are enabled, scoring data in the CVSS assessment relationships is provided by the primary identifier source only. For example, if a vulnerability has both a CVE and an EUVD identifier, the CVSS scores and vectors come from NVD (the CVE source). Secondary identifiers do not contribute additional assessment relationships.

Vulnerability Remediation Representation

SPDX 3 reports now include specific relationships to represent the remediation status of vulnerabilities:

- **VexAffectedVulnAssessmentRelationship**: Used for vulnerabilities with a "known affected" status, indicating the vulnerability is present and affects the component
- **VexFixedVulnAssessmentRelationship**: Used for vulnerabilities with a "fixed" status, indicating the vulnerability has been remediated
- **VexNotAffectedVulnAssessmentRelationship**: Used for vulnerabilities with a "known not affected" status, indicating the vulnerability does not affect the component
- **VexUnderInvestigationVulnAssessmentRelationship**: Used for vulnerabilities with an "under investigation" status, indicating the impact is still being assessed

Note: Unlike CycloneDX, SPDX 3 reports do not include CWE (Common Weakness Enumeration) information due to limitations in the current SPDX 3 specification.

These enhancements align with the SPDX 3.0.1 specification and provide you with more detailed information about vulnerabilities, their severity, and remediation status.

SPDX 3 status mapping

Black Duck SCA Vulnerability Status	SPDX VulnAssessmentRelationship relationshipType
`AFFECTED`	`affects`
Duplicate	N/A - Not included in report
Ignored	N/A - Not included in report
`MITIGATED`	`doesNotAffect` with the justification `"security_justificationType": "inlineMitigationsAlreadyExist"`
`NEEDS_REVIEW`	`underInvestigationFor`
`NEW`	`underInvestigationFor`
`NOT_AFFECTED`	`doesNotAffect`
`PATCHED`	`fixedIn`
`REMEDIATION_COMPLETE`	`fixedIn`
`REMEDIATION_REQUIRED`	`affects`
`UNDER_INVESTIGATION`	`underInvestigationFor`

CycloneDX Data Fields

Enhanced vulnerability information

The CycloneDX SBOM provides enhanced vulnerability information in CycloneDX SBOM reports. This improvement includes detailed vulnerability scores with their corresponding scoring systems, comprehensive information about weakness types through CWE references, and analysis details that show current status and planned remediation actions. By adding this security context to your CycloneDX reports, you can share more meaningful insights with stakeholders while maintaining a clear and actionable format.

What's included in your SBOM reports

Your SBOM reports now automatically filter vulnerabilities to show the most relevant information:

- Vulnerabilities marked as "Duplicate," "Ignored," or "New" are not included in the reports
- Only after changing a vulnerability's status from "New" to another status will it appear in your SBOM reports

Your CycloneDX reports now include:

- Vulnerability scores with their scoring system (like CVSS)
- Information about the type of weakness (CWE)
- Analysis details including current status and planned actions

Vulnerability Ratings

Each vulnerability now includes detailed scoring information:

- **Method**: Identifies the scoring system used (such as CVSSv31 or CVSSv4)
- **Vector**: Contains the complete CVSS vector string that details the characteristics of the vulnerability

For example:

```
"ratings": [
                {
                "method": "CVSSv31",
                "vector": "CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H"
                }
                ]
```

Common Weakness Enumeration (CWE)

Vulnerabilities now include associated CWE identifiers, providing insight into the type of weakness:

```
"cwes": [
                "CWE-79",
                "CWE-80"
                ]
```

Analysis Information

The analysis section provides details about the vulnerability's remediation status:

- **State**: Current status of the vulnerability (e.g., "resolved", "in_triage")
- **Justification**: Reasoning behind the assigned status (e.g., "component_not_present")
- **Response**: Actions taken or planned in response to the vulnerability (e.g., "will_not_fix", "update")

```
"analysis": {
                "state": "resolved",
                "justification": "component_not_present",
                "response": "update"
                }
```

These enhancements align with the CycloneDX 1.6 specification and provide you with more detailed information about vulnerabilities, their severity, and remediation status.

Data fields

Field	Description
`bomFormat`	Specifies the format of the BOM. This helps to identify the file as CycloneDX since BOMs do not have a filename convention nor does JSON schema support namespaces.
`specVersion`	The version of the CycloneDX specification a BOM used for the report.
`serialNumber`	A string formatted by "urn:uuid:"+ a randomly generated UUID number.
`version`	The version allows component publishers/authors to make changes to existing BOMs to update various aspects of the document such as description or licenses. When a system is presented with multiple BOMs for the same component, the system should use the most recent version of the BOM. The default version is '1' and should be incremented for each version of the BOM that is published. Each version of a component should have a unique BOM and if no changes are made to the BOMs, then each BOM will have a version of '1'.

`metadata`	`timestamp`: The date and time (timestamp) when the document was created. `tools`: Describes the tool(s) used in the creation of the BOM, which includes the name of the vendor who created the tool, the name of the tool itself, and the version of the tool. `authors`: The name of the person(s) who created the BOM. May also contain the email address of the contact if present. `component`: The component that the BOM describes; the name of the component, the component version, the type of component, and a `bom-ref` which can be used to reference the component elsewhere in the BOM. `properties`: The SBOM Type of the generated SBOM.
`components`	`author`: The person(s) or organization(s) that authored the component. `supplier`: The organization that supplied the component. `name`: The name of the component. This will often be a shortened, single name of the component. If this field is missing in the SBOM import, the filename of the SBOM will be used as the scan name in place of the missing metadata. `version`: The component's version. If there is no version information, this field is set as "Unknown". `description`: Specifies a description for the component. `licenses`: A list of all licenses associated to the component. If the license is a valid SPDX license, it will be displayed in the `id` field. If the license's SPDX id is not available it will be displayed in the `name` field. `cpe`: Specifies a well-formed CPE name that conforms to the CPE 2.2 or 2.3 specification. `purl`: The component package URL. `pedigree`: The `notes` field lists the license display text. it is especially useful for complex license cases. The licenses section list all licenses objects in a flat list. By using this field, it can pass the complex license info. `externalReferences`: This section contains the component url, e.g. host/components/[component UUID]/versions/[component version id]. `type`: Specifies the type of component. `bom-ref`: An optional identifier which can be used to reference the component elsewhere in the BOM. Every `bom-ref` should be unique.
`dependencies`	Defines the direct dependencies of a component. `ref`: References a component by the components `bom-ref` attribute `dependsOn`: The parent's identifier, either entity version UUID or enitity UUID if the version UUID is unavailable.
`vulnerabilities` (v1.4 only)	`id`: The primary identifier for the vulnerability, determined by which identifier types are enabled in your SBOM export template and the priority order CVE > EUVD > BDSA. The highest-priority enabled identifier that exists for the vulnerability is used. For example, if all three are enabled

	and a CVE exists, the CVE is the `id`; if no CVE exists but an EUVD does, the EUVD identifier is the `id`. `source`: The source of the primary identifier. The `name` will be NVD for CVE, EUVD for EUVD, or BDSA for BDSA, corresponding to whichever identifier is used as the primary `id`. `references`: Additional identifiers for the same vulnerability. When multiple identifier types are enabled in your SBOM export template and additional IDs exist beyond the primary, they appear here. For example, if CVE is the primary `id`, any corresponding EUVD and BDSA identifiers appear as references with their respective `source` `name`. References do not include duplicate score or vector metadata. `ratings`: Vulnerability scoring data from the primary identifier source. The `source` `name`, `score`, `severity`, `method`, and `vector` values come from whichever source provides the primary `id` (NVD for CVE, EUVD for EUVD, or BDSA for BDSA). Only one set of ratings is included — secondary identifiers listed in `references` do not contribute additional ratings. `description`: The description of the vulnerability from the primary source (NVD, EUVD, or BDSA).

CycloneDX status mapping

Black Duck SCA Vulnerability Status	CycloneDX Vulnerability Analysis State
`AFFECTED`	`exploitable`
Duplicate	N/A - Not included in report
Ignored	N/A - Not included in report
`MITIGATED`	`not_affected` with the justification `protected_by_mitigating_control`
`NEEDS_REVIEW`	`in_triage`
`NEW`	`in_triage`
`NOT_AFFECTED`	`not_affected`
`PATCHED`	`resolved`
`REMEDIATION_COMPLETE`	`resolved`
`REMEDIATION_REQUIRED`	`exploitable`
`UNDER_INVESTIGATION`	`in_triage`

Dependency relationships in SBOM reports

Dependency relationships indicate how a 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, the SBOM report will indicate that they will not actually be included in the released version of the project.

See the table below for usage types and how they are defined in the SBOM report. Please note that this applies only to SPDX 2.3 SBOM reports. CycloneDX SBOM reports do not indicate the usage type in its `dependencies` section.

Table 12: Usage types

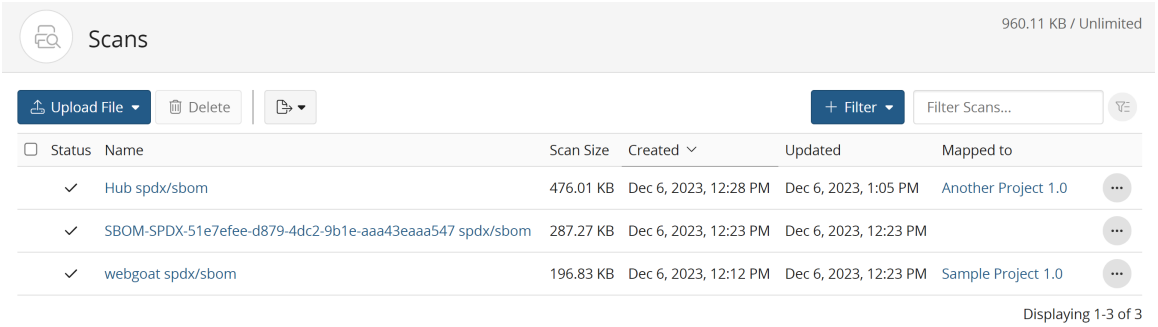
Usage	SPDX relationship used	Description	SPDX relationship comment
Dynamically linked	`DYNAMIC_LINK`	Dynamically linked components that are part of the distribution package, such as with DLLs or JAR files. For dynamically linked components that are not part of the distribution package, please choose "Prerequisite".	NONE
Statically linked	`STATIC_LINK`	A component that is not part of the source tree but linked into the project deliverable statically and distributed with your project.	NONE
Source code	`CONTAINS`	A component or snippet included in the project's source code directly.	NONE
Separate Work	`OTHER`	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.	Separate Work
Prerequisite	`HAS_PREREQUISITE`	Run-time dependencies or dynamically linked components that are not part of the distribution package.	NONE
Merely Aggregated	`OTHER`	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.	Merely aggregated
Implementation of a Standard	`OTHER`	Intended for cases where you implemented according to a standard. For example, a Java spec request that ships with your project.	Implementation of a Standard
Dev. Tool / Excluded	`DEV_TOOL_OF`	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.	NONE
Unspecified	`OTHER`	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.	Unspecified

Importing a SBOM file

If you have a SBOM file, you can import the file into Black Duck.

To import a SBOM file:

1. Log in to Black Duck SCA.
2. Click Scans.



3. In the Scans page, click **Upload File**.
4. Use the Upload Files dialog box to locate the SBOM file
5. Click **Close**.

Mapping the SBOM scan to a project

You can map the scan to a project in the following ways.

On the Scans page

1. Click the button at the end of the scan's row and selecting **Map to Project**. This opens the **Map Scan to Project Version** dialog box.
2. Enter the project's name in the **Project** field. Alternatively, you can create a new project for this scan.
3. Enter the project version in the **Version** field. You can also create a new project version for this scan.
4. Click the **Save** button to complete the mapping.

Preserving Component Relationships on SBOM Import

Black Duck SCA now preserves component relationships when importing SBOM (Software Bill of Materials) reports. This enhancement ensures that dependency information from imported SBOMs is retained and can be accurately reproduced when generating new SBOM reports from your project versions.

Why This Matters

CISA SBOM minimum requirements mandate that SBOMs include dependency relationship information. These relationships document how software components are included in your application, enabling you to:

- Build accurate dependency graphs showing direct and transitive dependencies
- Perform comprehensive vulnerability impact analysis
- Conduct thorough license compliance assessments
- Document backported or forked software components

According to CISA guidelines, the Dependency Relationship element reflects how a given software component was included in the target software. The inclusion relationship ("includes" or "included in") supports the capability to build a dependency graph and explicitly documents when components are derived from or are descendants of other software.

How It Works

When you import an SBOM report into Black Duck SCA, the system now:

1. **Preserves Relationship Data**: Component relationships from the original SBOM are stored and maintained within Black Duck.
2. **Supports Multiple Relationship Types**: The system recognizes and maps various SPDX relationship types to Black Duck SCA usage types, including:
   - Dynamic and static linking
   - Source code containment
   - Prerequisites
   - Development tools
   - And more
3. **Handles Complex Scenarios**: The implementation accounts for:
   - Components with multiple parent dependencies
   - Merging relationships when multiple scans are mapped to the same project version
   - Preventing duplicate relationships within imported SBOMs
4. **Maintains Compatibility**: Existing functionality, including scan unmapping and purge operations, continues to work as expected.

Configuration

Preserving component relationships on SBOM import is disabled by default. To enable this feature, you must configure the following setting:

```
blackduck.scan.sbom.preserve.relationships=true
```

Once this configuration is set, Black Duck SCA will automatically preserve relationship data from imported SBOM reports.

SPDX Relationship Type Mappings

Black Duck SCA automatically maps SPDX relationship types to internal usage types to ensure accurate representation:

Table 13: SPDX v2 Mappings

SPDX v2 RelationshipType	Black Duck SCA Usage mapping
DYNAMIC_LINK	DYNAMICALLY_LINKED
STATIC_LINK	STATICALLY_LINKED
CONTAINS	SOURCE_CODE
HAS_PREREQUISITE	PREREQUISITE

SPDX v2 RelationshipType	Black Duck SCA Usage mapping
DEV_TOOL_OF	DEV_TOOL_EXCLUDED
OTHER	Determined from the possible RelationshipType Comment: • SEPARATE WORK → SEPARATE_WORK • MERELY AGGREGATED → MERELY_AGGREGATED • IMPLEMENTATION OF A STANDARD → IMPLEMENTATION_OF_A_STANDARD • UNSPECIFIED → UNSPECIFIED
All other types default to DYNAMICALLY_LINKED	

Table 14: SPDX v3 Mappings

SPDX v3 RelationshipType	Black Duck SCA Usage mapping
HAS_DYNAMIC_LINK	DYNAMICALLY_LINKED
HAS_STATIC_LINK	STATICALLY_LINKED
CONTAINS	SOURCE_CODE
HAS_PREREQUISITE	PREREQUISITE
OTHER	Determined from the possible RelationshipType Comment: • SEPARATE WORK → SEPARATE_WORK • MERELY AGGREGATED → MERELY_AGGREGATED • IMPLEMENTATION OF A STANDARD → IMPLEMENTATION_OF_A_STANDARD • UNSPECIFIED → UNSPECIFIED
USES_TOOL with lifecycle DEVELOPMENT	DEV_TOOL_EXCLUDED
All other types default to DYNAMICALLY_LINKED	

Vulnerability Exploitability EXchange (VEX)

Vulnerability Exploitability eXchange (VEX) reports in Black Duck SCA offer a standardized way to communicate the exploitability status of vulnerabilities in your products. These machine-readable reports help organizations efficiently inform stakeholders about whether specific vulnerabilities affect their offerings.

An SBOM (Software Bill of Materials) serves as a static declaration of components in a product release. After release, it's important to determine the exploitability of any vulnerabilities, and VEX reports facilitate this process. While the SBOM content remains unchanged, VEX allows for on-demand updates regarding the status of vulnerabilities for specific project versions.

Important: To take advantage of this feature, you must have **Vulnerability Exploitability eXchange (VEX)** enabled on your product registration key.

Creating a Project Version VEX Report

Project Version VEX reports focus on vulnerabilities specific to individual project versions, enabling targeted remediation strategies. Use the steps below to generate a VEX report for a specific project version:

- Log in to Black Duck SCA.

- Select the desired project name using the **Watching** or **My Projects** dashboard.
- Select the version of the project for which you want to run the report.
- Select the **Reports** tab.
- Select **+ Create Vulnerability Report → Vulnerability Exploitability EXchange (VEX): CSAF 2.0**.
- Select a **Template**.
- Optionally, you can expand the **Template Details** section to see which options are enabled in the selected template.
- Select **Create** to generate the report.

  The following links appear when the report completes:

  - csaf-report_*YYYY-MM-DD_HHMMSS* (time stamp in system timezone) for a global version of the report.
- Select the link to view the report.

VEX Report Format

The VEX report utilizes the CSAF 2.0 (profile 5) format, aligning with industry standards for vulnerability reporting. It clearly indicates which products are unaffected by vulnerabilities, making it a valuable resource for product security teams. This feature facilitates effective communication about security risks, enabling organizations to address customer inquiries and maintain a robust security posture.

Reports will be generated in JSON format, implementing only the minimum required elements for CSAF v2, profile 5. Each report will include entries for CVEs or BDSAs within the selected projects that have one of the following BD status values:

- Under Investigation
- Needs Review
- Known Affected
- Known Not Affected
- Remediation Required
- Remediation Complete

VEX Document Metadata

The VEX report includes essential metadata that is automatically populated and structured according to the CSAF 2.0 specifications. Below are the key elements of the VEX document metadata:

- **Document metadata**
  - **CSAF Version**: Set to `2.0` (cannot be modified).
  - **Category**: Set to `csaf_vex` (cannot be modified).
  - **Publisher**
    - **Category**: Set to `vendor` (cannot be modified).
    - **Name**: Use the value from the existing Project Group SBOM Creator Organization field. If this field is set to the default value of "COMPANY NAME," a warning will be generated indicating that the default value is being used and should be changed. Users will have the option to Cancel or Continue:
      - **Cancel**: Returns to the report generation screen.

        - **Continue**: Generates the report using "COMPANY NAME" as the publisher name.
    - **Namespace**: A new text field will be added under the "Person" field in the Project Group SBOM fields for the Namespace. SBOM generation will utilize the BD namespace value to populate the namespace field in the SBOMs (for applicable SPDX/CycloneDX versions).
- **Title**: Defined by Black Duck and currently not modifiable. The title will read: "Vulnerability status report using the CSAF 2.0 Profile 5 specification."
- **Tracking**
    - **ID**: The ID will be the CSAF document filename (e.g., `csaf-report_all_projects_2025-07-22_143038`). The VEX Product ID can be configured in SBOM Fields.
    - **Current Release Date**: The date/timestamp of report generation (UTC).
    - **Initial Release Date**: The timestamp of report creation.
    - **Revision History**: One entry will be added for the current (latest) information:
        - **Date**: Report generation date/timestamp (UTC).
        - **Number**: Hardcoded value of 1 for the version.
        - **Summary**: Either "Latest information" or "Initial," depending on whether tracking both is necessary.
    - **Version**: Hardcoded value of 1 for the version.
    - **Generator Information:**
        - **Date**: Report generation date/timestamp (UTC).
        - **Engine**:
            - **Name:** "Black Duck HUB"
            - **Version:** The version of Black Duck HUB used to generate the report.
    - **Status**: Set to `draft`. (Note: Future support may be added for final and interim options, as referenced in section 3.2.1.12.7 of the CSAF 2.0 specification.)
- **Product Tree**: The `product_tree` section of a VEX report in SCA provides a hierarchical representation of the products that are affected by a specific vulnerability. It is structured to illustrate the relationships between different product versions and their corresponding product names.
    - **Branches**: The `product_tree` consists of branches that represent different levels of the hierarchy. Each branch can contain further branches, allowing for a nested structure that captures the relationship between parent products and their versions.
        - **Category**: Each branch includes a category that defines the type of information it represents. In the example provided, categories might include "product_name" and "product_version."
        - **Name**: The name of the product or product version.
        - **Product**: Each product branch contains relevant details such as:
            - **Name**: The name of the product or product version.
            - **Product ID**: A unique identifier for the product, which helps in distinguishing it from other products within the tree.

- **Vulnerabilties**: The Vulnerabilities section details specific vulnerabilities identified within the software components. This section is essential for understanding the security posture of the application and the associated risks.
  - **CVE Identifier**: If the vulnerability is a CVE, it is listed with its unique CVE ID (Common Vulnerabilities and Exposures). This identifier allows for easy tracking and reference to the vulnerability in external databases.
    - **Notes**: Accompanying each CVE entry are detailed notes that provide descriptions of the vulnerabilities. These descriptions explain the nature of the vulnerability, its potential impact, and any relevant technical details.
      - **Category**: Defines the type of note.
      - **Text**: The actual free-form text providing the information. If there is a related EUVD ID associated with this CVE, it will be displayed here.
    - **Product Status**: This section is essential for users to understand the current state of the products affected by vulnerabilities. It provides transparency regarding ongoing assessments and remediation efforts, helping organizations prioritize their actions to maintain security.
    - **Scores**: This section provides critical information regarding the severity of the identified vulnerabilities, utilizing the Common Vulnerability Scoring System (CVSS).
  - **IDs**: If the vulnerability has a BDSA ID, it will be found in this section.
- **Distribution**
  - **TLP**:
    - **Label**: TLP label associated with the component version.
- **Notes**: Provides additional information about the document itself. This can include general descriptions, frequently asked questions, summaries, or legal disclaimers.
  - **Category**: Note type identifier. Displays `legal_disclaimer` if the project group has a VEX Legal Disclaimer.
  - **Text**: The actual content of the legal disclaimer. Free-form text containing legal terms, conditions, limitations, and protections as defined by the organization.

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.

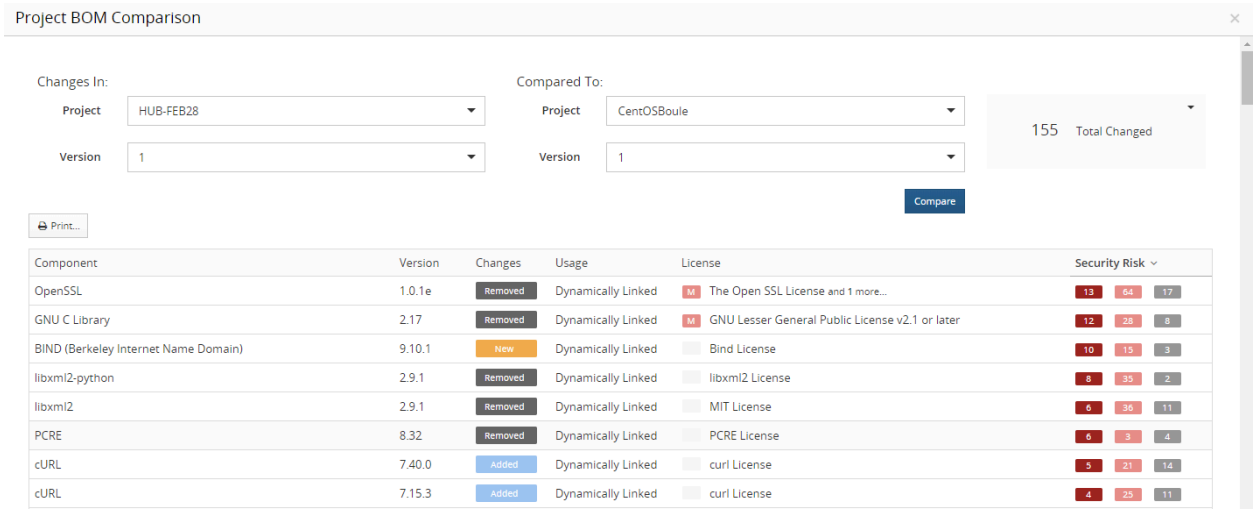
Note: You can only compare projects which you have permission to view.

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.



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:

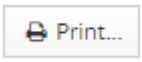
- There is only a top-level comparison of subprojects: the components in subprojects are not compared.
- If you selected to maintain component adjustments to all versions of a project, the Project Comparison window may show little to no changes between versions of the same project.
- Only confirmed snippets 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: • **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. For modifications to ignored components: • 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 **New**. • Components ignored in the "current" version but not ignored in the "compared to" version have a value of **Removed**. Note that for a modification to the version: • 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. 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: 
Usage	Usage of the component or subproject version in the "current" version of the BOM. Strikeout usage text shows the usage for this component version from the "compared to" version of the BOM.
License	Declared license of the component or subproject in use in the "current" version of the project. Strikeout license text shows the license for this component version from the "compared to" version of the BOM.

Column	Description
Security Risk	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. 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: • **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

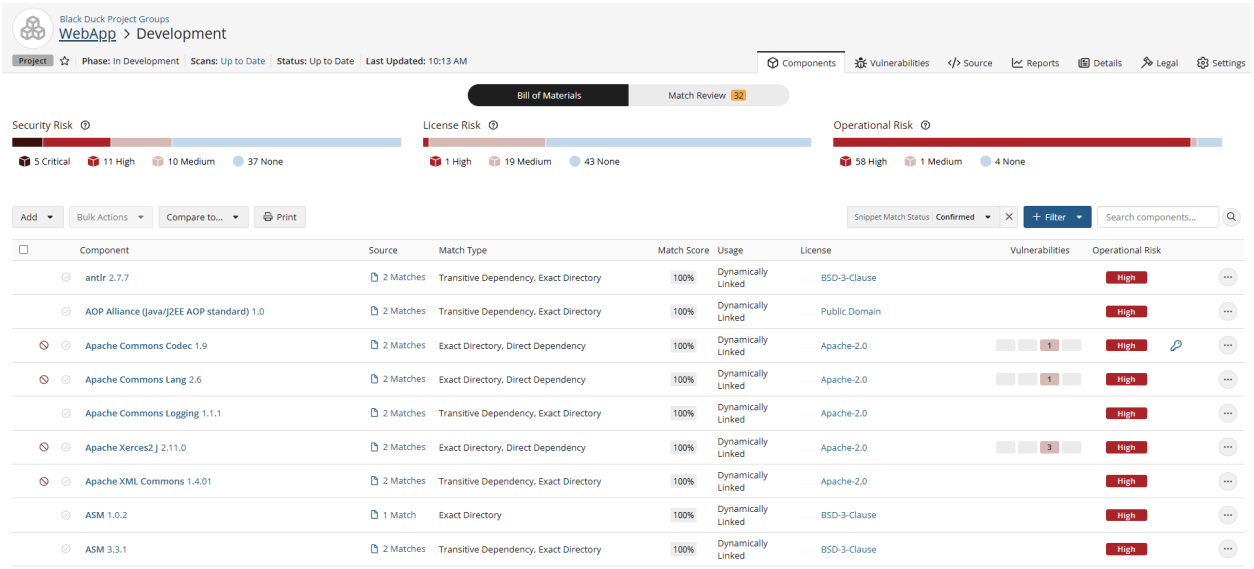
You can print a BOM.

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.



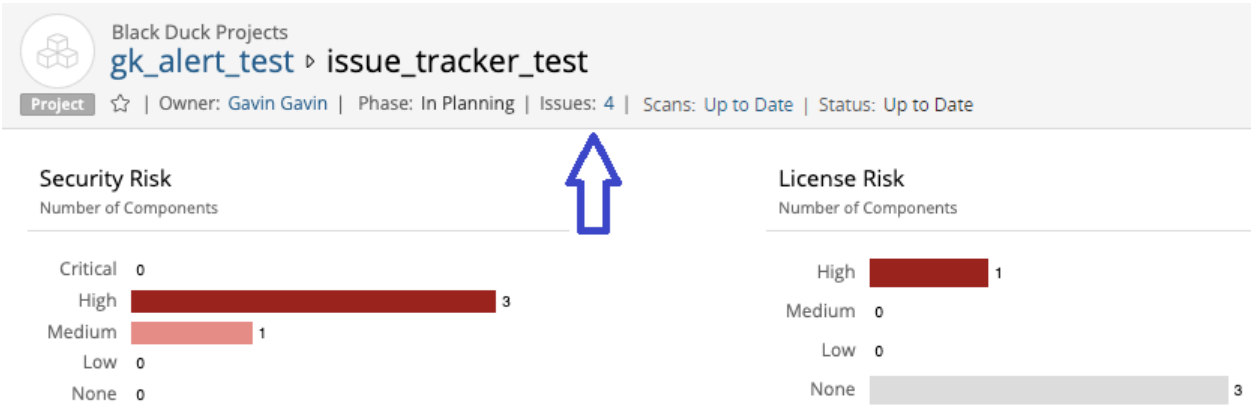
3. Optionally, filter the BOM so that the printout only shows the information you want to see.
4. Click 

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

Viewing issues in a project

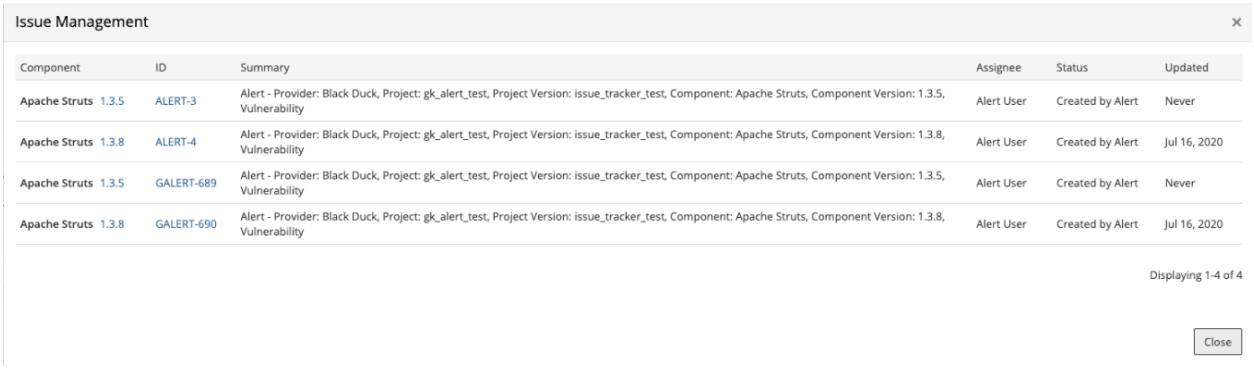
Black Duck provides information on the issues associated with a project version as monitored by an issue tracking system. Currently, this feature is supported using Black Duck Alert 6.2.0 and later.

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 Black Duck Alert. Once Black Duck 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 Global Project Administrator, Global Project Manager roles and all project members (users assigned to the project) can select the issues value to display the Issue Management table.



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 Black Duck Alert.

Viewing component versions with encryption

Attention: This feature is not enabled by default in Black Duck and must be activated by adding the feature to your Product Registration key.

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

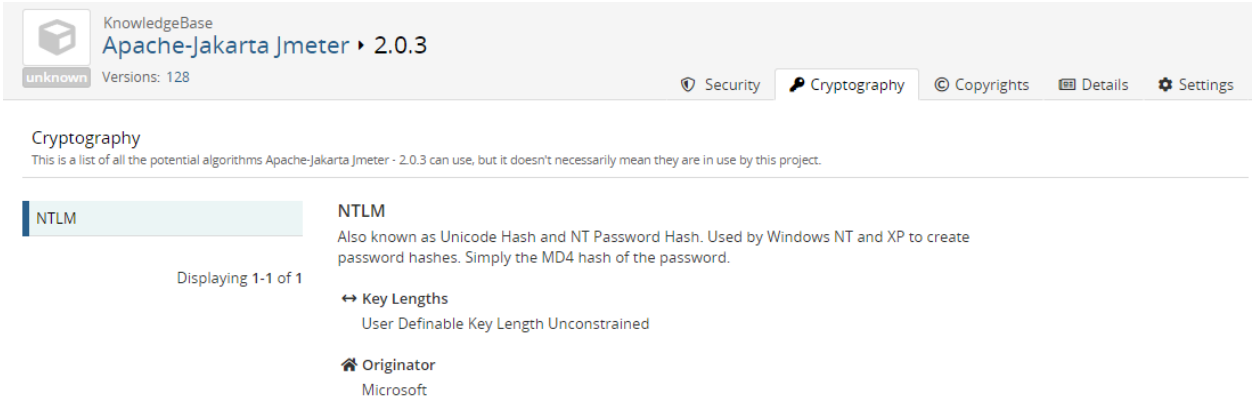
On the compliance side, whenever you send software out of the country – for example, on a computer, as source code, or compiled into an application that is for sale – depending upon where you live, you may be required to adhere to certain governmental regulations regarding the export of cryptography. 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 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.

Match Review

The Match Review feature provides a dedicated workspace for reviewing and managing component matches that require additional verification before being included in your project's Bill of Materials (BOM). This feature helps ensure the accuracy of component identification by allowing you to review and confirm matches that have some degree of uncertainty or ambiguity.

What Components Appear in Match Review

Components appear in the Match Review page when they meet specific criteria that indicate potential uncertainty in their identification:

- **Components with unknown versions** (from binary scans)
- **Components below the match score threshold** (configurable globally via component match review settings)
- **Unmatched components or external IDs** from package manager and SBOM import scans

- **Ignored components** from the BOM when the **Ignored** filter is activated

Important: Components in Match Review are not included in reports or SBOMs until they are reviewed and promoted to the main BOM page.

Accessing Match Review

To access the Match Review page:

1. Navigate to your project version
2. Go to the **Components** section
3. Select the **Match Review** tab

Match Review Page Overview

The Match Review page contains the following features; the **review categories filters** and the **match review table**.

At the top of the page, you'll find **review category filters** that display the count of components in each review state: **Unknown Version**, **Low Match Score**, **Unmatched**, and **Ignored**. Clicking on any category will filter the table to show only components in that specific state, allowing you to focus on particular types of matches that require attention.

Important: Ignored components are not displayed in the Match Review table unless the Ignored filter is activated.

The central **match review table** displays all components requiring review in a structured format. Each row represents a component with columns showing key information:

- **Policy Violation** ⦸: Indicates whether the component has any associated policy violations
- **Component/ID**: Displays the component name and identifier information
- **Match Type**: Indicates the type of match (e.g., exact match, binary, SBOM)
- **Score**: Shows the match score for the component match
- **License**: Displays license information associated with the component
- **Vulnerabilities**: Shows the number of vulnerabilities identified for the component
- **Operational Risk**: Indicates the operational risk level of the component
- **Actions**: Provides an options menu for performing review actions on individual components.

Match Review Sidebar

Selecting a component or ID opens the Match Review sidebar. The Match Review sidebar displays comprehensive details about the selected component, enabling users to make informed decisions regarding matches.

In cases where multiple potential matches exist, the sidebar also shows a list of alternative matches. Each alternative entry includes key identifying information, and users can choose to match the component to any of these alternatives directly from the sidebar. This makes it easier to evaluate available options and quickly correct or refine the match without navigating away from the review workflow.

The panel is organized as follows:

1. **Component/ID**: The name and identifier of the component are displayed at the top for quick reference.
2. **Action Buttons**: Below the component name, users can take actions on the match with available buttons:

- **Match**: Unmatched components offer additional capabilities for handling internal or proprietary components:
  - **Local Matching (Project Version Specific)**
    - **Local match to existing KB/custom component**: Affects only this project version
    - **Local match to new custom component**: Create a new custom component for this project version only
  - **Global Matching (Requires a user with both the Component Manager and Project Manager roles)**
    - **Global match to existing custom component**: Future scans with the same external ID will automatically match to this component
    - **Global match to new custom component**: Create a new custom component that will be used for future matches across all projects
- **Confirm**: Approve the match and promote it to the BOM.
- **Edit**: Modify the match details if necessary.
- **Ignore**: Mark the match as a false positive.
- **Unignore**: If the component was ignored in the BOM, it can be unignored in Match Review. This returns the component to the BOM. Components that were ignored in Match Review remain there as either ignored or unignored.

3. **Match Type**: The type of match is indicated, which may include:
   - Unknown Version
   - Low Match Score
   - Unmatched
   - Ignored
4. **Vulnerabilities and Licenses**: This section displays any associated vulnerabilities and licenses for the component.
5. **Source Information**:
   - **Match Score**: Displays the score level of the match.
   - **KB Artifacts Matched**: Indicates the number of Knowledge Base artifacts matched.
   - **Version Ambiguity**: Shows the number of component versions involved.
   - **Match ID**: Lists the identifier for the match.
   - **Type**: Describes the match type.
   - **File Path**: Provides the path related to the match.
6. **Fields Section**:
   - **SBOM Fields**: Displays any SBOM Fields activated.
   - **Custom Fields**: Indicates if there are any custom fields associated with the component.
7. **Comments Section**: Users can add and view comments related to the match, facilitating collaboration and discussion.
8. **Details Section**:
   - **Component Links**: Provides a URL link to more information about the component.

- **Description**: Offers a brief description of the component.
- **Tags**: Lists any relevant tags associated with the component.

Review Actions

When reviewing components, you have several options available:

- **Confirm Match**: Approve the component and promote it to the main BOM/Components page
- **Edit**: Manually change the match to a different component and/or version
- **SBOM Fields**: Access and edit SBOM-specific field information
- **Comments**: Optionally add comments for any review action
- **Ignore**: Mark as a false positive and move to ignored state
- **Unignore**: If the component was ignored in the BOM, it can be unignored in Match Review. This returns the component to the BOM. Components that were ignored in Match Review remain there as either ignored or unignored.
- **Generate Support Bundle**: If you find discrepancies in component identification, licensing, or vulnerability data in your project's Bill of Materials (BOM), you can generate a KnowledgeBase (KB) support bundle directly from here. This bundle automatically gathers all necessary component data for the support team, streamlining the investigation process and reducing the need for additional communication when opening a support case.

Match Review supports bulk operations for efficiency:

- **Confirm**: Approve multiple selected components simultaneously
- **Comment**: Add a comment to multiple selected components simultaneously
- **Ignore**: Mark multiple components as false positives
- **Unignore**: Mark multiple previously ignored components as active again
- **Generate Support Bundle**: If you find discrepancies in component identification, licensing, or vulnerability data in your project's Bill of Materials (BOM), you can generate a KnowledgeBase (KB) support bundle directly from here. This bundle automatically gathers all necessary component data for the support team, streamlining the investigation process and reducing the need for additional communication when opening a support case.

Note: Bulk actions are not supported for unmatched components, except for the ignore action.

Important Considerations

When using the Match Review feature, it's essential to note the following:

- **Exclusion from Reports**: Items listed in the Match Review are not included in the current reports generated by Black Duck SCA. This includes:
  - **Notices Report**: Components marked for review will not appear in the Notices File report.
  - **Version Details > Components Report**: Components in Match Review are excluded from this report as well.
  - **SBOM Report**: Items from Match Review are not reflected in the SBOM report.
- **Inclusion in Vulnerabilities**: Although components in Match Review are excluded from reports, their associated vulnerabilities **are** included in the project version's **Vulnerabilities** tab. This means the

Vulnerabilities tab may display risks from components that have not yet been confirmed and promoted to the BOM.

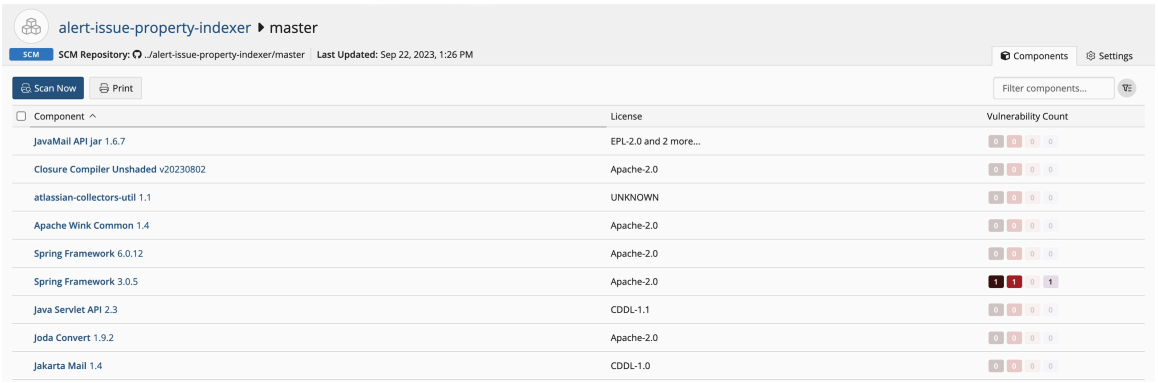
About SCM read-only BOMs

Once a GitHub repository has been scanned, the results create a read-only bill of materials (BOM).

To view a read-only BOM:

1. Select the project name using the **SCM** 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:



What's contained in a read-only BOM?

The read-only BOM is composed of the following sections:

- The header bar contains the project's name and version. It also contains the GitHub repository location and the date when the project was last scanned.
- The data table displays the following information:
  - The **Components** column lists the components found in the project. Clicking the component link displays the origin IDs where this component was found. You can also filter the component by using the *Filter components...* field.
  - The **License** column displays the license associated to the component.
  - The **Vulnerability Count** column displays the vulnerabilities linked to the component. Clicking the vulnerability count opens a modal containing a list of all the vulnerabilities associated to the selected component.

Note: The information presented in this BOM cannot be edited.

What can you do with the read-only BOM?

You can:

- Manually reinitiate a scan by clicking the **Scan Now** button. This will update the project version with any changes that were made in the project repository. Note that initiating a rescan will unset any ignored components.
- Print the BOM by clicking the **Print** button.

- Delete a project version BOM by clicking the **Settings** tab and then clicking the **Delete Version** button.

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 Black Duck KB.

Use the following pages to identify and manage risk in projects:

- Dashboard pages
- Project version page/**Components** tab
- Project version page/**Vulnerabilities** tab

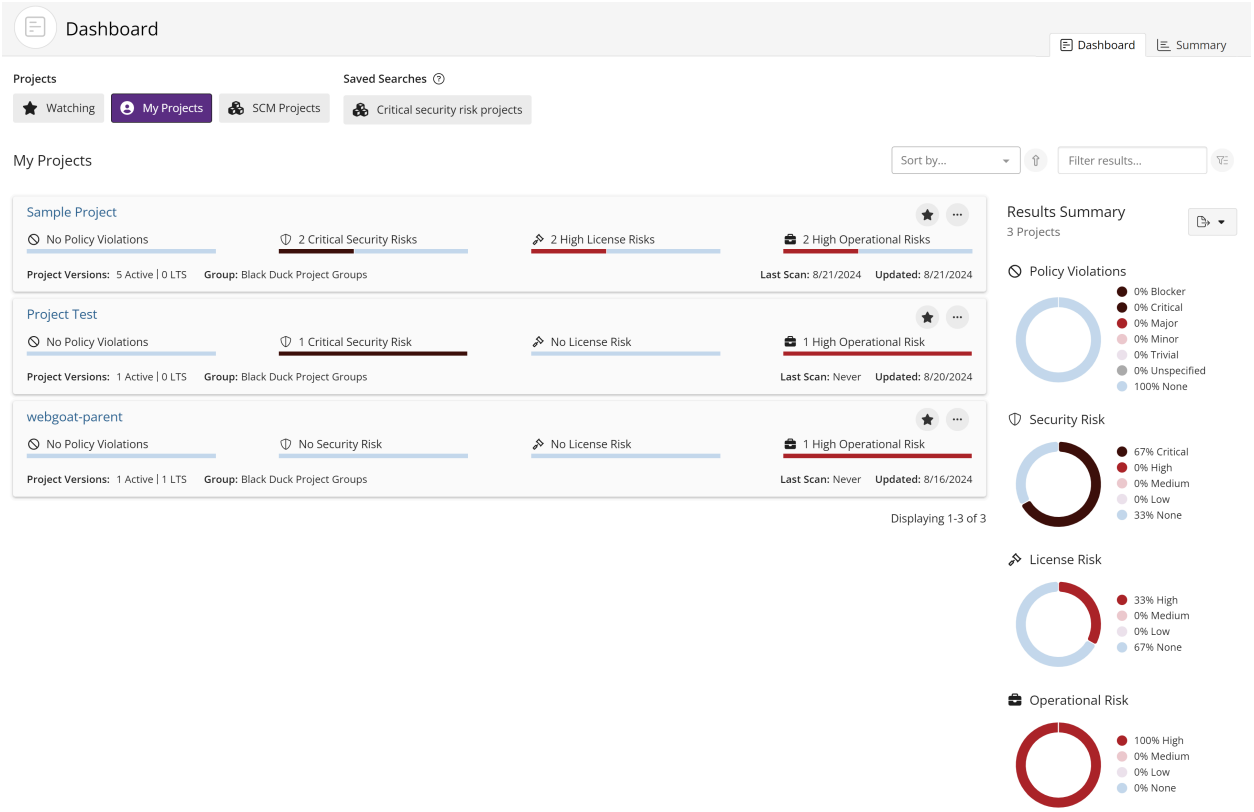
Note that the security risk values shown use CVSS v3.x or CVSS v4.x scores, depending on which security risk calculation you selected; by default, CVSS v4.x scores are shown.

Dashboards

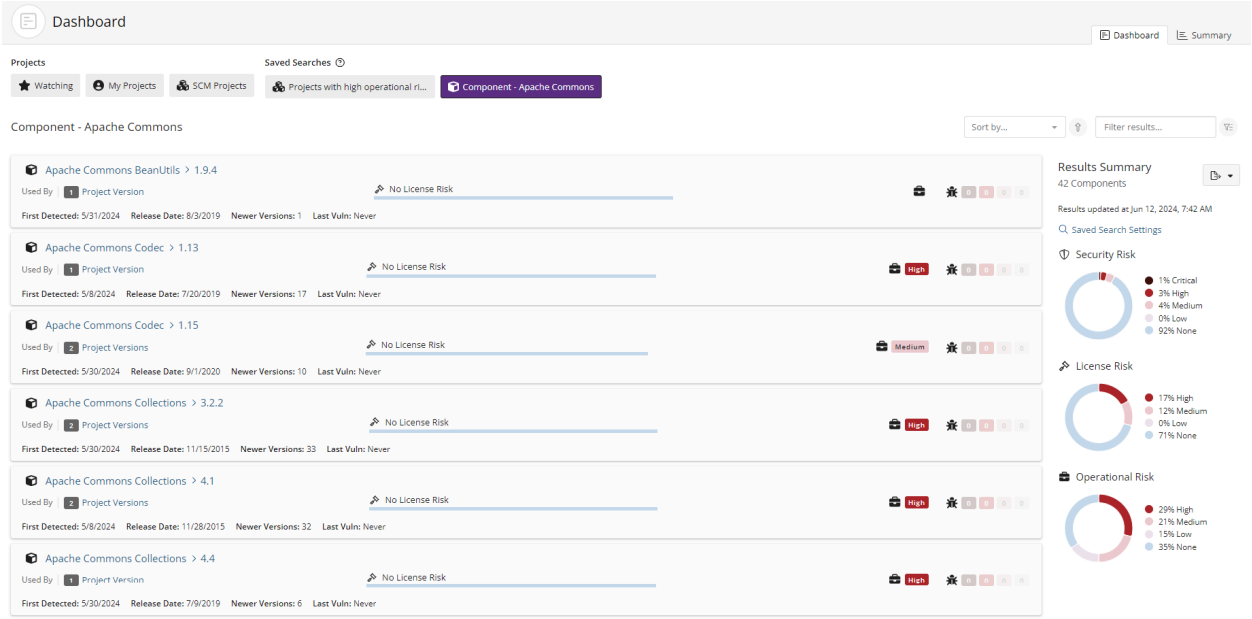
Dashboards provide a high-level overview of risk from different perspectives.

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.

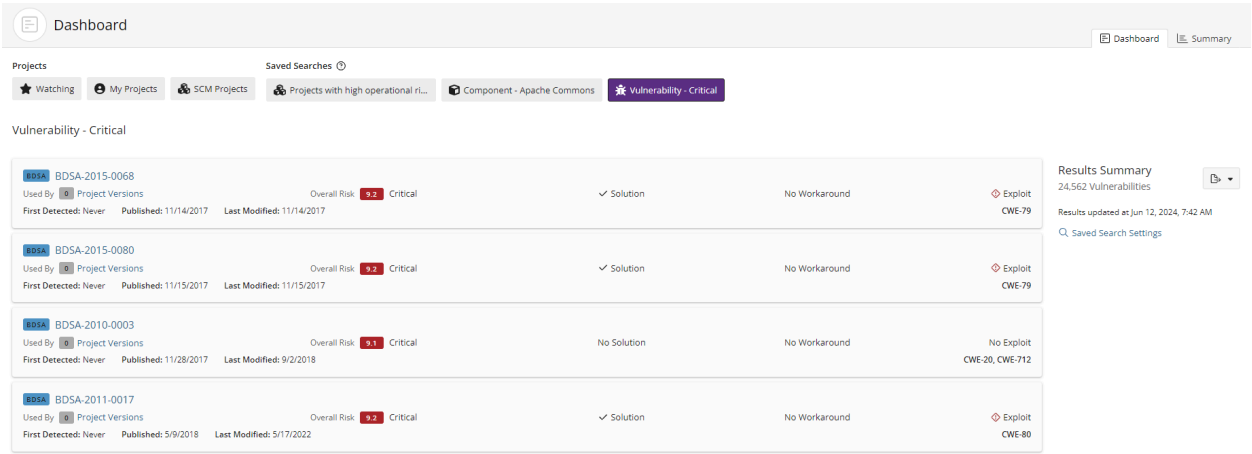
- You can view the projects that interest you by using the **Watching** or **My Projects** dashboard or create a custom dashboard by saving your project search results.



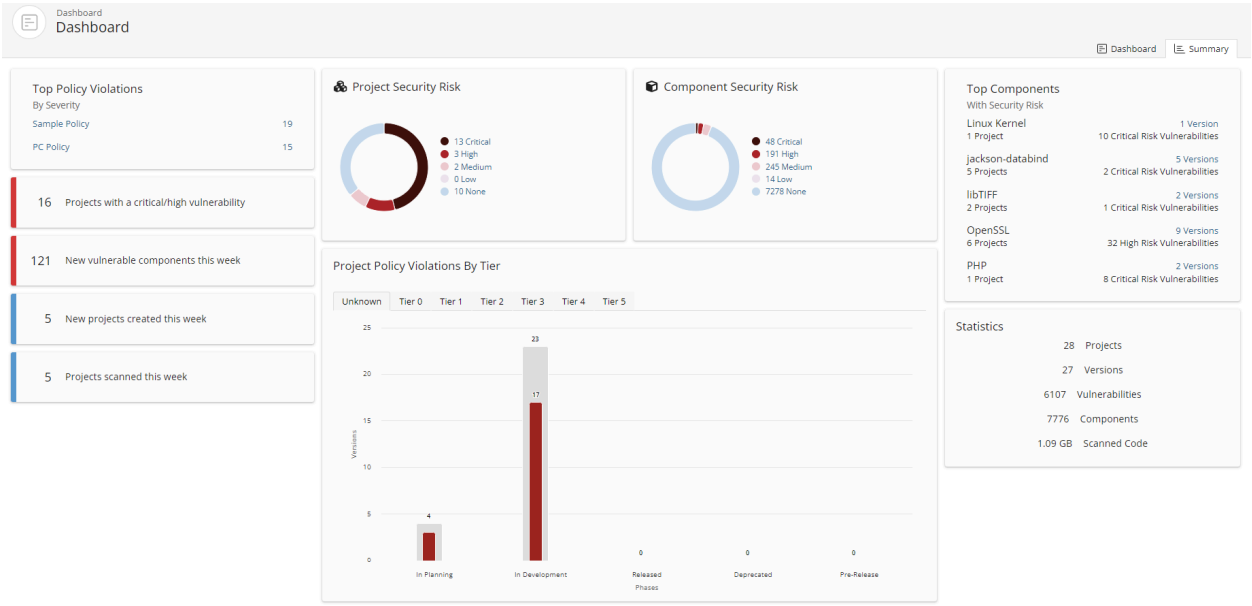
- Create a saved component search to view the components that interest you that are used in one or more projects.



- Create a saved vulnerability search to view the vulnerabilities that interest you.



- Use the Summary Dashboard to view the overall health of the projects you have permission to view and identify areas of concern.



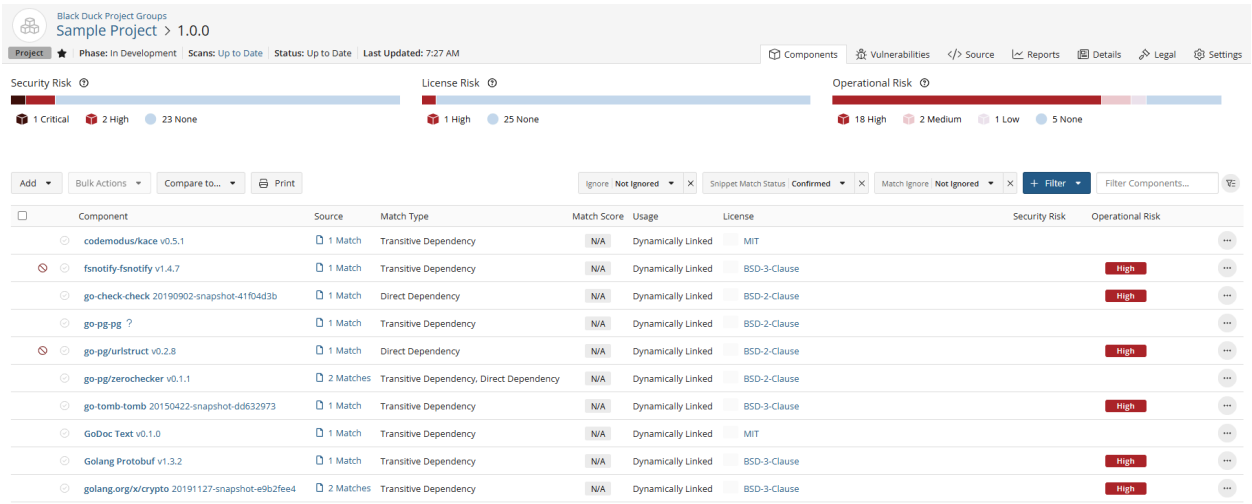
Note:

- 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 

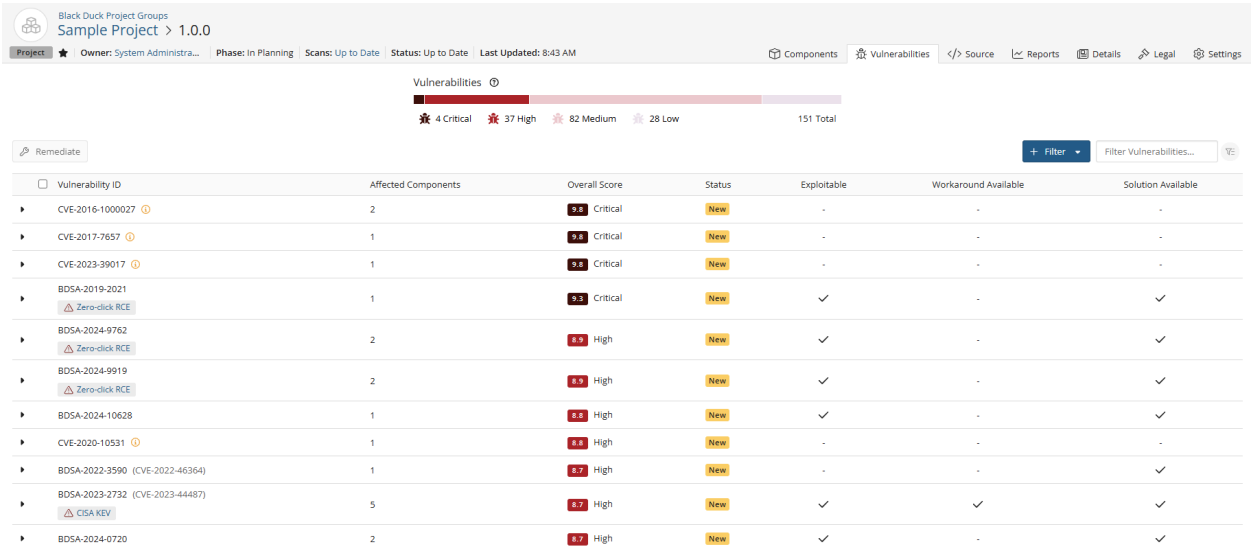
or the logo in the upper left corner of the navigation bar to view the last dashboard (Dashboard or Summary) you viewed.

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/ **Vulnerabilities** tab to view the security vulnerabilities of each severity associated with the components used in a project version.



Viewing the health of your projects

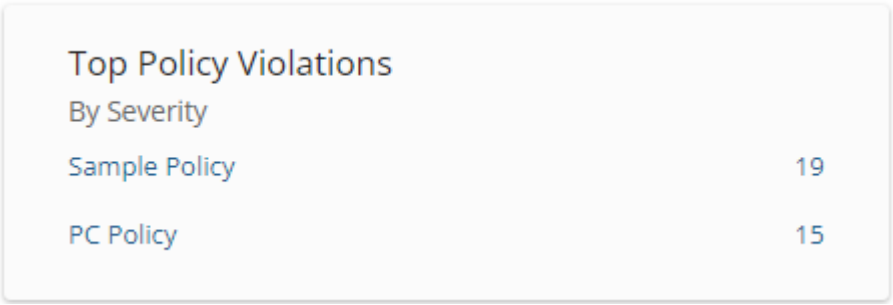
Use the **Summary** tab to view the overall health of your projects and identify areas of concern. The page consists of widgets that provide business critical information which you can use to quickly assess areas where you need to focus your attention.



Note: The **Summary** tab only displays information for the projects you have permission to view.

The following describes each widget shown on the **Summary** tab and, where available, how to view additional information. Note that the security risk values shown use CVSS v3.x or CVSS v4.x scores, depending on which security risk calculation you selected; by default CVSS v3.x scores are shown.

Top Policy Violations



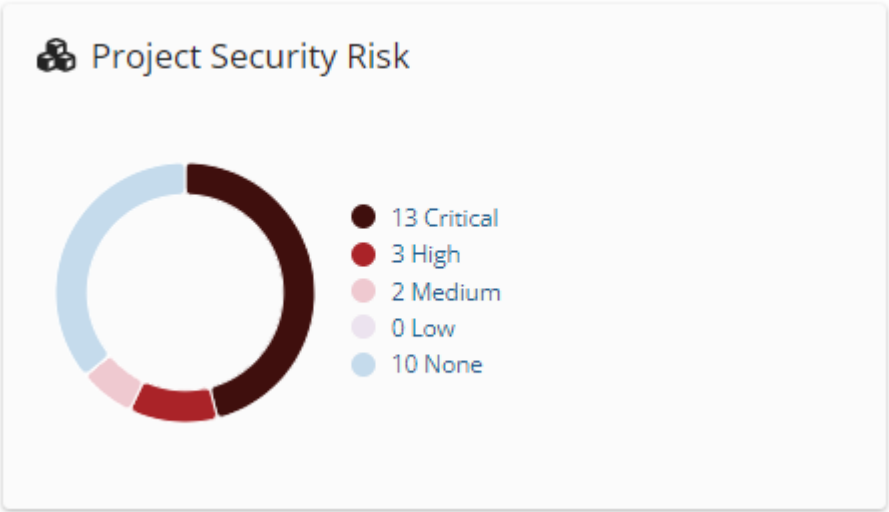
The **Top Policy Violations** widget displays up to the top five policy violations across all projects that you have permission to view.

Policy rules are listed by severity level and then by the number of policy violations, in descending order. If policy rules do not have severity levels assigned to them, the widget displays the top five policy violations, in descending order by the number of violations.

- 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.

Project Security Risk

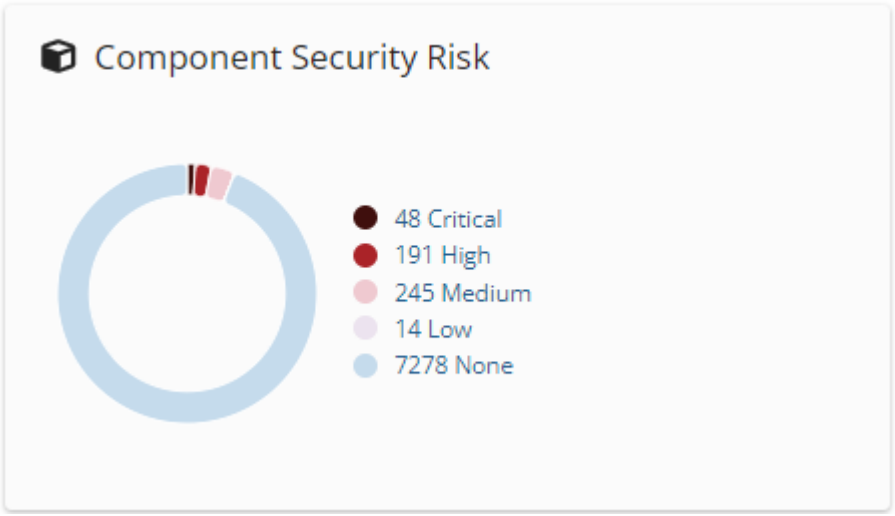


The **Project Security Risk** widget displays the number of projects you have permission to view for each level of security risk.

Note that this widget counts the highest security risk level for a project, not all security levels affecting a project. For example, if a project has medium and low security risks, it is counted as a project with medium security risk; it is not included as a project with low security risks.

Hover over the graph to view the number of projects with that level of security risk.

Component Security Risk

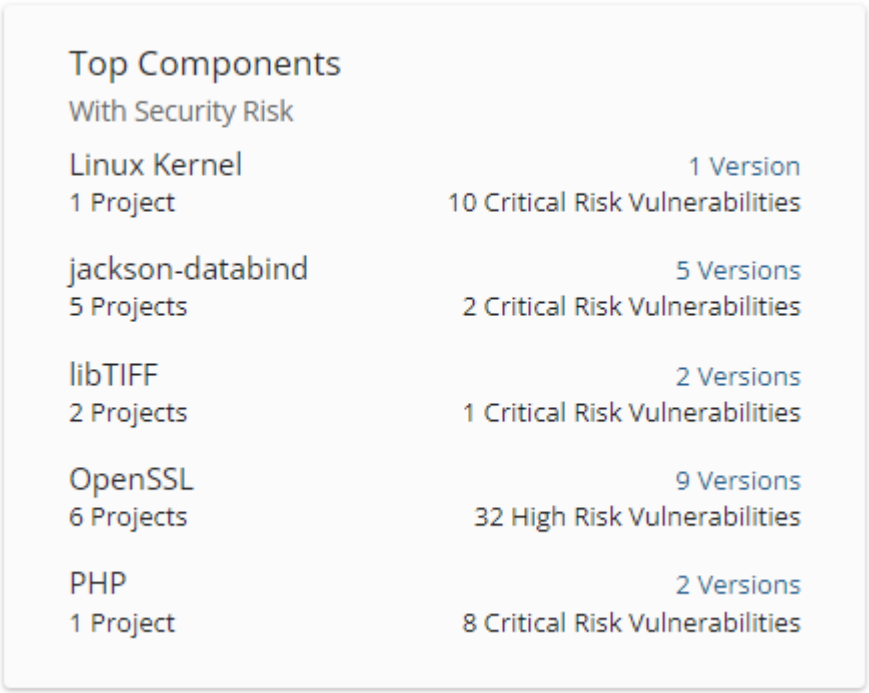


The **Component Security Risk** widget displays the number of components in projects you have permission to view for each security risk level.

Note that the widget counts only the highest security risk for a component. For example, if a component has medium and low security risks, it is counted as one component with a medium security risk.

Hover over the graph to view the number of components with that level of security risk.

Top Components



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:

- 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.

Components are organized by security risk, with those components with the highest risk listed first.

Select the specific version or number of versions to view the Component Version Details page.

Projects with critical/high vulnerabilities

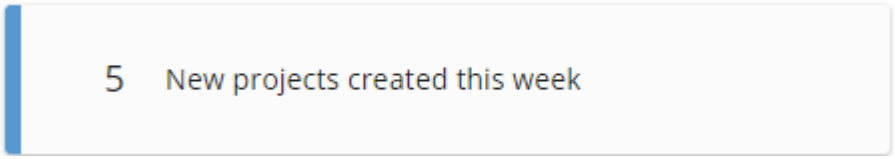


The **Projects have a critical/high vulnerability** widget displays the number of projects with versions that contain components with a critical and/or high security risk.

New vulnerable components this week

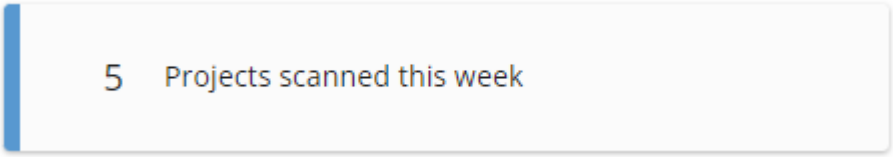
The **New vulnerable components this week** widget displays the number of components the Black Duck KB mapped a vulnerability to in the past seven days, including today.

New projects created this week



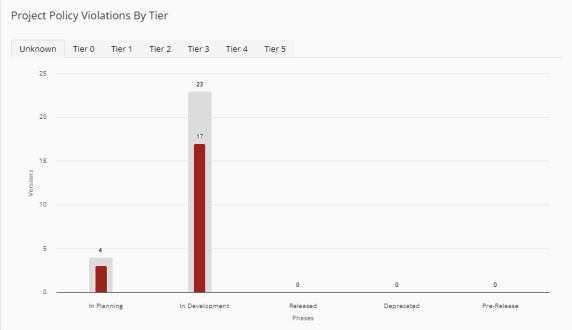
The **New projects created this week** widget displays the number of projects that you have permission to view that have been created in the past seven days, including today.

Projects scanned this week



The **Projects scanned this week** widget displays the number of projects with scans from the past seven days, including today.

Project Policy Violations by Tier

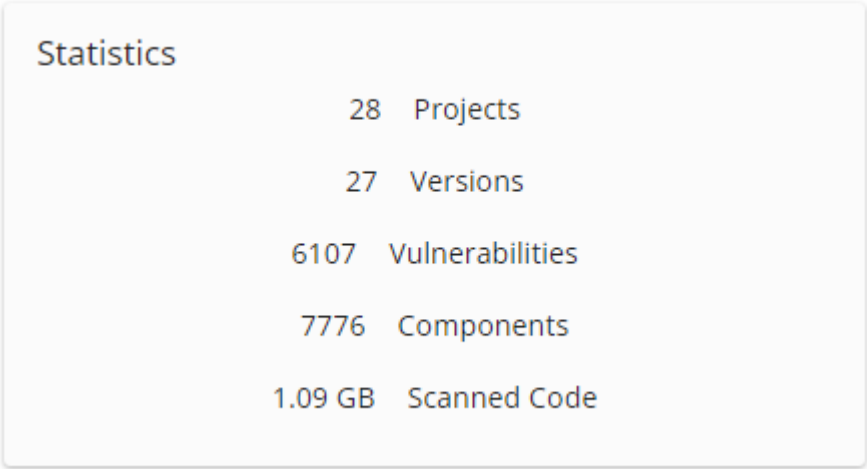


The **Project Policy Violations by Tier** widget displays the total number of projects by phase that have a policy violation, grouped by tiers.

- 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.

Statistics



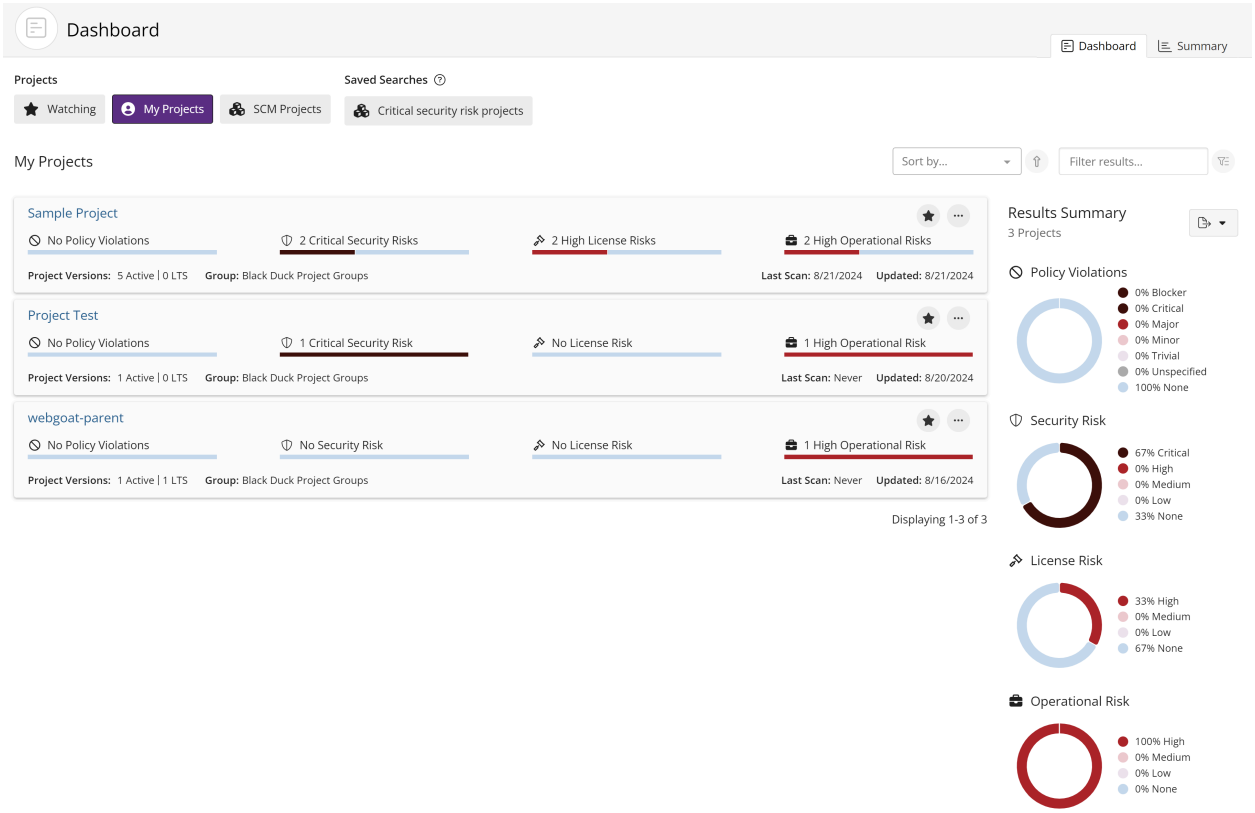
The **Statistics** widget displays the following information:

- **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, *including* ignored components.
- **Scanned Code** lists the number of GBs scanned for all scans.

Viewing overall risk

Project level

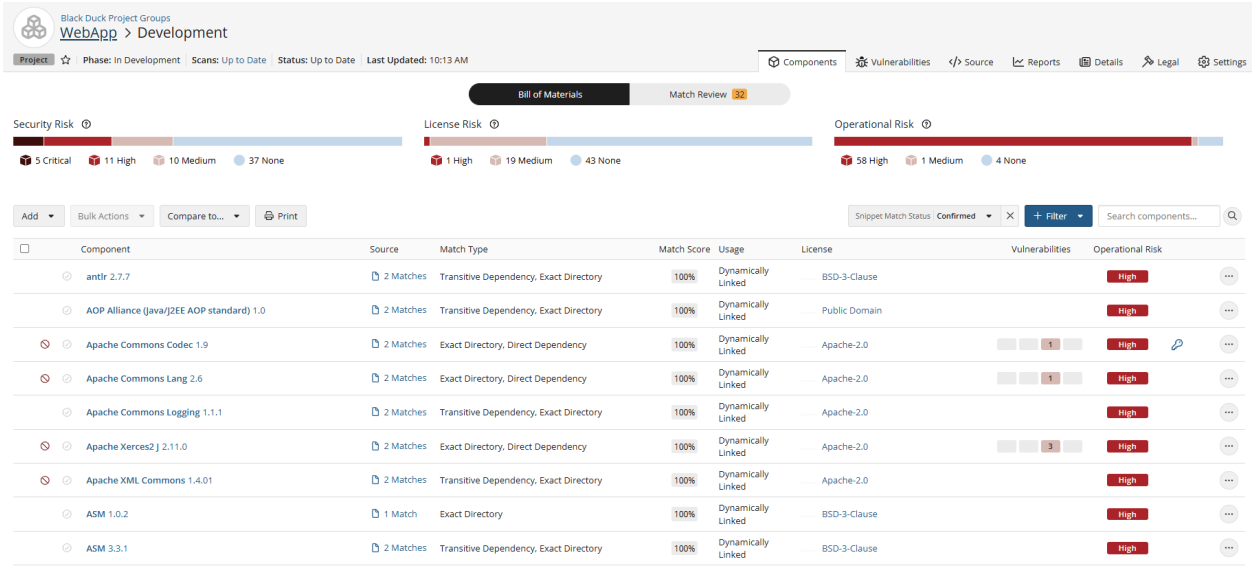
The **Watching**, **My Projects**, and saved search dashboards show the overall risk across all projects where you are a project team member.



Click here for more information about using this page to understand security vulnerabilities associated with your projects.

Project version level

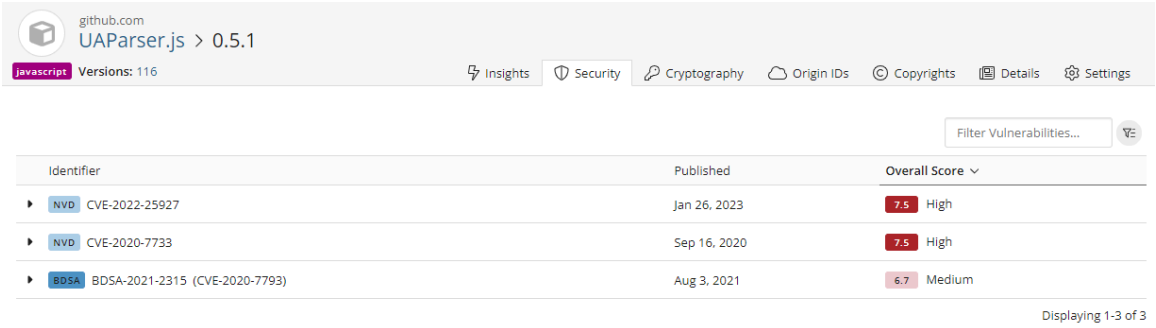
Risk information for a specific project version is shown on the project version's **Components** tab.



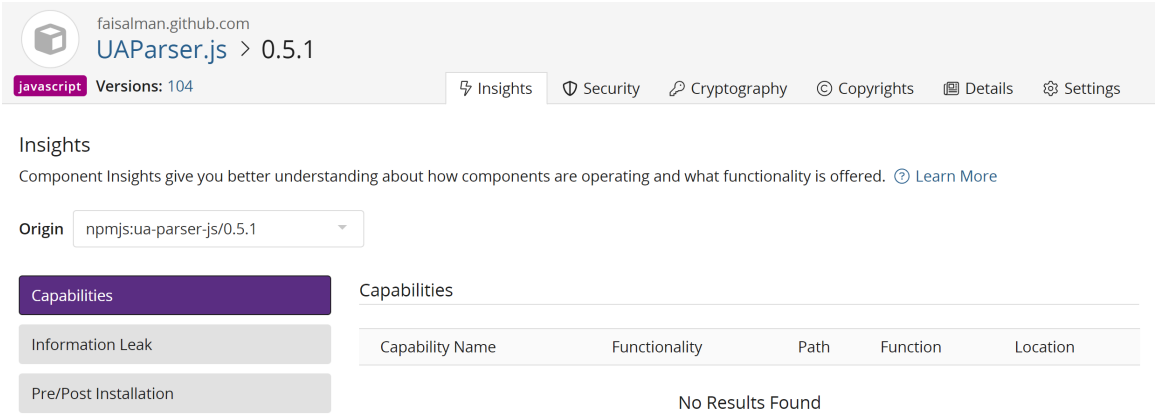
Also known as the BOM page, this tab shows all type of risks associated with each component in the project version's BOM.

Component version level

Risk information for a specific component version is shown in the **Security** tab.



Additionally, if a component version's origin has been flagged with further Component Intelligence risks, you can click the menu button found at the end of the component's row and select **Insights** to view the component version's Insights page.



On this page, you can view a component version's insights:

Capabilities: Native functionality offered by the component. Capabilities currently detectable:

- Network Communications: The software component possesses the ability to communicate over the network.
- System Operation: The software component possesses the ability to execute system level commands.
- Cryptography: The software component possesses the ability to secure communications.
- Serialization: The software component possesses the ability to convert data structures to byte streams.
- File System Access: The software component possesses the ability to interact with the local file system.
- Compression: The software component possesses the ability to compress data.
- Security operations (Sanitizers & Validators): The software component possesses the ability to sanitize and/or validate input data.

Information Leak: Detected valid IPv4/6 addresses including domains identified within the codebase.

Pre/Post Installation: Various component configuration findings and observations. Observations may signify security and/or operational concerns.

Understanding the types of risk

There are three types of risk being assessed across all projects, project versions, and component versions:

- **Security Risk**. Project and project version security risk is based on the vulnerabilities associated with the components that comprise the project and the project version's BOM. Component version security risk is 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 v3.x or CVSS v4.x scores, depending on which security risk calculation you selected; by default, CVSS v3.x scores are shown.

  Possible risk categories are Critical, High, Medium, Low, and None.

- **License Risk**. Refers to the legal and compliance challenges that arise from using licensed software. These risks include failing to meet license obligations, dealing with conflicting licenses, and unintentionally exposing proprietary code.

  License risk is assigned one of four categories of overall risk: High, Medium, Low, and None.

  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 are High, Medium, Low, and None.

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 versions 2, 3.x, and 4.x scores. By default, Black Duck displays CVSS v4.x scores.

CVSS scores have the following values:

- None: 0.0
- Low risk: 0.1 - 3.9
- Medium risk: 4.0 - 6.9
- High risk: 7.0 - 8.9

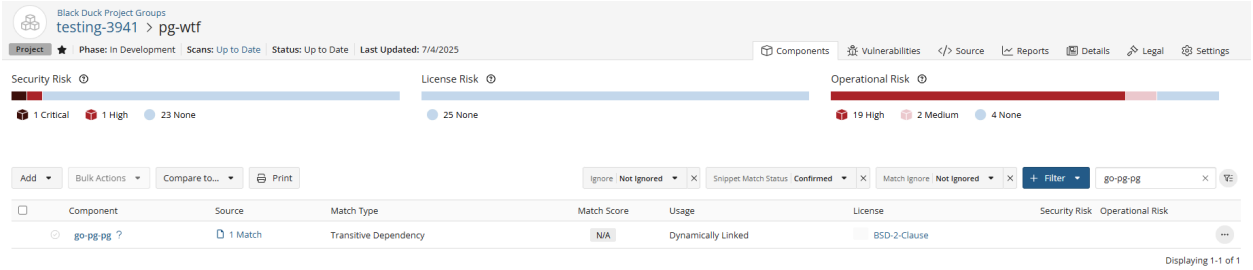
- Critical risk: 9.0 - 10.0

For more information on the Common Vulnerability Scoring System, see the CVSS 3.x and 4.x specification documents.

Note: CVSS v2 will no longer be supported as the highest-priority CVSS ranking. However, it will still be displayed if no CVSS v3.x or CVSS v4.x score exists for a vulnerability.

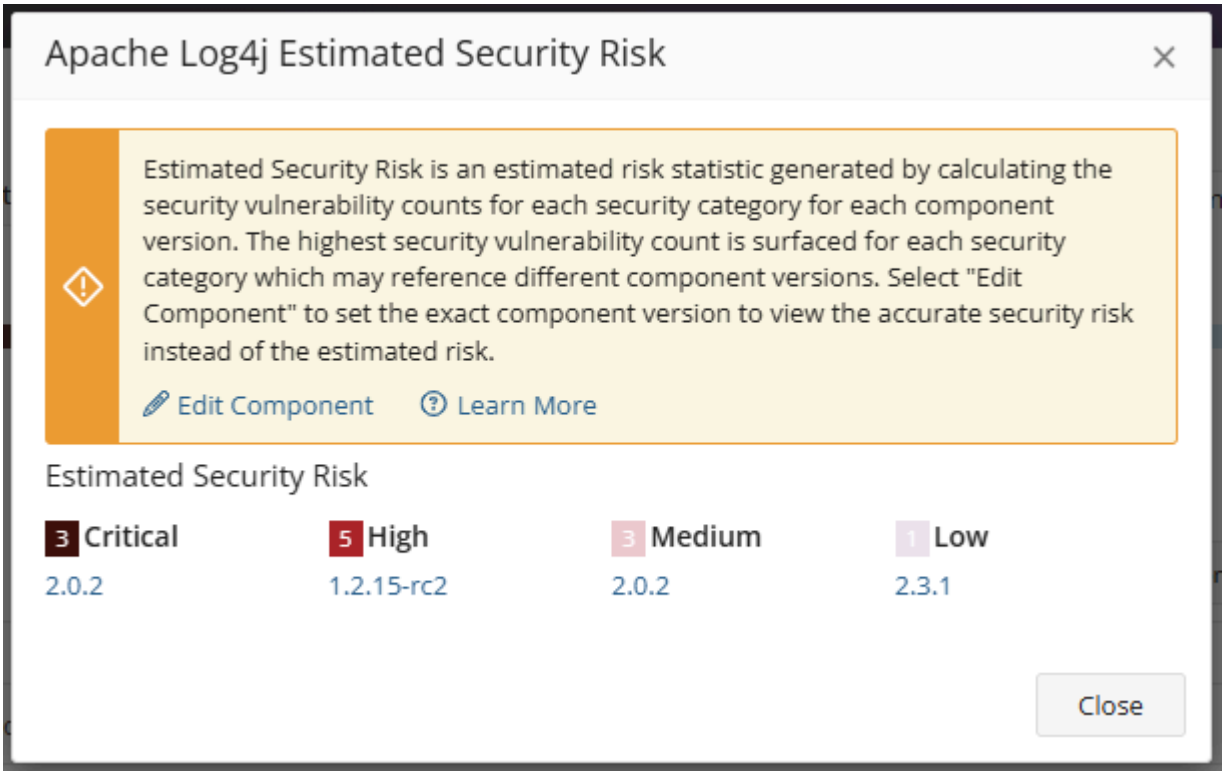
Estimated Security Risk

This estimated risk statistic is used when a component in the BOM has an unknown version. Such components are marked with a ? icon next to the component name.



Estimated security risk is formulated by looking at all the versions of a component sorted by security vulnerability severity category and calculating the maximum vulnerability count for each severity category for each component version. The maximum vulnerability count for each severity category of the component with no version specified is shown in the components's **Security Risk** column.

Clicking the security risk icons for a component with an unknown version displays a popup which provides more information on which component versions are known to fall into each specified risk category.



The highest severity category counts may reference different component versions. For example:

- Component version 1.1 has 2 Critical, 3 High, 15 Medium, 4 Low
- Component version 1.2 has 2 Critical, 4 High, 12 Medium, 1 Low

The estimated security risk by severity category for this component with unknown versions would return as 2 Critical, 4 High, 15 Medium, 4 Low on the BOM.

Users should choose the exact version used in the application to view the accurate risk instead of the estimated risk. This estimated risk information is provided to help prioritize what components to review first. Users are encouraged to use estimated risk information in conjunction with BD Policy Management to further prioritize what components to triage first based on their company's security policies.

Note: The information presented is only a statistical data estimation. As a result, the estimated security risks will not have CVE data.

Suggested workflow

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).
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 interest 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/**Vulnerabilities** 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.

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 **Vulnerabilities** 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.

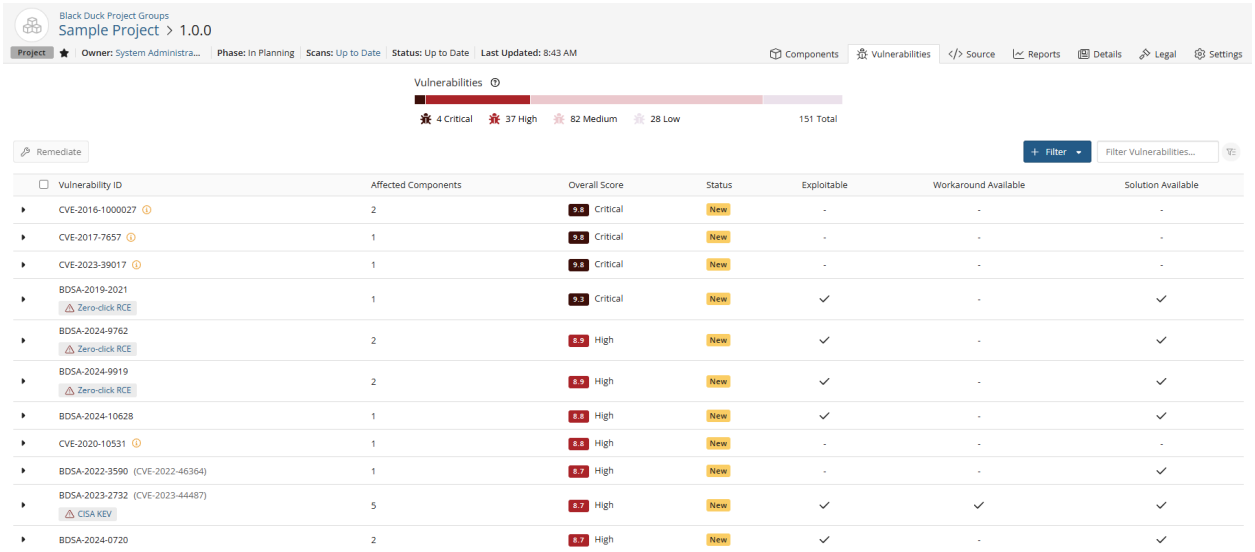
For example, suppose vulnerability X is found; NVD identifies it as CVE1 and BDSA identifies it as BDSA1. NVD has also found vulnerability X in component version origin A and component version origin B but BDSA has only found it in component version origin B (BDSA has either decided NVD is incorrect or not evaluated it). If your BOM has component version origin A, the project version's **Vulnerabilities** 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 **Vulnerabilities** tab to view the security vulnerabilities associated with the components used in a project version.

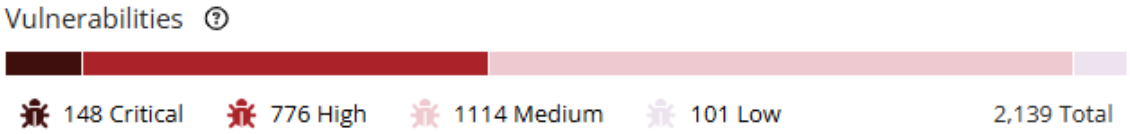
The information shown uses CVSS v3.x or CVSS v4.x scores, depending on which security risk calculation you selected; by default CVSS v3.x scores are shown.



This page has these sections:

- Vulnerabilities graph
- Filters
- Vulnerabilities table

Vulnerabilities graph



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

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 Vulnerabilities graph to view all vulnerabilities that share the same level of risk.

Filters

Use the **Filter components** field to view specific components. Click Add filter 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

This section displays all vulnerabilities identified in the selected project version. For each entry, you'll see the vulnerability ID, the number of affected components, the overall risk score, current status, and indicators for exploitability, available workarounds, and available solutions.

Click ▸ next to a vulnerability to reveal additional details.

The vulnerabilities table lists the following information for each vulnerability:

Column	Description
Vulnerability ID	The identifier, value associated with this vulnerability, and any vulnerability tags (if applicable). 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. Users with the appropriate role can also use this section to remediate the vulnerability by clicking the 🔧 icon next to the affected component.
Affected Components	Displays the number of components in the project version affected by this specific vulnerability.
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. • 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 confirmation of the technical details of a vulnerability, the existence of any patches or workarounds, and the availability of exploit code or techniques. The Base score reflects the overall basic characteristics of a vulnerability that are constant over time and user environments: • Attack Vector (AV) • Attack Complexity (AC) • Priviledges Required (PR) • User Interaction (UI) • Scope (S) • Confidentiality (C) • Integrity (I) • Availability (A) • Exploit Code Maturity (E) • Remediation Level (RL) • Report Confidence (RC) For more information, see the CVSS specification document section on Exploitablility Metrics. 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. The Impact score reflects the possible impact of successfully exploiting the vulnerability, considering the integrity, availability, and confidentiality impacts.
Status	Remediation status of this vulnerability. Possible values are: Duplicate, Ignored, Needs Review, New, Mitigated, Patched, Remediation Complete, or Remediation Required.
Exploitable	Indicates whether an exploit for this vulnerability is available: • – No exploit available • ✔ Exploit available

Column	Description
Workaround Available	Indicates whether a workaround for this vulnerability is available: • – No workaround available • ✔ Workaround available
Solution Available	Indicates whether a solution for this vulnerability is available: • – No solution available • ✔ Solution available

Viewing component version vulnerabilities

Use the component version page's **Vulnerabilities** tab to view the vulnerabilities associated with specific component versions in Black Duck SCA. Understanding these vulnerabilities helps you assess security risks and prioritize remediation efforts effectively.

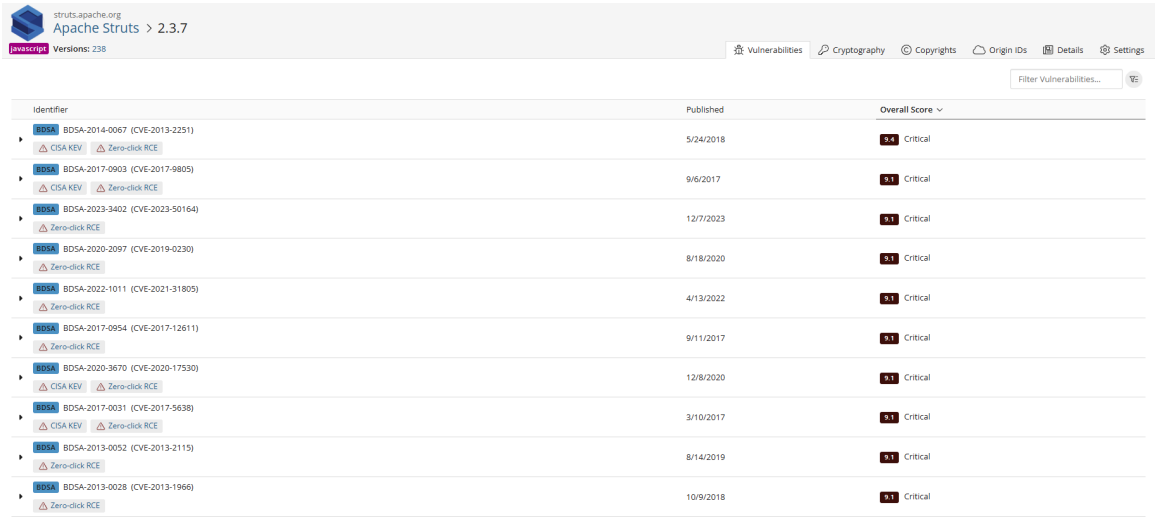
The information shown uses CVSS v3.x or CVSS v4.x scores, depending on which security risk calculation you selected; by default CVSS v3.x scores are shown.

Accessing a component versions vulnerability page

You can access a component version's vulnerability page through multiple pathways:

- **From the Project Version Page**: Navigate to the Project Version page and click on a specific component listed under the **Components** tab to open its Component Version page. Then select the **Vulnerabilities** tab to view its associated vulnerabilities.
- **Using the Find Page**: Search for the component directly on the **Find** page by entering its name or other identifying details. Select the desired component from the search results to view its Component Version page. Then select the **Vulnerabilities** tab to view its associated vulnerabilities.

Vulnerabilities list

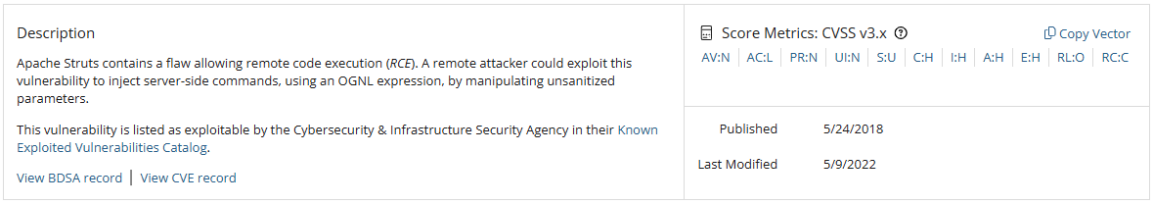


This Vulnerabilities list displays all identified vulnerabilities linked to the component version which includes key details such as:

- **Vulnerability ID**: The unique identifier assigned to a specific security vulnerability. It serves as a reference point for tracking and managing vulnerabilities across various systems and databases. Formats for Vulnerability IDs include:
  - **CVE (Common Vulnerabilities and Exposures)**: Example: `CVE-2023-12345`. CVEs are standardized identifiers used globally to catalog publicly disclosed vulnerabilities.
  - **BDSA (Black Duck Security Advisory)**: Example: `BDSA-2023-4567`. BDSAs are proprietary identifiers used by Black Duck to provide enhanced information about vulnerabilities, including detailed analysis and remediation guidance.
- **Published**: The date when a vulnerability was officially disclosed or published by its source, such as the National Vulnerability Database (NVD) or a Black Duck Security Advisory (BDSA). This date marks when the vulnerability became publicly known and available for organizations to assess and address.
- **Overall Score**: The composite metric used to evaluate the severity and potential impact of a security vulnerability. It typically integrates various factors to provide a comprehensive view of the risk posed by the vulnerability.

Viewing detailed vulnerability information

Select a vulnerability to expand more information about it:



- **Description**. A brief description of the vulnerabilities found for this component version.
- **Score Metrics**. See Exploitability Metrics for more information.
- **Published**. The date this vulnerability was discovered and published.
- **Last Updated**. Date when a vulnerability for the component was last updated in Black Duck (such as updates from Black Duck KnowledgeBase, a user manually changing the associated vulnerability, and so on).

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 Black Duck 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. Black Duck Detect analyzes the code

When the **--detect.impact.analysis.enabled** property in Black Duck Detect to set to **true**, Black Duck 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, Black Duck Detect creates a file which will be used by 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 Black Duck 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 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.

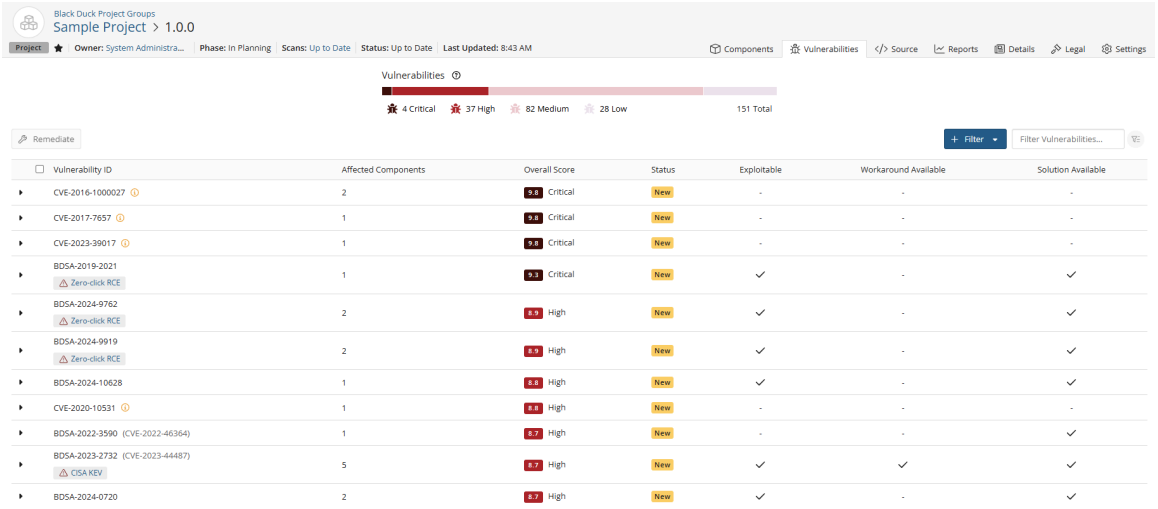
Black Duck displays the vulnerabilities for a component that have a method that is being called in the project version's **Vulnerabilities** tab.

Viewing reachable vulnerabilities

To view reachable vulnerabilities:

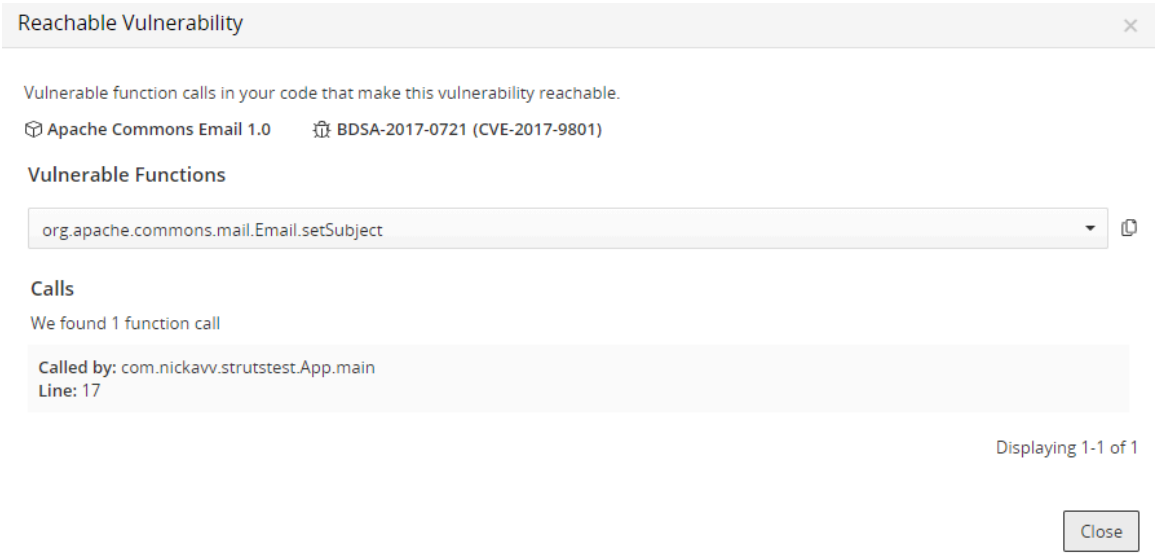
1. Set the **--detect.impact.analysis.enabled** property in Black Duck Detect to **true** to enable vulnerability impact analysis.

2. Once scanning completes and Black Duck has built the BOM, open the project version's **Vulnerabilities** tab which lists the security vulnerabilities for this project version.
3. Select a vulnerability from the table to view a table which lists the vulnerabilities for this component.



⚠ Reachable 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 ⚠ Reachable to view more information for an individual BDSA record.



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 Black Duck Detect version 6.5 or later (and Black Duck Detect Desktop that uses Black Duck Detect 6.5 and later). Set the **--detect.impact.analysis.enabled** property in Black Duck Detect to **true** to enable vulnerability impact analysis.
- This feature is for Java applications only. Black Duck Detect looks at Java `.class` files only.
- Black Duck 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 SCA provides detailed information on a security vulnerability depending on whether you are viewing:

- BDSA record
- CVE record
- EUVD

Black Duck Security Advisories

Black Duck Security Advisories (BDSAs) are a proprietary vulnerability data feed curated by the Black Duck Cybersecurity Research Center (CyRC). They provide more comprehensive coverage of vulnerabilities compared to the National Vulnerability Database (NVD), offering timely insights on severity, impact, and exploitability.

BDSAs include actionable remediation guidance, detailing fixed versions, patches, known exploits, and workarounds. Additional validated references can be found under the Technical page of each BDSA record.

The CyRC team conducts thorough research, cross-checking vulnerabilities against affected component versions for accuracy. If a BDSA is not linked to a CVE record, it means further research has determined that the component is not impacted by the vulnerability. BDSAs are updated frequently, often on an hourly basis, to reflect new vulnerabilities.

In contrast to BDSAs, NVD CVE records are not typically cross-validated and may update more slowly. Users should view BDSA records as complementary insights that enhance understanding and decision-making regarding open source security vulnerabilities.

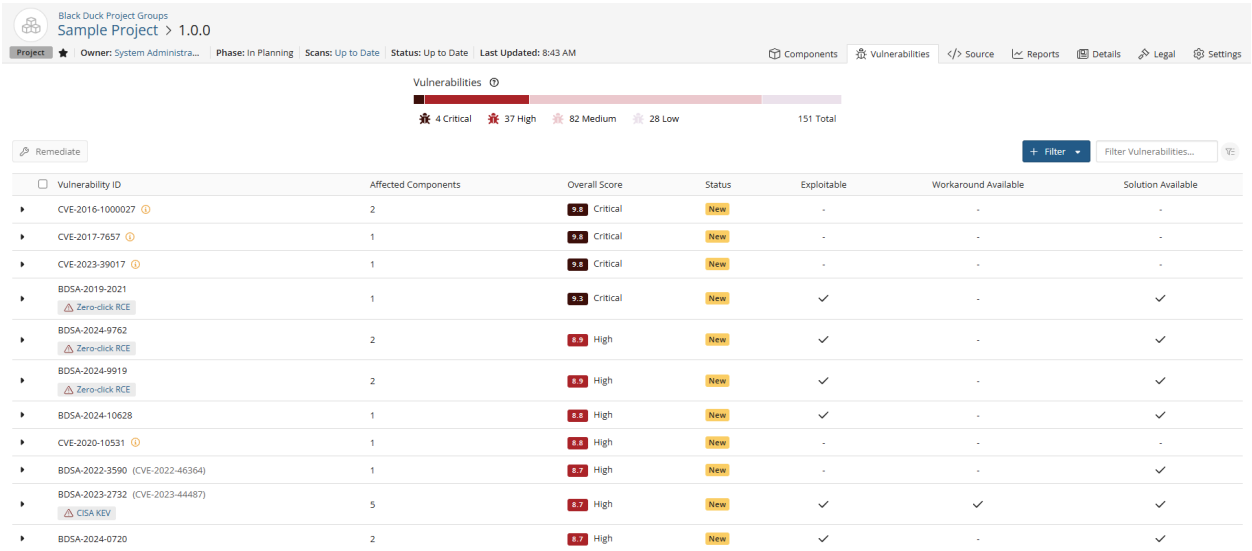
Viewing a BDSA record

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 **Vulnerabilities** tab for a project version to view the vulnerabilities for a project version BOM.



Vulnerabilities with identifiers beginning with BDSA are proprietary findings curated by the Black Duck Security Advisory (BDSA) team.

Click ▶ next to the vulnerability to reveal additional details. Then click the **View BDSA record** link to open the full 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:

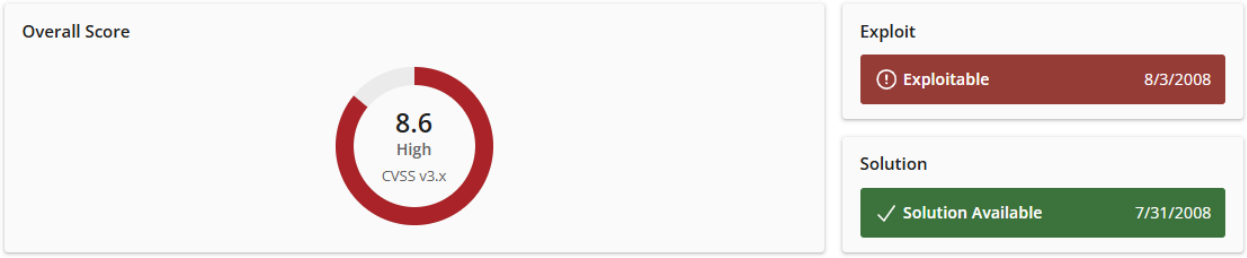
- The header bar displays the following information:



  - **BDSA ID**: The unique identifier assigned to the BDSA, which serves as a reference for the specific vulnerability.
  - **Related Records**: This section lists links to associated Common Vulnerabilities and Exposures (CVE) and European Union Vulnerability Database (EUVD) records, offering users additional context and remediation options related to the same vulnerability or component.
  - **Published**: The date when the National Vulnerability Database (NVD) published the CVE. This indicates when the vulnerability was officially recognized and made public.
  - **Updated**: The last modified date by the NVD, reflecting when the CVE record was last updated with new information or corrections.
  - **Under Review**: Some BDSAs display the status "This vulnerability is currently under review with Black Duck." Reviews are carried out on a cyclic basis for a defined period — sometimes up to a year — depending on the nature of the vulnerability and whether further information may become available. For example, embargoed vulnerabilities may have no technical information available for six to nine months after initial disclosure. Once all relevant information has been gathered, analyzed, and included in the BDSA, the Under Review status is removed. The review process is essential for

ensuring scoring accuracy, as the publication of exploit code can impact temporal metrics and the overall severity rating. The **Updated** date always reflects the most recent changes to the record.

- The **Overall Score** graphic provides the following information: represents the



Shown here are the:

  - **Common Vulnerability Scoring System** (CVSS): Score for the BDSA.
  - **Exploit**: If a known exploit exists, this link provides a technical description of the vulnerability, detailing how it can be exploited. It includes examples illustrating how a specially crafted message can lead to arbitrary content injection into the HTTP response, highlighting the conditions necessary for a successful attack. If there is no known exploit, the label will simply state "No Known Exploit," indicating that there are currently no publicly available methods to exploit this vulnerability.
  - **Solution**: This link offers guidance on remediating the vulnerability. It includes references to relevant advisories, vendor upgrades, and patches, along with a proof-of-concept example that illustrates how the vulnerability operates. If there is no known solution, the label will state "No Known Solution," indicating that no remediation steps are currently available.

- The **Description** section provides a detailed overview of the vulnerability associated with the BDSA. It includes information on the nature of the vulnerability, its potential impact on affected systems, and the conditions under which it can be exploited. The description may also outline the software or components affected, attack vectors, and any known mitigations or recommendations for addressing the vulnerability. This information is crucial for users to understand the context and significance of the CVE, facilitating informed decision-making regarding remediation strategies.
- The **Vulnerability Tags** section highlights specific attributes or classifications associated with the vulnerability record, providing context and important details to help users understand the nature and implications of the vulnerability. Each tag may include a **Solution** section, which outlines available fixes, patches, and references to relevant commits and versions.
  - **Zero-click Remote Code Execution (RCE)**. This vulnerability can result in the execution of code on the system, triggered by a remote attacker without requiring or relying on any third party action.
  - **Malicious Code Identified**. This software contains code with malicious intent and is designed to have harmful or destructive consequences if executed within your system.
  - **Embargoed Vulnerability Details**. Technical details of this vulnerability are currently under embargo and the details are not published by the vendor at this time. An embargo remains in place for a fixed period. The BDSA record will be reviewed and updated with further details where possible once the embargo has been lifted.
  - **Unconfirmed Vulnerability**. This vulnerability does not have a code-based fix because the vendor has decided that the behavior of the component is intended and does not believe there is a vulnerability. The vendor may have resolved this issue by providing clarification in their documentation.
  - **Automated Security Advisory (ASA)**. Automated Security Advisories are automatically created by Black Duck's Cyber Security Research Center using automated AI tools. ASAs are created from various trusted security feeds such as the GitHub Security Advisories (GHSA) feeds along

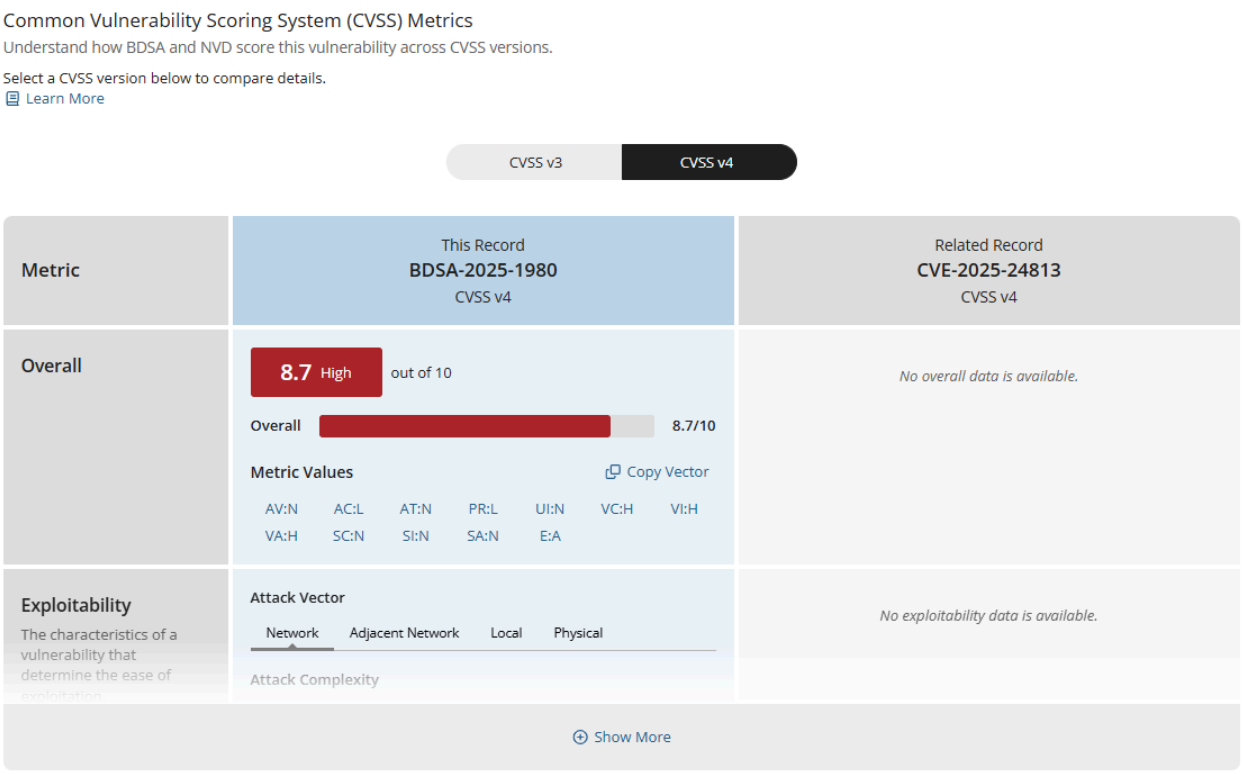
with automated vetting using AI tooling. These advisories are designed to supplement the BDSAs identified and verified by our Cyber Security Research Center.

- **CISA Known Exploited Vulnerability**. This vulnerability is listed in the Cybersecurity & Infrastructure Secrity Agency's (CISA) catalog. All federal civilian executive branch agencies are required to remediate vulnerabilities in the KEV catalog within prescribed timeframes. Please visit CISA's Known Exploited Vulnerability Catalog page for more information.
- **AI Assisted**. AI Assisted Security Advisories are automatically created by the Black Duck Cybersecurity Research Center using automated AI tools. These BDSAs have not been independently verified by the BDSA team but are created using automated processes and generative AI assistance. These advisories are designed to supplement the BDSAs identified and verified by the Cybersecurity Research Center.

  Note: This tag has been deprecated and will no longer be included in new BDSA vulnerabilities.
- **Potential Remote Code Execution**. This vulnerability has the potential to cause a remote code execution vulnerability however this has not been proven in the wild or via a known exploit.
- **Remote Code Execution Requiring User Input**. This vulnerability can lead to a remote code execution vulnerability however it requires specific user input to be provided.
- **Rapid Review**. BDSA Rapid Review vulnerabilities are produced en masse to expand our BDSA coverage for lower popularity components. These BDSAs are created with AI and automation assistance, including generating the text description. These have not been researched independently by our Vulnerability Analysis team but have been verified for affected component version ranges. These BDSAs enable proactive vulnerability notification for an expanded coverage of less popular OSS components using highly scalable technology.

- The **Scores and Metrics** section displays the scores for the related BDSA and NVD records (if applicable), based on the Common Vulnerability Scoring System (CVSS). Select a value above the graph to view the information in the graph and details below.

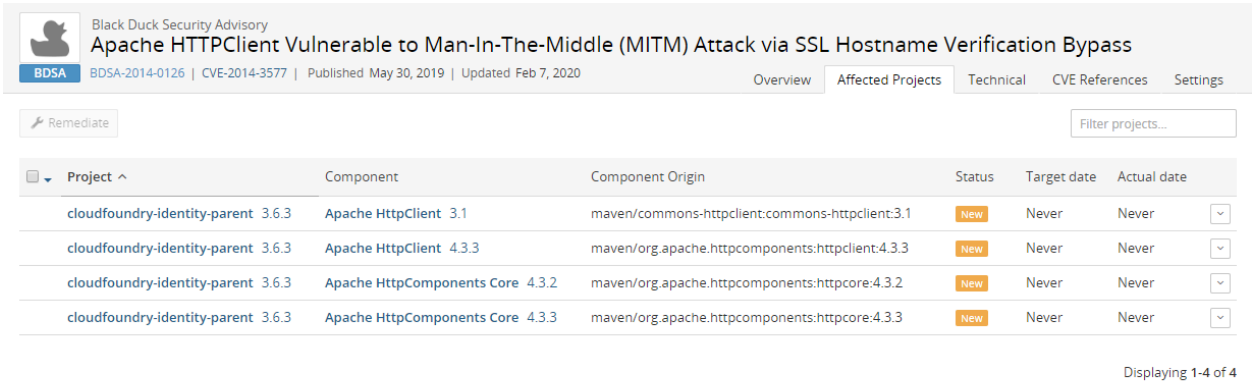
  This section may also display a comparative, side-by-side graph if the vulnerability also has a NVD record.



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.



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:

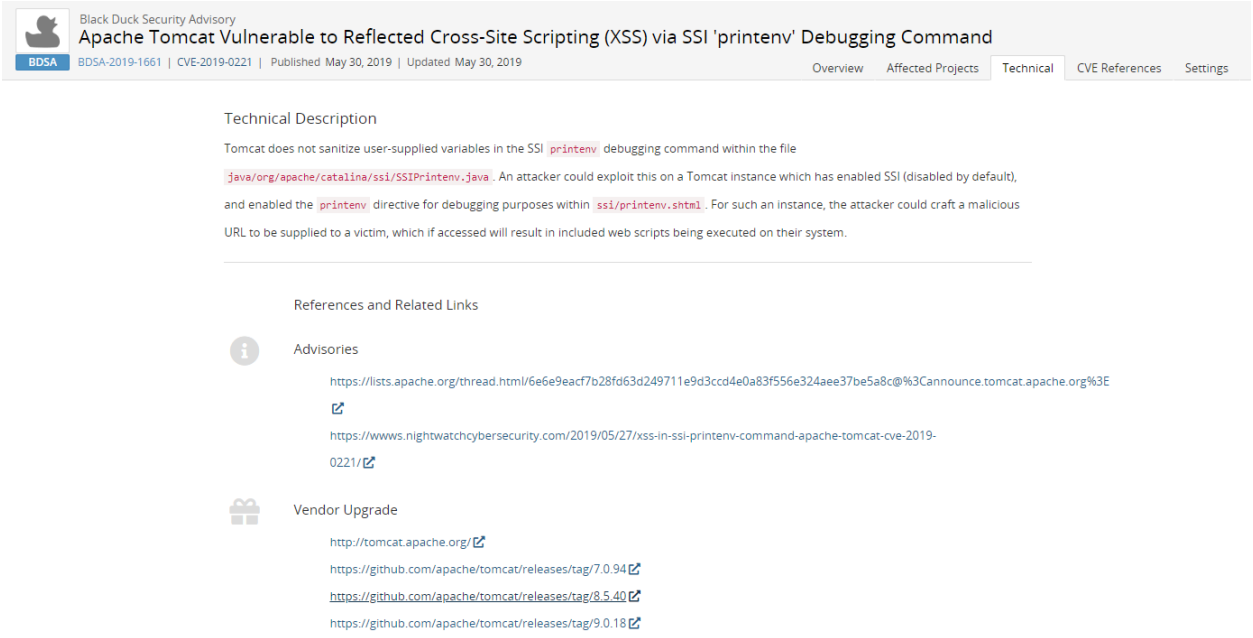
- **View all vulnerabilities** to view all vulnerabilities affecting this project version.
- **View related files to view** to display the **Source** tab filtered to display the affected files.

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.



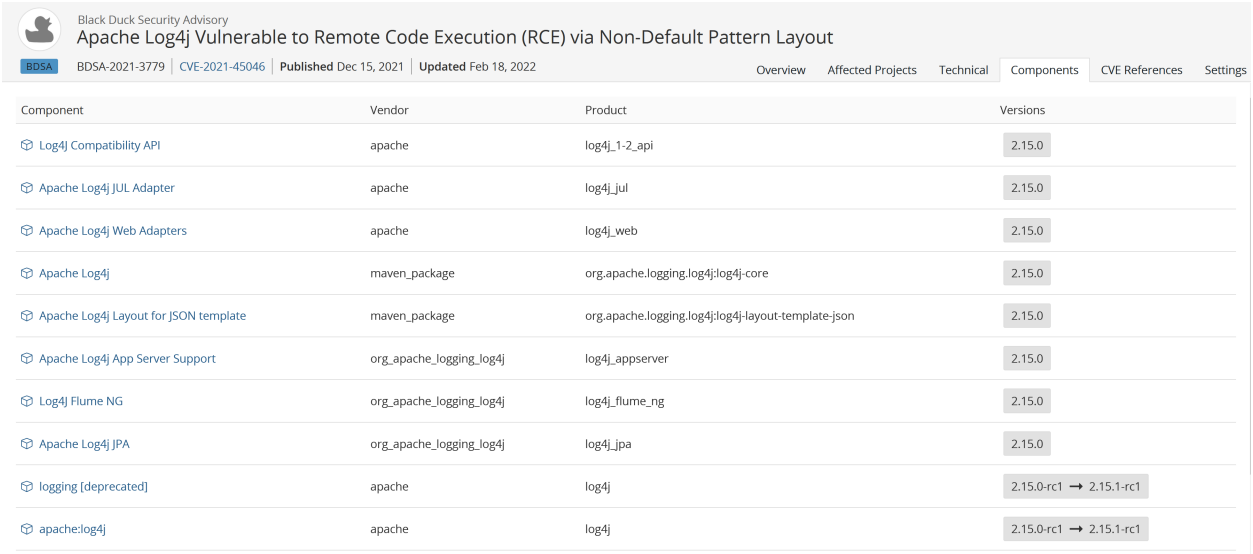
Included in the **References and Related Links** section is a list of Key Events:

- 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.

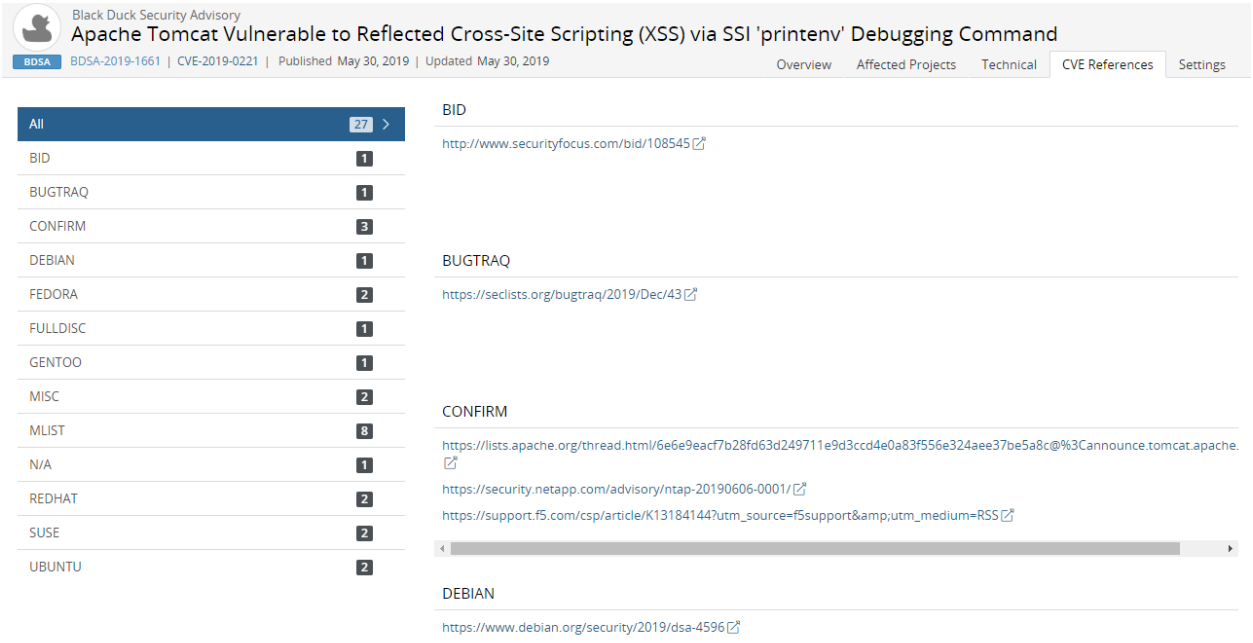
Components tab

Select the **Components** tab to view a list of all known component versions affected by this particular BDSA vulnerability record.



CVE References tab

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



Settings tab

Use this tab to manage the global remediation for this vulnerability. Click here for more information.

CVE record

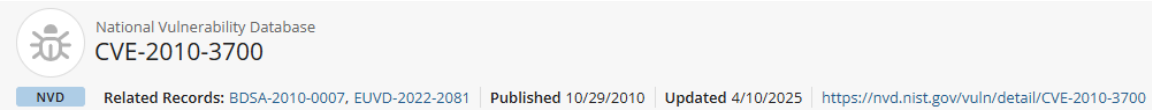
Vulnerabilities are linked to components by the Common Vulnerabilities and Exposures numbers (CVEs), as reported in the National Vulnerabilities Database (NVD) maintained by the National Institutes of Standards and Technology (NIST).

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 header bar displays the following information:



  - **CVE ID**: The unique identifier assigned to the CVE, which serves as a reference for the specific vulnerability.
  - **Related Records**: This section lists links to associated Black Duck Security Advisories (BDSA) and European Union Vulnerability Database (EUVD) records, offering users additional context and remediation options related to the same vulnerability or component.
  - **Published**: The date when the National Vulnerability Database (NVD) published the CVE. This indicates when the vulnerability was officially recognized and made public.
  - **Updated**: The last modified date by the NVD, reflecting when the CVE record was last updated with new information or corrections.
  - **URL**: A direct link to the NVD webpage for the CVE, providing users with access to detailed information, including descriptions, impact assessments, and mitigation strategies.
- The **Overall Score** graphic represents the Common Vulnerability Scoring System (CVSS) score for the CVE.



This score quantifies the severity of the vulnerability on a scale from 0 to 10, with higher scores indicating a greater level of risk. The CVSS score is calculated based on several factors, including the exploitability of the vulnerability, the impact on confidentiality, integrity, and availability, as well as the environmental context in which the vulnerability exists.

- The **Description** section provides a detailed overview of the vulnerability associated with the CVE ID. It includes information on the nature of the vulnerability, its potential impact on affected systems, and the conditions under which it can be exploited. The description may also outline the software or components affected, attack vectors, and any known mitigations or recommendations for addressing the vulnerability. This information is crucial for users to understand the context and significance of the CVE, facilitating informed decision-making regarding remediation strategies.

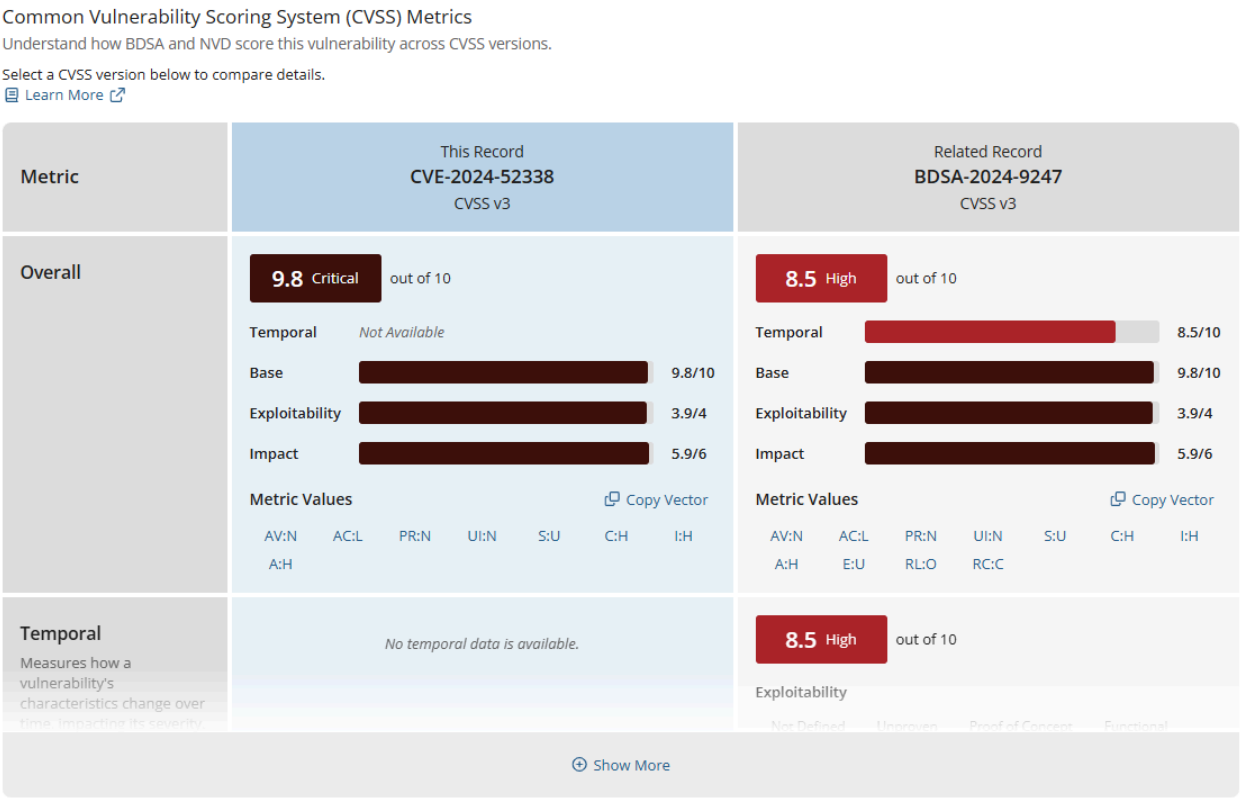
  Additionally, this section may contain the following information:

  - If the CVE has **CISA Known Exploited Vulnerability**, it will be displayed here. This section highlights vulnerabilities listed in the Cybersecurity & Infrastructure Security Agency's (CISA) Known Exploited Vulnerabilities (KEV) catalog. Federal civilian executive branch agencies are required to remediate vulnerabilities in the KEV catalog within prescribed timeframes. Information in this section includes:
    - **Vulnerability Title:** A brief description of the vulnerability.
    - **Added:** The date the vulnerability was added to the KEV catalog.
    - **Due Date:** The deadline for remediation.
    - **Action:** Recommended actions for remediation.
  - If the CVE has a related **EUVD ID**, it will be displayed here. This section provides additional vulnerability information sourced from the European Union Vulnerability Database, enhancing the context and coverage available for the CVE.

    The EUVD ID section contains the following details:

    - **EUVD ID:** The unique identifier assigned to the vulnerability in the EUVD.
    - **Link:** A direct link to the corresponding EUVD webpage for detailed information.
    - **Score:** The vulnerability severity score as assessed by the EUVD.
    - **Vector:** The CVSS vector string describing the characteristics of the vulnerability.
    - **Published:** The date when the EUVD published the vulnerability.
    - **Updated:** The date when the EUVD last updated the vulnerability information.
- The **Scores and Metrics** section displays the scores for the related BDSA and NVD records (if applicable), based on the Common Vulnerability Scoring System (CVSS). Select a value above the graph to view the information in the graph and details below.

  This section may also display a comparative, side-by-side graph if the vulnerability also has a BDSA record.



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.



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:

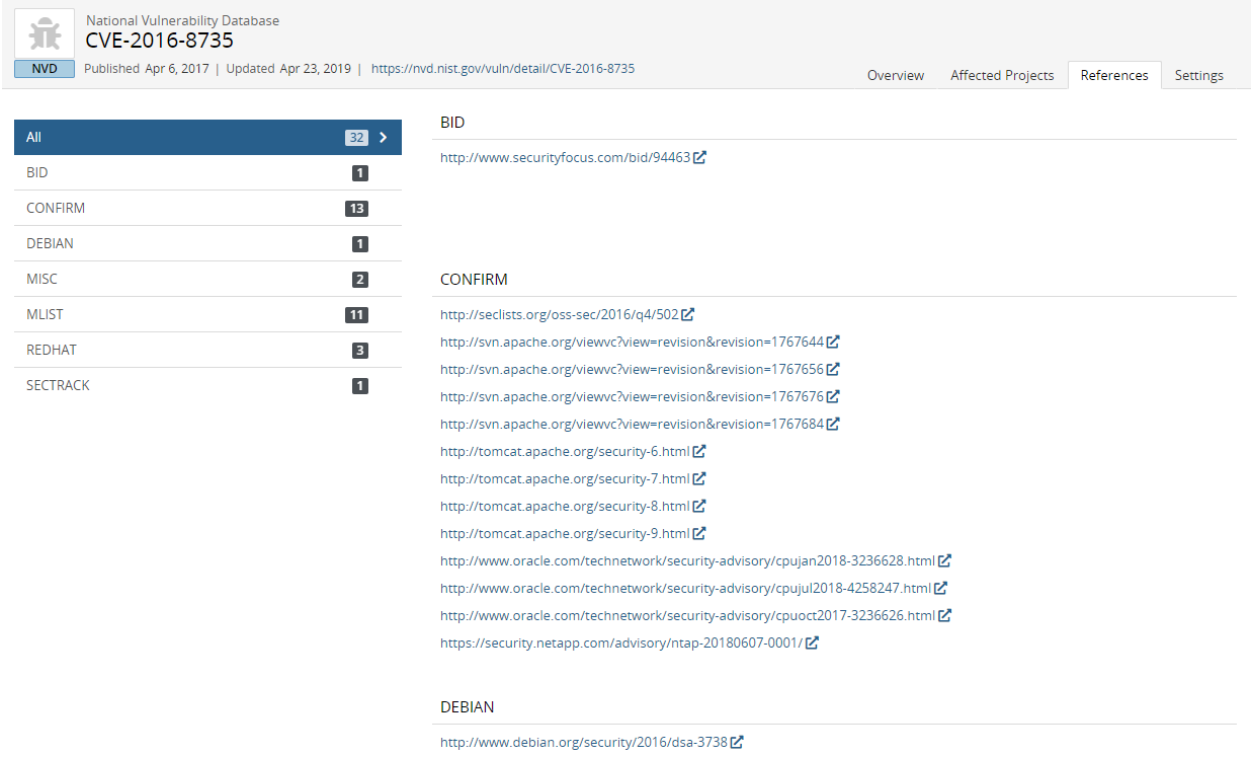
- **View all vulnerabilities** to view all vulnerabilities affecting this project version.
- **View related files to view** to display the **Source** tab filtered to display the affected files.

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.



Settings tab

Use this tab to manage the global remediation for this vulnerability. Click here for more information.

European Vulnerability Database (EUVD) Records

The European Vulnerability Database (EUVD) is a resource that catalogs vulnerabilities reported in software products across Europe. It serves as a valuable reference for security teams seeking to understand the vulnerabilities impacting their systems, including those not covered by other databases.

Finding EUVD Records in Black Duck SCA

In Black Duck SCA, when viewing a vulnerability record (such as a CVE), you may see an EUVD ID displayed in the page header if a related EUVD record exists. This ID indicates that there are additional details available in the EUVD regarding the vulnerability.



Accessing EUVD Information

When you click on the EUVD ID link in the vulnerability record, you will be redirected to the EUVD website. There, you can find comprehensive information about the related vulnerability, including descriptions, severity ratings, and any remediation recommendations.

CVE Numbering Authorities (CNA)

Black Duck SCA now ingests NVD CNA (CVE Numbering Authorities) vulnerability scores, enhancing our ability to provide comprehensive vulnerability assessments. When the NVD has not provided its own score for a CVE record, a secondary CNA score will be displayed in Black Duck SCA

Benefits of CNA Scores

The incorporation of CNA scores means there will be fewer CVE vulnerabilities without a CVSS score. Additionally, the speed at which vulnerability scores are published will improve, as CNA scores are often published ahead of NVD scores. This improvement allows organizations to respond more swiftly to vulnerabilities.

User Experience

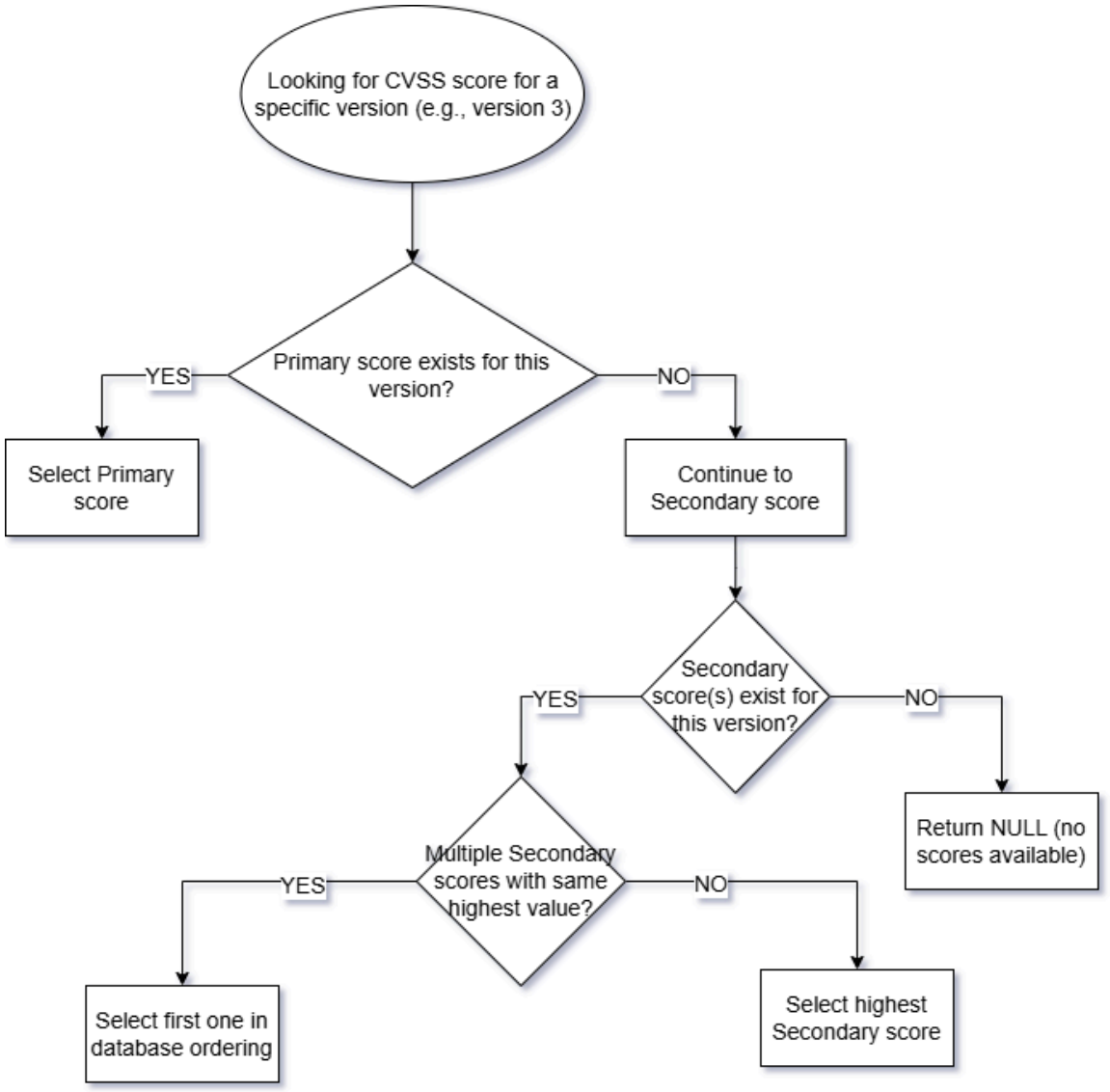
Customers do not need to make any changes to see the new CNA scores; this feature was implemented as part of an update to our KnowledgeBase. Users will benefit from CNA scores automatically without needing to upgrade.

Future Enhancements

In a future Black Duck SCA release, we will enhance our UI, APIs, and vulnerability reports to highlight the origin of CVE scores, including the CNA name. This will provide greater transparency and clarity in vulnerability management.

Understanding CNA Scores

See the flow diagram below, which explains when and how a CVE score is chosen. Note that customers cannot change the security risk ranking around CNA score priority.



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
Duplicate	No	This vulnerability is a duplicate.
Ignored	No	The vulnerability has been ignored.
Known Affected	Yes	Actions are recommended to remediate or address this vulnerability.

Remediation Status	Included in Security Risk Calculation?	Definition
Known Not Affected	No	A justification must be provided to ensure that the decision is properly documented.
Mitigated	No	The vulnerability has been mitigated.
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.
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.
Remediation Required	Yes	Remediation is required for the component version.
Remediation Complete	No	Remediation for the vulnerability is complete.
Under Investigation	Yes	It is not yet known if the product versions are affected by the vulnerability, but an update will be provided in a later release.

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.

Important: If a CVE record has a related BDSA record (or vice versa), it cannot be remediated unless that vulnerability record type is prioritized in the Security Risk Ranking. This is due to the fact that the non-prioritized vulnerability record is not being used as a determinant and is not used to calculate security risk.

Note: You can select any value for the remediation status. Selecting **Remediation Complete**, **Mitigated**, **Patched**, **Known Not Affected**, 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.

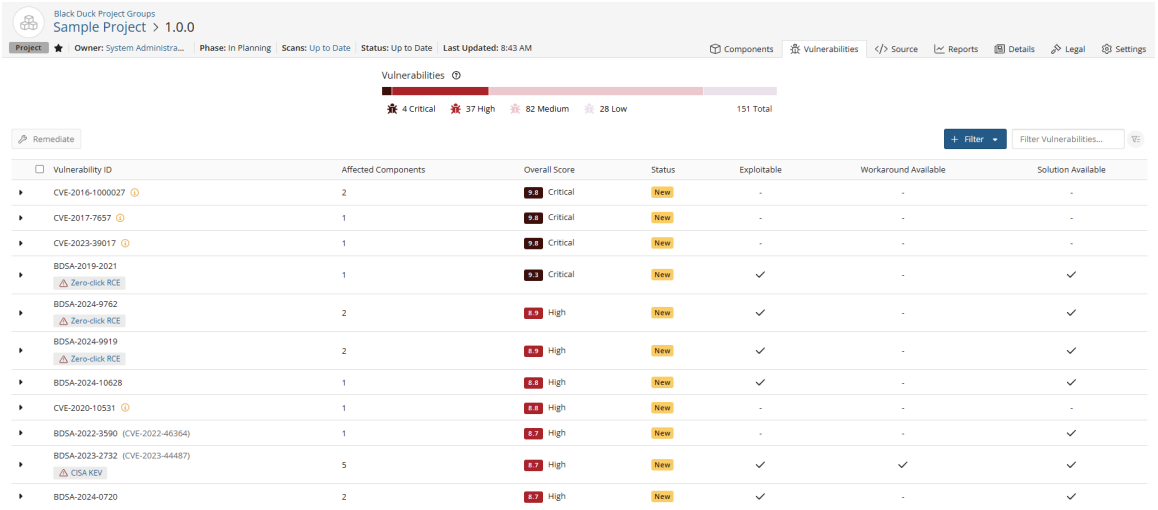
Only users with the appropriate role can remediate vulnerabilities.

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 SCA.

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 **Vulnerability** tab which lists all vulnerabilities for this project version.



5. Click the ▶ icon in the table next to the vulnerability to expand the row. A brief description, additional information, and fields to remediate the vulnerability appear.
6. Click the 🔧 next to a component to open the **Remediate Vulnerability** modal. From here, you can update the following fields:
   - **Status**. See the table above for status definitions.
   - **Status Justification**. If "Known Not Affected" is the selected remediation, you must provide a justification for it. You can choose from the following:
     - **Component not present**. The vulnerable component is not included in the product.
     - **Vulnerable code not present**. The specific vulnerable code is not present in the component.
     - **Vulnerable code cannot be controlled by adversary**. The vulnerable code is present but cannot be controlled or exploited by an attacker.
     - **Vulnerable code not in execute path**. The vulnerable code exists but cannot be executed in the product's operational context.
     - **Inline mitigations already exist**. Built-in security measures are already in place, preventing the exploitation of the vulnerability.
   - **Remediation**: If "Known Affected" is the selected remediation, you must provide a remediation option. You can choose from the following:
     - **Mitigation**: The product includes built-in protections that prevent exploitation of the vulnerability.
     - **No fix planned**: The vulnerability will not be fixed.
     - **None available**: There are no fixes available for this vulnerability.
     - **Vendor fix**: There is an update by the vendor to address the vulnerability.
     - **Workaround**: There is a workaround available to fix the problem.

- **Target Date**. This refers to the planned or intended deadline by which a vulnerability is expected to be remediated. It's often set based on the severity of the vulnerability, risk assessments, and your organization's security policies.
- **Actual Date**. This is the date when the vulnerability is actually remediated or mitigated. It may differ from the target date depending on various factors, such as resource availability, complexity of the issue, or unexpected delays.
- **Comments**. Any additional details or information regarding the vulnerability.

Remediating vulnerability by origin

To remediate vulnerabilities for this and/or additional project versions by origin, click the ▸ icon in the table next to the vulnerability to expand the row and then click the **View BDSA record** or **View CVE record** link in the **Description** box. Doing so will open the vulnerability's BDSA record or CVE record page.

From here, you can view the projects affected by this vulnerability by selecting the **Affected Projects** tab. See the BDSA record or CVE record links above for more detailed instructions on how to remediate vulnerabilities by origin.

Note: When viewing projects on this tab, you may see rows that are hashed out and cannot be selected for remediation. This is due to the project having a linked type of security vulnerability record, either BDSA or CVE. If that vulnerability record is not prioritized in the Security Risk Ranking, a Remediation Plan cannot be undertaken for that project. Consider viewing the prioritized security vulnerability record to update the Remediation Plan.

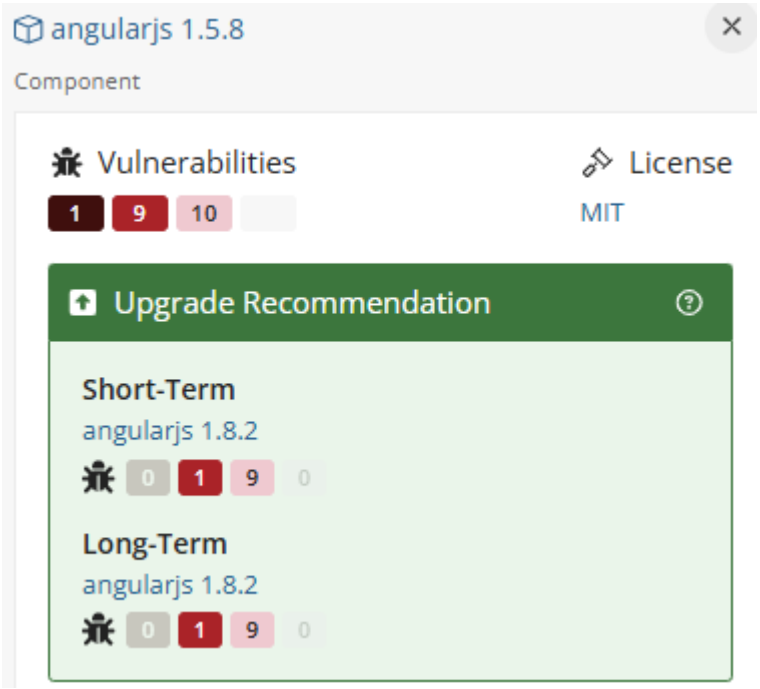
Getting remediation guidance for components with security vulnerabilities

When Black Duck identifies a component version with known security vulnerabilties, it attempts to find safer alternative versions of that component. These are presented as a **Upgrade Recommendation** within the component slide-out panel on the project version's BOM.

After reviewing this information, you may want to explore whether alternative component versions are available—particularly ones that resolves the security vulnerability affecting the version used in your BOM.

Upgrade recommendations

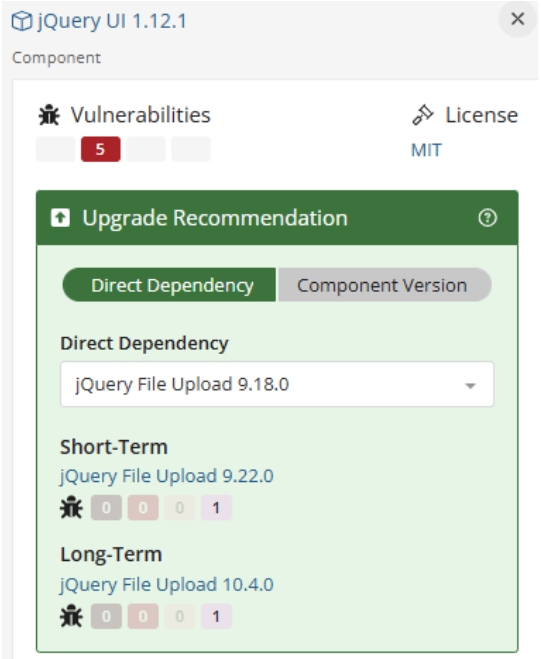
The simplest way to minimize or resolve security risk is to upgrade the version of the used component with fewer vulnerabilities.

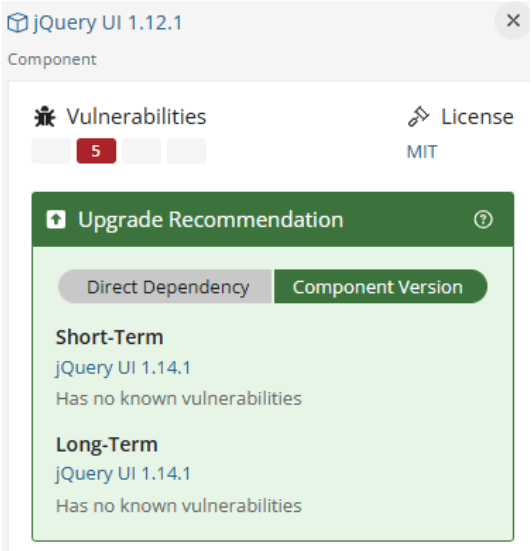


If your project version contains any component versions which have known vulnerabilities or are simply out of date, the Upgrade Recommendation section will display options you can explore to mitigate risk:

- **Short-Term** recommendations provides a short-term upgrade path as it is typically the same major version as the version currently used in your BOM.
- Unlike the short term upgrade recommendation, **Long-Term** recommendations usually requires a major version upgrade. This may require more planning and/or engineering work to implement.

If the component is a transitive dependency, you will see two additional options, **Direct Dependency** and **Component Version**:





- **Direct Dependency**: Provides a short- and long-term upgrade recommendation for the root node of a transitive dependency.
- **Component Version**: Short-term typically reflects upgrading a component's minor or patch version. Long-term will focus on upgrading a component's major version.

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.

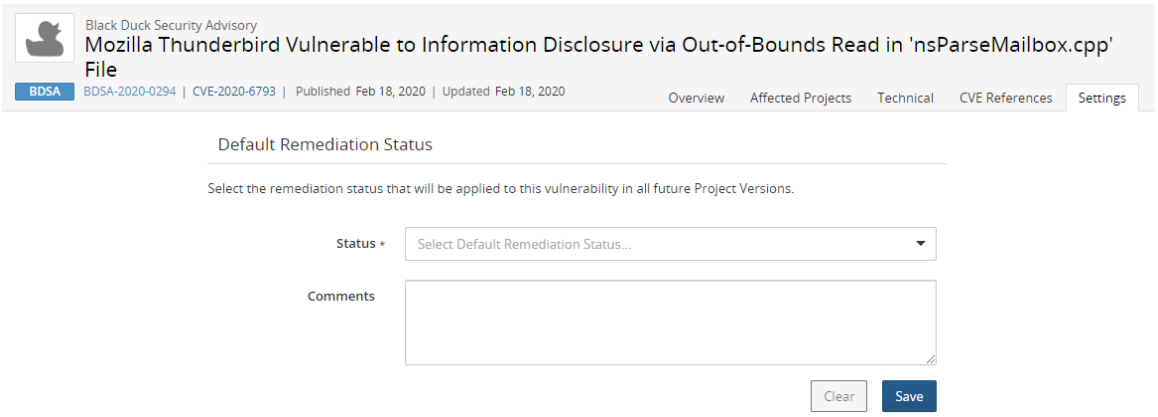
Note: Any component in your existing BOMs that has that vulnerability retains its current remediation status. The global remediation status only applies to *new* instances of the vulnerability.

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.

1. Find the vulnerability you want to remediate globally, For example, you can:
   - Use the **Vulnerabilities** tab for a specific project version.
   - Use the Search feature.
2. 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.



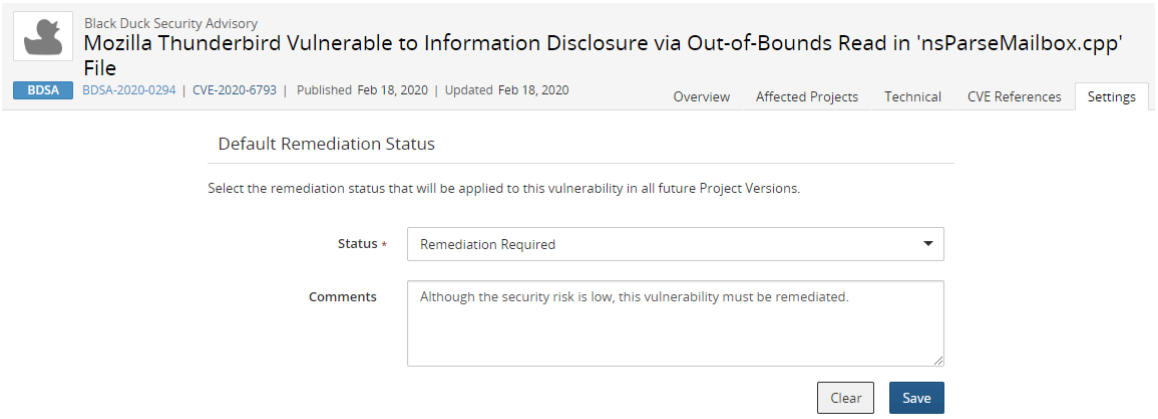
4. Select a default status and optionally, enter a comment. This comment appears when viewing the description, as described below.
5. Click **Save**.

   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.



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.

Finding data in Black Duck

In Black Duck, there are two ways to search for information: through the **Find** menu item or by using the search field at the top of the page.

Find page

The **Find** page allows you to conduct targeted searches in your projects, for components and vulnerabilities found in scans, as well as though the Black Duck KnowledgeBase. 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: These are the components that comprise your projects and are viewable in your BOM.

- 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.
- Black Duck KnowledgeBase: The Black Duck KnowledgeBase is a comprehensive database of open source software (OSS) components and known vulnerabilities. It allows you to search for components and vulnerabilities beyond those present in your current projects, providing valuable insights into potential security risks and component details across the wider OSS landscape.

Search field

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.

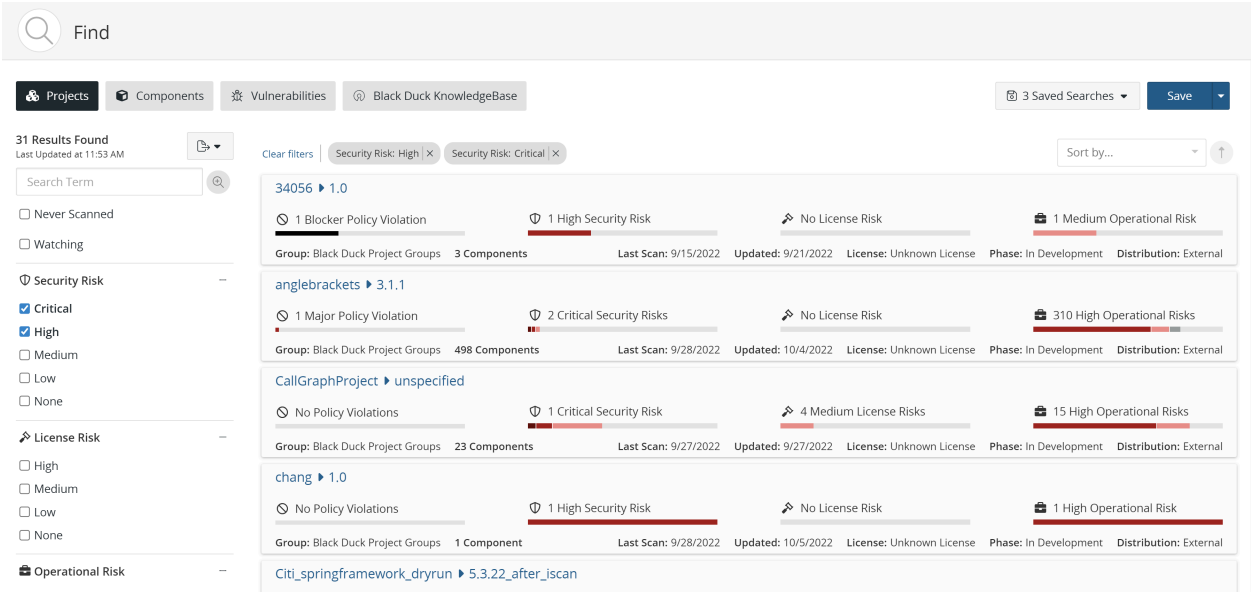
Searching for projects

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.

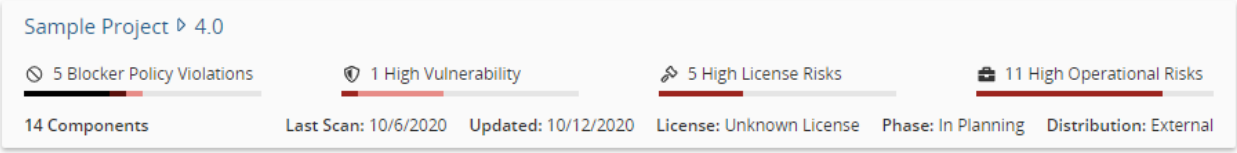


You can also type your search term in the Search field located at the top of the application and press **Enter** or click . The Find page appears displaying the search results. Note that entering a global search term

initiates a new search and resets any filters you previously selected. Select the **Projects** tab and filters to refine the results, as described below.

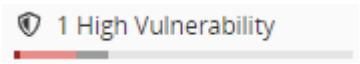
About the search results

Search results show all project versions that meet your search criteria. The following information is shown for each project version:

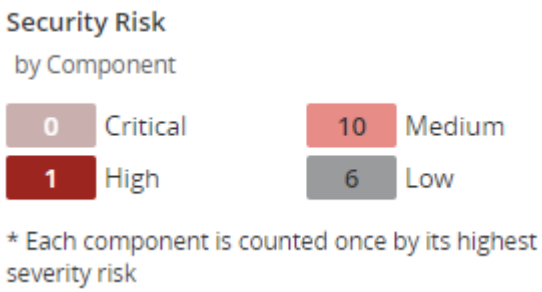


- Use the bars to quickly view the number of components with the highest level of security, license, or operational risk.

  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.

  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.

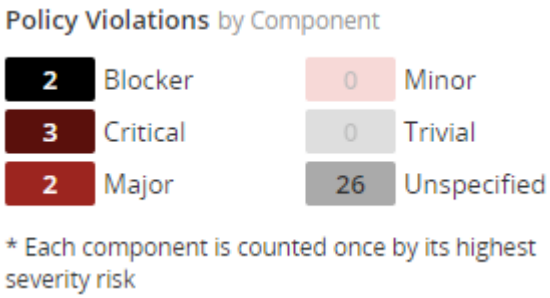
- Use the bar to see the number of components with the highest policy severity level for this project version.

  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.

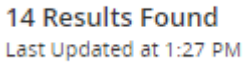
  

  Note: The text shown states the number of components with the highest policy severity level for this project version, not all policy severity levels affecting this project version.

- 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:



- 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 Violations. Severity level of the policy rule.
- IaC Issues: Select from Open, Dismissed, or Not Reported.
- Last Scanned. Select a time period when this project was last scanned.
- Not Scanned Since. Select the time period since this project was last scanned.
- Project Group. Select one or more project groups.
- Release Phase. Select one or more release phases.
- Tags. Select one of more tags.

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.

Exporting to CSV

You can export your search results to CSV which converts the individual rows to tabular data. To do so, click the button and select CSV.

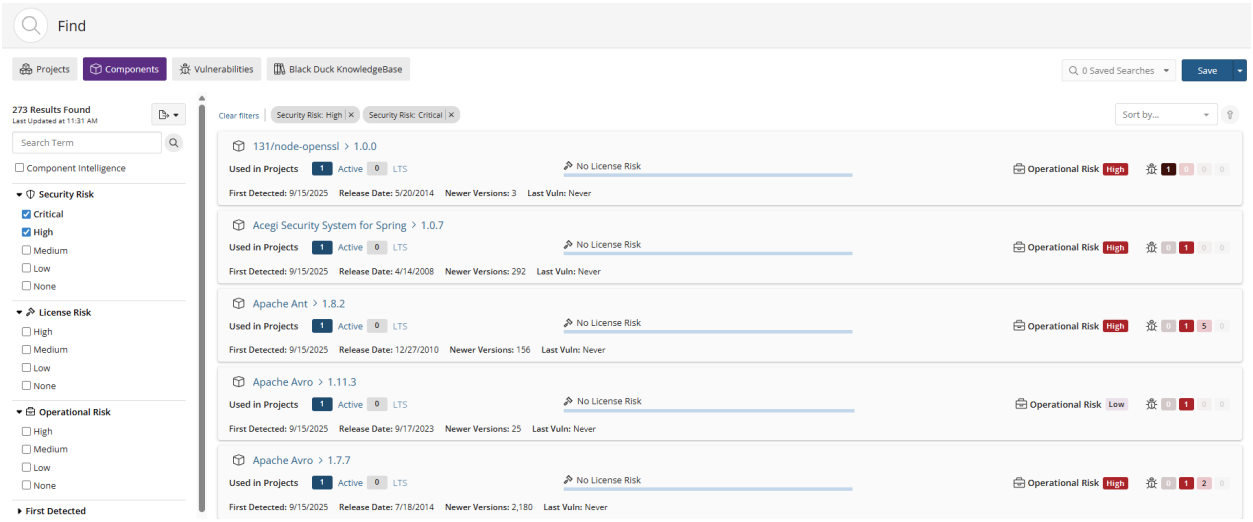
Searching for components

You can search for component versions used in your BOMs.

To search for components :

1. Click  to open the **Find** page.
2. Select the **Components** tab to find component versions used in your projects.
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.



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.

Using search filters

Filters that appear depend on whether you are searching for components used in your BOMs or searching 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.

Component filters

Use the following filters to narrow your results when searching components used in your BOM:

- **Component Intelligence**. Check to display all components containing suspicious events or incidents where it is highly likely that malware or malicious code has been identified. See Operational Risk for more information on Component Intelligence.
- **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.
- **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**. Select a license from the list.
- **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.

Components search results

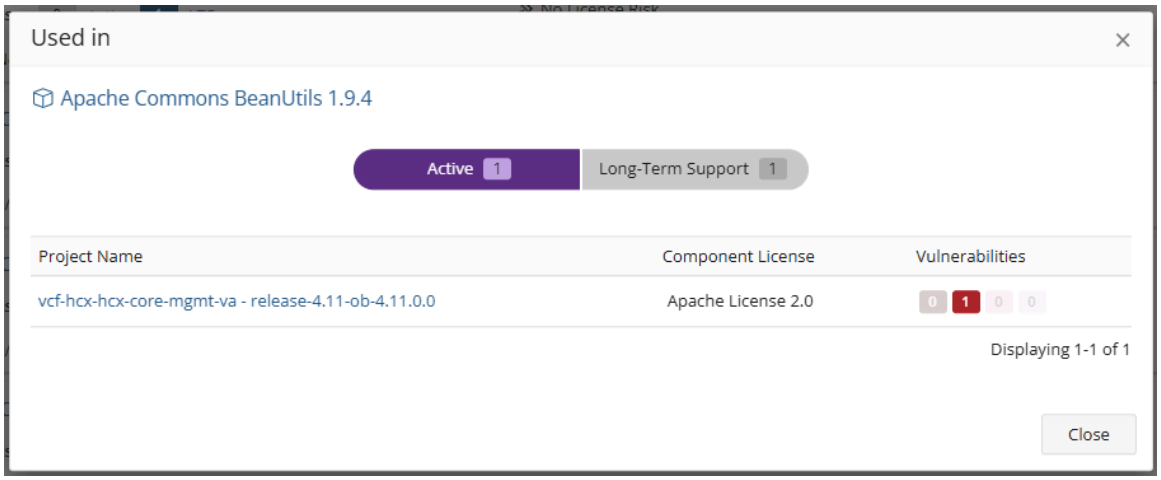
The following information is shown for each component in your BOM that meets your search criteria.



- Select the component name/version to display the *Component Name Version* page.
- View the number of Active and LTS project versions that use this component version as shown by the value next to **Used in Projects**.



Select **Active** or **LTS** to open the **Used in** dialog box.



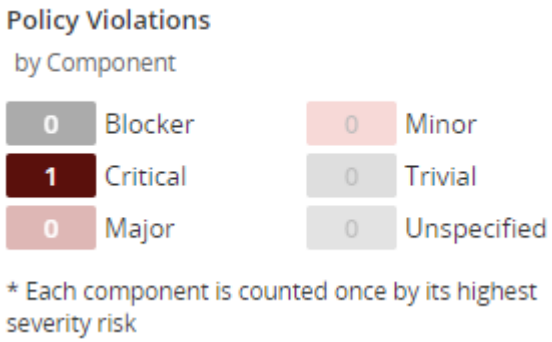
This dialog box now displays the project versions that utilize this version of the component, including those in Long-Term Support (LTS) projects.

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.
License	License for this component version.
Vulnerabilities	Lists the vulnerabilities for each severity level, from left to right: Critical, High, Medium, and Low.  Select a value to access the Vulnerabilities tab on the *Component Version* page. This tab provides a detailed list of vulnerabilities linked to the specific version of the component.

- 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:

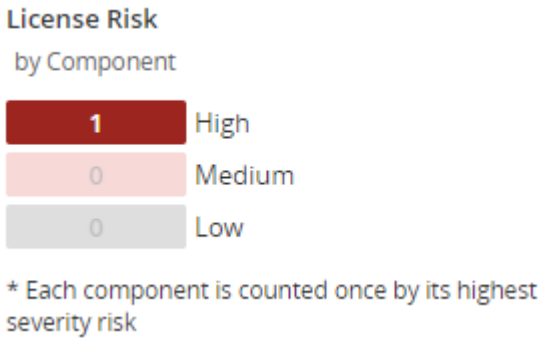


Note: A component is only counted once with the highest policy severity level, not all policy severity levels affecting this component.

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



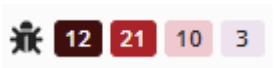
Select the bar to view the number of components in each risk category.



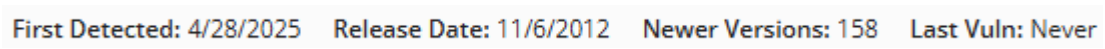
- View the operational risk for this component version:



- View the number of vulnerabilities by severity associated with this component version.



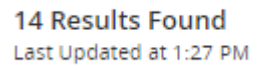
- Additional information relevant to the vulnerability:



  - **First Detected**. The specific date when a vulnerability was initially identified in a component within your project. This timestamp indicates when Black Duck SCA first associated the vulnerability with the component version during a scan or Knowledge Base (KB) update.

- **Release Date**. The date when a vulnerability was officially disclosed or published by its source, such as the National Vulnerability Database (NVD) or a Black Duck Security Advisory (BDSA). This date marks when the vulnerability became publicly known and available for organizations to assess and address.
- **Newer Versions**. The number of newer releases of a component or library that may address known vulnerabilities, improve functionality, or enhance performance.
- **Last Vuln**: The most recent date when a vulnerability was added to the list of vulnerabilities associated with a specific component version.

- View the number of results found and the time the database was last updated:



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.

Notice: Ignored components are excluded from the **Find** → **Component** page, meaning they will not be counted in the component totals displayed there.

Exporting to CSV

You can export your search results to CSV which converts the individual rows to tabular data. To do so, click the button and select CSV.

Searching for vulnerabilities

You can search Black Duck for published security vulnerabilities. Searching by vulnerability is an efficient way to:

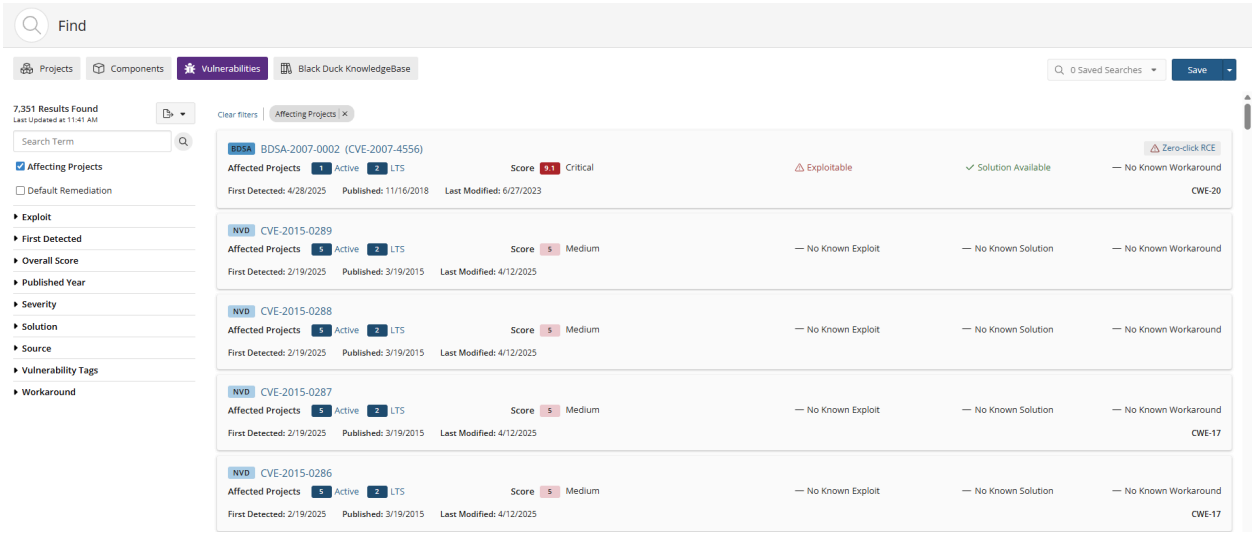
- Identify if a new or existing security vulnerability affects a component that is included in your projects.
- Review the severity of the security vulnerability to determine if remediation is required.
- Create a custom vulnerability dashboard so that you can focus on the vulnerabilities that are important 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.



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:

- Click ▸ to display the filter values.
- 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 of 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. When selected, this filter limits results to vulnerabilities found in your Active and LTS projects. When cleared, results include vulnerabilities from both your projects and the Black Duck KnowledgeBase.
- Default Remediation. Selecting this filter displays vulnerabilities that are automatically remediated.
- Exploit. Select whether a vulnerability is Exploitable or has No Known Exploit.
- First Detected. When the vulnerability first appeared in a BOM.
- 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. Select from Critical, High, Medium, or Low as determined by the selected security configuration.
- Solution. Select whether a solution is available for a vulnerability.

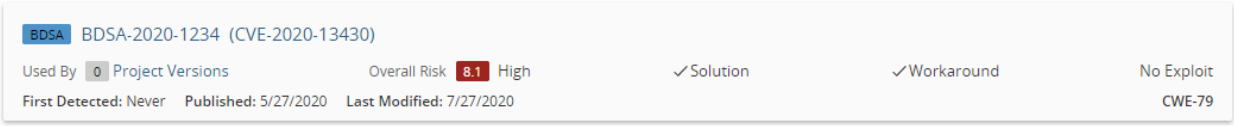
- Source. BDSA or NVD.
- Vulnerability Tags. Select one or more vulnerability tags.

  Note: If searching for CISA Known Exploited Vulnerabilities, you must also check the Affecting Projects checkbox to display results.
- Workaround. Select whether a workaround is available for a vulnerability.

Attention: Filter options and tags are not all supported for KnowledgeBase vulnerability queries and some queries are only supported for local vulnerabilities affecting project versions (using the checkbox). This means different behavior may be observed when using the Find page with and without the **Affecting Projects** checkbox enabled.

About the search results

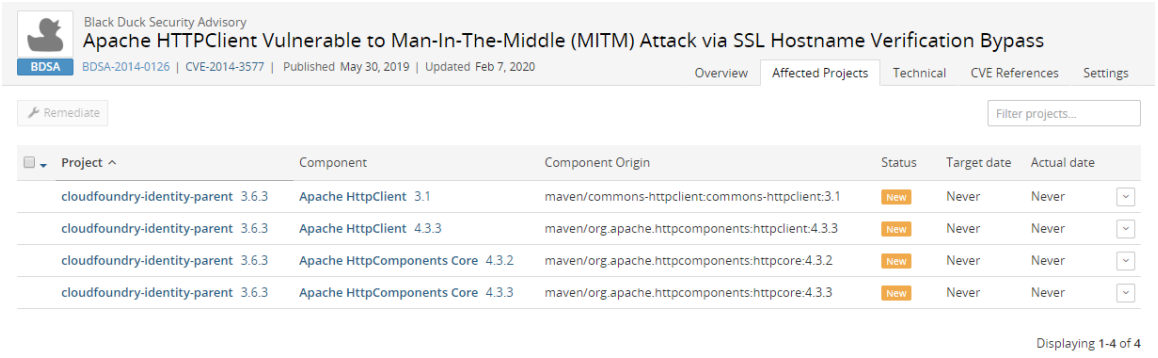
Search results show all vulnerabilities that meet your search criteria. The following information is shown for each vulnerability:



- 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**.

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

- 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. Note that this date displays the last time the vulnerability was modified in the KnowledgeBase. It does not necessarily mean the vulnerability information was updated itself.
  - Common Weakness Enumeration (CWE) number for this security vulnerability.

Note: Search results are limited to a maximum of 10,000 items.

Exporting to CSV

You can export your search results to CSV which converts the individual rows to tabular data. To do so, click the button and select CSV.

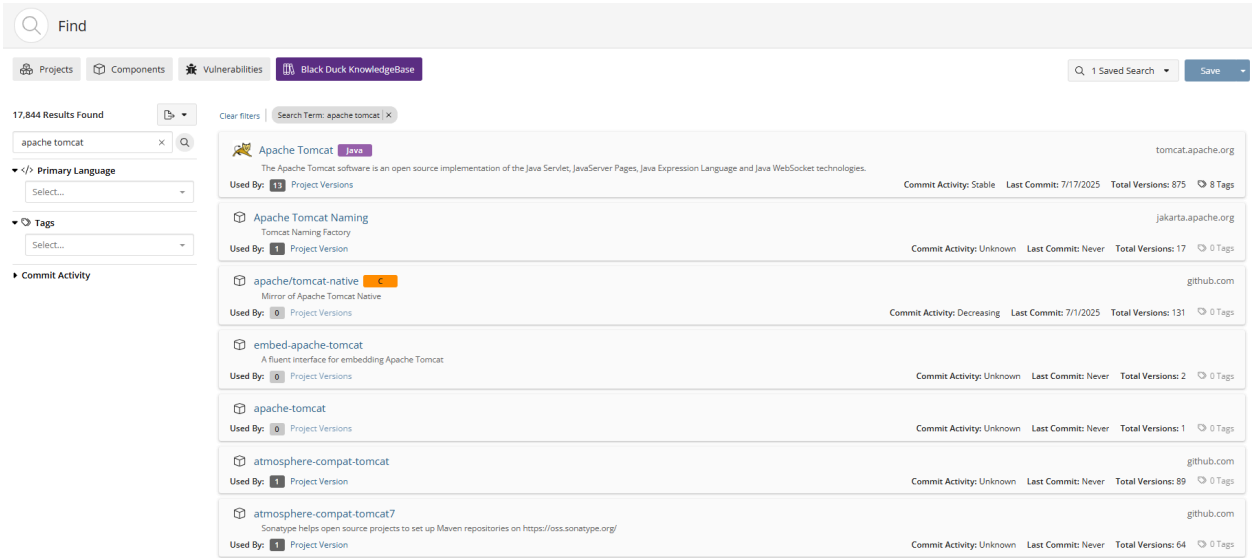
Searching the Black Duck KnowledgeBase

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. Select the **Black Duck KnowledgeBase** tab. Note that using the Black Duck KnowledgeBase tab to search for components will not display any custom components used in your projects. Use the Components tab to include these in your search results.
3. Type your search term in the Search field and/or optionally, select any filters to refine your search.
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.



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.

KnowledgeBase filters

Use the following filters to narrow your results when searching 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.

Important:

Primary Language and **Tags** are dynamic search filters. Enter one or more characters in the filter field to retrieve matching values. These filters do not pre-populate with all available languages or tags. Primary language, tags, and commit activity information is provided by Black Duck Open Hub.

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**.



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	Lists the vulnerabilities for each severity level, from left to right: Critical, High, Medium, and Low.  Select a value to display the **Vulnerabilities** tab of Black Duck KB *Component Name Version* page, which lists the vulnerabilities associated with this version of the component.

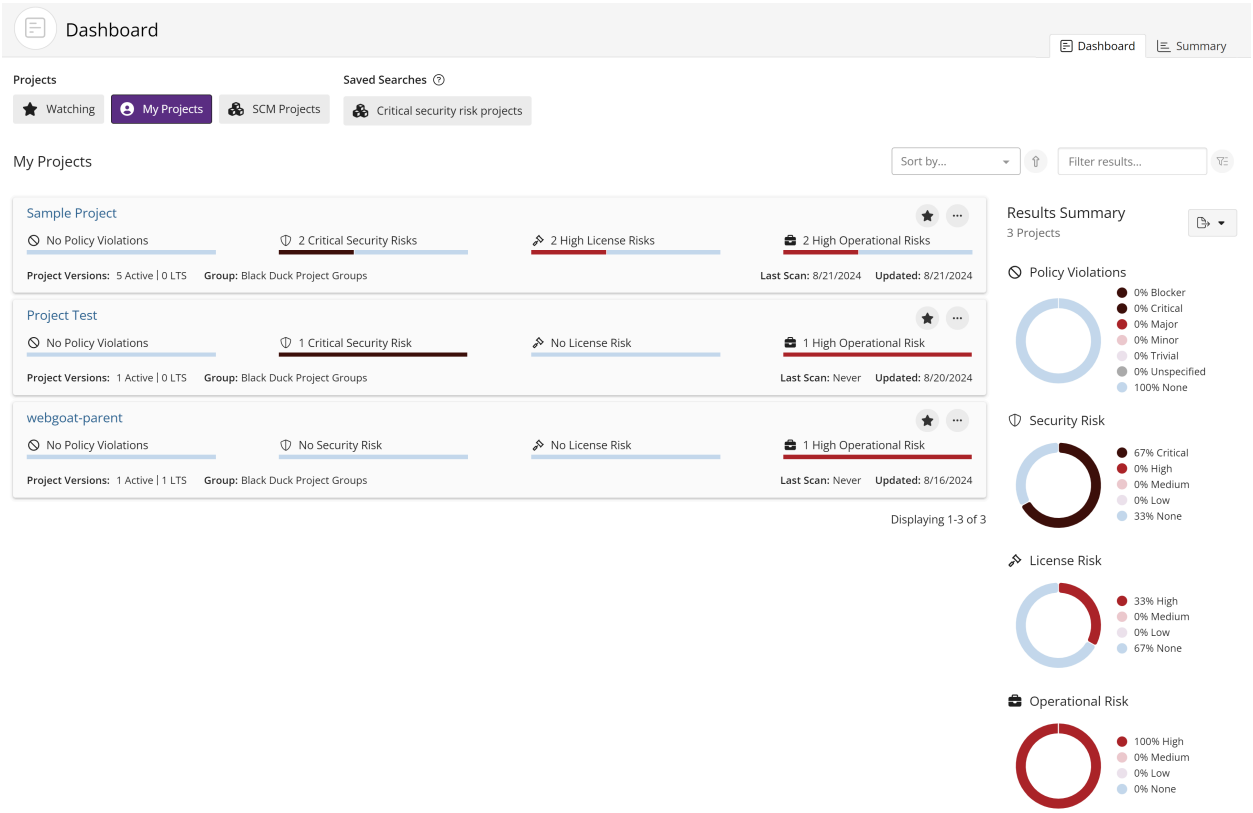
- For each component, the search results show:
  - Commit Activity.
  - Last commit date.
  - Total number of versions for this component.
- Select **Tags** to view the tags for this component.
- The URL in the upper right corner is the URL for this component.

Exporting to CSV

You can export your search results to CSV which converts the individual rows to tabular data. To do so, click the button and select CSV.

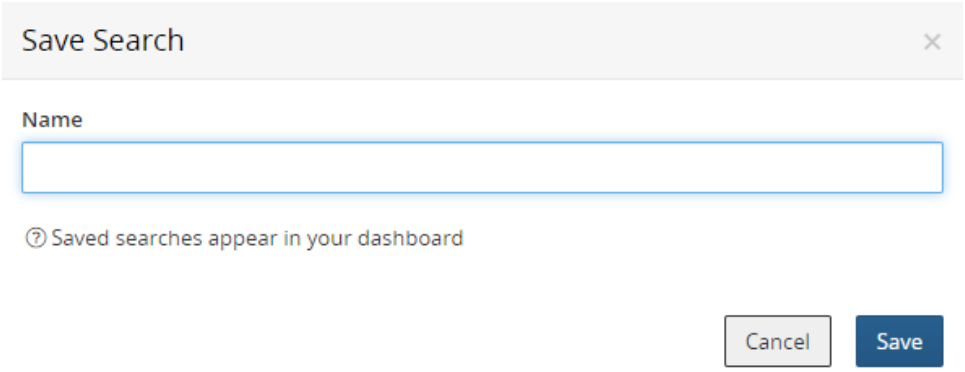
Saving and managing search results

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

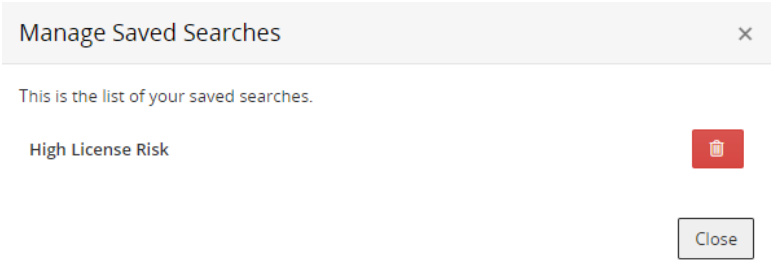
1. Click Find. The Find page appears.
2. Select the saved search you wish to modify from the list of saved searches. The Find page displays the search results for that saved search.
3. Optionally, modify the filters for this saved search.
4. Click **Update**.

Renaming saved searches

1. Click Find. 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 Find. 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**.

Generating global reports

The **Reports** page provides the means of generating reports, granting critical insights into the security status of your organization's projects, specifically addressing known vulnerabilities and their remediation. These reports are essential for tracking and managing vulnerabilities effectively.

To access the Reports page:

1. Log in to Black Duck SCA.
2. Click 

.

From here, you can generate a report by clicking **+ Create new report**.

The following options are available:

- 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.

Note: Reports include subproject information *if* you have permission to the subproject.

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.

Running a Vulnerability Remediation report at the global level

To run a Vulnerability Remediation report at the global level:

1. Log in to Black Duck SCA.
2. Click 

.
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-remedation-report_all_assigned_projects_*YYYY-MM-DD_HHMMSS* (time stamp in system timezone) for a global version of the report
- vulnerability-remedation-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.

Running a Vulnerability Remediation report at the project level

To run a Vulnerability Remediation report at the project level:

1. Log in to Black Duck SCA.
2. Click the desired project name on the Dashboard.
3. Click the desired project version.
4. Click the **Reports** tab.
5. Click **+ Create new report**. The Create New Report dialog box appears.
6. Select **Vulnerability Remediation Report** from the **Report Type** list.
7. 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.
8. 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.
9. 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.
10. Optionally, select one or more remediation statuses.
11. Click **Confirm** to run the report.
    One of the following links appears when the report completes:
    - vulnerability-remedation-report_all_assigned_projects_*YYYY-MM-DD_HHMMSS* (time stamp in system timezone) for a global version of the report
    - vulnerability-remedation-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.
12. 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.

Running a Vulnerability Status report at the global level

To run a Vulnerability Status report at the global level:

1. Log in to Black Duck SCA.
2. Click 

.
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.

Running a Vulnerability Status report at the project level

To run a Vulnerability Status report at the project level:

1. Log in to Black Duck SCA.
2. Click the desired project name on the Dashboard.
3. Click the desired project version.
4. Click the **Reports** tab.
5. Click the **+ Create New Report**.

6. Select **Vulnerability** from the report list.
7. Select **Vulnerability Status Report** from the **Report Type** list.
8. 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.
9. 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.
10. 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.
11. 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.

Running a Vulnerability Update report at the global level

To run a Vulnerability Update report at the global level:

1. Log in to Black Duck SCA.
2. Click 

.
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. The CSV report will contain four reports:
   - new-remediated-vulnerabilities_YYYY-MM-DD_HHMMSS.csv

     This report will contain all new remediations for the selected project(s) within the specified time frame. A vulnerability is newly remediated if the remediation was created after the vulnerability was created and it is in the specified time range.
   - new-vulnerabilities_YYYY-MM-DD_HHMMSS.csv

     This report will contain all new vulnerabilities for the selected project(s) within the specified time frame. A vulnerability was associated to a BOM component whether or not it is remediated and the bom association was in the specified time range.
   - updated-remediated-vulnerabilities_YYYY-MM-DD_HHMMSS.csv

     This report will contain all updated remediations for vulnerabilities that occurred within the specified time range for the selected project(s). A vulnerability has an updated remediation if the remediation update happened after the remediation was created and is in the specified time range.
   - updated-vulnerabilities_YYYY-MM-DD_HHMMSS.csv

     This report will contain all updated vulnerabilities for the selected project(s) within the specified time frame. A vulnerability is updated if the the updated dateTime is within the specified time range and this vulnerability risk was updated after the risk was created.

Note: You can use the native print functionality of your web browser to print the HTML version of the report.

Running a Vulnerability Update report at the project level

To run a Vulnerability Update report at the project level:

1. Log in to Black Duck SCA.
2. Click the desired project name on the Dashboard.
3. Click the desired project version.

4. Click the **Reports** tab.
5. Click **+ Create new report**. The Create New Report dialog box appears.
6. Select **Vulnerability Update Report** from the **Report Type** list.
7. 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.
8. 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.
9. 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.
10. 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.
11. Select the link to view the report.

    If you selected CSV as the report format, download the report and extract the zip file. The CSV report will contain four reports:
    - new-remediated-vulnerabilities_YYYY-MM-DD_HHMMSS.csv

      This report will contain all new remediations for the selected project(s) within the specified time frame. A vulnerability is newly remediated if the remediation was created after the vulnerability was created and it is in the specified time range.
    - new-vulnerabilities_YYYY-MM-DD_HHMMSS.csv

      This report will contain all new vulnerabilities for the selected project(s) within the specified time frame. A vulnerability was associated to a BOM component whether or not it is remediated and the bom association was in the specified time range.
    - updated-remediated-vulnerabilities_YYYY-MM-DD_HHMMSS.csv

      This report will contain all updated remediations for vulnerabilities that occurred within the specified time range for the selected project(s). A vulnerability has an updated remediation if the remediation update happened after the remediation was created and is in the specified time range.
    - updated-vulnerabilities_YYYY-MM-DD_HHMMSS.csv

      This report will contain all updated vulnerabilities for the selected project(s) within the specified time frame. A vulnerability is updated if the the updated dateTime is within the specified time range and this vulnerability risk was updated after the risk was created.

Note: You can use the native print functionality of your web browser to print the HTML version of the report.

Vulnerability Exploitability EXchange (VEX)

Vulnerability Exploitability eXchange (VEX) reports in Black Duck SCA offer a standardized way to communicate the exploitability status of vulnerabilities in your products. These machine-readable reports help organizations efficiently inform stakeholders about whether specific vulnerabilities affect their offerings.

An SBOM (Software Bill of Materials) serves as a static declaration of components in a product release. After release, it's important to determine the exploitability of any vulnerabilities, and VEX reports facilitate this process. While the SBOM content remains unchanged, VEX allows for on-demand updates regarding the status of vulnerabilities for specific project versions.

Important: To take advantage of this feature, you must have **Vulnerability Exploitability eXchange (VEX)** enabled on your product registration key.

Creating a Global VEX Report

Global VEX reports summarize vulnerabilities across all projects in your Black Duck SCA environment, aiding in high-level assessments and remediation decisions. Follow the steps below to generate a global VEX report:

1. Log in to Black Duck SCA.
2. Click 

.
3. Click **+ Create new report**. The Create New Report dialog box appears.
4. Select **Vulnerability Exploitability EXchange (VEX): CSAF 2.0** from the **Vulnerability Report Type** list.
5. Select a project from the **Projects** field.
6. Select the desired **Project Phases**. The VEX report will include an entry for each CVE or BDSA within the selected project(s) as long as they have one of the following Black Duck SCA status values:
   - Under Investigation
   - Needs Review
   - Known Affected
   - Known Not Affected
   - Remediation Required
   - Remediation Complete
7. Select a **Template**.
8. Optionally, you can expand the **Template Details** section to see which options are enabled in the selected template.
9. Click **Save** to run the report.

   The following links appear when the report completes:

   - csaf-report_*YYYY-MM-DD_HHMMSS* (time stamp in system timezone) for a global version of the report.
10. Select the link to view the report.

VEX Report Format

The VEX report utilizes the CSAF 2.0 (profile 5) format, aligning with industry standards for vulnerability reporting. It clearly indicates which products are unaffected by vulnerabilities, making it a valuable resource for product security teams. This feature facilitates effective communication about security risks, enabling organizations to address customer inquiries and maintain a robust security posture.

Reports will be generated in JSON format, implementing only the minimum required elements for CSAF v2, profile 5. Each report will include entries for CVEs or BDSAs within the selected projects that have one of the following BD status values:

- Under Investigation
- Needs Review
- Known Affected
- Known Not Affected
- Remediation Required
- Remediation Complete

VEX Document Metadata

The VEX report includes essential metadata that is automatically populated and structured according to the CSAF 2.0 specifications. Below are the key elements of the VEX document metadata:

- **Document metadata**
  - **CSAF Version**: Set to `2.0` (cannot be modified).
  - **Category**: Set to `csaf_vex` (cannot be modified).
  - **Publisher**
    - **Category**: Set to `vendor` (cannot be modified).
    - **Name**: Use the value from the existing Project Group SBOM Creator Organization field. If this field is set to the default value of "COMPANY NAME," a warning will be generated indicating that the default value is being used and should be changed. Users will have the option to Cancel or Continue:
      - **Cancel**: Returns to the report generation screen.
      - **Continue**: Generates the report using "COMPANY NAME" as the publisher name.
    - **Namespace**: A new text field will be added under the "Person" field in the Project Group SBOM fields for the Namespace. SBOM generation will utilize the BD namespace value to populate the namespace field in the SBOMs (for applicable SPDX/CycloneDX versions).
  - **Title**: Defined by Black Duck and currently not modifiable. The title will read: "Vulnerability status report using the CSAF 2.0 Profile 5 specification."
  - **Tracking**
    - **ID**: The ID will be the CSAF document filename (e.g., `csaf-report_all_projects_2025-07-22_143038`). The VEX Product ID can be configured in SBOM Fields.
    - **Current Release Date**: The date/timestamp of report generation (UTC).
    - **Initial Release Date**: The timestamp of report creation.
    - **Revision History**: One entry will be added for the current (latest) information:

        - **Date**: Report generation date/timestamp (UTC).
        - **Number**: Hardcoded value of 1 for the version.
        - **Summary**: Either "Latest information" or "Initial," depending on whether tracking both is necessary.
    - **Version**: Hardcoded value of 1 for the version.
    - **Generator Information:**
        - **Date**: Report generation date/timestamp (UTC).
        - **Engine**:
            - **Name:** "Black Duck HUB"
            - **Version:** The version of Black Duck HUB used to generate the report.
    - **Status**: Set to `draft`. (Note: Future support may be added for final and interim options, as referenced in section 3.2.1.12.7 of the CSAF 2.0 specification.)
- **Product Tree**: The `product_tree` section of a VEX report in SCA provides a hierarchical representation of the products that are affected by a specific vulnerability. It is structured to illustrate the relationships between different product versions and their corresponding product names.
    - **Branches**: The `product_tree` consists of branches that represent different levels of the hierarchy. Each branch can contain further branches, allowing for a nested structure that captures the relationship between parent products and their versions.
        - **Category**: Each branch includes a category that defines the type of information it represents. In the example provided, categories might include "product_name" and "product_version."
        - **Name**: The name of the product or product version.
        - **Product**: Each product branch contains relevant details such as:
            - **Name**: The name of the product or product version.
            - **Product ID**: A unique identifier for the product, which helps in distinguishing it from other products within the tree.
- **Vulnerabilties**: The Vulnerabilities section details specific vulnerabilities identified within the software components. This section is essential for understanding the security posture of the application and the associated risks.
    - **CVE Identifier**: If the vulnerability is a CVE, it is listed with its unique CVE ID (Common Vulnerabilities and Exposures). This identifier allows for easy tracking and reference to the vulnerability in external databases.
        - **Notes**: Accompanying each CVE entry are detailed notes that provide descriptions of the vulnerabilities. These descriptions explain the nature of the vulnerability, its potential impact, and any relevant technical details.
            - **Category**: Defines the type of note.
            - **Text**: The actual free-form text providing the information. If there is a related EUVD ID associated with this CVE, it will be displayed here.
        - **Product Status**: This section is essential for users to understand the current state of the products affected by vulnerabilities. It provides transparency regarding ongoing assessments and remediation efforts, helping organizations prioritize their actions to maintain security.

        - **Scores**: This section provides critical information regarding the severity of the identified vulnerabilities, utilizing the Common Vulnerability Scoring System (CVSS).
      - **IDs**: If the vulnerability has a BDSA ID, it will be found in this section.
    - **Distribution**
      - **TLP**:
        - **Label**: TLP label associated with the component version.
- **Notes**: Provides additional information about the document itself. This can include general descriptions, frequently asked questions, summaries, or legal disclaimers.
  - **Category**: Note type identifier. Displays `legal_disclaimer` if the project group has a VEX Legal Disclaimer.
  - **Text**: The actual content of the legal disclaimer. Free-form text containing legal terms, conditions, limitations, and protections as defined by the organization.

VEX status mapping

Black Duck SCA Vulnerability Status	CSAF VEX Status
AFFECTED	`known_affected`
Duplicate	N/A - Not included in report
Ignored	N/A - Not included in report
`MITIGATED`	`known_not_affected` with the justification `inline_mitigations_already_exist`
`NEEDS_REVIEW`	`under_investigation`
`NEW`	`under_investigation`
`NOT_AFFECTED`	`known_not_affected`
`PATCHED`	`fixed`
`REMEDIATION_COMPLETE`	`fixed`
`REMEDIATION_REQUIRED`	`known_affected`
`UNDER_INVESTIGATION`	`under_investigation`

Notes:

- The results were validated by generating VEX documents that have vulnerabilities with each of the Black Duck SCA status values that we have
- The order of the Black Duck SCA Vulnerability Status in the table above is the order that is represented in the UI
- The value in the CSAF VEX Status is the actual value in 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:

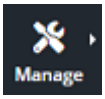
- Reports older than 30 days are automatically deleted.
- The system retains up to 20 reports, per user, across all project versions. If a user creates more than 20 reports, the system automatically deletes the oldest reports and retains the 20 newest reports.

Managing Black Duck

Managing Components

Users with the **Component Manager** role can use the **Components** page to create and manage custom components, component versions, Black Duck KnowledgeBase components, and unmatched origins.

Open the Components page

1. Click 

> Components.

The **Components** page provides access to:

- The **Components** tab, which lists available components.
- The **Component Versions** tab, which lists component versions.
- Filtering and search tools.
- Actions for creating custom components and KnowledgeBase components.
- The **Unmatched Origins** page.

Components tab

The **Components** tab displays a list of managed components.

Select a component name to open the component details page.

If a component contains multiple versions, expand the component to view its versions. Select a version to open the component version details page.

The table includes the following information:

Description
Component Name of the component. Select the component name to open the **Overview** tab of the *Component Name* page. If there are multiple versions for this component, select > to display the versions. Select a version to open the **Overview** tab of the *Component Name* > *Version* page. indicates that there is a note for this component or component version. Hover over the icon to view the information.
License for this component.
Source for this component. Possible values are: • Custom. A custom component. • KnowledgeBase. An unmodified Black Duck KnowledgeBase component.

Description
• Modified KnowledgeBase. A modified Black Duck KnowledgeBase component.
Approval Status The approval status assigned to the component. You can change the status directly from the component's **Options** menu. Possible values are: • Unreviewed • In Review • Reviewed • Approved • Limited Approval • Rejected • Deprecated

For custom components, click the to open the **Options** menu at the end of a row to perform component-level actions.

From the Options menu,you can:

- Update the component's approval status:
  - Approved
  - Deprecated
  - In Review
  - Limited Approval
  - Rejected
  - Reviewed
  - Unreviewed
- Delete the component.

Note: The available actions may vary depending on the component type and whether the component is currently in use.

Component Versions tab

The **Component Versions** tab displays a list of component versions.

Select a version to open the component version details page.

The table includes the following information:

Description
Component The component version. **Version** Select the component name to open the **Overview** tab of the *Component Name* page. Select the version to open the **Overview** tab of the *Component Name* > *Version* page.
License for this component version.
Source for this component version. Possible values are: • Custom. A custom component version. • KnowledgeBase. An unmodified Black Duck KnowledgeBase component version.

Description
• Modified KnowledgeBase. A modified Black Duck KnowledgeBase component version.
Approval Status The approval status assigned to the component version. You can change the status directly from the component's **Options** menu. Possible values are: • Unreviewed • In Review • Reviewed • Approved • Limited Approval • Rejected • Deprecated
Last Updated When this component version was last modified and the user who last modified it.

For custom components versions, click the to open the **Options** menu at the end of a row to perform component-level actions.

From the Options menu,you can:

- Update the component version's approval status:
  - Approved
  - Deprecated
  - In Review
  - Limited Approval
  - Rejected
  - Reviewed
  - Unreviewed
- Delete the component version.

Note: The available actions may vary depending on the component type and whether the component version is currently in use.

About Custom Components

Custom components enable you to represent software components in your BOM that are not available from the KnowledgeBase, such as proprietary, commercial, internal, or otherwise untracked components. Users with the Component Manager role can create and manage custom components and add them to project BOMs.

Users with the **Component Manager** role can create and manage custom components and their versions, and then add them to project BOMs.This helps ensure that your BOM accurately reflects all software used by your project.

Custom components can contain information such as version details, licensing information, and descriptive metadata. Depending on how a custom component version is configured, it can also be associated with known security vulnerabilities. Custom component vulnerabilities are included in BOM risk calculations, policy evaluation, notifications, and reports when vulnerability data is available for that component version.

If you require a version of an open source component that is not available in the KnowledgeBase, contact Customer Support.

Managing custom components

Component Managers can use the following pages to manage custom components and custom component versions.

Page	Description
Custom Component Overview	View information about a custom component, including approval status, description, tags, custom fields, and component versions. Create additional versions for the component.
Custom Component Settings	View and update component details, manage custom fields and SBOM metadata, and delete a custom component.
Custom Component Version Overview	View information about a custom component version, including project usage, vulnerability counts, licensing information, approval status, and additional metadata.
Custom Component Version Vulnerabilities	View vulnerabilities associated with a custom component version. Vulnerabilities are determined by the CPE assigned to the component version.
Custom Component Version Origin IDs	View and manage origin identifiers associated with a custom component version. Origin IDs can be used to automatically match scan results to the component version.
Custom Component Version Settings	View and update component version details, manage licenses, custom fields, SBOM metadata, origin identifiers, and CPE values, and delete a custom component version.

Note: Component Managers must have permission to view projects before they appear on the Custom Component Version Overview page.

Understanding risk for custom components

Black Duck SCA evaluates custom components differently from KnowledgeBase components.

License risk

Custom components display license risk based on the license assigned to the component version.

Security risk

Custom component versions can display security risk when vulnerabilities are associated with the version. Associated vulnerabilities can be viewed from the custom component version's Vulnerabilities page.

Operational risk

Operational risk is not calculated for custom components.

Custom components in the BOM

When a custom component is added to a BOM:

- The match type is **Manually Added**.
- Policy Managers can create policy rules that apply to custom components.
- You can search for custom components by using the **Component Source** filter and selecting **Custom Component**.
- Project Version reports identify custom components separately from KnowledgeBase-managed components.
- Custom components do not have origins.

Project Version reports include a **Source/Type** field that identifies whether a component is a custom component or a KnowledgeBase-managed component.

Related tasks

- Create a custom component or custom component versions
- Managing custom components
- Manage custom component versions

Managing Custom Components

Users with the **Component Manager** role can create, modify, and delete custom components. A custom component represents a software component that is not managed by the Black Duck KnowledgeBase.

Custom components can contain one or more component versions. Component Managers can maintain component information, track version-specific details, organise components with tags and statuses, and manage the lifecycle of custom components and their versions.

Tasks

- Create a custom component.
- View custom component information to edit or delete a custom component.
- Create additional versions for a custom component.
- Manage component tags and statuses.

Unmatched component auto-creation on SBOM import

In an SBOM management workflow, the SBOM is the input and all of the components included in the SBOM need to be persisted in the SBOM management solution so that visibility isn't lost, regardless if there is a match to the KnowledgeBase. This feature provides the option to automatically populate unmatched components in the BOM with custom components of the same name in an SBOM import.

To use unmatched component auto-creation:

1. Log in to Black Duck SCA.
2. Click 

.
3. Click the **Upload File** button and select either **SBOM-SPDX** or **SBOM-CycloneDX**.
4. Upload the SBOM file(s).

5. Check the **Unmatched Component Auto-Creation** checkbox at the bottom of the Upload SBOM dialog box.
6. Map the scan to a project or create a new project for the scan.

Important: Components in the SBOM import must have an associated package URL (PURL) in order to be automatically populated in the BOM.

Tip: Unmatched components will automatically use the default license as configured in the System Settings.

Viewing auto-populated components in the BOM

Once your scan is mapped to a project, you can view the BOM by clicking the project's name in the **Mapped To** column of the Scans page or by navigating to the project on the Dashboard. Unmatched components that have been auto-populated will be included in the BOM report and can be found by using the Source/Type → Custom component filter on the project version page.

About Black Duck KnowledgeBase components

The Black Duck® KnowledgeBase™ (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 Black Duck KB includes more than 9 million unique components from more than 50,000 sites and contains detailed data on more than 260,000 actively traced vulnerabilities across billions of lines of code. The Black Duck KB includes detailed data for more than 2,900 unique licenses, including the full license text and dozens of encoded attributes and obligations for each license. Black Duck connects to a version of 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 Black Duck KB that supports Black Duck SCA.

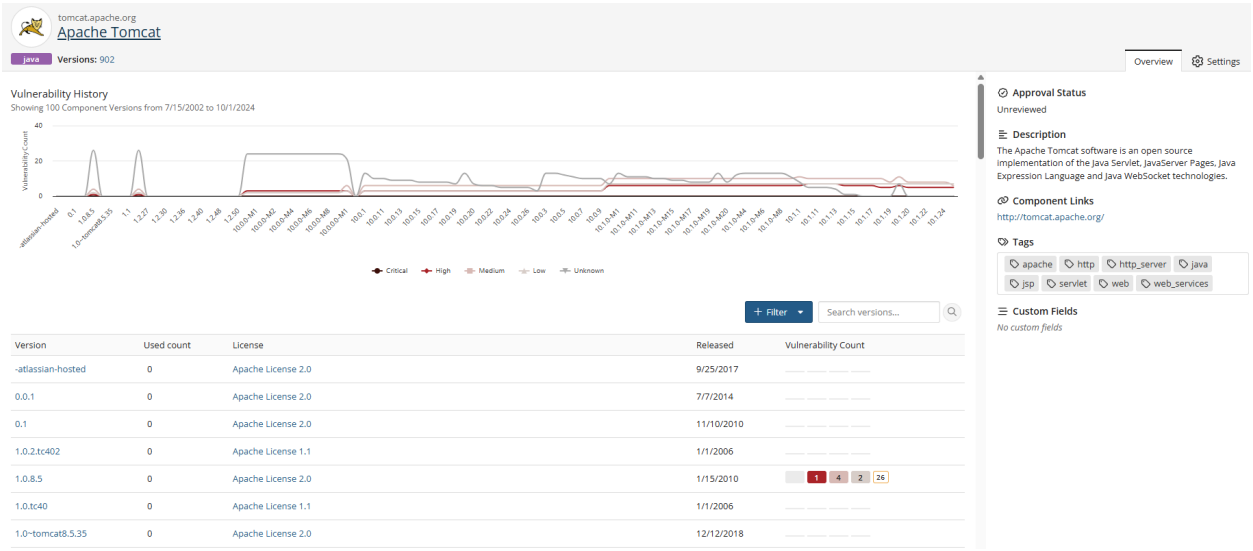
The Black Duck KB provides information about OSS components at the component level and at the component version level.

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.



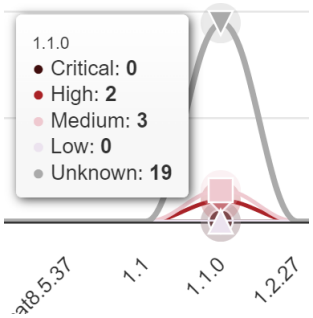
This page is comprised of two tabs: an **Overview** tab and a **Settings** tab.

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 Black Duck KB.

A graph at the top of the page shows a history of high, medium, and low vulnerabilities for each version of this component. Use this graph to quickly view vulnerability information for component versions.

- Select **Previous** or **Next** to view older or newer versions.
- Hover over a data point in the graph to view the version, release date, and number of vulnerabilities for this version:



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

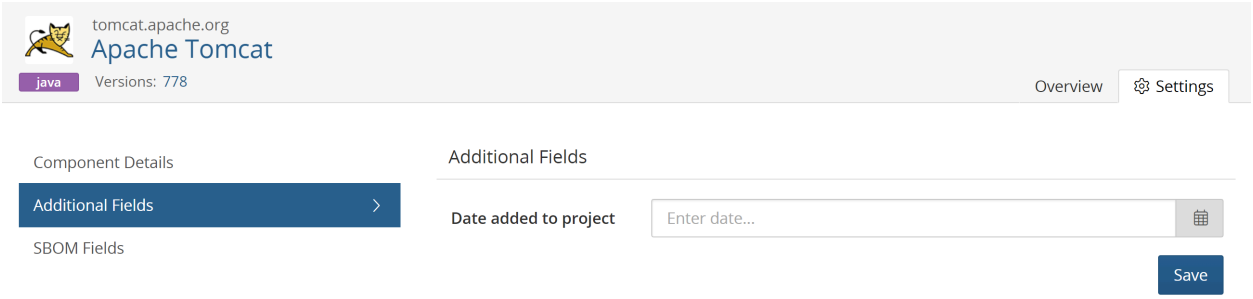
The following information is available for each version:

Column	Description
Version	Release number of this version of the component. Select the version number to display the *Component Name > Version* page.
Used Count	Number of project version BOMs in which this version of the OSS project is used. **Tip:** Select the number to go to the **Overview** 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.

Column	Description
License	Declared license of this version of the OSS component. Other license types include: • "Unknown" indicates that the OSS component version's license is not known. • "License Not Found" indicates that although researched by Black Duck, no declared license was found for the component. • "No License" indicates that Black Duck has found a declaration of 'No License' for the component. For known licenses, select the license name to view license details and license text.
Released	The date this version of the OSS component was released.
Security Risk	A graph which shows the number of high risk, medium risk, low risk, and unknown vulnerabilities associated with this version of the OSS component.

About the Settings tab

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

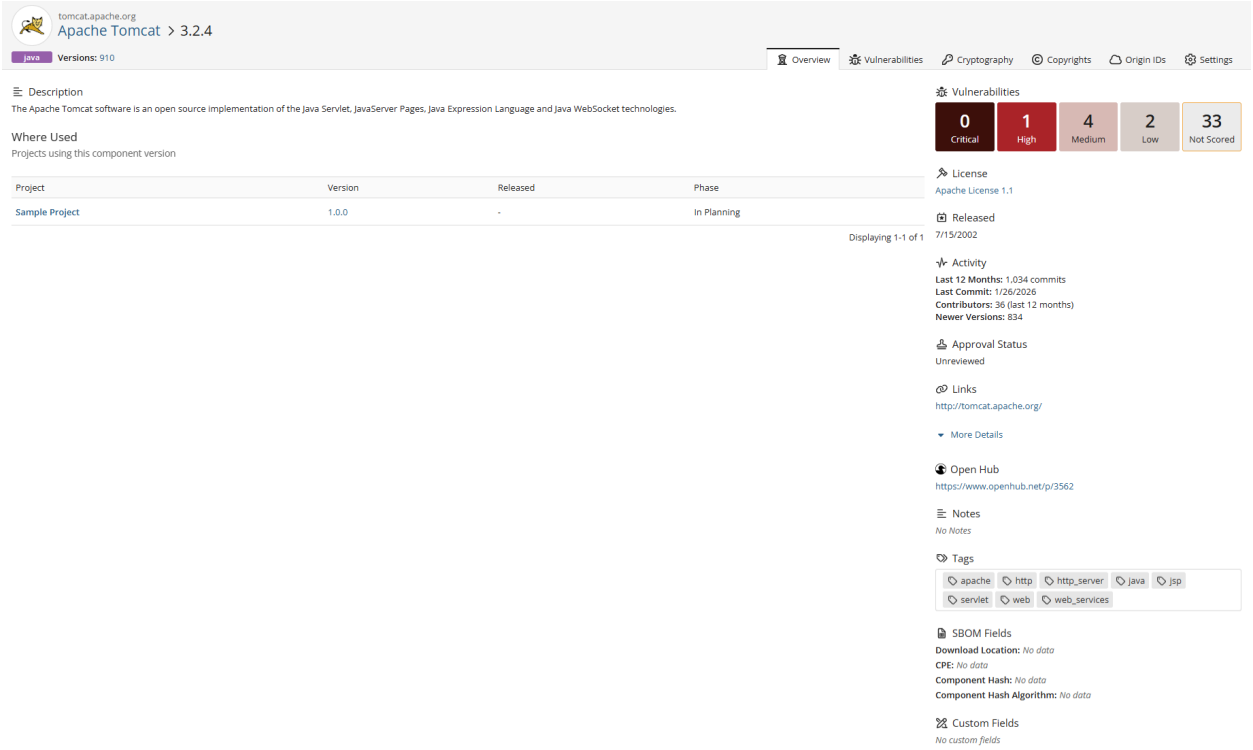


Users with the Component Manager role can use the **Settings** tab to edit the description, URL, notes, and status for this KB component. Click here for information on editing component information and here for information on modifying a component's status.

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 **Overview** tab provides the following information:



- **Description**: A brief summary of the component's purpose and functionality.
- A **Where Used** table that lists the projects and respective versions in which this version of the component is used, excluding any Bills of Materials (BOMs) that contain the component if it has been marked as ignored.
- **Vulnerabilities**: Count of vulnerabilities associated with this version.
- **Licenses**: Information on the licenses that apply to this component.
- **Released**: The date when this version was published.
- **Activity**: Insights into commit activity and trends over the past 12 months.
  - **Last 12 months**: The number of commits.
  - **Last Commit:** This indicates the date of the most recent commit made to the component, providing insight into the latest updates or changes.
  - **Contributors:** The total number of contributors who have contributed to the component in the past year. This reflects the level of community involvement and support for the component.
  - **Newer Versions:** This section displays the number of newer versions that have been released since this version. It helps users understand if there are updates available that may include important features or security fixes.
- **Approval Status:** Current Approval Status of this version.
- **Links:** Links to the component resources, if available.

Expand the **More Details** link to see additional information:

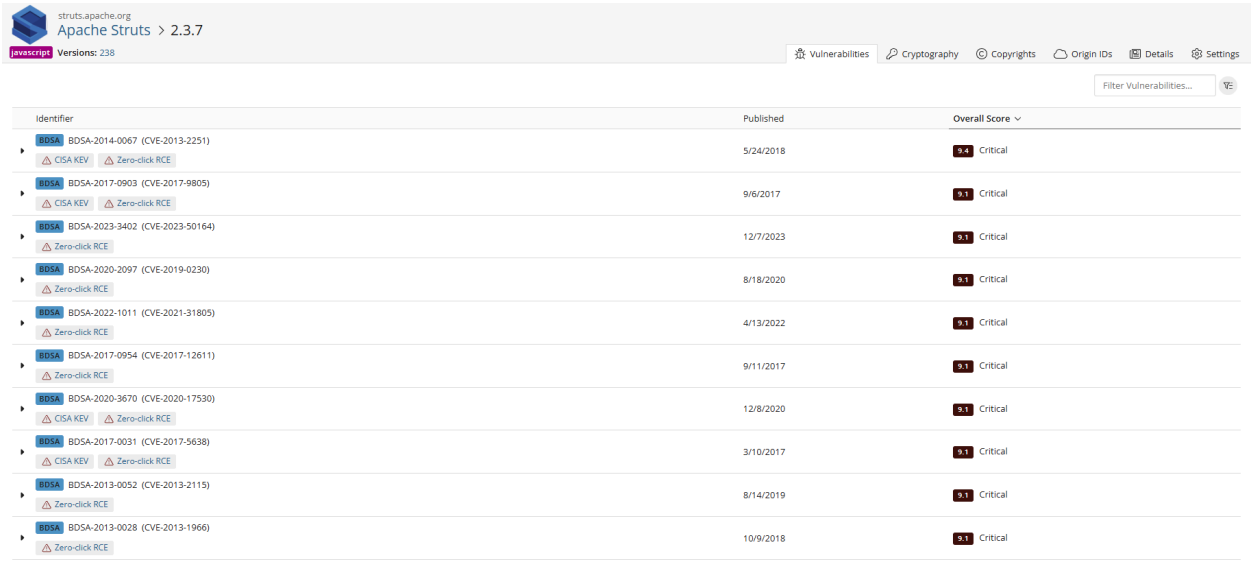
- **Open Hub**: Links to Open Hub, if available.
- **Notes**: Any notes for this component version.
- **Tags**: Relevant tags associated with the component, if available.

- **SBOM Fields:** Details on Software Bill of Materials fields, if present.
- **Custom Fields:** Any additional custom fields relevant to the component.

The table contains the following information:

Column	Description
Project	Name of the project that uses this version of the OSS component from 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 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 Black Duck KB *Component Name Version* page, the **Vulnerabilities** tab displays the list of vulnerabilities associated with this version of the OSS component from Black Duck KB.



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. • For BDSA, the Temporal, Base, Exploitability, and Impact scores are shown.

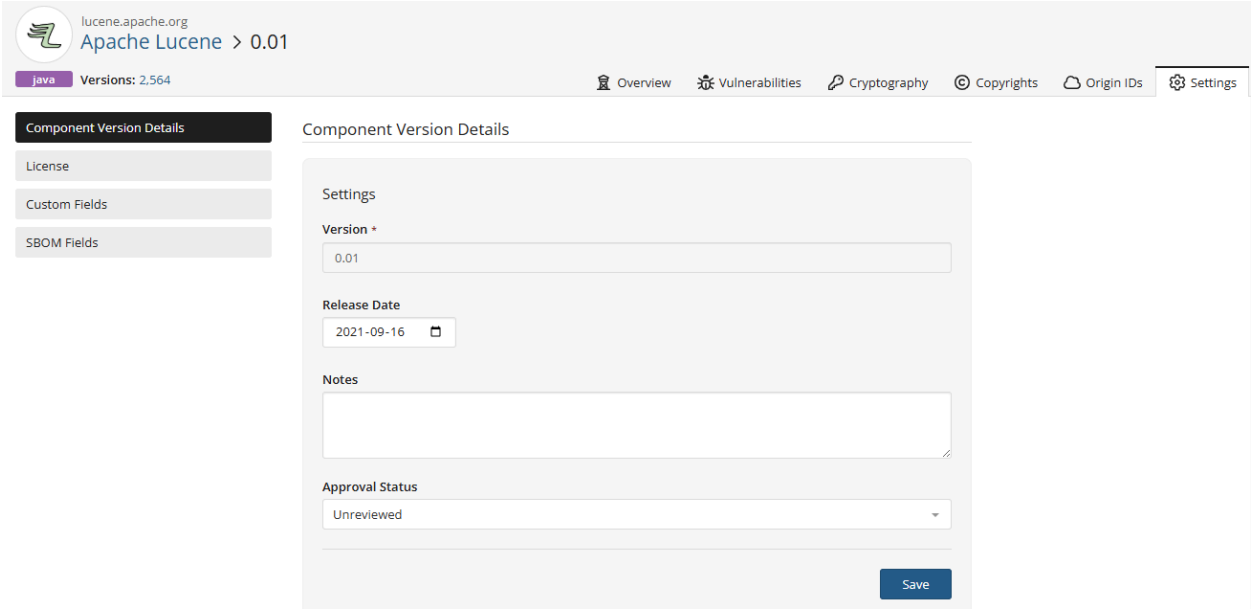
Column	Description
	• For NVD, the Base, Exploitability, and Impact scores are shown. The Temporal score represents time-dependent qualities of a vulnerability, taking into account the confirmation of the technical details of a vulnerability, the existence of any patches or workarounds, and the availability of exploit code or techniques. The Base score reflects the overall basic characteristics of a vulnerability that are constant over time and user environments: • Attack Vector (AV) • Attack Complexity (AC) • Priviledges Required (PR) • User Interaction (UI) • Scope (S) • Confidentiality (C) • Integrity (I) • Availability (A) • Exploit Code Maturity (E) • Remediation Level (RL) • Report Confidence (RC) For more information, see the CVSS specification document section on Exploitablility Metrics. 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. The Impact score reflects the possible impact of successfully exploiting the vulnerability, considering the integrity, availability, and confidentiality impacts.

The **Cryptography** tab shows information on component versions that have encryption algorithms. Click here for more information. This tab will only appear if you have **Cryptography** enabled on your Product Registration key.

The **Origin IDs** tab lists all known external IDs and Package URLs (PURLs) associated with a specific component version.

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 **Overview** tab.



Users with the Component Manager role can use the **Settings** tab to edit information for this KB component version.

- Select **Component Version 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.
- Select **Custom Fields** to edit any custom values or properties set by the Custom Fields Administrator.
- Select **SBOM Fields** to edit any SBOM values or properties set by the Custom Fields Administrator.

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.

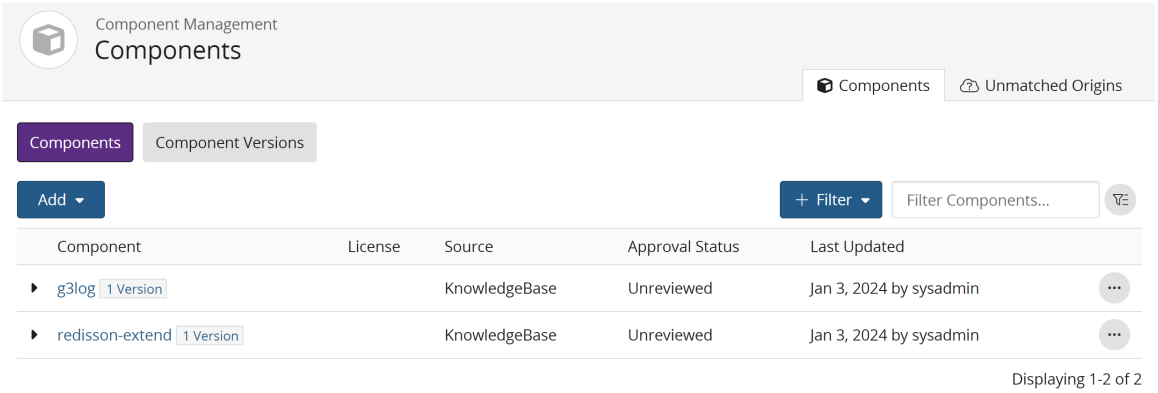
Note: Setting the status of a KB component and all versions listed in the Component Management table to **Unreviewed** removes the KB component and its versions from the Component Management table. Note that this does not apply to those KB components and versions shown with a source of **Modified KnowledgeBase**.

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 

> **Components**.

   The **Components** tab appears.



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 

> **Components**.

   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 

> **Components**.

   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 **Overview** 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**.

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 

> **Components**.

   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 

> **Components**.

   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

The KnowledgeBase Feedback Service allows users to submit feedback on component and vulnerability data in Black Duck, helping to improve the accuracy of the Black Duck KnowledgeBase (KB). Feedback is sent when specific adjustments are made to components or matches identified by the KB.

When feedback is sent:

- **Component adjustments**: Feedback is submitted when the component itself is adjusted or when a component's license is modified. Other component adjustments do not trigger feedback submission. Other component adjustments do not trigger feedback submission.
- **File adjustments**: Feedback is sent only when the component matched to a specific file is changed.
- **BOM adjustments**: Feedback is also triggered when making adjustments to a component, version, origin, origin ID, or license for a match made by the KB.
- **Unmatched files**: If you manually identify an unmatched file to a component, feedback is sent. However, feedback is not sent for manually added components that do not have associated files.

Note: No customer-identifiable information is transmitted to the KB.

About unmatched origin components

Components with unmatched origins are components with origin IDs that Black Duck identified during a package manager scan but could not be mapped to a component version. You can manually map any given origin ID to a custom component version.

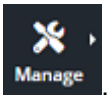
Components with unmatched origins can be found in these locations in Black Duck:

- Scans page: Click the desired scan and then click the View BOM Import Log button to view the components with unmatched origins specific to this scan.
- Project version BOM page: Click the Unmatched link on the top right of the BOM report to view the components with unmatched origins specific to this project version.
- Unmatched Origins page: Users with the Component Manager role can click the Manage button and then select Unmatched Origins.

Managing unmatched origin components

The Unmatched Origins page allows users with the Component Manager role to manage components with unmatched origins that are found throughout the projects on your server.

Getting to the Unmatched Origins page

1. Log in to Black Duck as a user with the Component Manager role.
2. Click 

.
3. Select **Unmatched Origins**.

What's on the Unmatched Origins page?

On the Unmatched Origins page, you will find a table with the following information:

- **Origin ID**: The namespace and package ID for the component package.
- **Type**: The type of origin identifier. Possible values are:
  - **External ID**: Comes from Package Management scan with format `namespace:external_id`.
  - **Package URL**: Comes from SBOM scan with proper Package URL format. (`pkg:...`)
- **Occurences**: Number of code locations or scans this package ID is found. Please note that this will only display occurrences for scans performed after upgrading to Black Duck 2023.7.0. Occurrences appearing in scans performed prior to this upgrade will not be tallied in this total.
- **Mapping**: The custom component to which this package is mapped to. Once a new scan is performed and this component is found, the entry will be removed from this table and added to the resulting BOM report. Clicking the component in the BOM will open the Source tab and display the mapped package.

Note: Matching to custom components requires the use of Detect 7 or higher and is currently only supported for package manager scans.

Filtering the unmatched origins table

The following filtering options are available:

- **Mapped**: Display origin IDs that are either mapped or unmapped to a project version.
- **Origin Type**: Display origin IDs of either external ID or package URL types.

You can also use the **Filter origins...** text field to search for specific origin IDs.

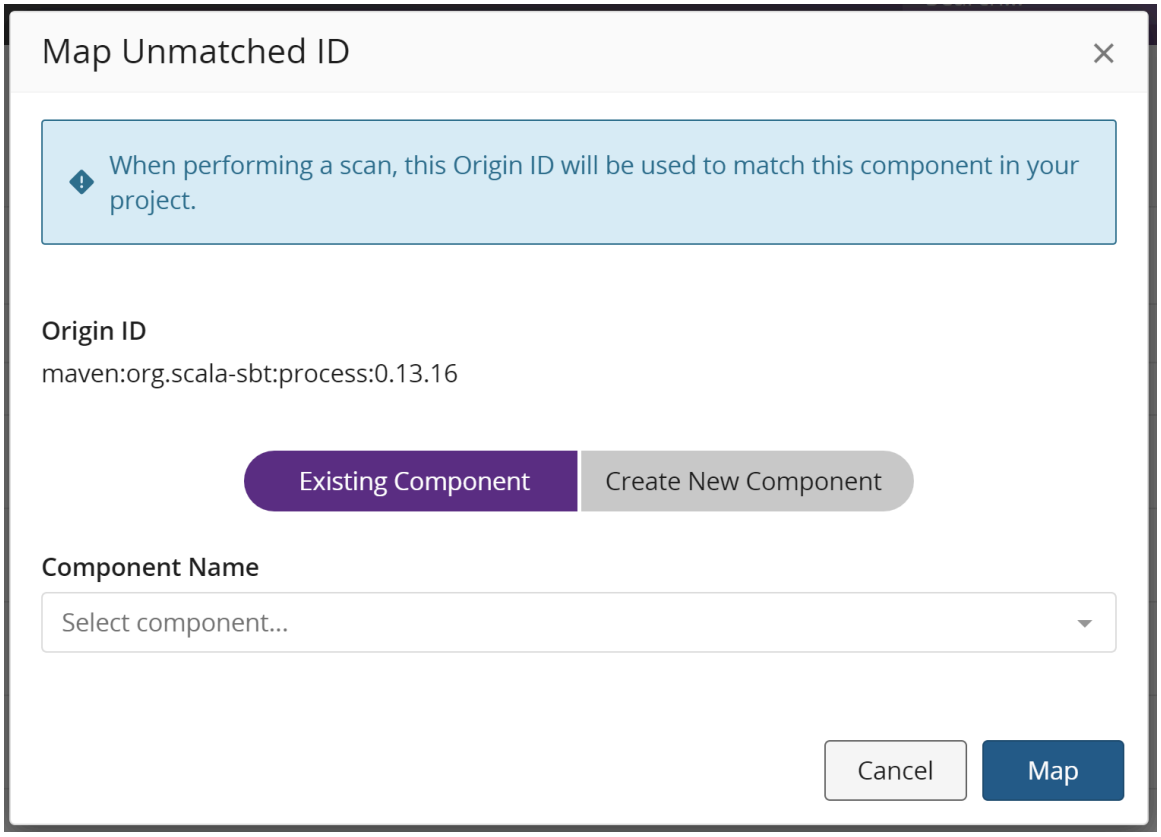
Mapping unmatched origin components

To map a component:

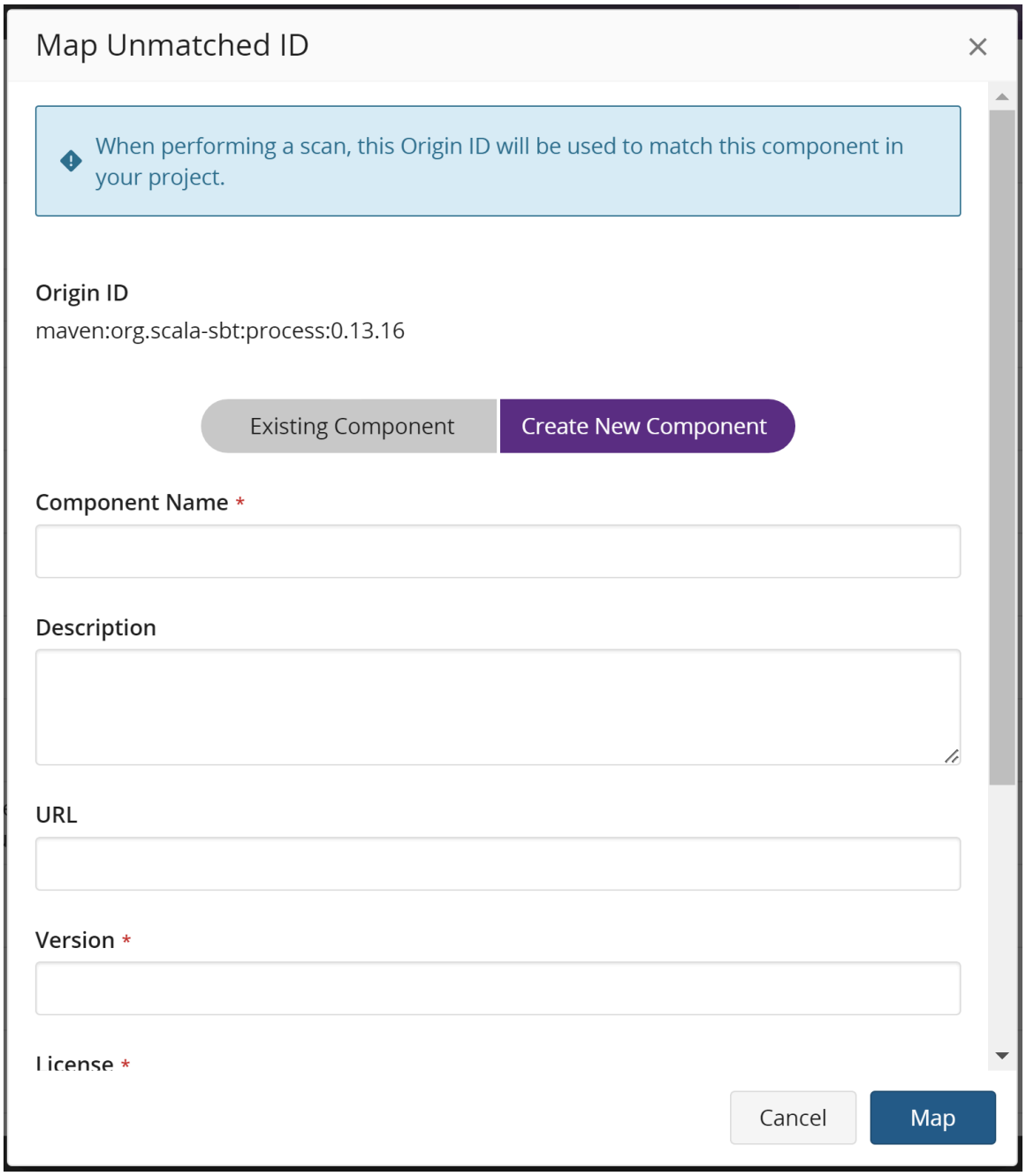
1. Click at the end of the component's row. The Map Unmatched ID dialog box appears.
2. Select either Existing Component or Create New Component.
   - Existing Component: Click the **Component Name** dropdown box, select a custom component from the list, and then select a version from the **Version** dropdown menu.

     You can instead create a new version for the custom component by clicking **+ Create New Version** from the dropdown menu. Doing so will add new options to the dialog box. Enter the new version in the **New Version Name** field. Click the **License** dropdown to select a license from the list displayed. Optionally, you can add a release date and approval status.

     Click the **Map** button when you are satisfied with your selections to complete the mapping.



   - Create New Component: Enter a name in the **Component Name** field for the custom component. Enter a version and select a license from their respective fields. Optionally, you can also add a description, URL, release date, and approval status. Click the **Map** button create the custom component and complete the mapping to the new component.



To unmap a component:

1. Click at the end of the component's row. The Remove Origin ID dialog box appears.



2. Click the **Remove** button to confirm the unmapping.

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 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".

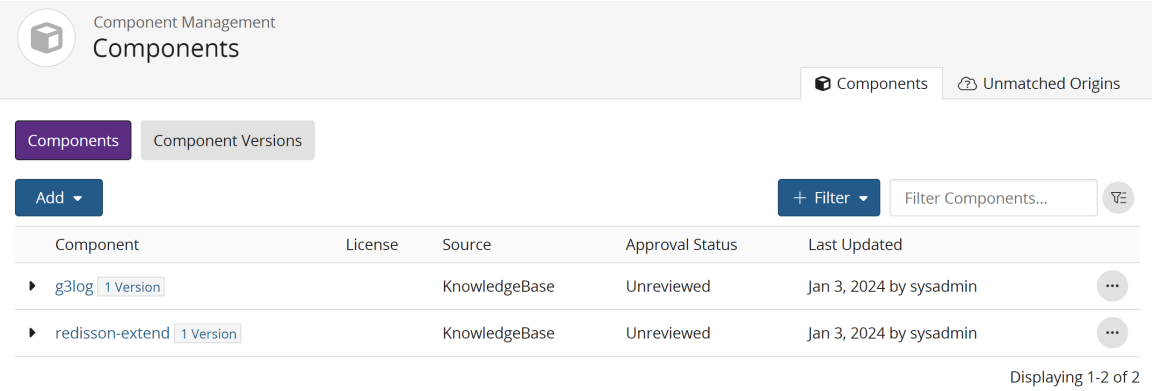
Note that the status of a component is independent of the status of its versions.

To modify the status for a component:

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

> **Components**.

   The **Components** tab appears.



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:
     a. Select the component you wish to modify from the **Components** tab.

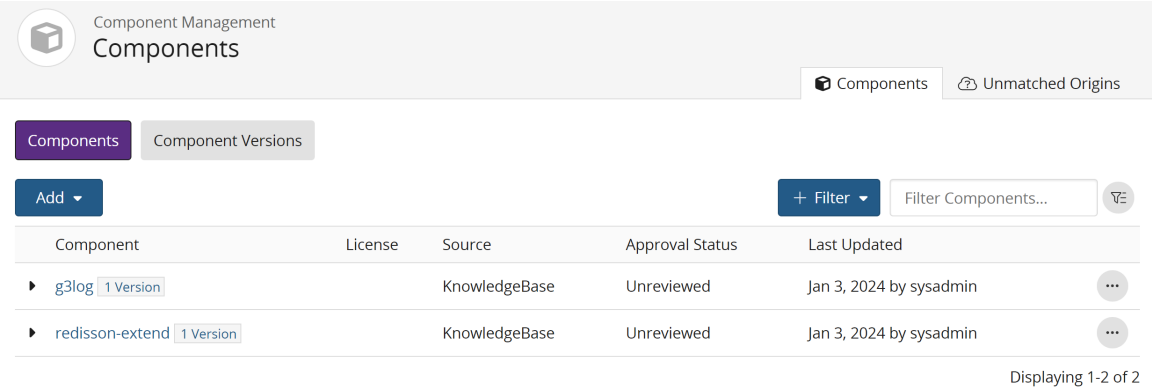
        The **Overview** tab of the *Component Name* page appears.
     b. Select the **Settings** tab.
     c. Select a status from the **Approval Status** list and click **Save**.

To modify the status for a component version:

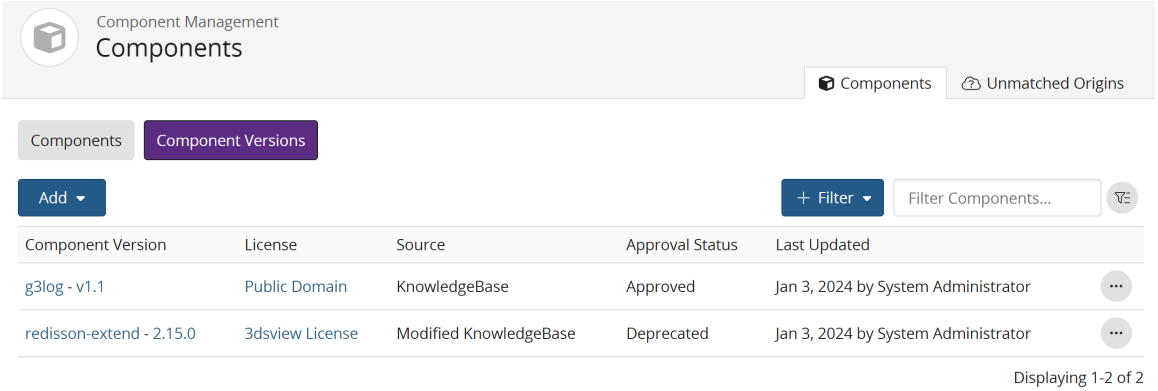
1. Log in to Black Duck with the Component Manager role.
2. Click 

> **Components**.

   The **Components** tab appears.



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



4. Do one of the following:

- Click in the row of the component version that you want to change the status and select a status from the list.
- Modify the status using the **Settings** tab in the *Component Name* > *Version* page:
  a. Select the component version you wish to modify from the **Component Versions** tab.

     The **Overview** tab of the *Component Name* > *Version* page appears.

  b. Select the **Settings** tab.

  c. Select a status from the **Approval Status** list and click **Save**.

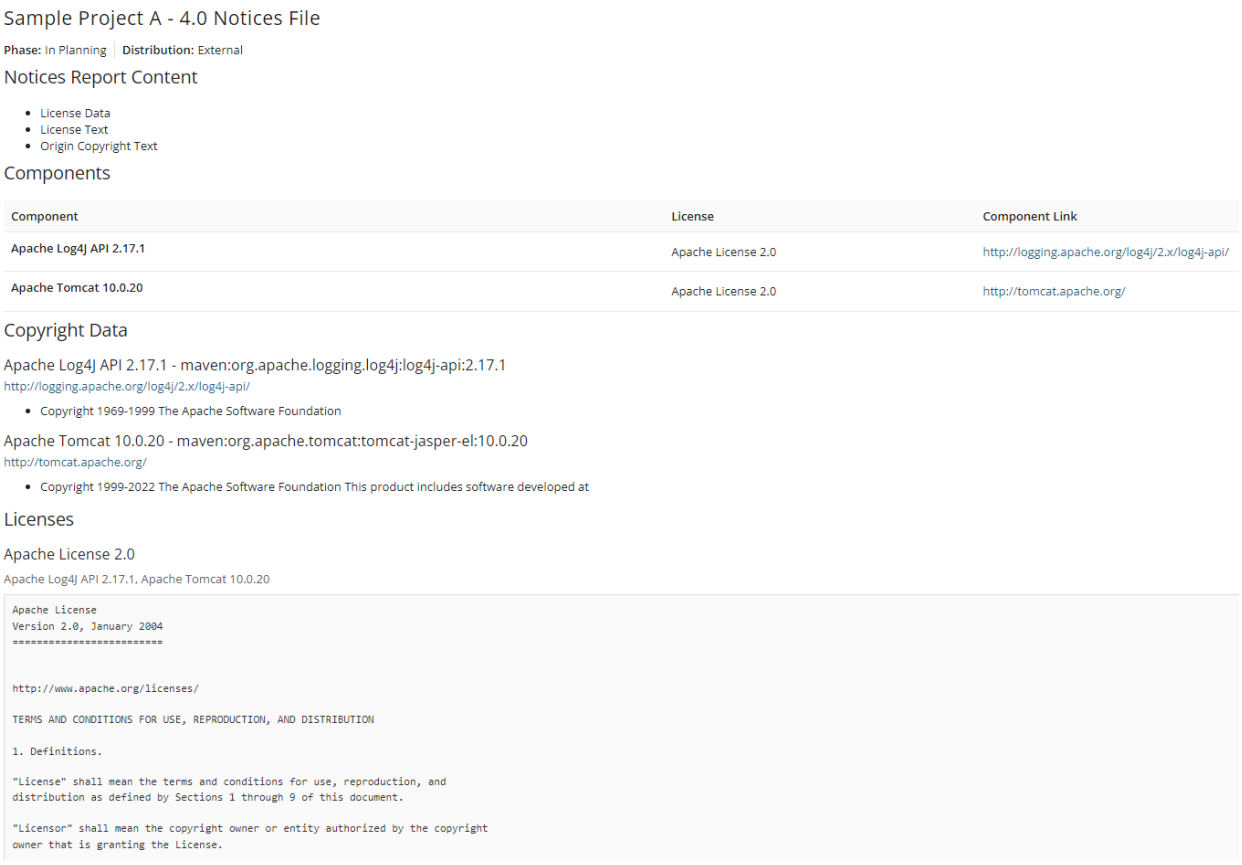
Managing open source licenses

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 A - 4.0 Notices File

Phase: In Planning | **Distribution:** External

Notices Report Content

- License Data
- License Text
- Origin Copyright Text

Components

Component	License	Component Link
Apache Log4J API 2.17.1	Apache License 2.0	http://logging.apache.org/log4j/2.x/log4j-api/
Apache Tomcat 10.0.20	Apache License 2.0	http://tomcat.apache.org/

Copyright Data

Apache Log4J API 2.17.1 - maven:org.apache.logging.log4j:log4j-api:2.17.1
http://logging.apache.org/log4j/2.x/log4j-api/

- Copyright 1969-1999 The Apache Software Foundation

Apache Tomcat 10.0.20 - maven:org.apache.tomcat:tomcat-jasper-el:10.0.20
http://tomcat.apache.org/

- Copyright 1999-2022 The Apache Software Foundation This product includes software developed at

Licenses

Apache License 2.0
Apache Log4J API 2.17.1, Apache Tomcat 10.0.20

```
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 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 2,900 open sources licenses that can range from those with few restrictions and obligations to those with many restrictions and obligations.

Depending upon the nature of these restrictions and obligations, some licenses are deemed to be riskier than others, as they require more management and care to ensure compliance with the license terms. Typically, the riskiest licenses are those that are reciprocal in nature. Reciprocal licenses, often pejoratively called “Viral Licenses”, are those in which the license terms can extend beyond the open source code itself and can try to apply to other code as well. The other code could be modifications to the open source, or even simply code that uses the open source code in a way that triggers the reciprocal nature of the licenses. Once triggered, it is possible that in order to be in compliance to the license, developers who create software applications may need to treat the entire application as under the open source license and comply with all these obligations for the entire application. This could include the obligation to provide all the source code for the application (not just the open source) and allowing people who receive the application to modify and redistribute it without restrictions. This may be in conflict with a proprietary license model.

Please note, the legal aspects of managing open source licenses can be complicated and often it is best to seek legal counsel when making decisions about open source licenses and creating policies regarding their use. Legal counsel can best help determine if the license rights, restrictions, and obligations apply in a particular scenario. However, in order to help customers manage these risks in a simple and effective way, Black Duck categorizes open source licenses into license families for purposes of risk calculations and the definition of open source policy rules. These families range from those that are highly reciprocal to those with few obligations and restrictions. These license families, called KnowledgeBase licenses are:

- 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.

- 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.

- 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.

- 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.

- AGPL (Affero General Public License)

  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.

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)	• GNU Affero General Public License v3 or later
Reciprocal	• GNU General Public License (GPL) 2.0 or 3.0 • Sun GPL with Classpath Exception v2.0
Weak Reciprocal	• 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	• Apache 2.0 • Artistic License • BSD License 2.0 (2-clause Simplified, 3-clause, New, or Revised)

License Family	Examples
	• Do What The F*ck You Want To Public License • ISC License • Microsoft Public License • MIT License • Zlib-Libpng License
Unknown	N/A

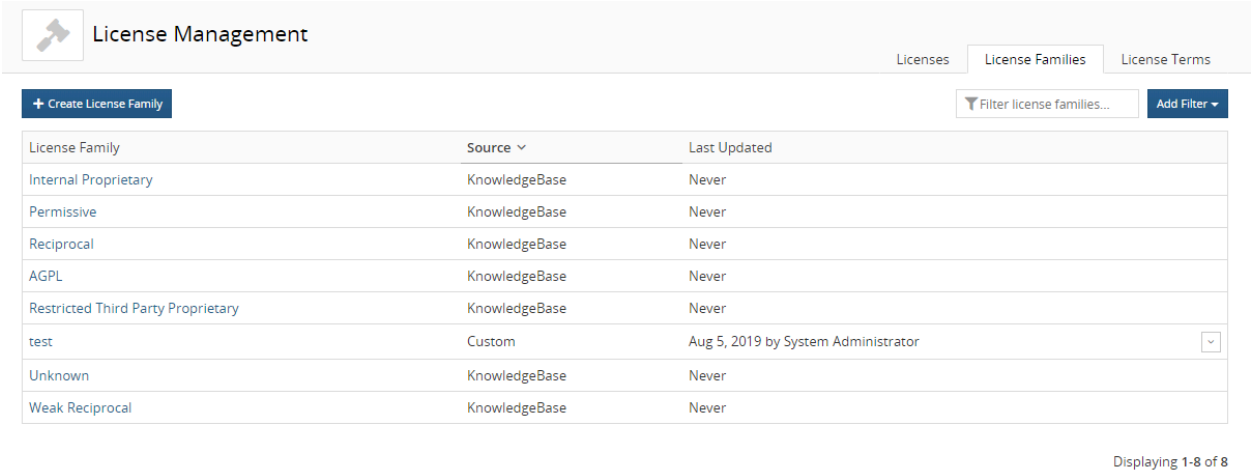
Managing license families

Users with the License Manager role can use the License Families 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 Families page:

1. Log in to Black Duck with the License Manager role.
2. Click 

→ **Licenses Families**.

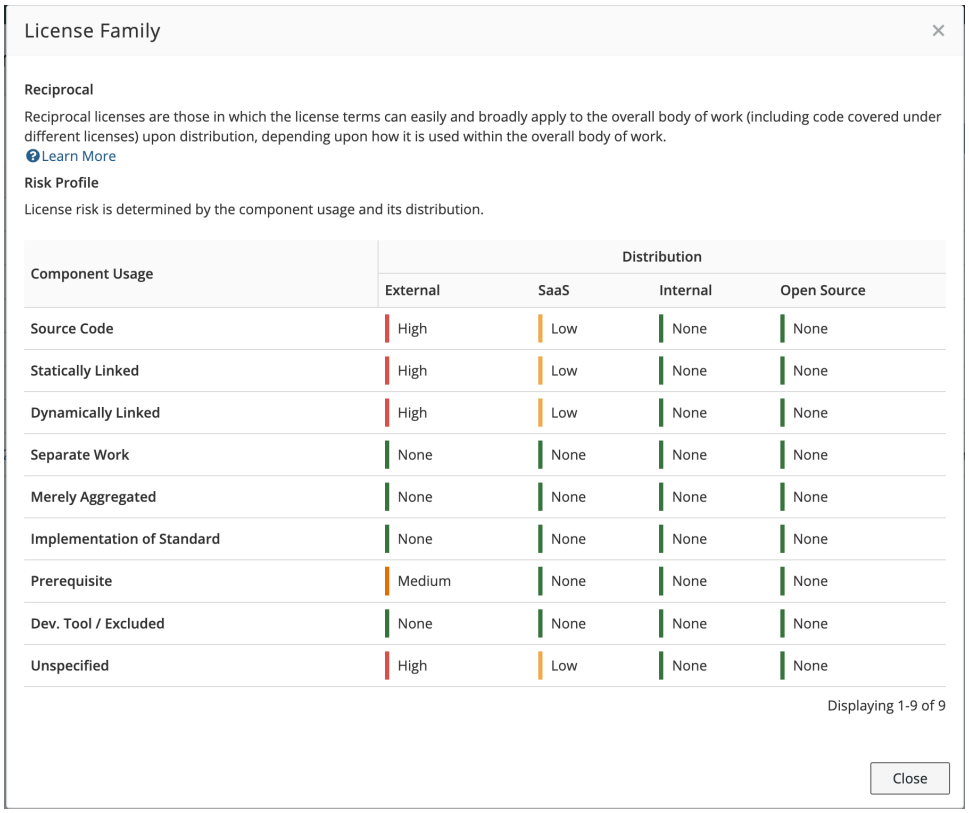


The table contains the following information:

- **License Family**. The name of the license family category.
- **Source**. The source for this license. Possible values are:
  - KnowledgeBase. From Black Duck KnowledgeBase.
  - Custom. Custom license family.
- **Last Updated**. The 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 **License Family Source** filter to limit the information shown on this page. Filter options are Custom or KnowledgeBase.

3. Click the desired license family. This will open a modal displaying the following information:



- **License family category**. The description for this license family. Categories include Restrictive Third Party Proprietary, Permissive, Reciprocal, Internal Proprietary, Unknown, Weak Reciprocal, AGPL.
- **Risk Profile**. The license risk as determined by the component usage and its distribution.

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

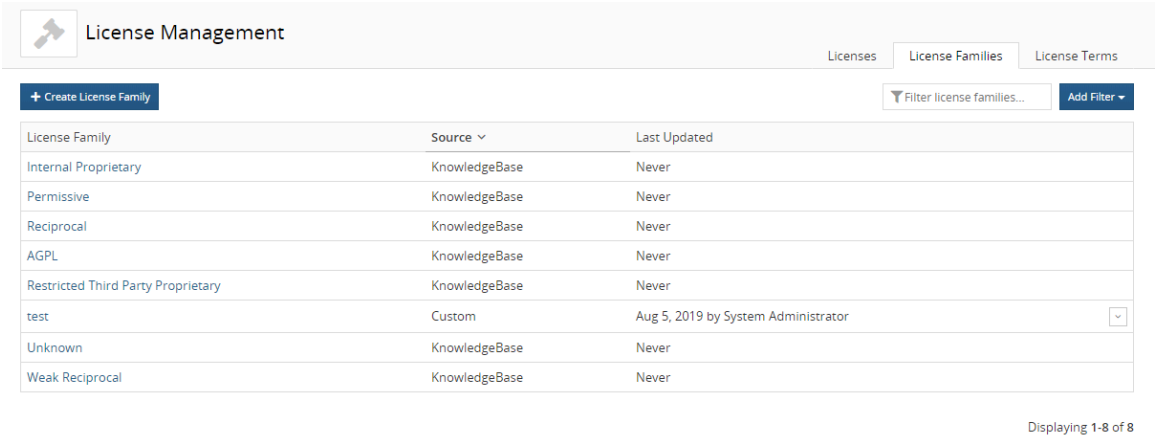
Only users with the License Manager role can create custom license families.

To create a custom license family:

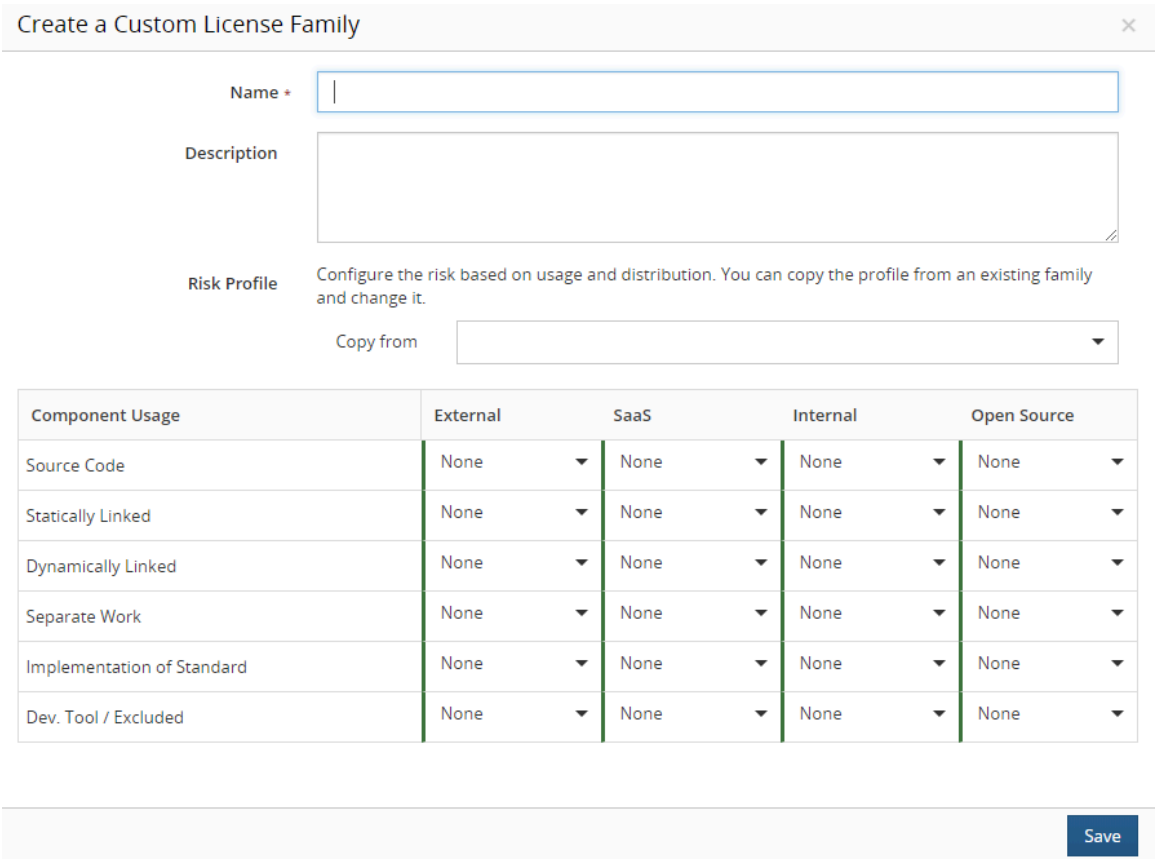
1. Log in to Black Duck with the License Manager role.
2. Click Manage > **Licenses**.

   The License Management page appears.

   Select the **License Families** tab to display all license families.



3. Click **Create License Family** to open the Create a Custom License Family dialog box.



4. Enter a name for this license family.
5. Optionally, enter a description.
6. Optionally, modify the license risk values. By default, the license risk is None for all usages and distributions. You can select a license family to use as a baseline for the license risk by selecting one from the **Copy from** list. You can then use these license risk values for the custom license family or modify the values by using the drop downs to modify the license risk by usage and distribution. Possible license risk values are: none, low, medium, and high.
7. Click **Save**.

Editing custom license families

Custom license families can be edited by users with the License Manager role.

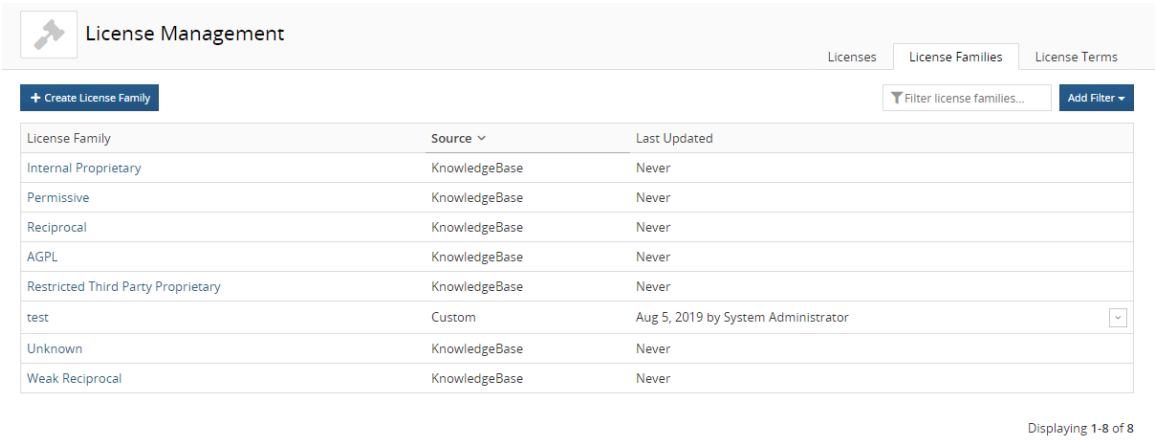
Note: Adjusting the risk profile for a license family will not change the calculated license risk for components on existing BOMs. Changes will only be reflected on project versions when the BOM is recalculated, such as during rescans or when assigning a scan to a project version.

To edit a custom license family:

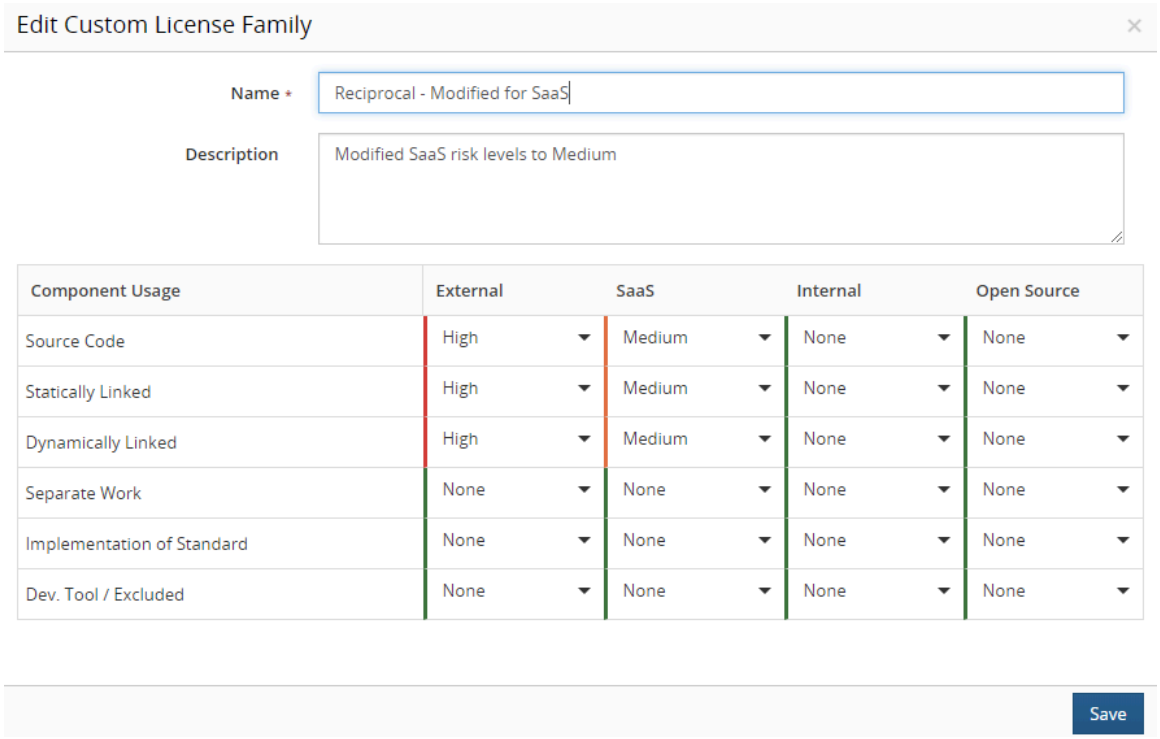
1. Log in to Black Duck with the License Manager role.
2. Click 

> **Licenses**.

   The License Management page appears.
3. Select the **License family** tab to display all license families.



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.

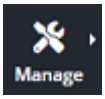


5. Modify the information shown for this custom license family.
6. Click **Save** in the Edit Custom License dialog box. The username of the user who edited this license family and the date appears in the **Last Updated** column.

Deleting custom license families

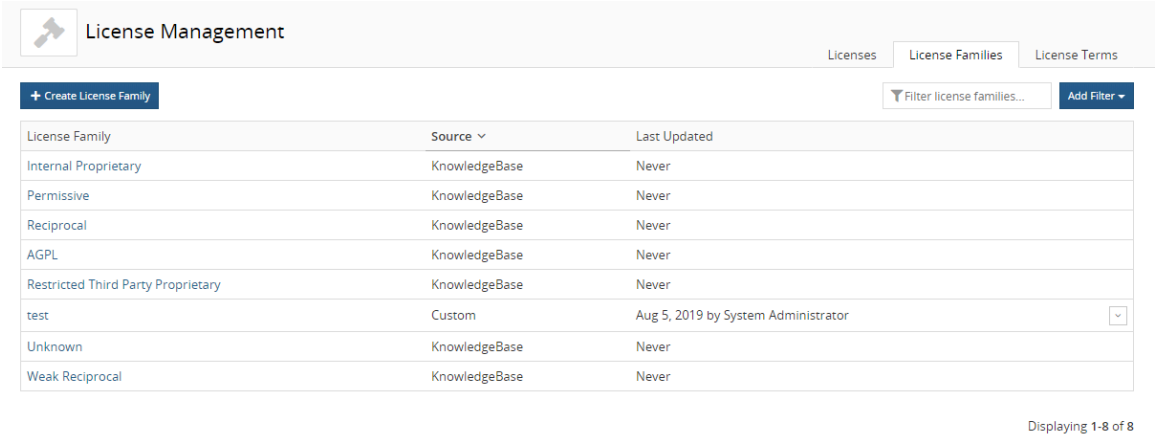
You cannot delete a license family that is being used by a license in a BOM.

You also cannot delete licenses provided by Black Duck KnowledgeBase.

1. Log in to Black Duck with the License Manager role.
2. Click 

> **Licenses**.

   The License Management page appears.
3. Select the **License Family** tab to display all license families.



4. Click and select **Delete** in the row of the custom license family that you want to delete to display a confirmation dialog box.

   An error message appears if you try to delete a custom license family that is currently being used by a license.

5. Click **Delete** to confirm.

Viewing licenses

The **Licenses** tab in the License Management page displays custom licenses you have created and the licenses from 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:

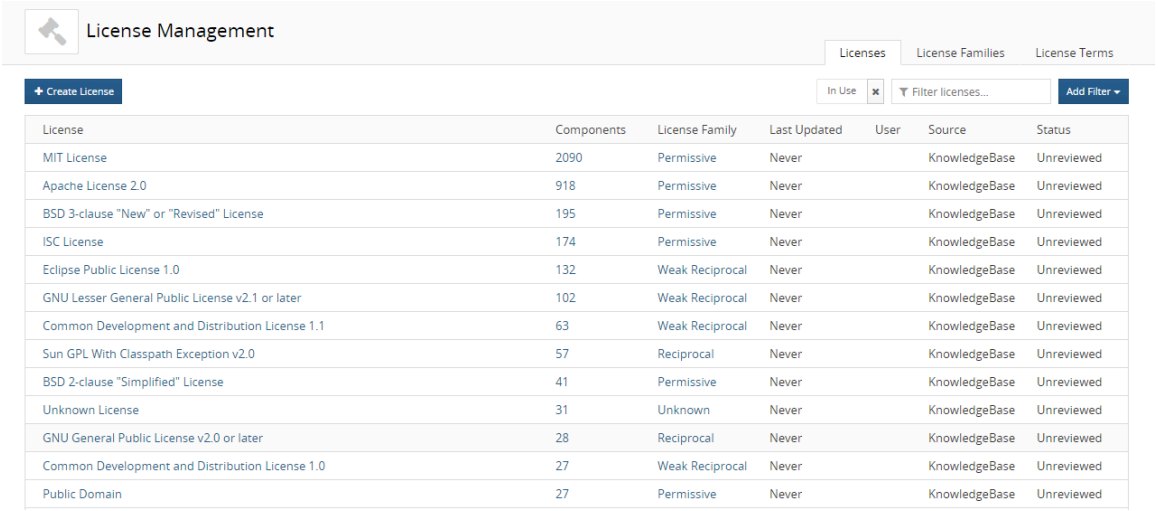
- Create, edit, or delete custom licenses.
- Edit KnowledgeBase licenses.
- View the full text of custom and Black Duck KnowledgeBase licenses.
- View the number of components in your projects that use a specific license.

Note: Edits made locally by a BOM manager 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.

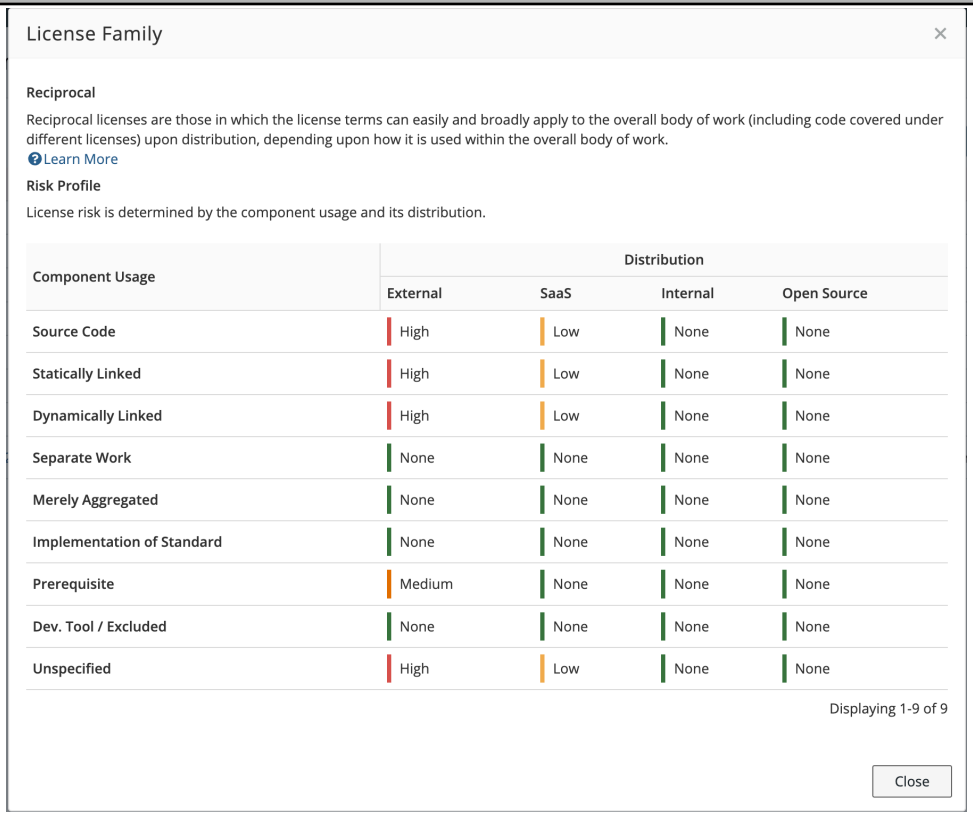


License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

Select the **Licenses** tab to view a table with the following information.

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	License name. Select the name to display the *License Name* page. Use the: • **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	Number of components or subprojects in all projects that have this license. **Note:** The value shown here does *not* include projects assigned with this custom license. Select the component value to display a page which lists the component versions or subprojects where this license is used.
License Family	The license family for this license. Select a license family to view a definition and risk profile for that license family:

Column	Description
	
Last Updated	Time, if updated today, or date that the license was last updated.
User	Username of the user who created or last updated the license. This field is empty for licenses from Black Duck KnowledgeBase that have not been edited.
Source	Source for this license. Possible values are: • KnowledgeBase. From Black Duck KnowledgeBase. • Modified KnowledgeBase. An edited Black Duck KnowledgeBase license. • Custom. Custom license.
Status	The review status for the license. Possible values are: • 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.

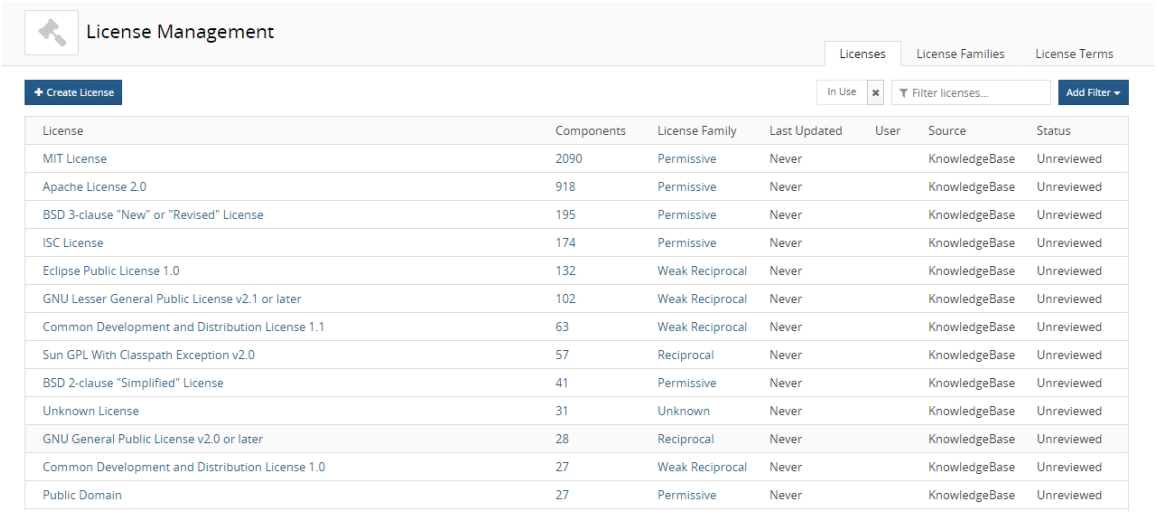
Note: For KnowledgeBase licenses, if the license is one that is modified for individual components (like the BSD or MIT license), then the template license text is shown here. However, when viewing the license text in the context of a component (such as viewing the component's license in a BOM), the 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 

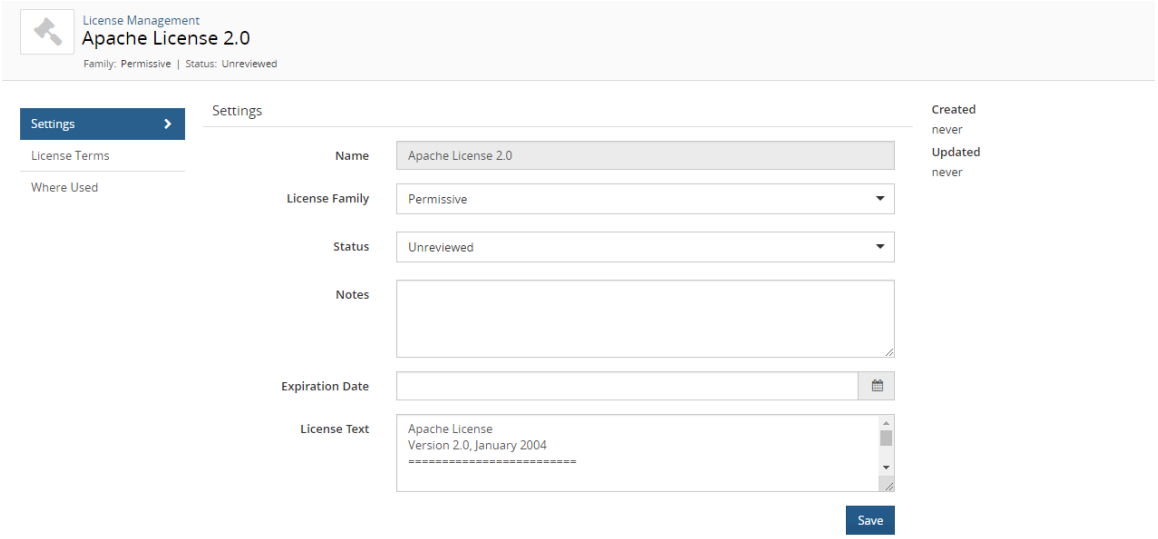
> **Licenses**.

   The License Management page appears.



License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

3. In the **Licenses** tab, select the license name to display the *License Name* **Settings** tab which displays the license text:



With the appropriate role, you can also view the license text in a BOM.

Viewing license use

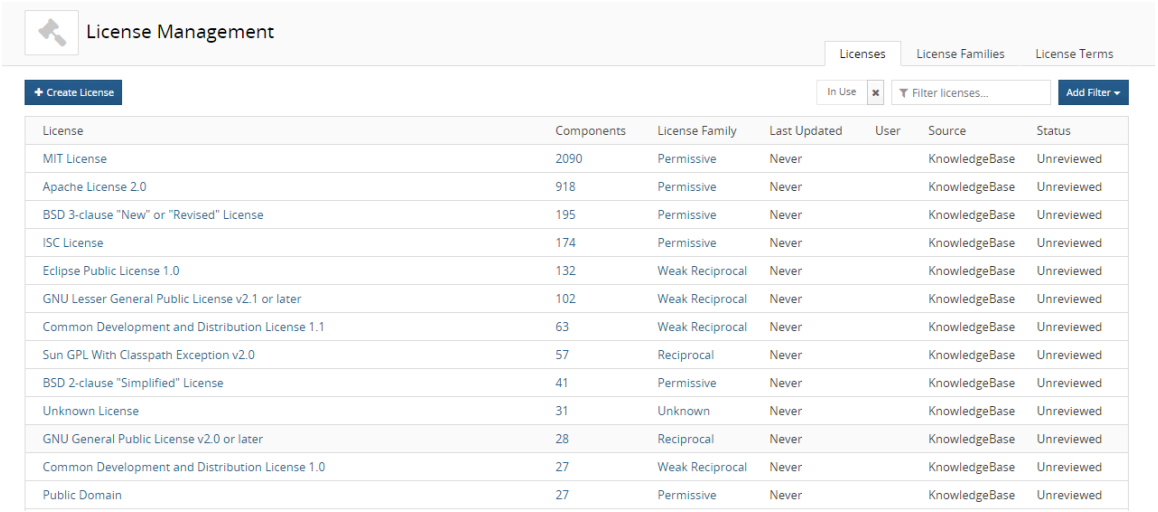
You can view the component and subproject versions where a specific license is used.

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:

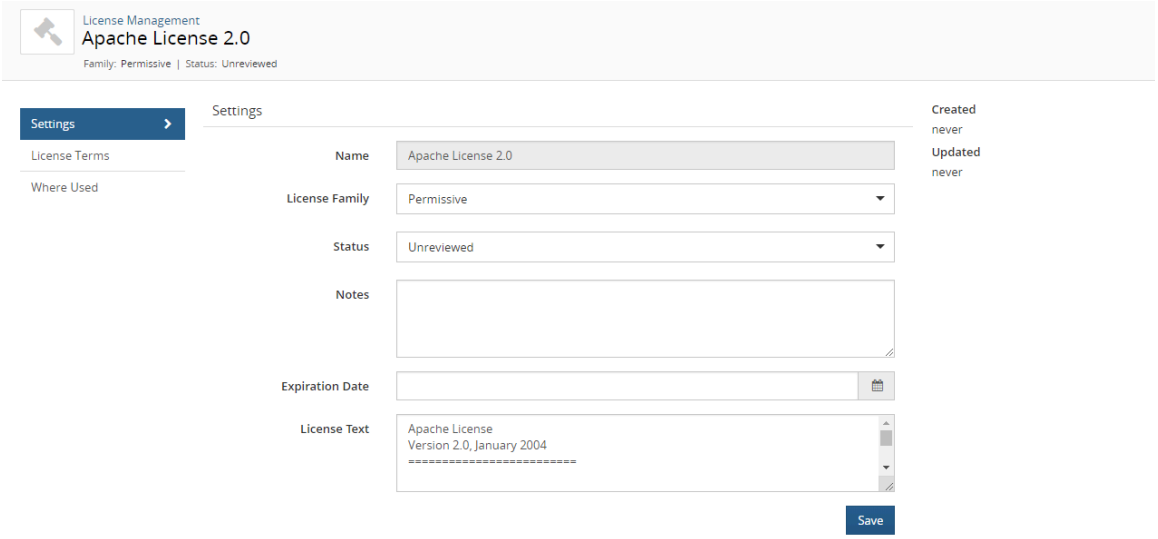
1. Log in to Black Duck with the License Manager role.
2. Click Manage > **Licenses**.

   The License Management page appears.

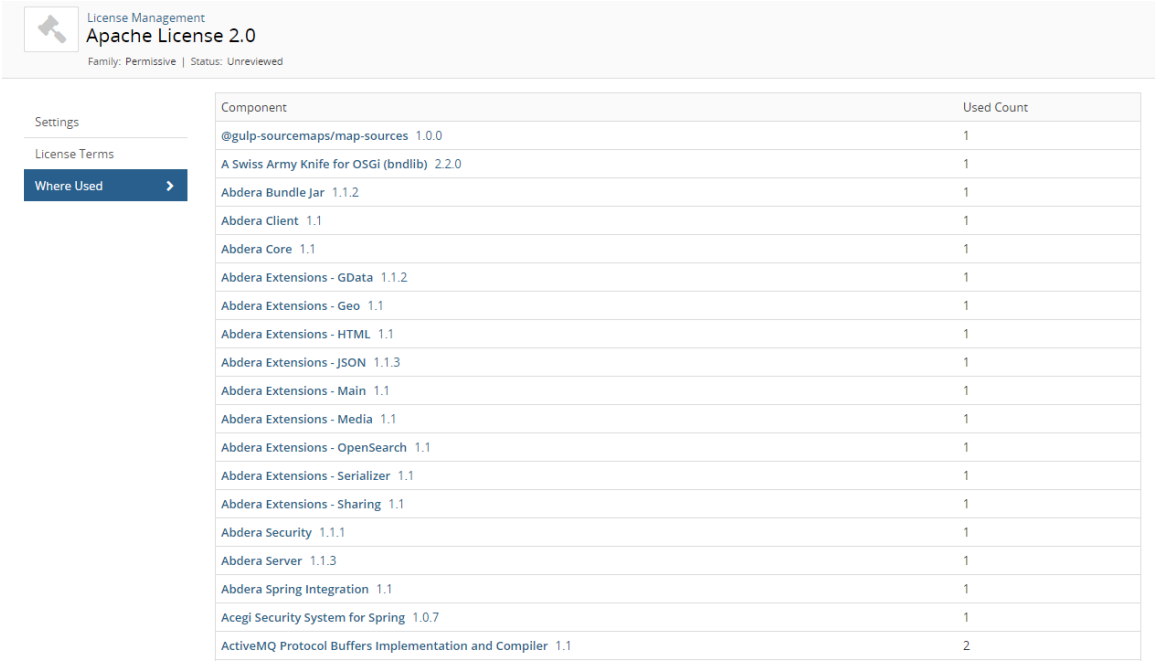


License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

3. Select the license name to display the *License Name* **Settings** tab.



4. Select the **Where Used** tab.



- 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 Black Duck KB.
- Select the component version to display the **Overview** 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 **Overview** 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 and subprojects in the project version's BOM.

There are 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 Black Duck, no declared license was found for the component; and "No License" which indicates that Black Duck 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.

Subproject license risk

If your project contains subprojects, the license risk is determined the subproject's license risk. A subproject's license is determined when it is added to the project.

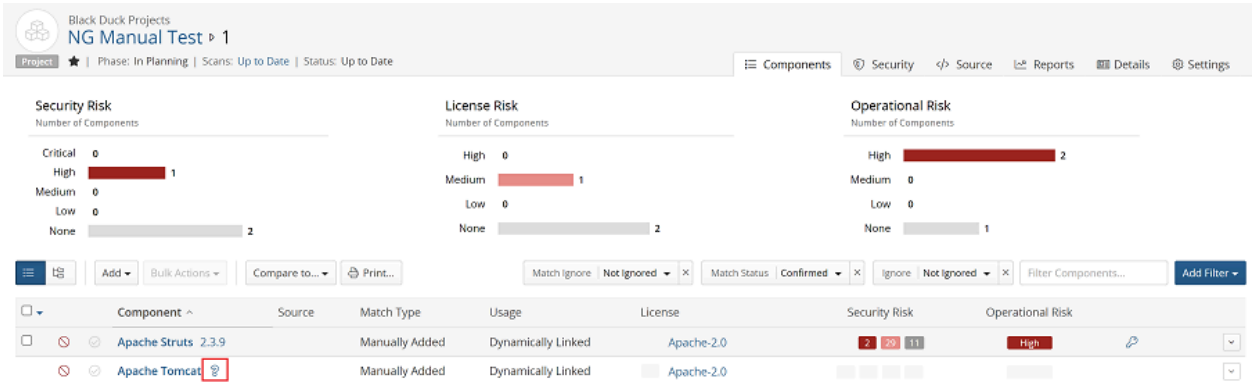
Notice: Black Duck 2023.10.0 introduces Enhanced license risk aggregation as a Limited Customer Availability Feature which improves the way subproject risk is determined. When enabled, the License Risk displayed for a subproject in your project's BOM will be determined by the subproject's license risk and the highest license risk of its components which reduces the possibility that license risk could be missed when using subproject hierarchies.

Important: When modifying a subproject's distribution type after it has been added to a project, the license risk of the parent project may not necessarily change to reflect the modification. The parent project's distribution takes precedence when calculating license risk.

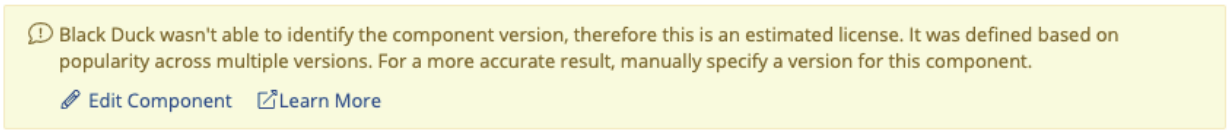
Estimated licenses

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:



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

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
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
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

License Family	External Projects	SaaS Projects	Internal Projects	Open Source Projects
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
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
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

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
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

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
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

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
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

Usage	External Projects	SaaS Projects	Internal Projects	Open Source Projects
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 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 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 or disabling deep license data

Enabling this checkbox will apply deep license data to your non-snippet components matches and allow visibility to embedded licenses which may exist in your components beyond declared licenses. Deep license data is enabled at the project level.

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.

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.
3. Click the **Project Details** tab.

4. Check the **Apply Deep License Data to Bill of Materials** checkbox or clear the checkbox to disable this feature.

   Apply Deep License Data to Bill of Materials

   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.

5. Click **Save**.

Enabling or disabling deep license data to snippet component matches

If enabled, component snippet matches are included in the deep license data calculation.

1. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the **Settings** tab.
3. Click the **Project Details** tab.
4. Check the **Apply Deep License Data to Snippet Component Matches** checkbox or clear the checkbox to disable this feature.

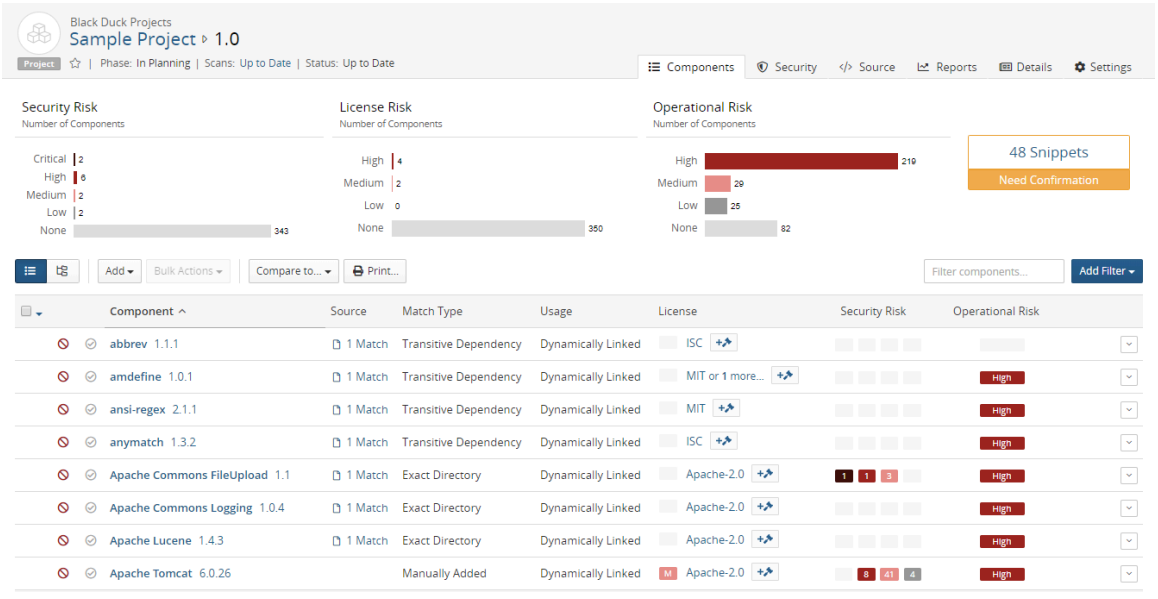
   Apply Deep License Data to Snippet Component Matches

   If enabled, component snippet matches are included in the deep license data calculation.

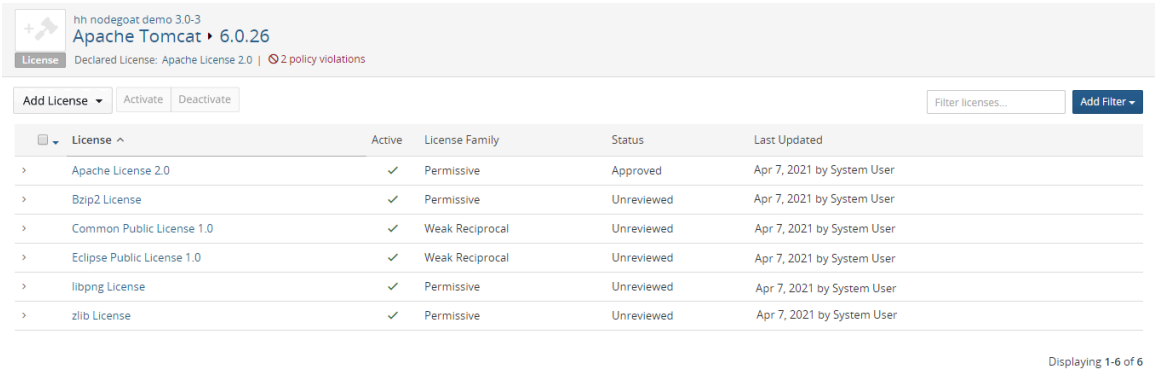
5. Click **Save**.

Reviewing deep license data

1. Open the project version BOM to view the components which have deep license data.



2. Components with  have deep license data. Click  to open the *Component Name Version* Deep License page.



This page displays the following information:

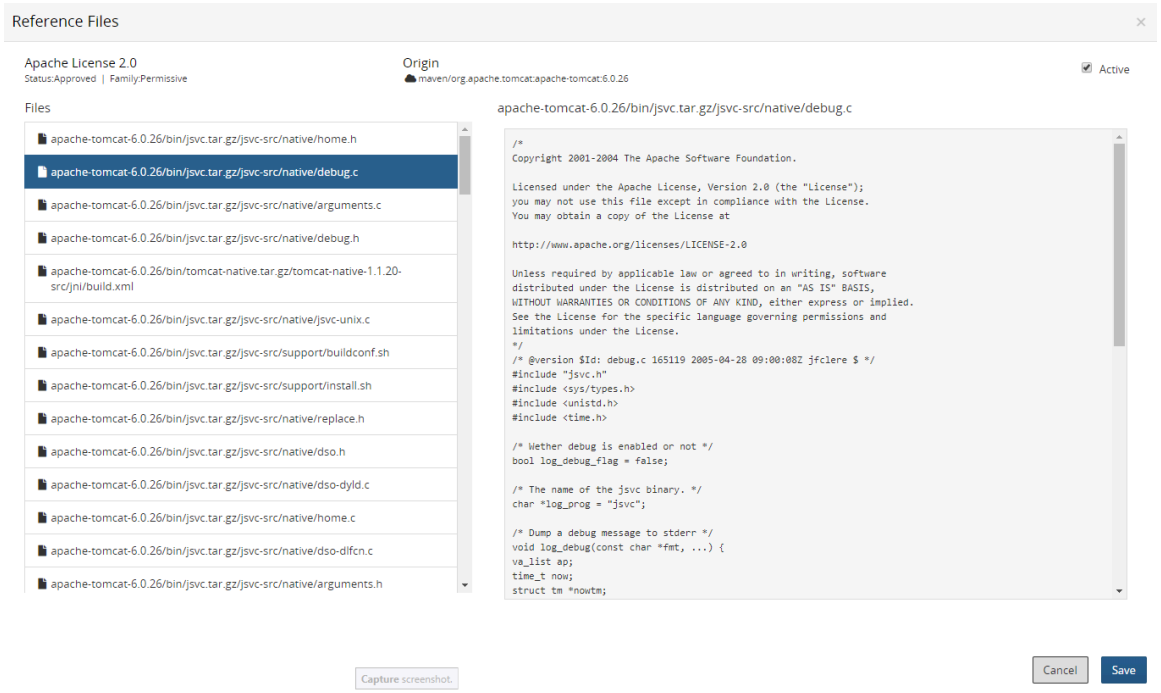
Column	Description
License	License name. Select the name to display the *License Name* page which displays the license text. Click > to view the origins for this license.
Active	Indicates whether this license is active. Active licenses are included in the calculation of license risk and policy violations.
License Family	The license family for this license.
Last Updated	Date and user who last updated the information on this page.
Status	The review status for the license. Possible values are: • Unreviewed • In Review • Reviewed • Approved • Limited Approval • Rejected • Deprecated

3. From this page:

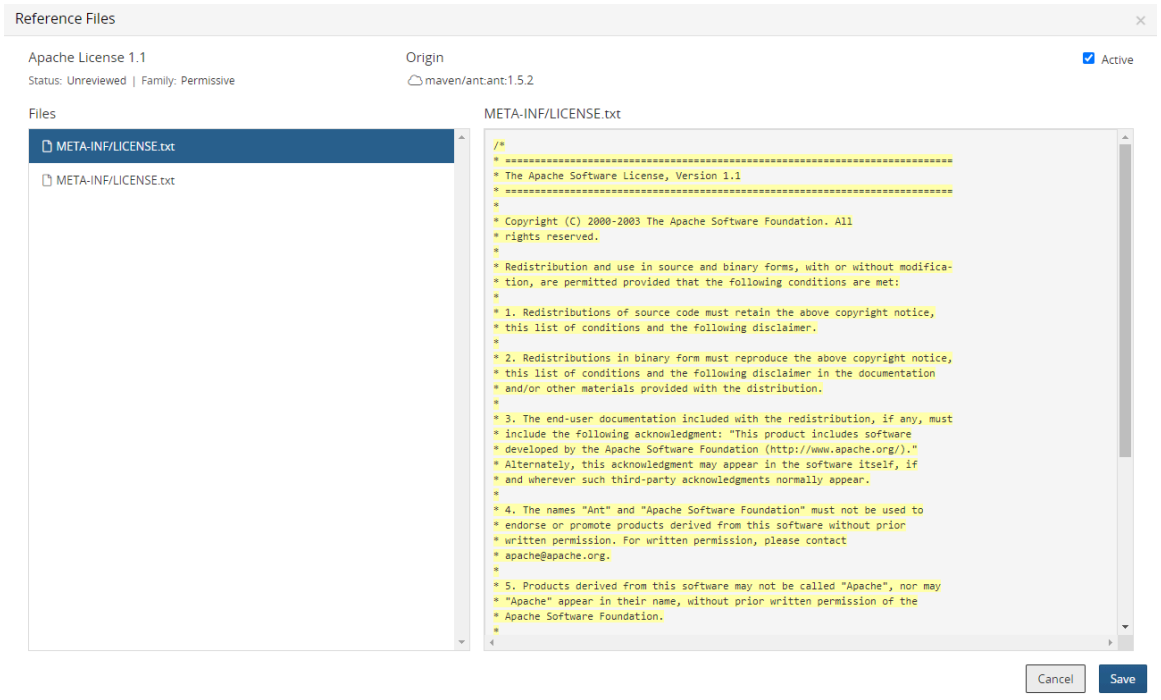
- View the evidence for the inclusion of this deep license.

  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."

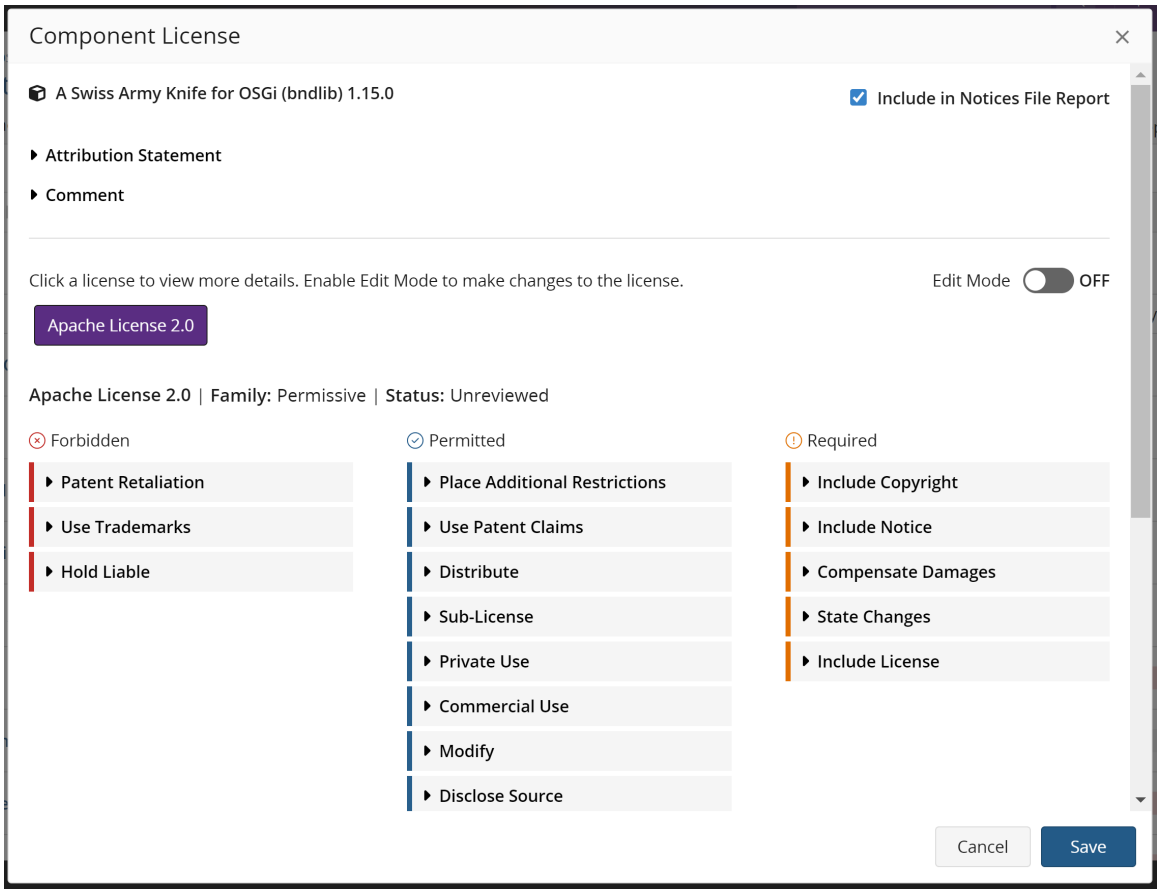
- Activate or deactivate the deep license. By default, all deep licenses are active.

  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**.

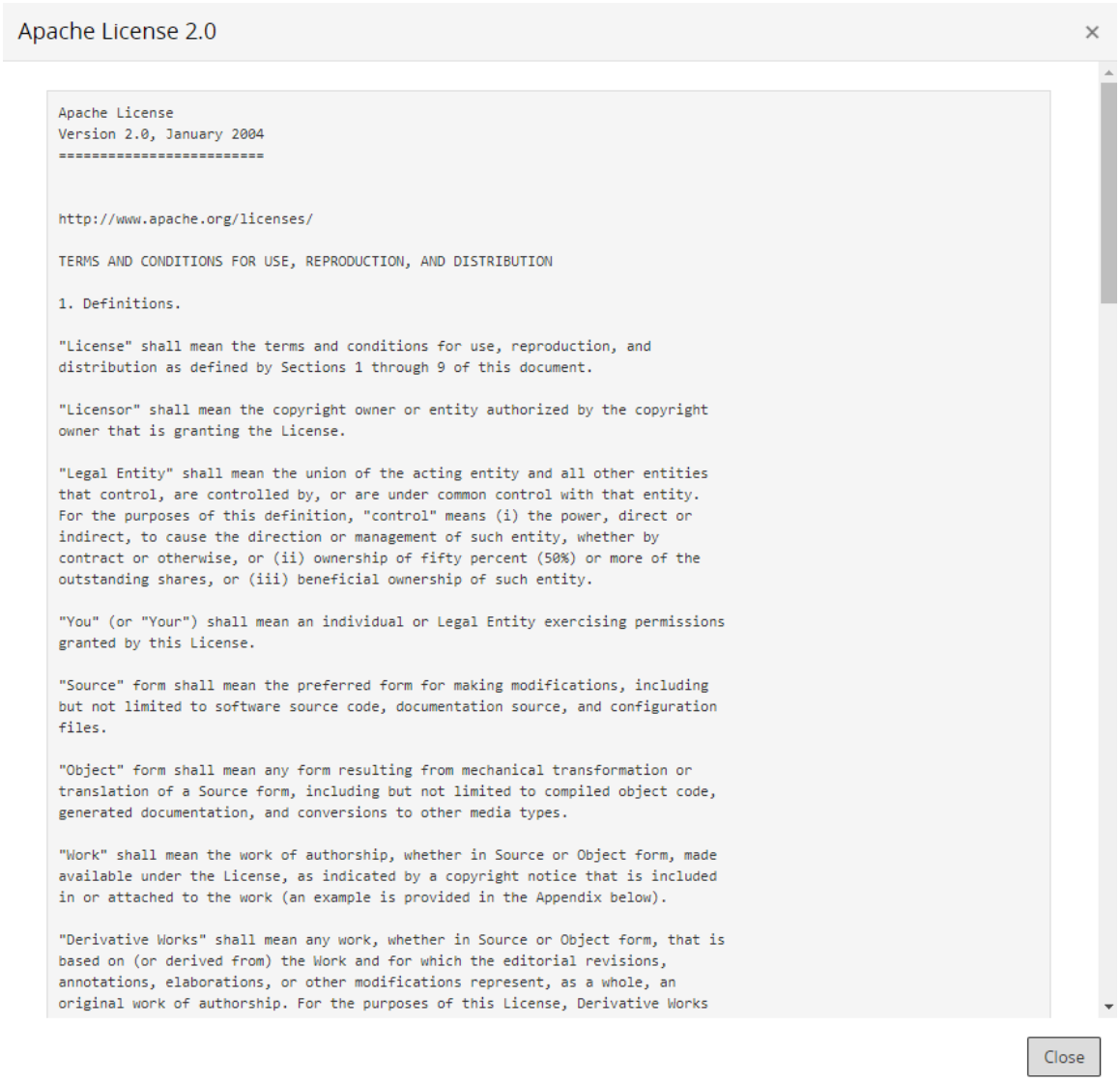
- Selecting or clearing the **Active** option located in the upper right-corner of the Reference Files dialog box.

- 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.

  - View deep license text by selecting the license name from the table.



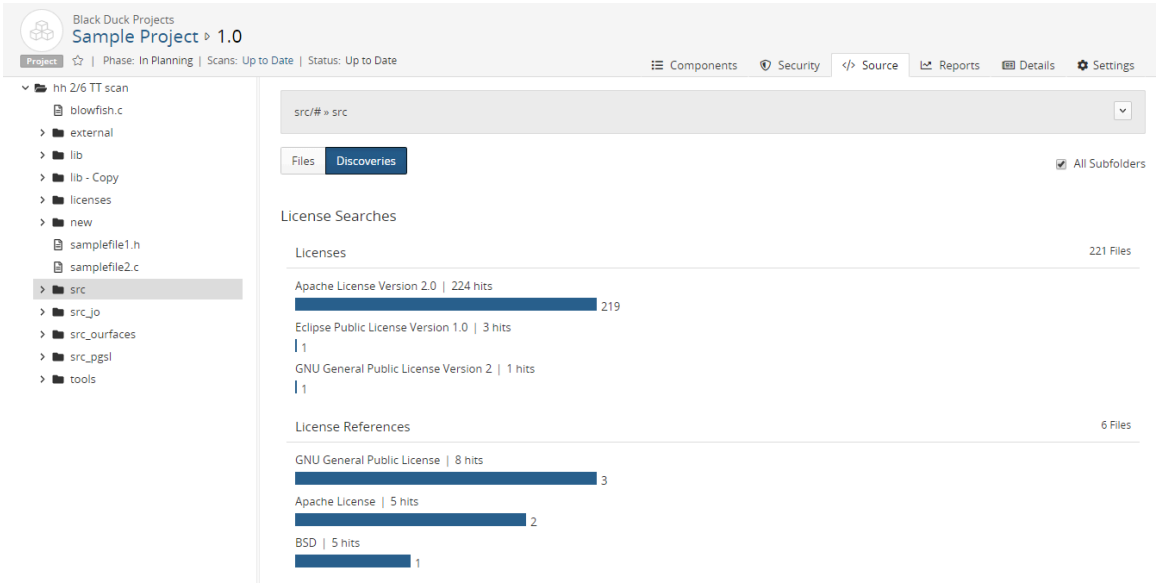
Detecting embedded licenses

Black Duck can detect instances of embedded open source licenses not declared by 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.



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.

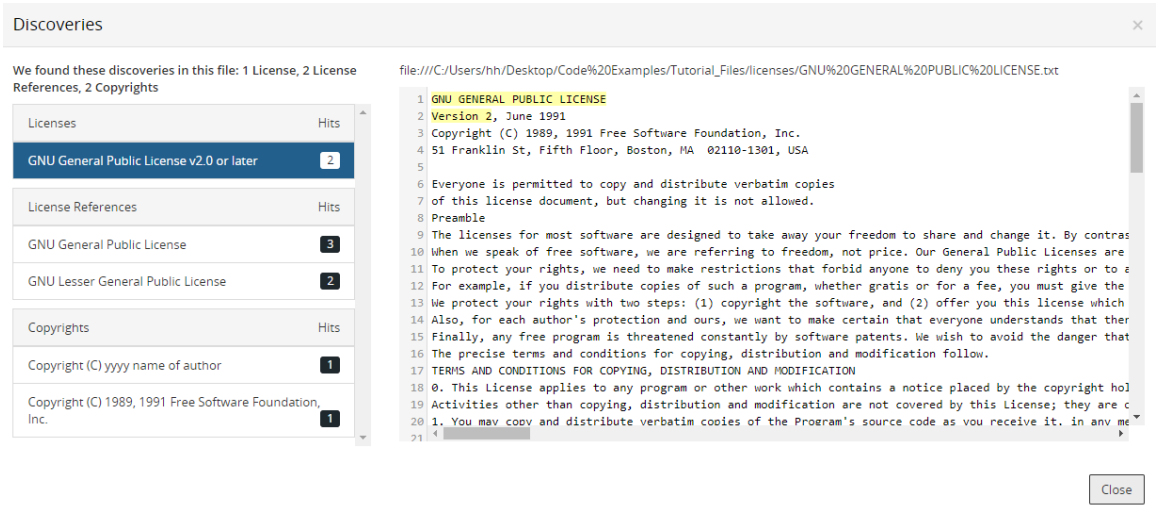
For each license statement found, Black Duck displays the number of:

- "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 was 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.



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:

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:

- bdsl
- 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
- Black Duck Detect Desktop Black Duck Detect
- Black Duck 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 Black Duck Detect Desktop or Black Duck Detect

Use the **--detect.blackduck.signature.scanner.license.search** property to enable deep license data detection. This property is available in Black Duck Detect version 6.2 and later.

Reviewing embedded licenses

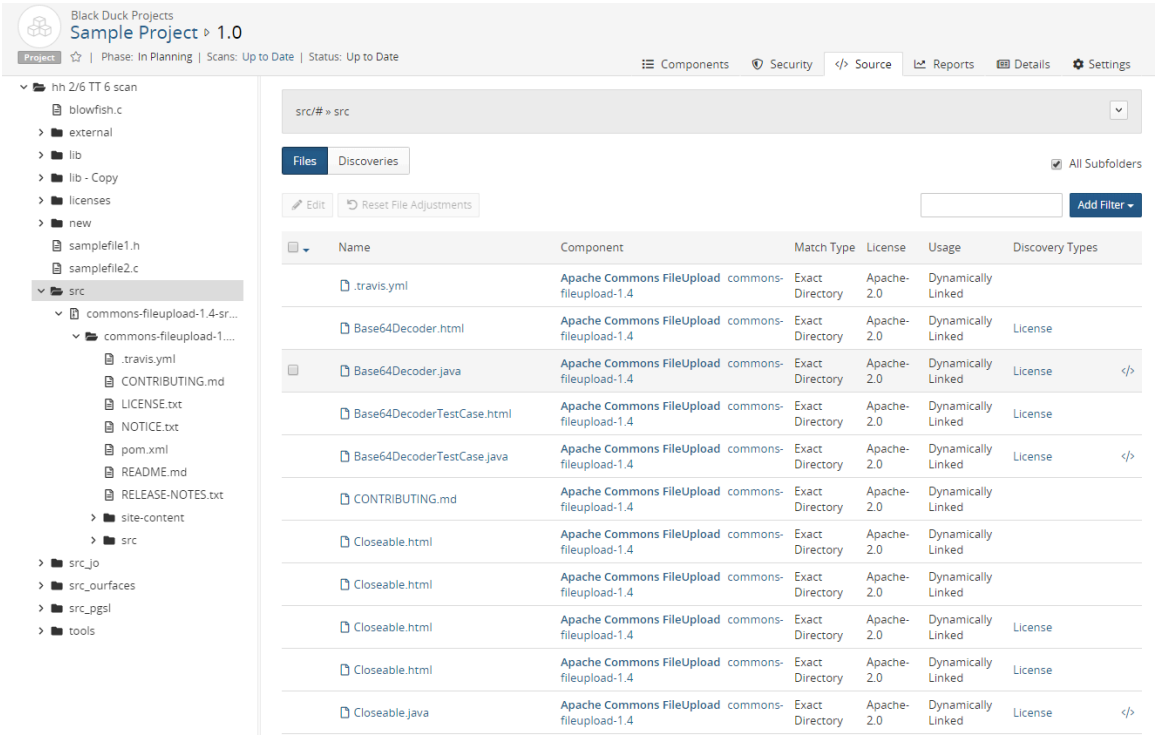
Black Duck displays the location of these licenses in your code tree.

To review embedded licenses :

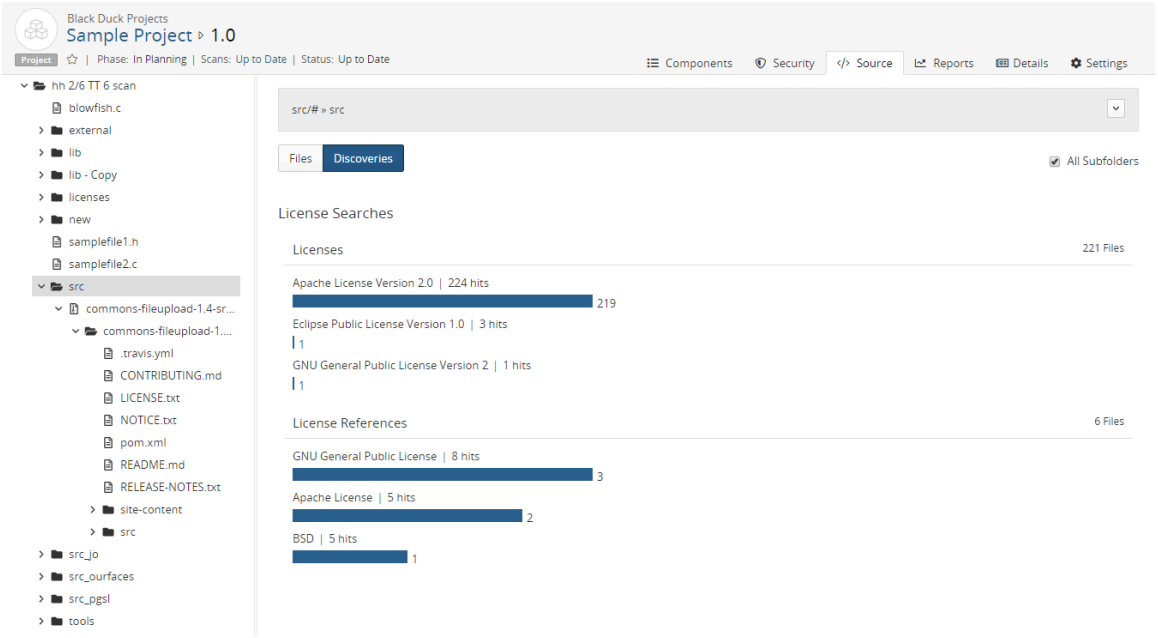
1. After enabling license search, select the **Source** tab from your project version BOM page.
2. Select a folder in the code tree that you want to determine if there are embedded licenses.

   Optionally, select **All Subfolders** to view information for all subfolders.

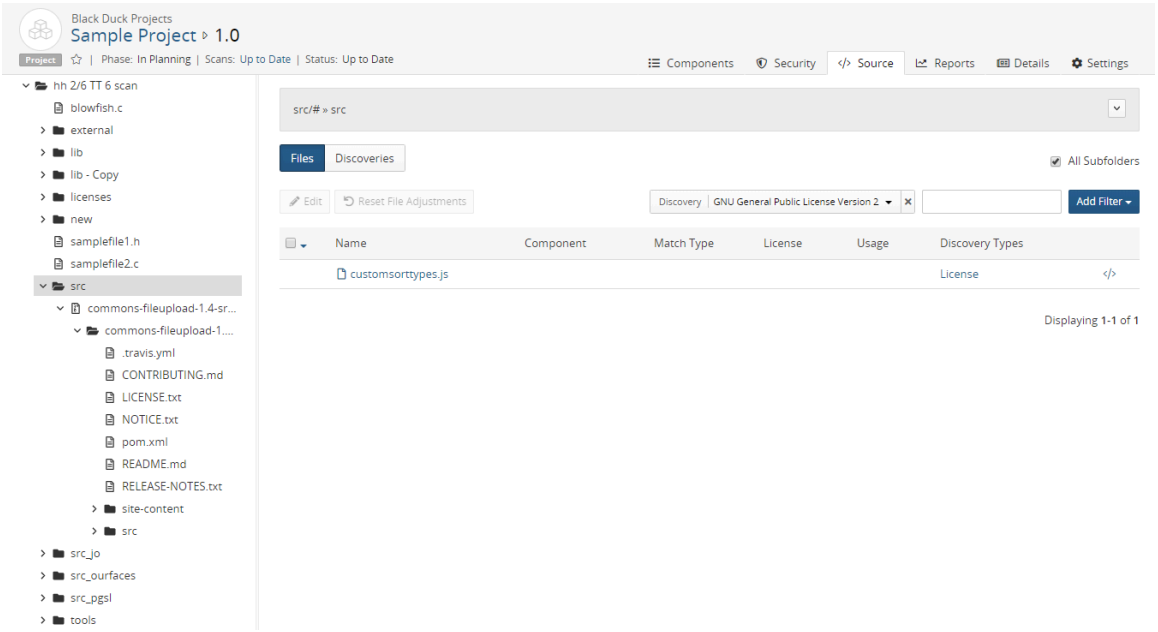
   The table displays information in the table for the selected location. By default the **Files** option is selected.



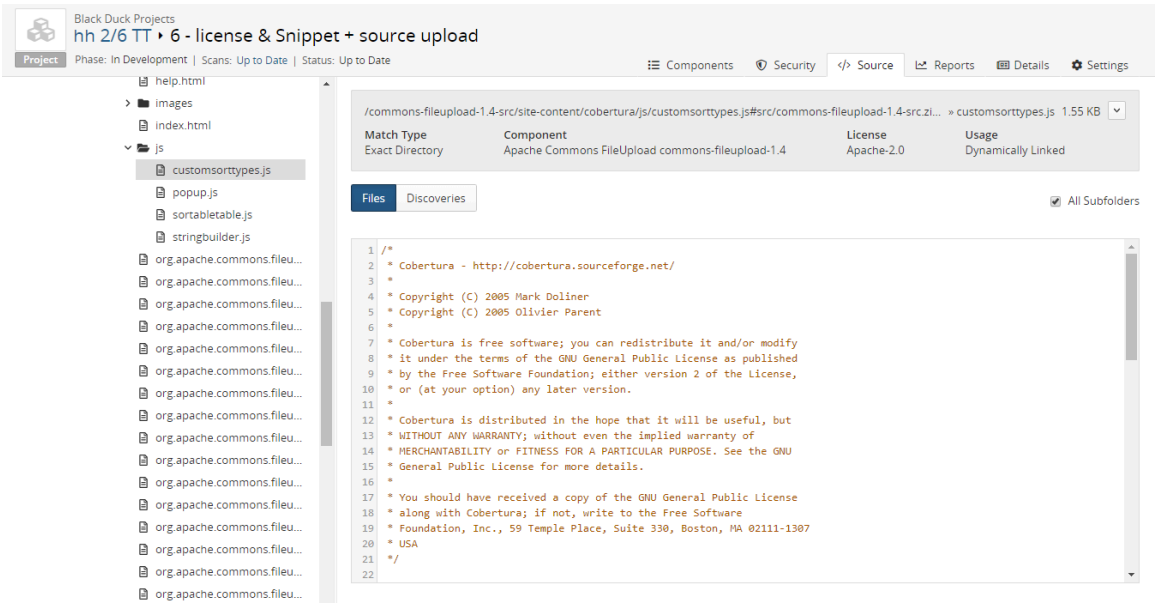
3. Select **Discoveries** to view the list of embedded licenses for this location.



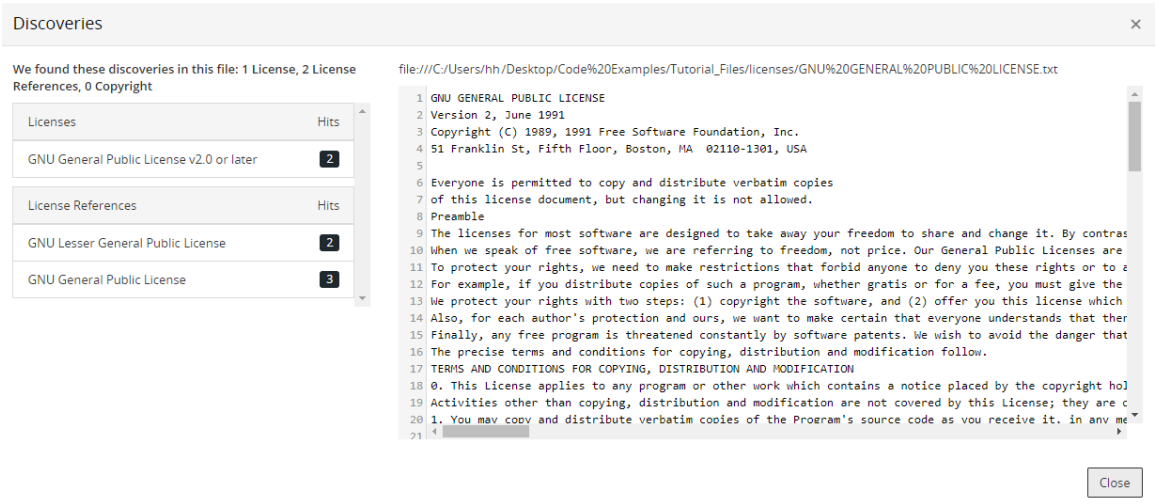
4. Select a license to view the **Source** tab filtered to display the files that contain the selected embedded license text.



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.



5. Select a type of discovery (**License** or **License Reference**) from the **Discovery Type** column to open the Discoveries dialog box.

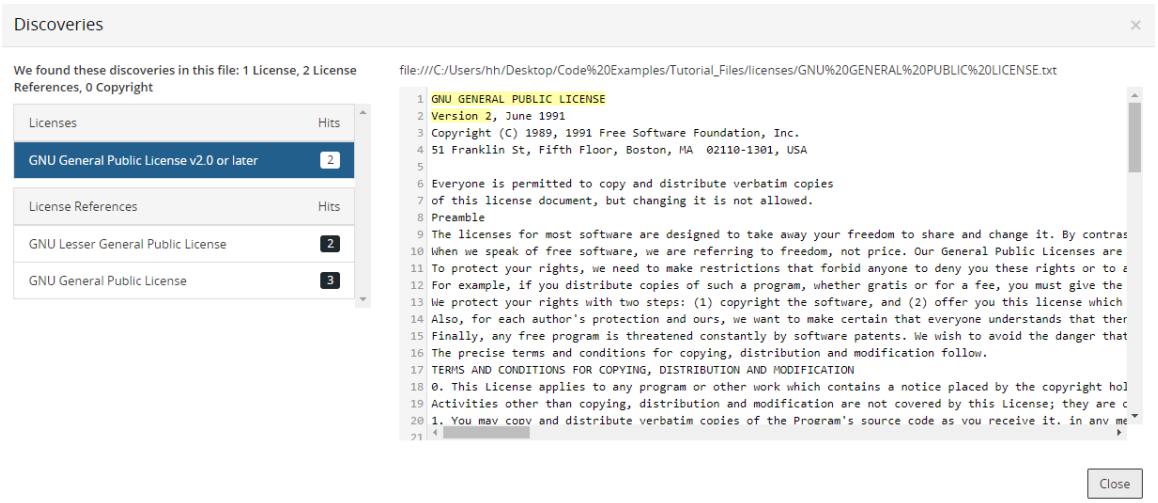


The Discoveries dialog box shows all licenses and license references found for the selected file.

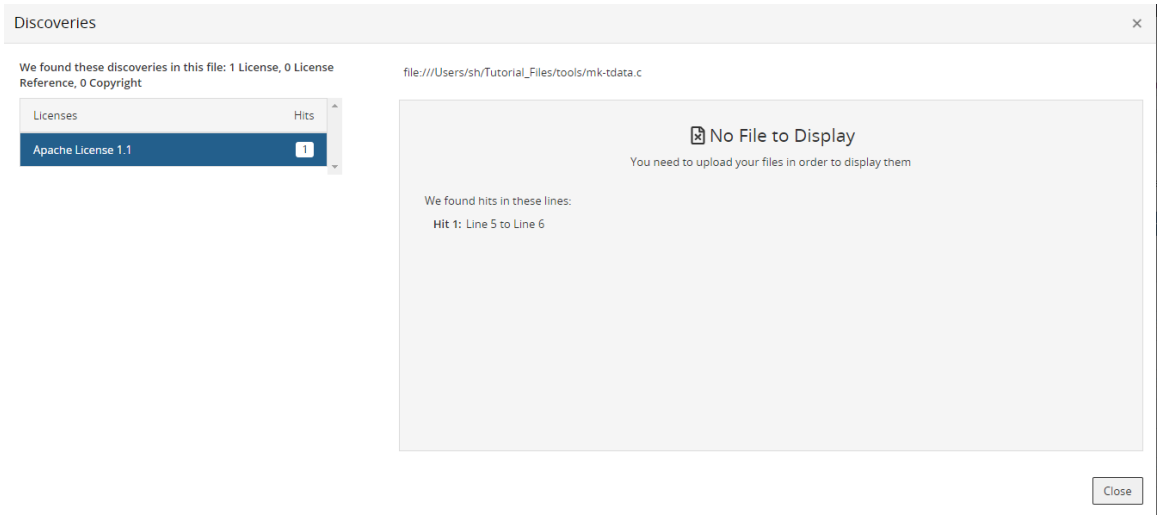
The information that appears here depends on whether you uploaded source files.

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 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:

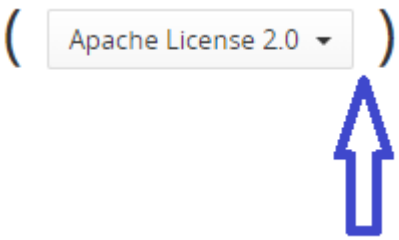
1. To modify a single license:
   a. Click ▾ located next to the license name and select the license from the list of suggestions.
   b. Do one of the following:
      - Click ✓ to confirm this selection.
      - Click 🗑 to delete this license and the operand.
2. To add a license to the existing license(s):
   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:



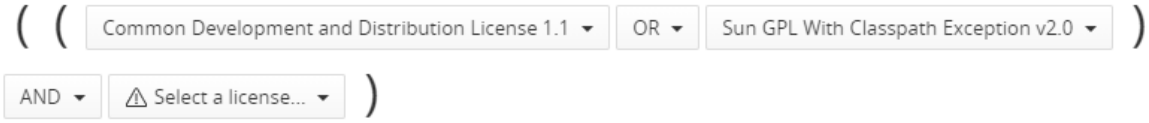
- To add a license at the original license level, select the license by placing the cursor within the parentheses of that license.



Click **Add License**. The licenses is added at the level of the original license:



For example, when added to an existing multi-license scenario, the following appears:



- 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.



Click **Add License**. The license is added at the same level as the existing licenses:



b. Optionally, click next to the operand to change it. Possible values are AND or OR.

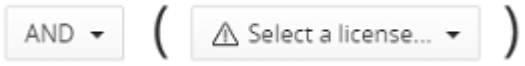
c. Click located next to the license name and select the license from the list of suggestions.

- Click to confirm this selection.
- Click to delete this license and the operand.

d. Repeat as necessary.

3. To add a multi-license scenario (for example, License A AND (License B OR License C):

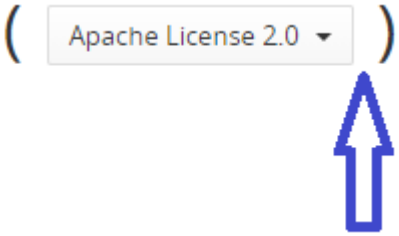
a. Click **Add Group**. Black Duck adds the following at the root level:



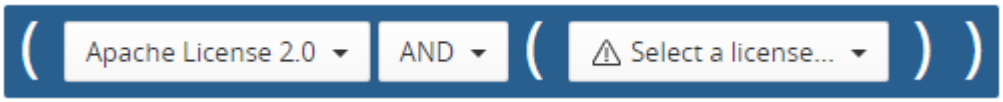
When added to a single license, the following appears:



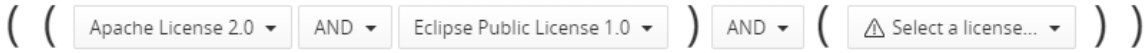
- To add a group at the original license level, select the license by placing the cursor within the parentheses.



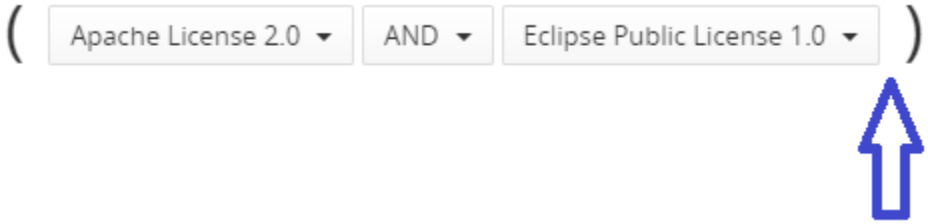
Click **Add Group**. The following appears:



When added to an existing multi-license scenario, the following appears:



- 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:



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**.
   The assigned license is updated. If the new license carries a different type of license risk than the previous one, the license risk calculations for the component and for the project version are updated in project version BOMs. A appears in the table row in the BOM to indicate that a manual adjustment was made to this component.

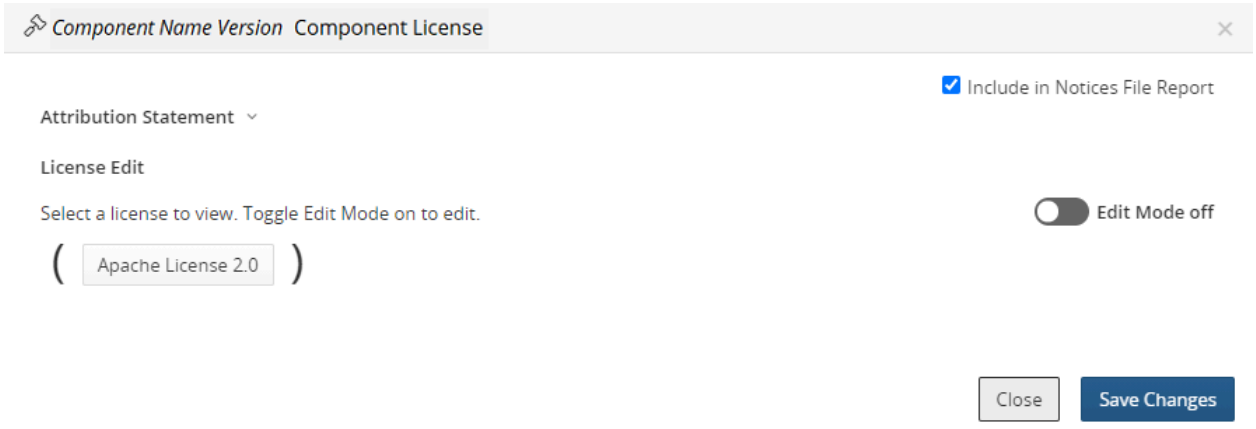
   When viewed in the BOM, the license obligations for the revised license(s) will appear when you re-open the *Component Name Version* Component License dialog box.

Reverting BOM-level license edits

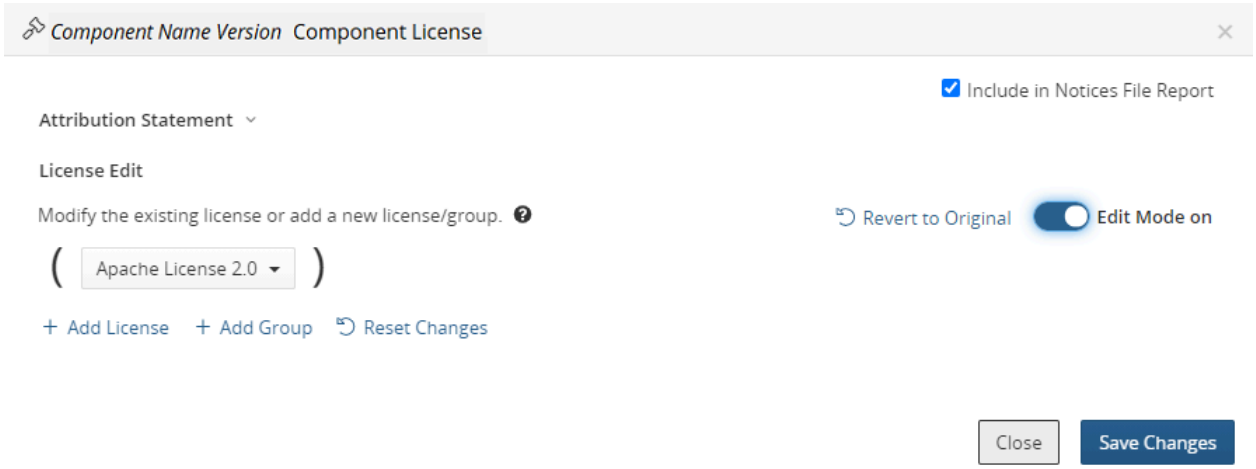
If you selected a different license for a component when editing licenses in the BOM, you can revert the license to its original license as defined in Black Duck KnowledgeBase.

To revert to an original license:

1. Log in to Black Duck SCA.
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**.
   The license is reverted. If the license carries a different type of license risk than the previous one, the license risk calculations for the component and for the project version are updated in the BOM.

About custom licenses

If you discover that a license that you use for a component in your BOM is not available from 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 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.

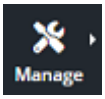
Custom licenses:

- 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 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 Black Duck KnowledgeBase licenses used in all the projects in your organization.

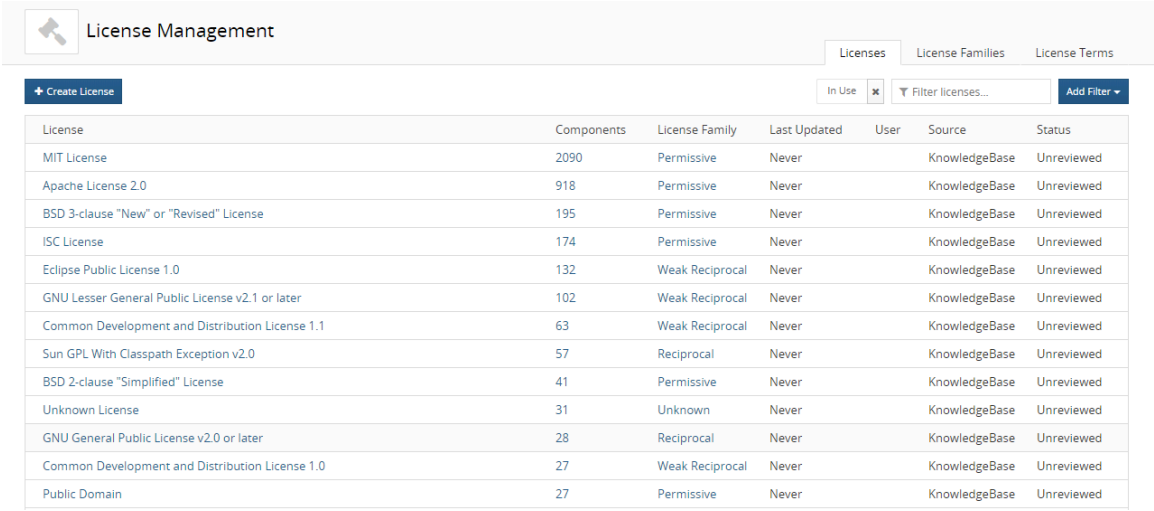
Creating custom licenses

Only users with the License Manager role can create custom licenses.

1. Log in to Black Duck with the License Manager role.
2. Click 

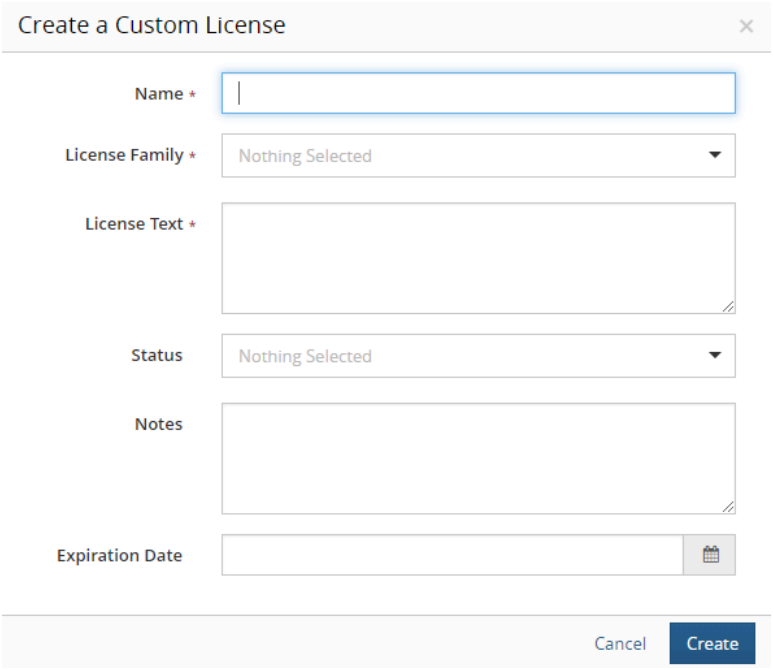
> **Licenses**.

   The License Management page appears.



License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

3. Click **Create License** to open the Create a Custom License dialog box.



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, 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 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:

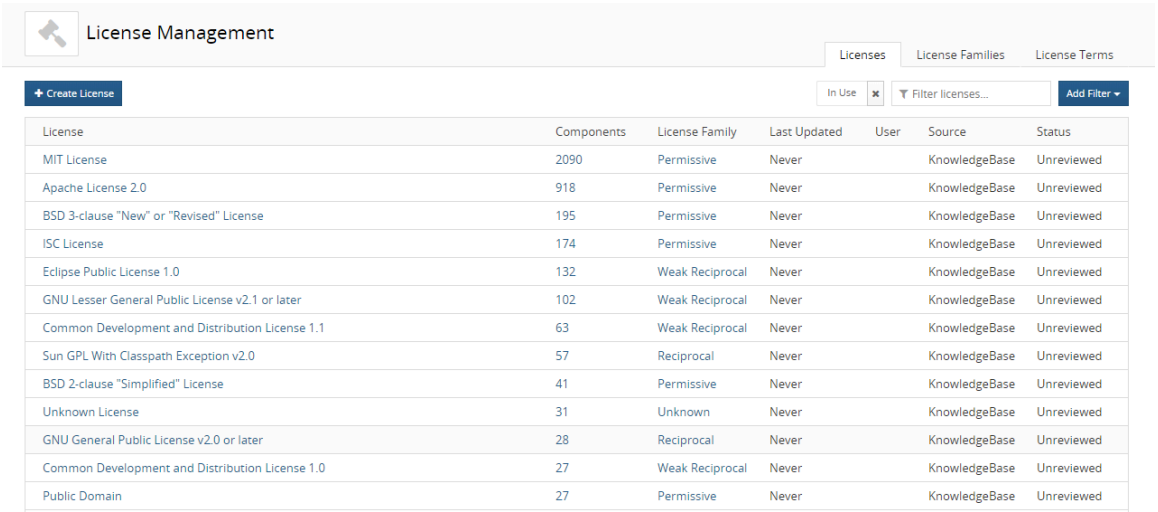
- Edits to the license family and license name are always propagated to the custom licenses used in BOMs.
- Edits to the license text *may or may not* be propagated to the custom licenses used in BOMs:
  - If the BOM Manager or Project Manager *edited the license text*, the edits made by the License Manager *are not* propagated to the version of the custom license used in the BOM.
  - If the BOM Manager or 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 

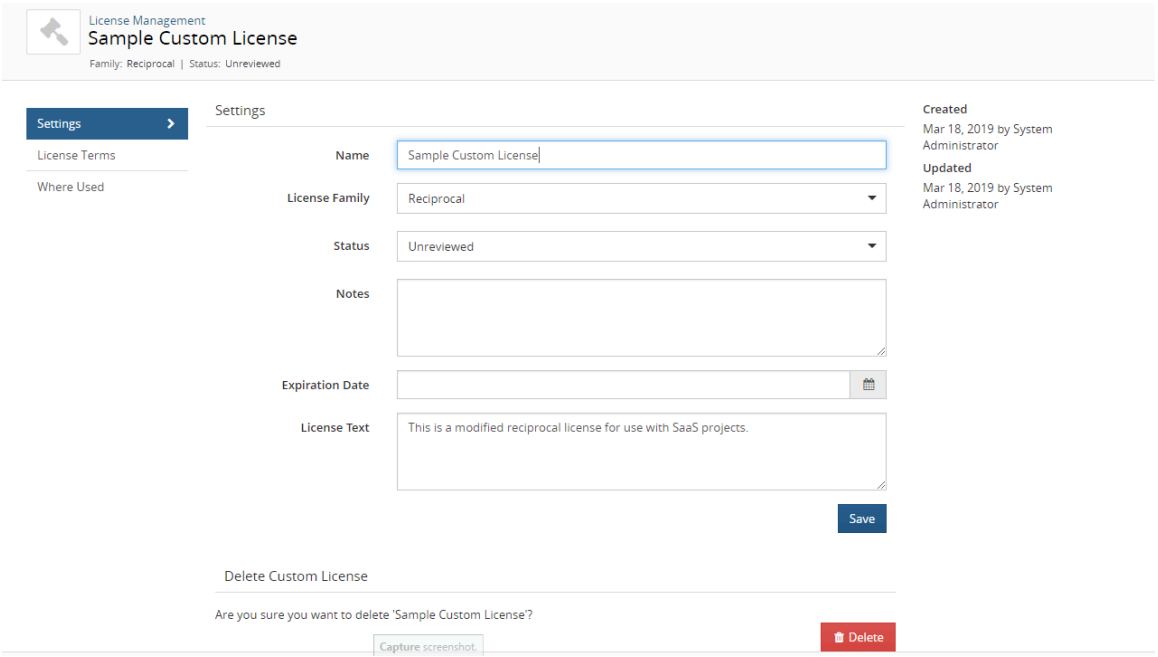
> **Licenses**.

   The License Management page appears.

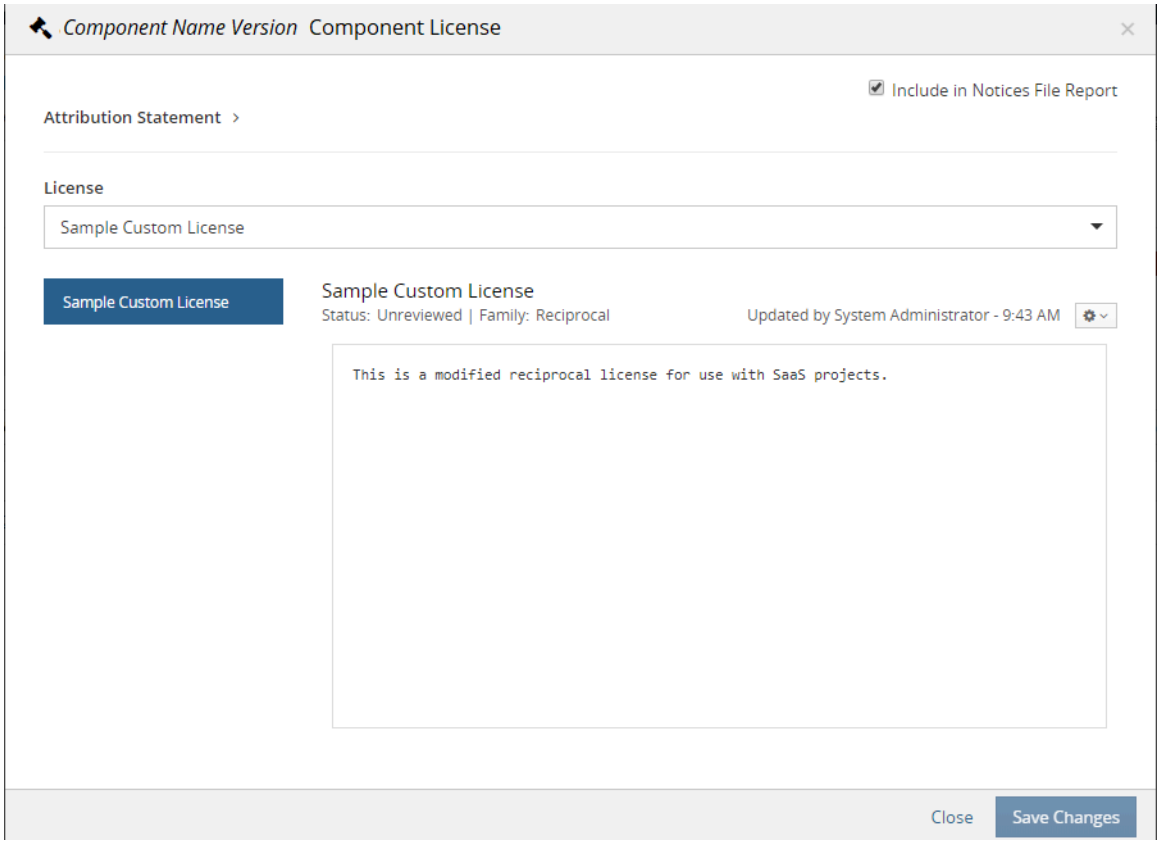


License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

3. Select the license name to display the *License Name* **Settings** tab.



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

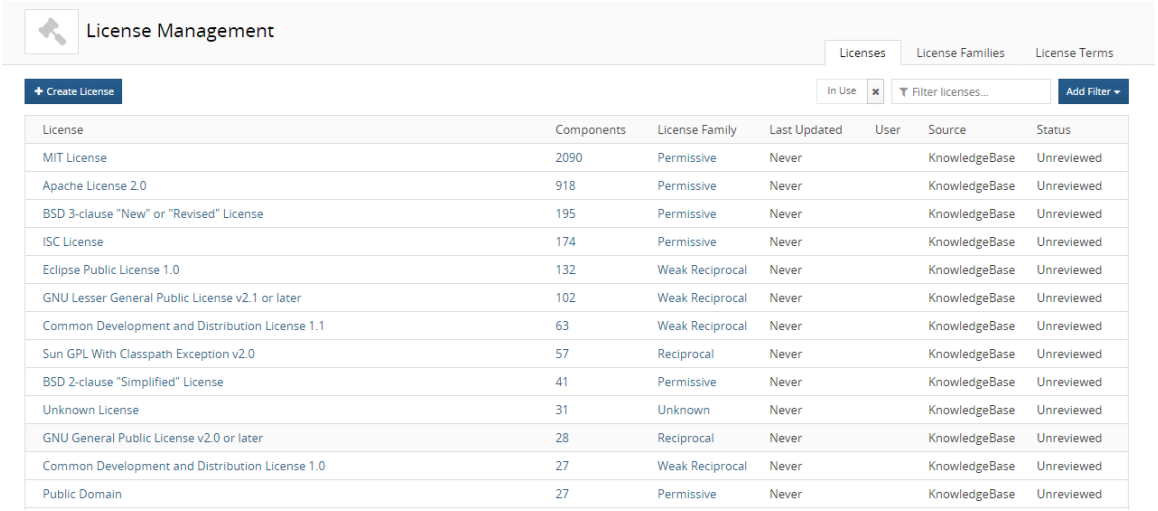
You cannot delete a license that is being used in a BOM.

You also cannot delete licenses provided by Black Duck KnowledgeBase.

1. Log in to Black Duck with the License Manager role.
2. Click 

> **Licenses**.

   The License Management page appears.



License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

3. 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.

4. 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 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.

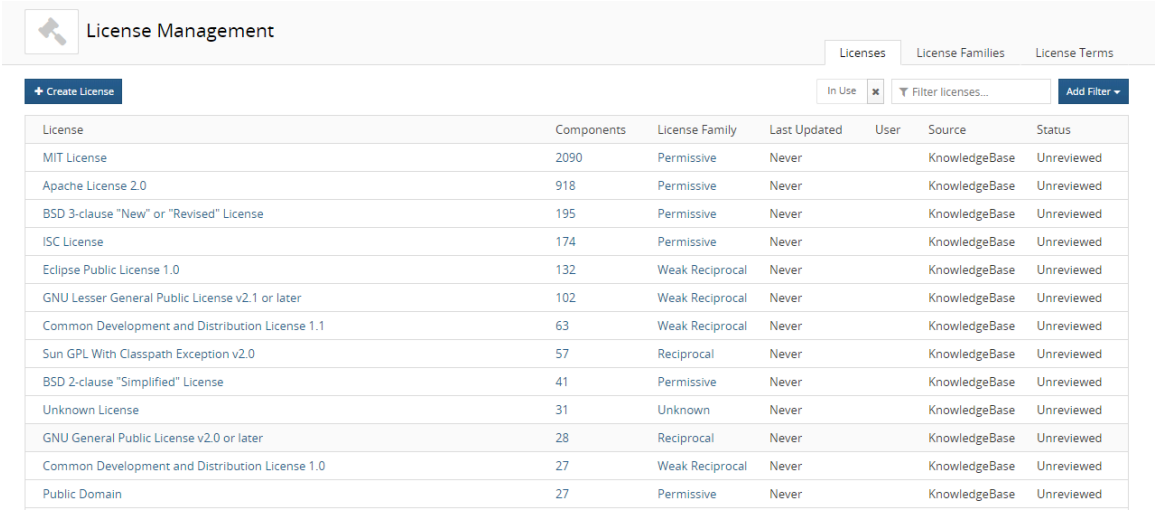
The **License Terms** tab shows all license terms for custom and KnowledgeBase licenses.

To view the License Terms tab:

1. Log in to Black Duck with the License Manager role.
2. Click 

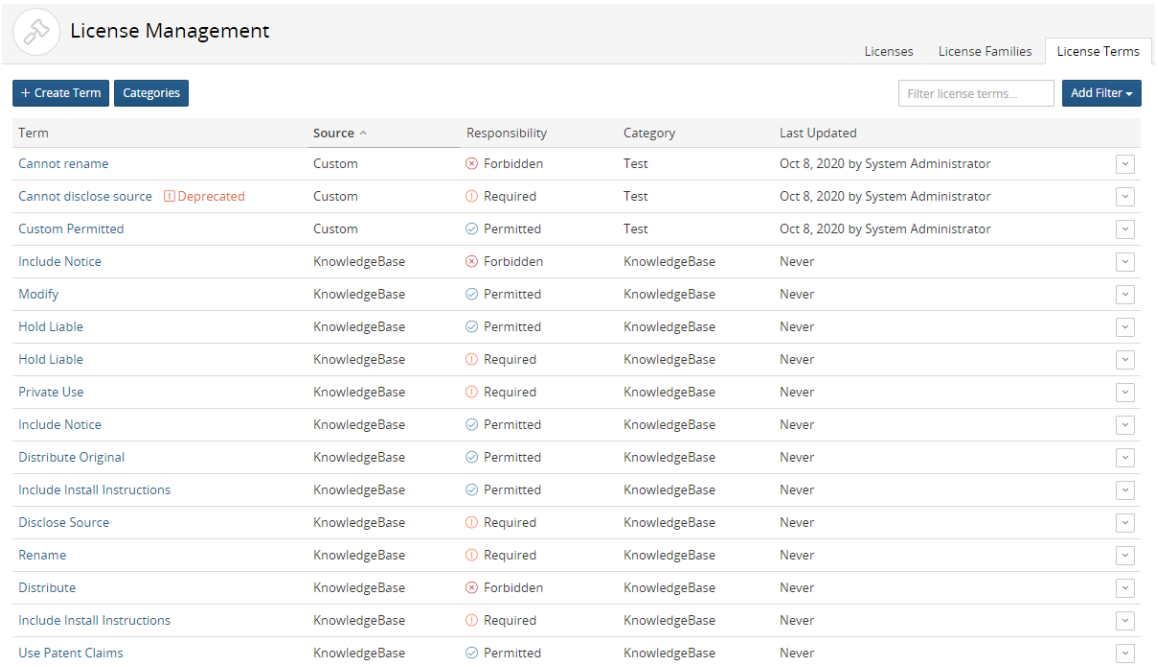
> **Licenses**.

   The License Management page appears.



License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

3. Select the **License Terms** tab.



Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source Deprecated	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	Term name. Hover over the name to view the description for this term. For custom license terms, select the name to open the Edit a License Term dialog box.
Source	The source for this term. Possible values are: • KnowledgeBase. This is a standard term from Black Duck KnowledgeBase. • Custom. A license term you created.
Responsibility	Responsibility for this license term. Possible values are:

Column	Description
	• Permitted • Forbidden • Required
Category	Category for this license term. License terms from Black Duck KnowledgeBase have **KnowledgeBase** as the category. Custom license terms list the category defined when adding the term.
Last Updated	Date that the license term was last updated and the username of the user who updated this term. The column lists **Never** for KnowledgeBase license terms.

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.

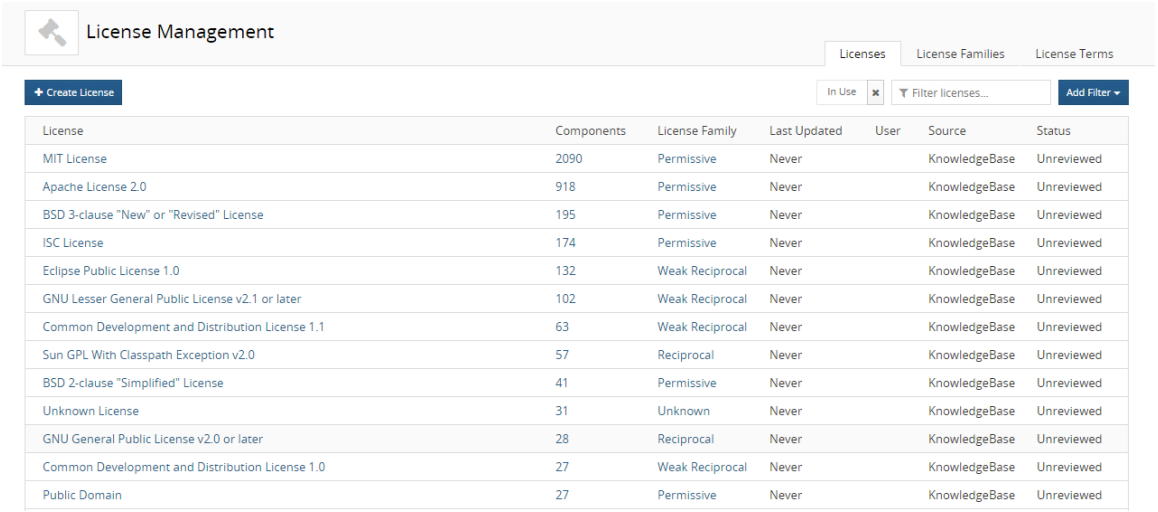
Note: License obligation will not appear in the UI, if the information is unavailable from OpenHub,

To view the license terms from the License Management page:

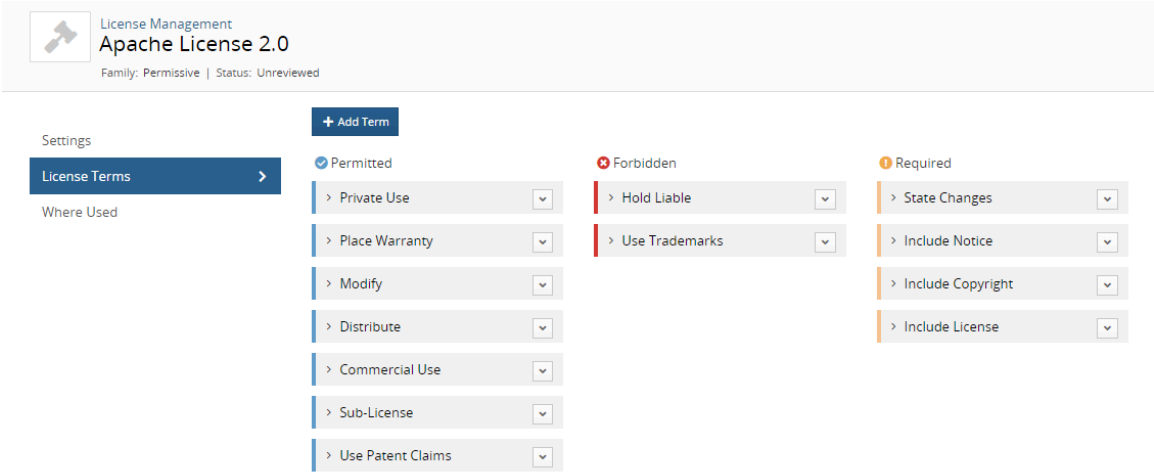
1. Log in to Black Duck with the License Manager role.
2. Click 

> **Licenses**.

   The License Management page appears.



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.

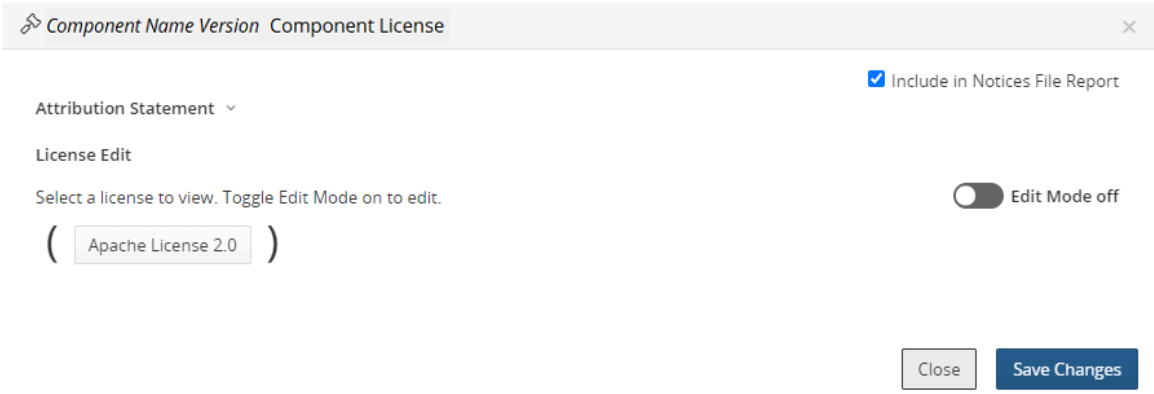


If available, select > to view additional information.

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:

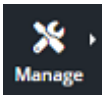
- It may take time for license term fulfillment requirements to appear on the **Legal** tab.
- Policy managers can create a policy rule that will trigger a violation when there are unfulfilled license terms.

  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

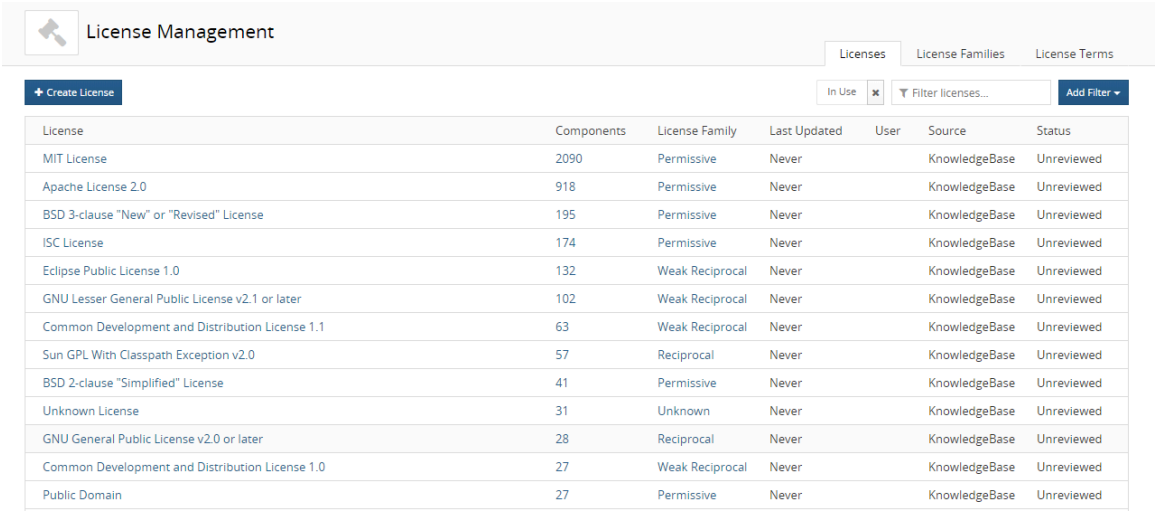
License Managers can indicate a license term is required when using the *License Name* **License Terms** tab which shows all terms for a specific license.

To define the fulfillment requirement when viewing a license:

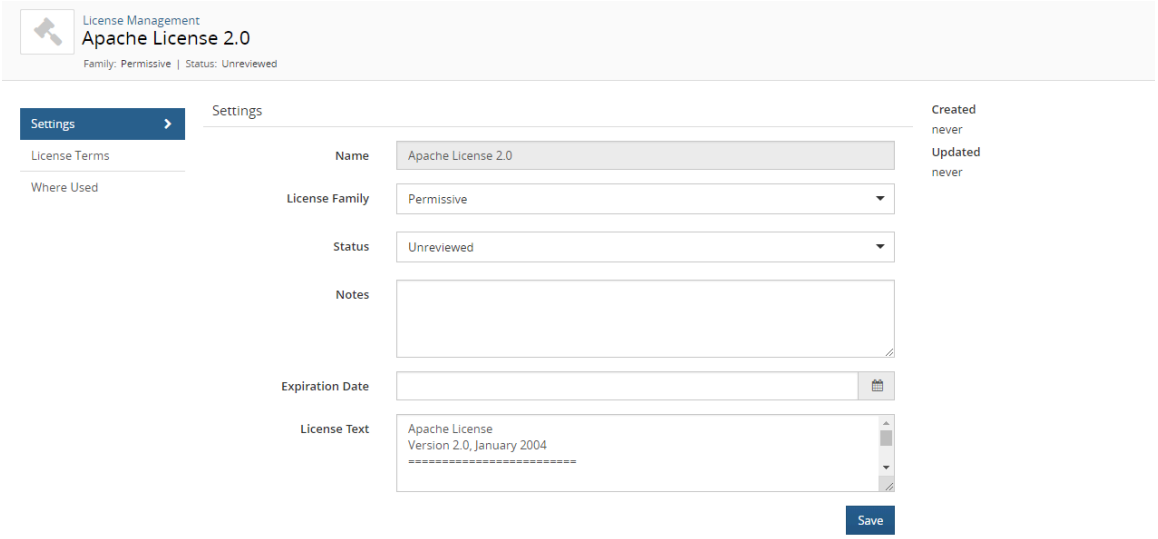
1. Log in to Black Duck with the License Manager role.
2. Click 

> **Licenses**.

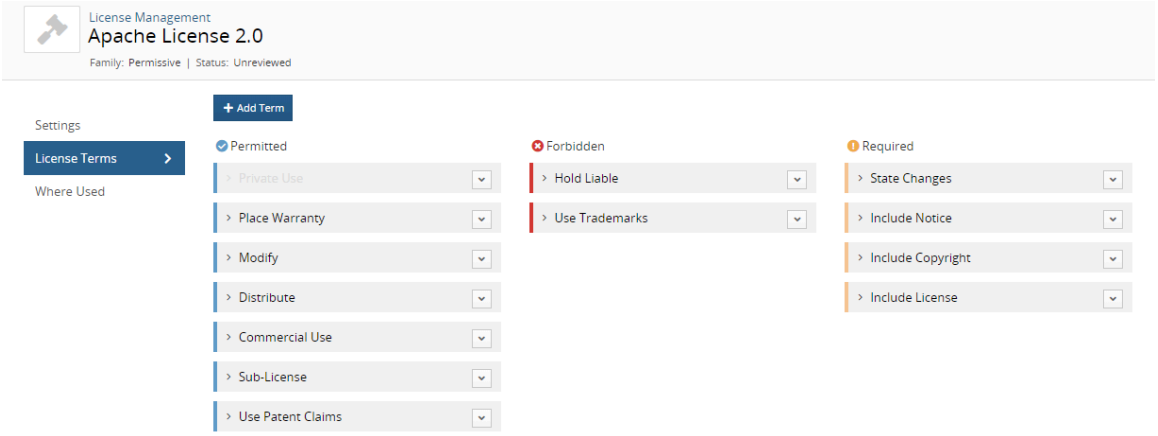
   The License Management page appears.



3. In the **Licenses** tab, select the license name to display the *License Name* **Settings** tab.

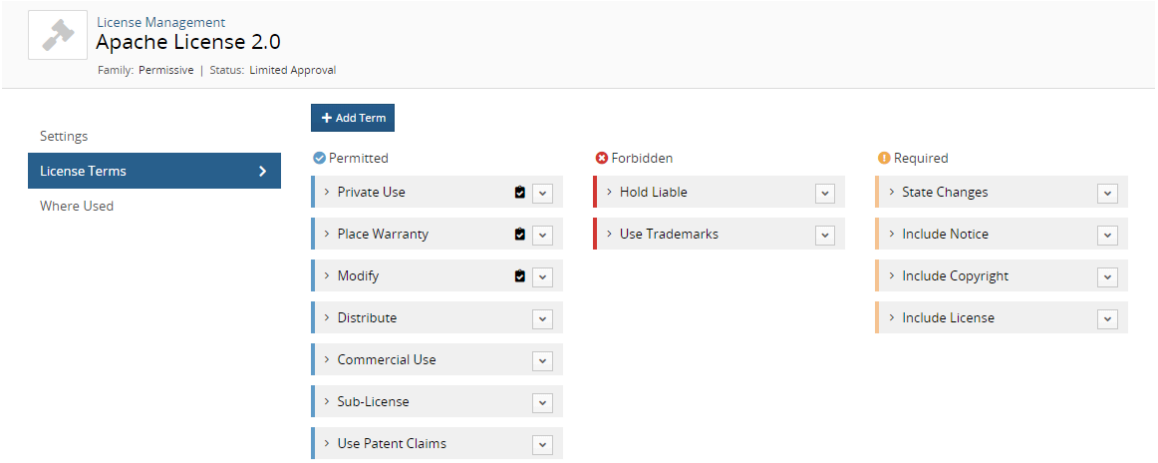


4. Select the **License Terms** tab to view the terms associated with this tab.



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.



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.

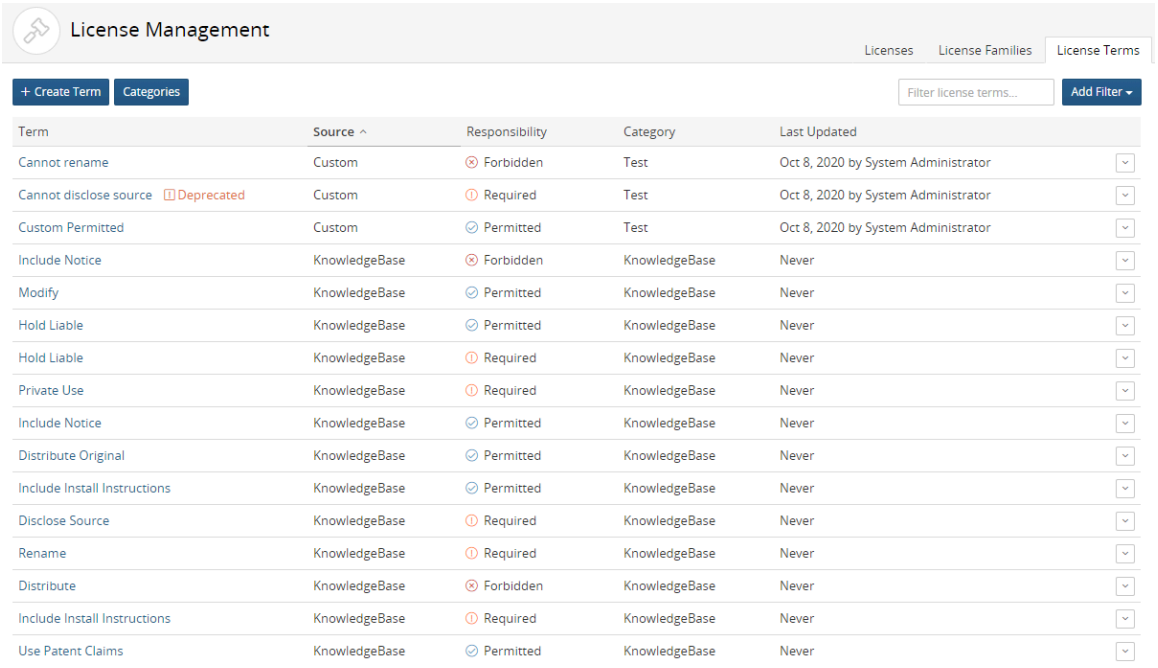
To create a license term:

1. Log in to Black Duck with the License Manager role.
2. Click 

> **Licenses**.

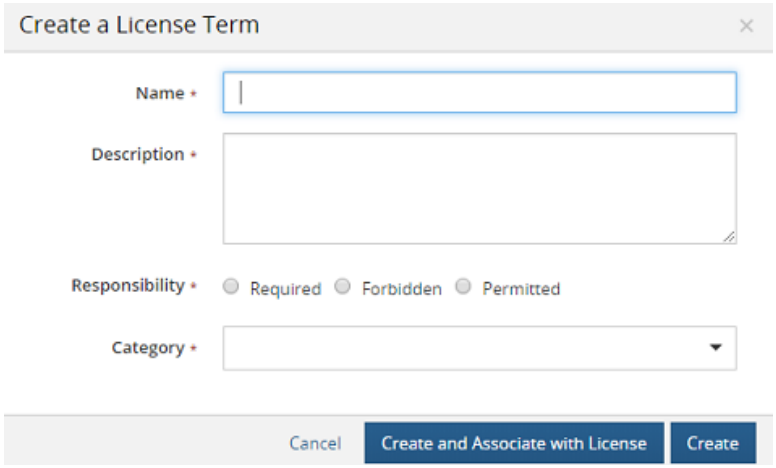
   The License Management page appears.

   Select the **License Terms** tab to display all license terms.



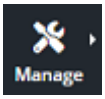
Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source Deprecated	Custom	Required	Test	Oct 8, 2020 by System Administrator
Custom Permitted	Custom	Permitted	Test	Oct 8, 2020 by System Administrator
Include Notice	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.



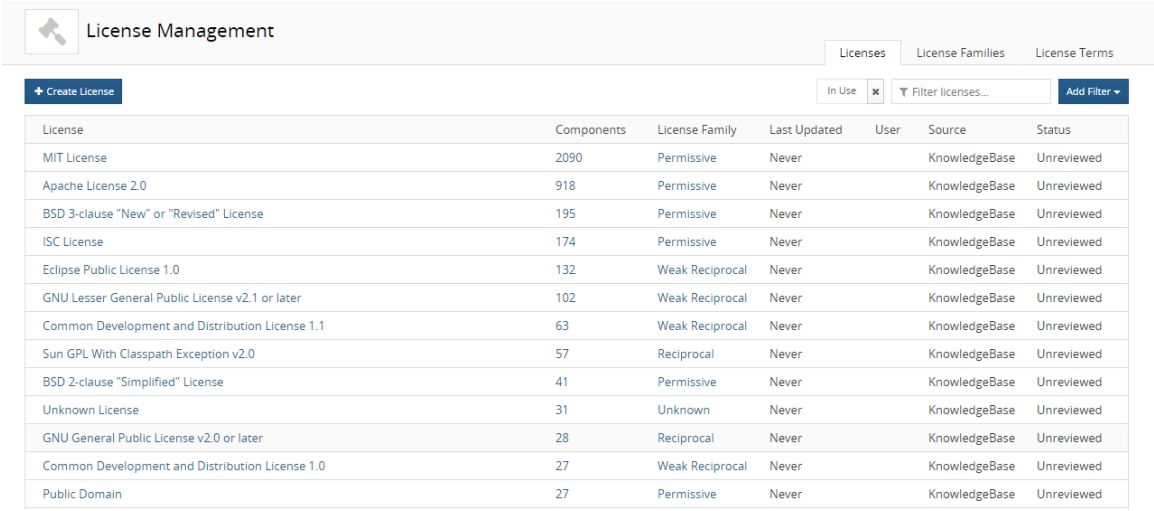
4. Complete the information in the dialog box:
   - **Name**.
   - **Description**.
   - **Responsibility**. Select whether this responsibility is required, forbidden, or permitted.
   - **Category**. Select a category for this license term. Optionally, create a new category by entering text in the field 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:

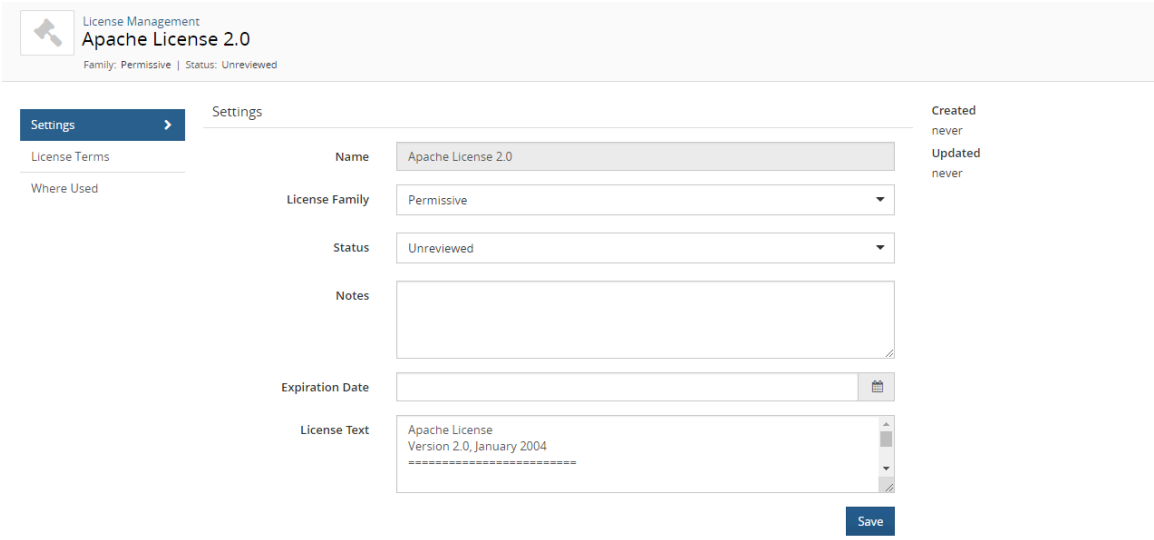
1. Log in to Black Duck with the License Manager role.
2. Click 

> **Licenses**.

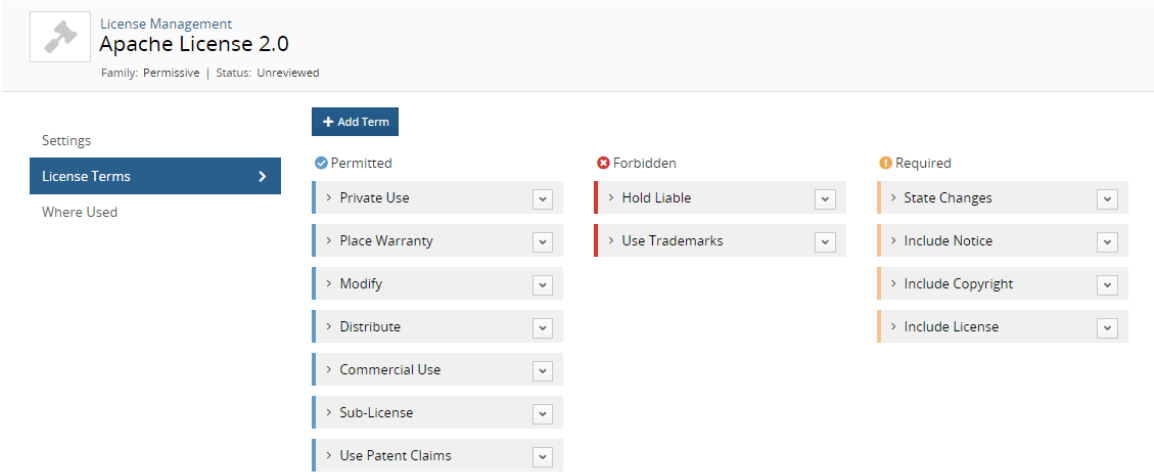
   The License Management page appears.



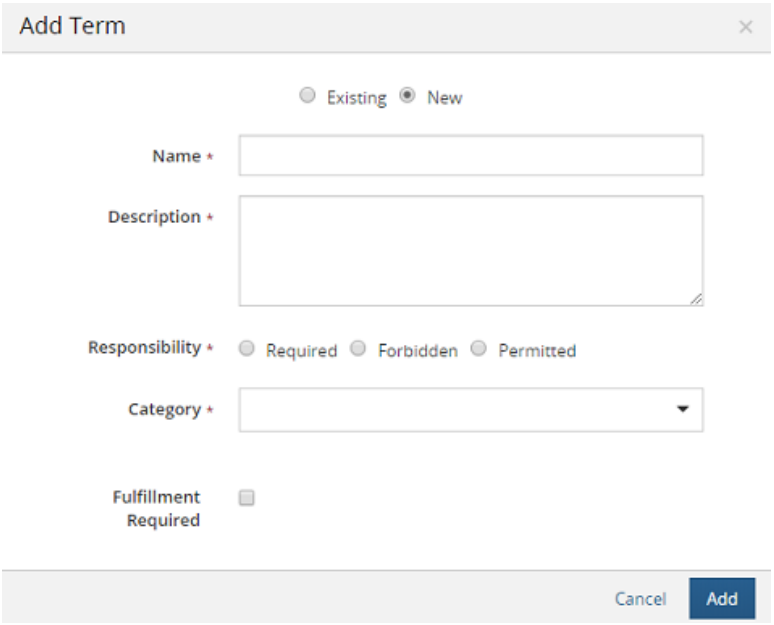
3. In the **Licenses** tab, select the license name to display the *License Name* **Settings** tab.



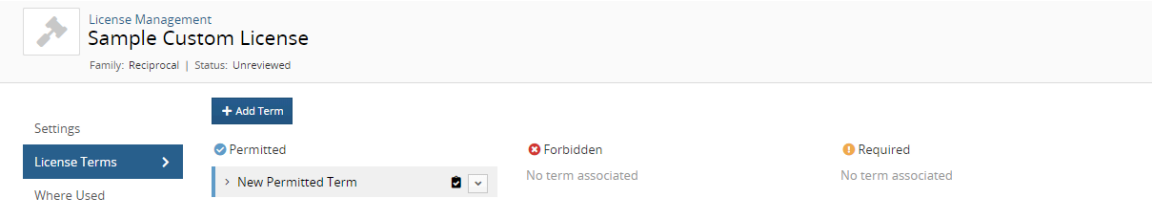
4. Select the **License Terms** tab to view the terms associated with this tab.



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.



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.



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 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.

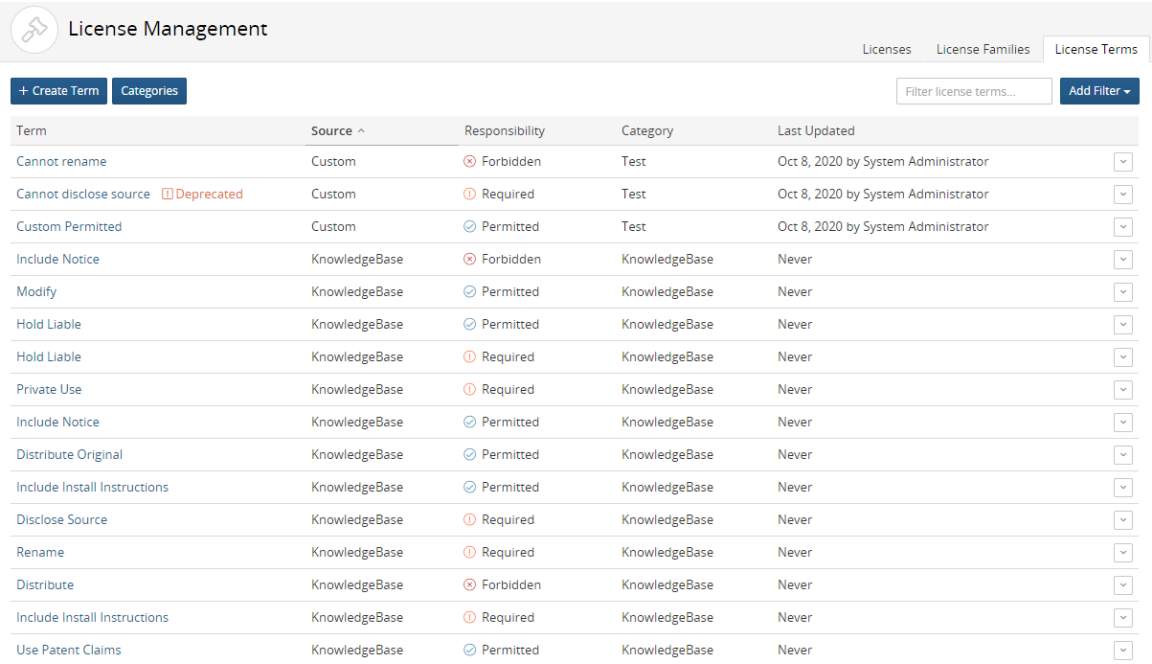
To create a category:

You can also create a category when creating a license term.

1. Log in to Black Duck with the License Manager role.
2. Click Manage > **Licenses**.

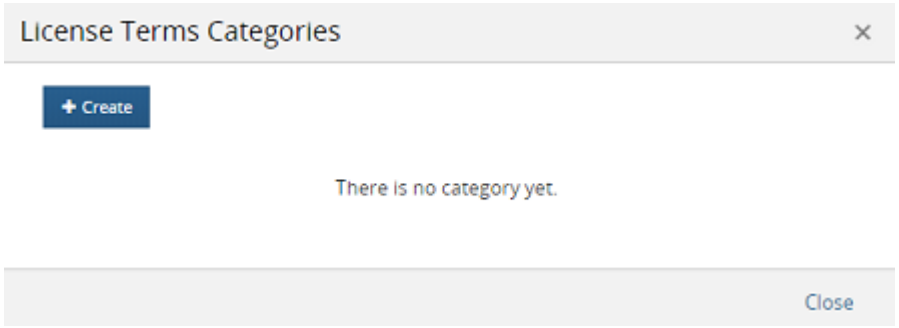
   The License Management page appears.

   Select the **License Terms** tab to display all license terms.



3. Click **Categories**.

The License Terms Categories dialog box appears.



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.

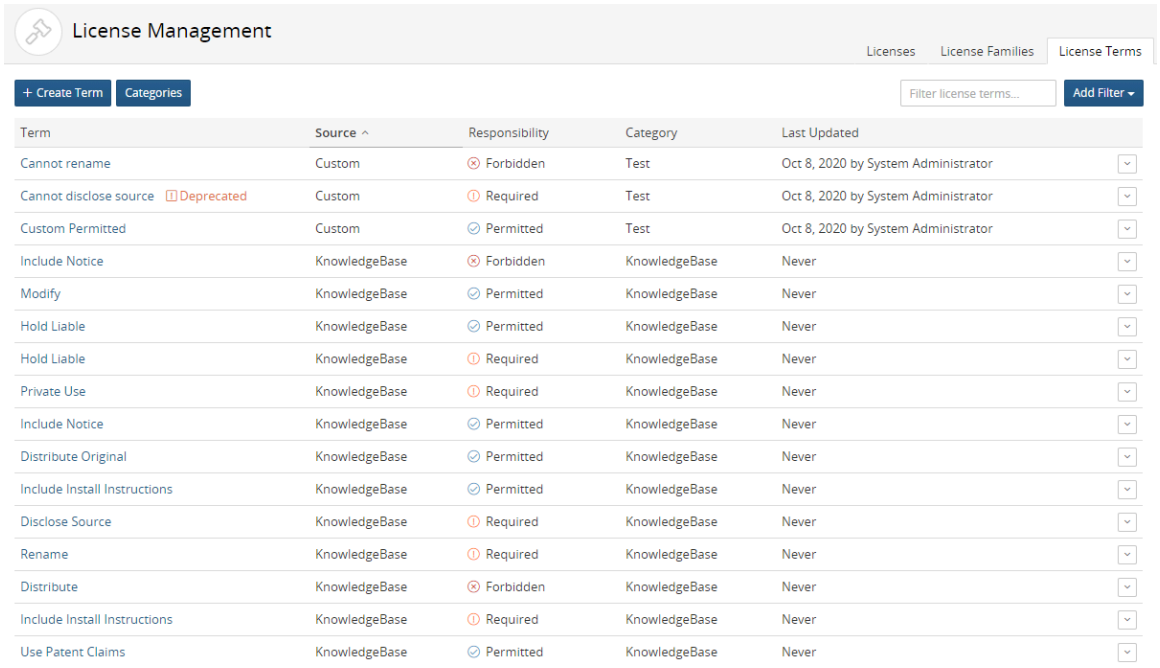
To delete a category:

You cannot delete a category that is in use.

1. Log in to Black Duck with the License Manager role.
2. Click Manage > **Licenses**.

   The License Management page appears.

   Select the **Terms** tab to display all license terms.



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

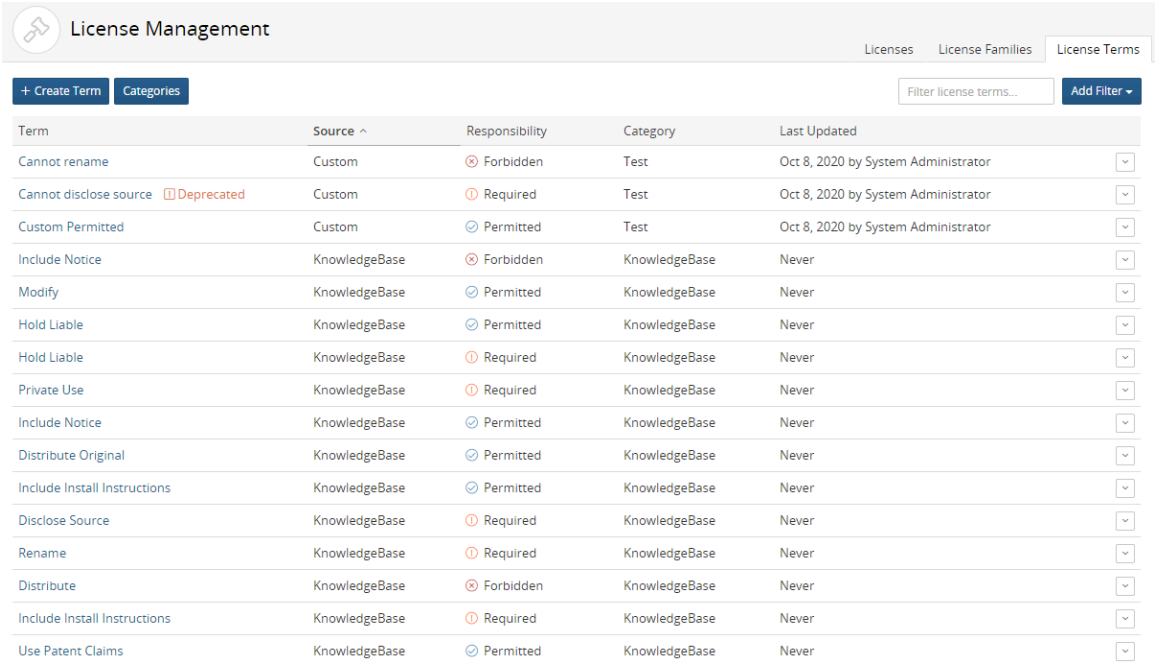
You can associate a new license term you created or an existing KnowledgeBase term to one or more custom or KnowledgeBase licenses.

When a license term is associated to a license, that term will appear to users when viewing licenses terms, for example, in the BOM.

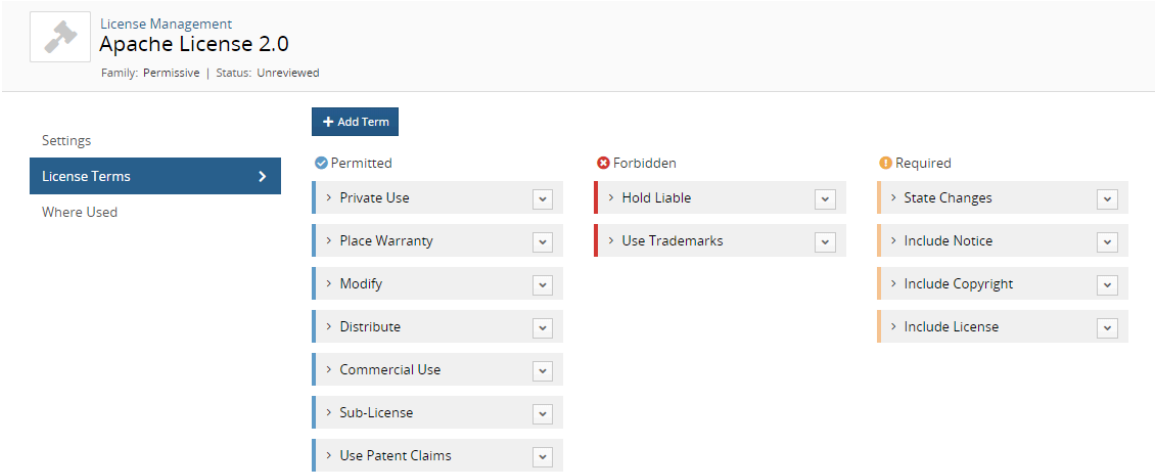
Only users with the License Manager role can associate a license term to a license.

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:



- Using the **License Terms** tab for an individual license:



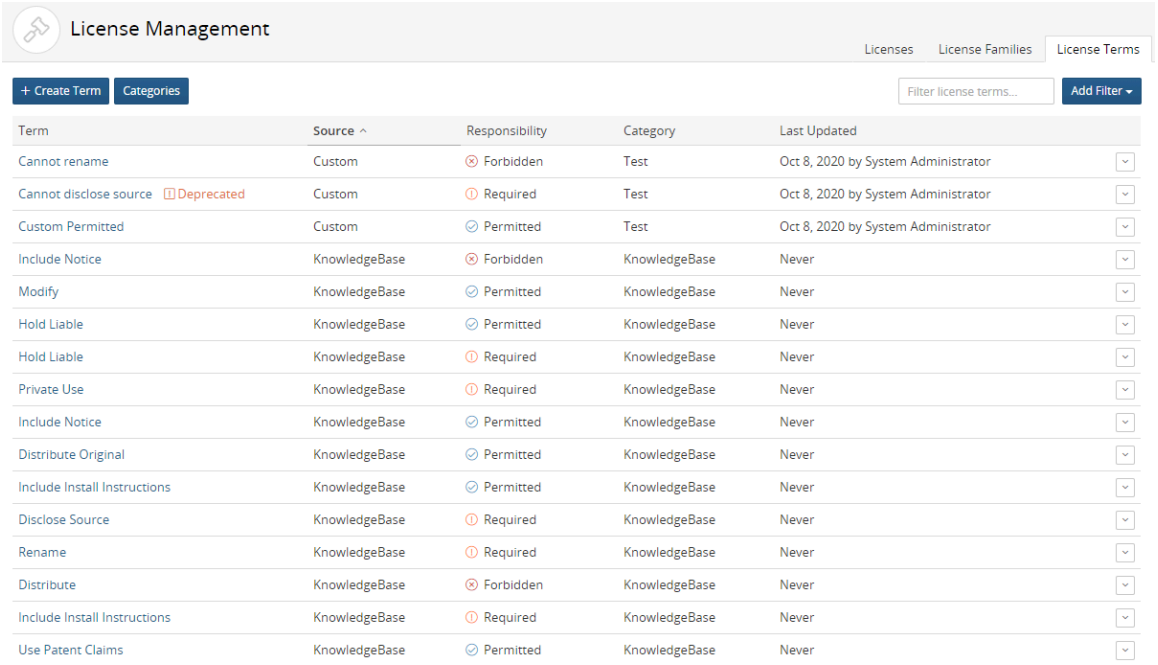
To associate a license term to one or more licenses:

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 > **Licenses**.

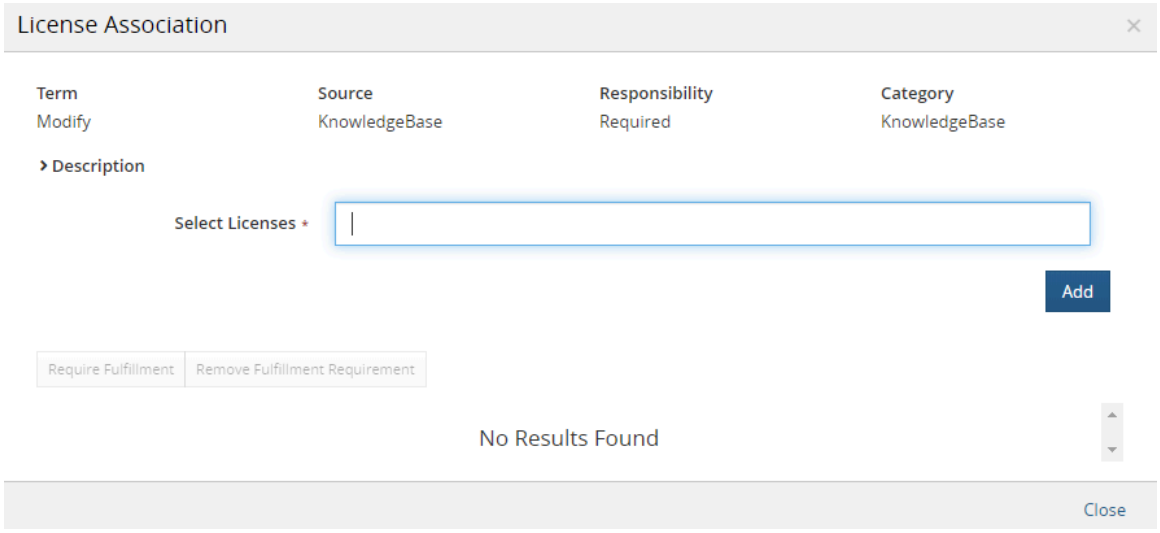
   The License Management page appears.

   Select the **License Terms** tab to display all license terms.



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:

   a. Select the check box next to the license where fulfillment of this term is required.

   b. Click **Require Fulfillment**. The Fulfillment Required icon ( ) appears in the table for the license where this term is required.

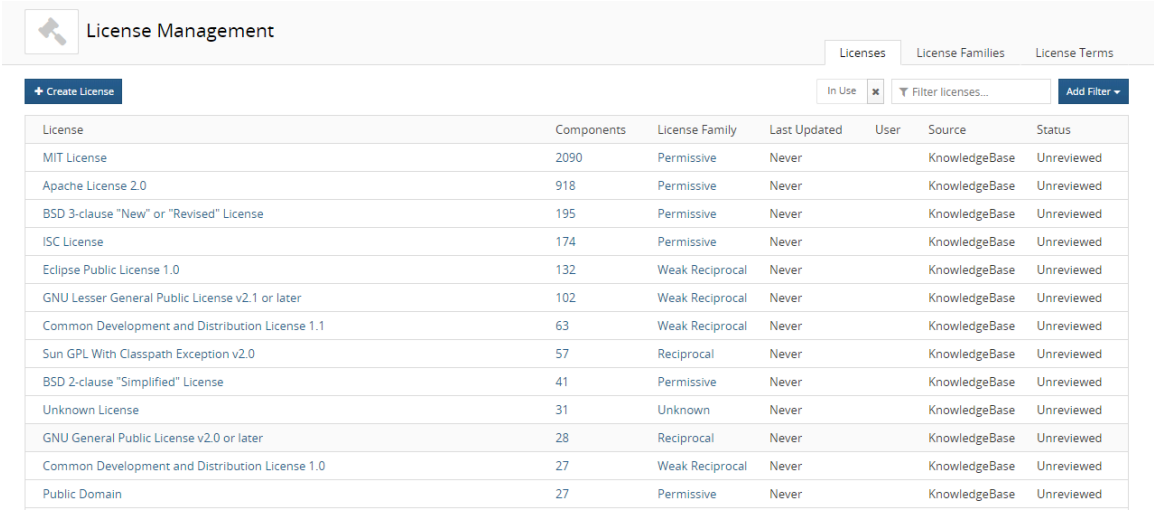
      Click **Remove Fulfillment Requirement** to remove the requirement that this term must be fulfilled.

6. Click **Close**.

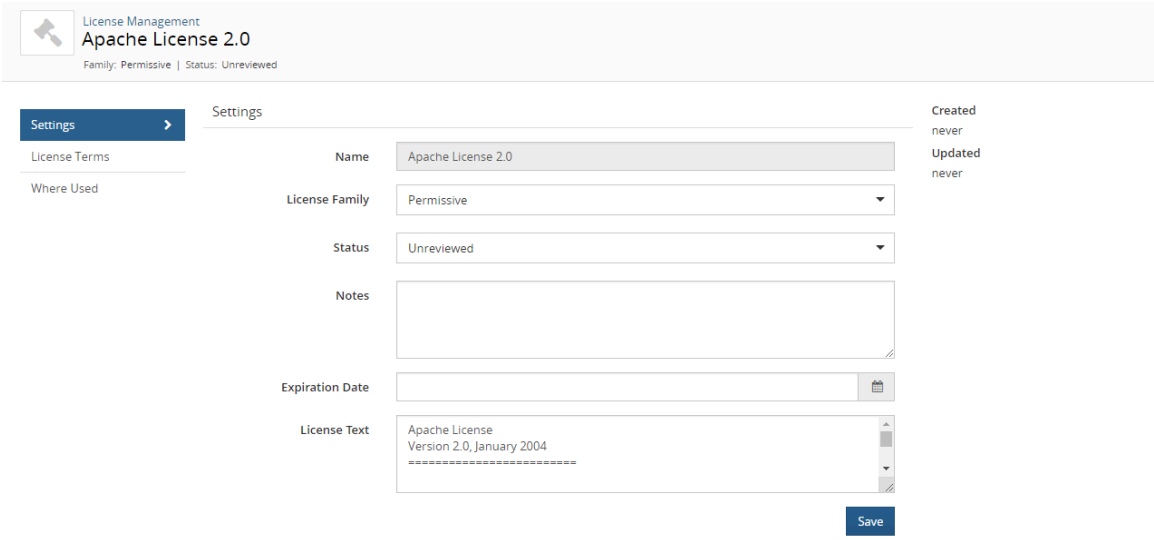
To associate an existing license term to a specific license:

1. Log in to Black Duck with the License Manager role.
2. Click Manage > **Licenses**.

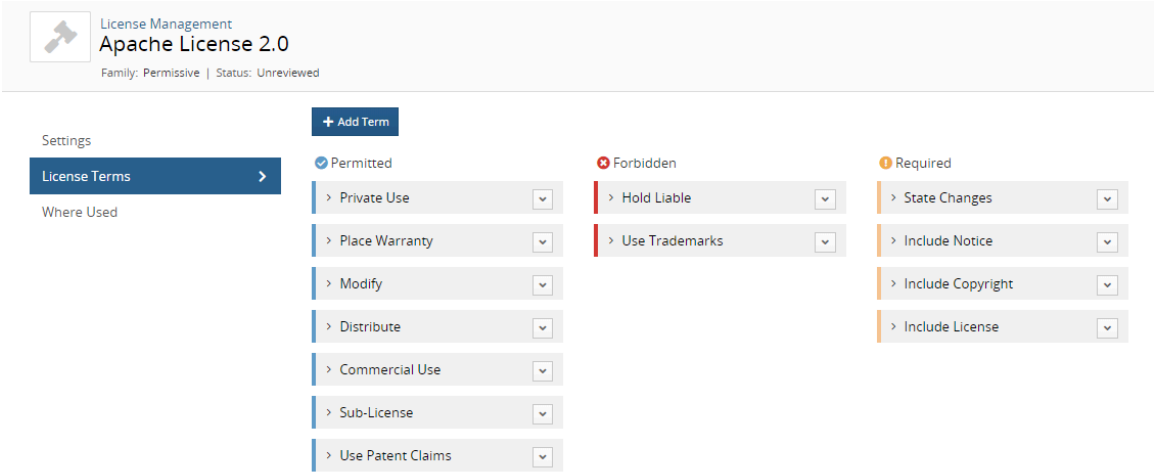
   The License Management page appears.



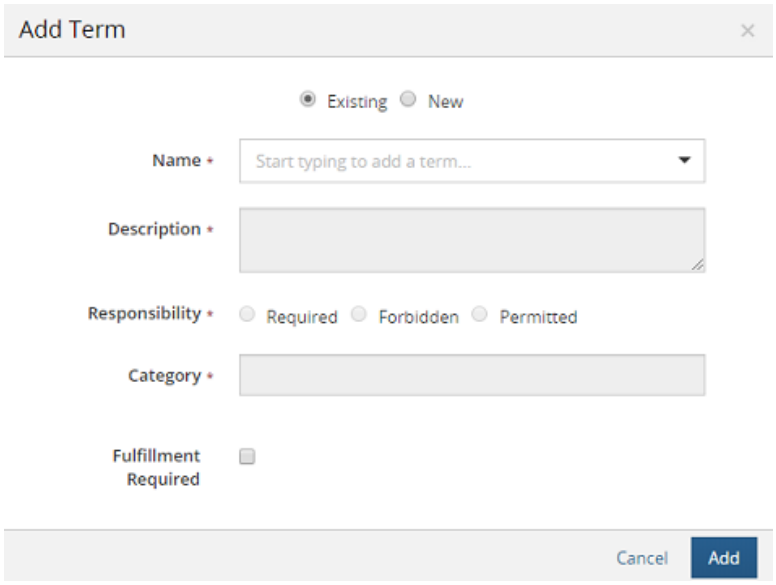
3. In the **Licenses** tab, select the license name to display the *License Name* **Settings** tab.



4. Select the **License Terms** tab to view the terms associated with this tab.



5. Click **Add Term** to open the Add Term dialog box.
6. Select **Existing** to add an existing license term.



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.

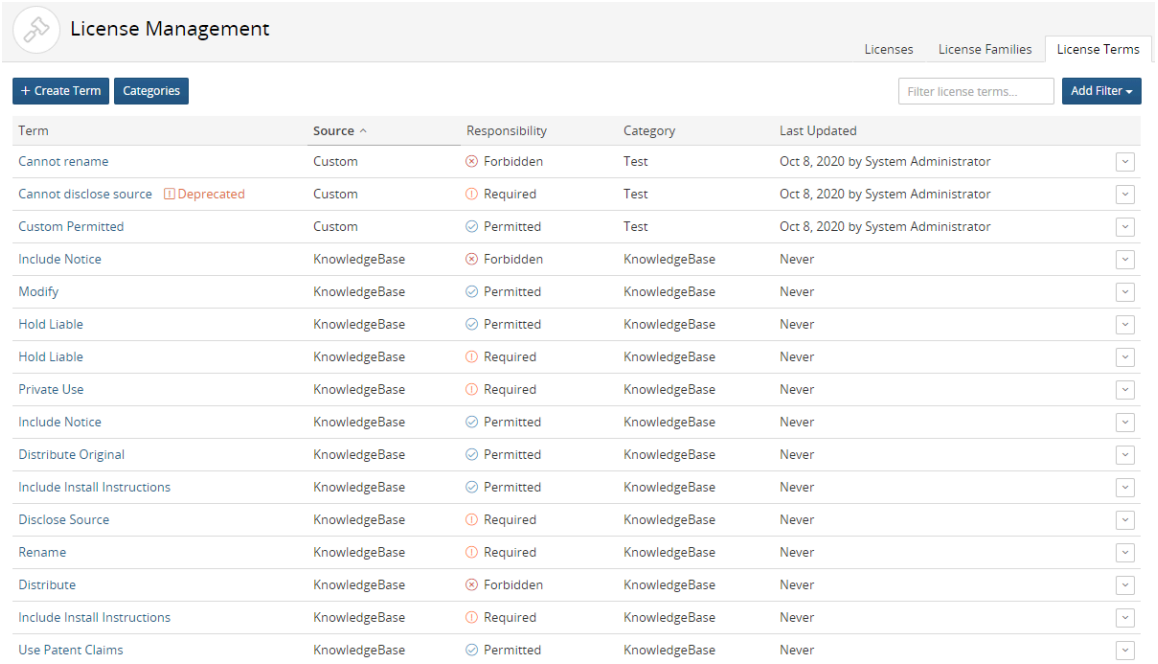
Only users with the License Manager role can edit license terms.

To edit a license term:

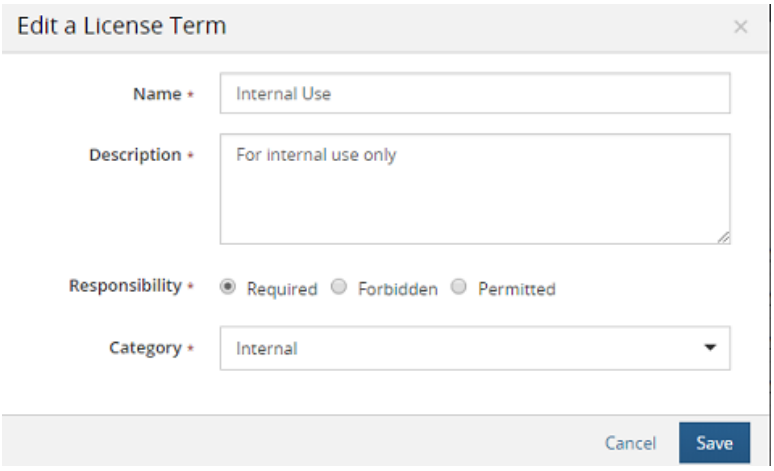
1. Log in to Black Duck with the License Manager role.
2. Click Manage > **Licenses**.

   The License Management page appears.

   Select the **License Terms** tab to display all license terms.



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.

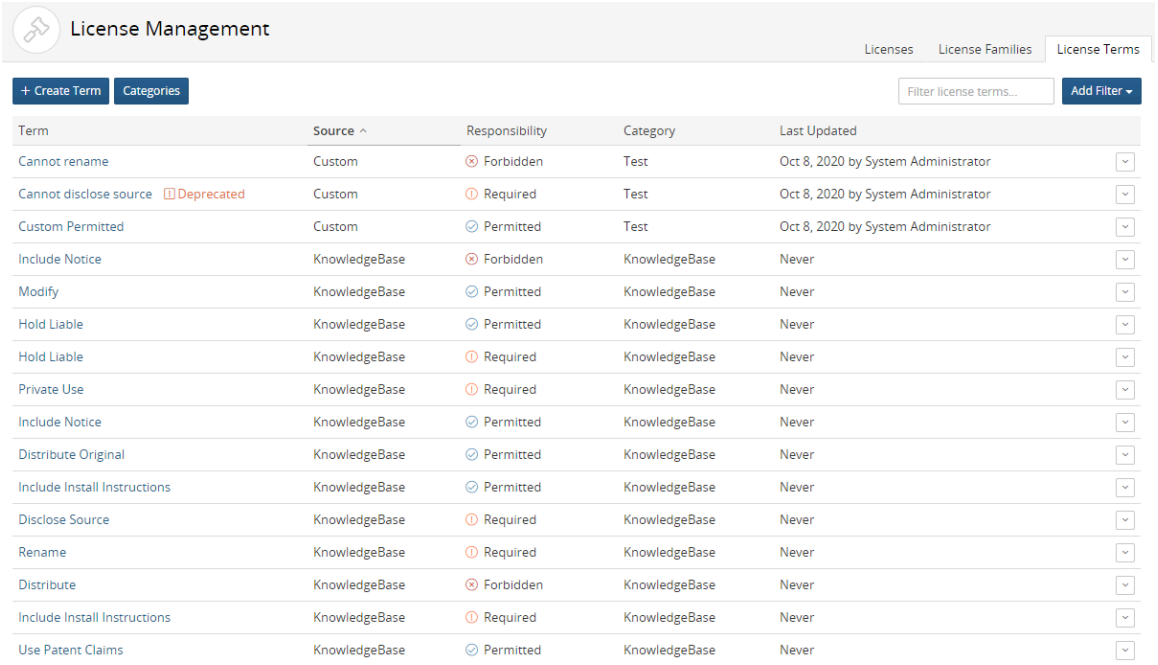
To delete a license term:

1. Log in to Black Duck with the License Manager role.
2. Click 

> **Licenses**.

   The License Management page appears.

   Select the **License Terms** tab to display all license terms.



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.

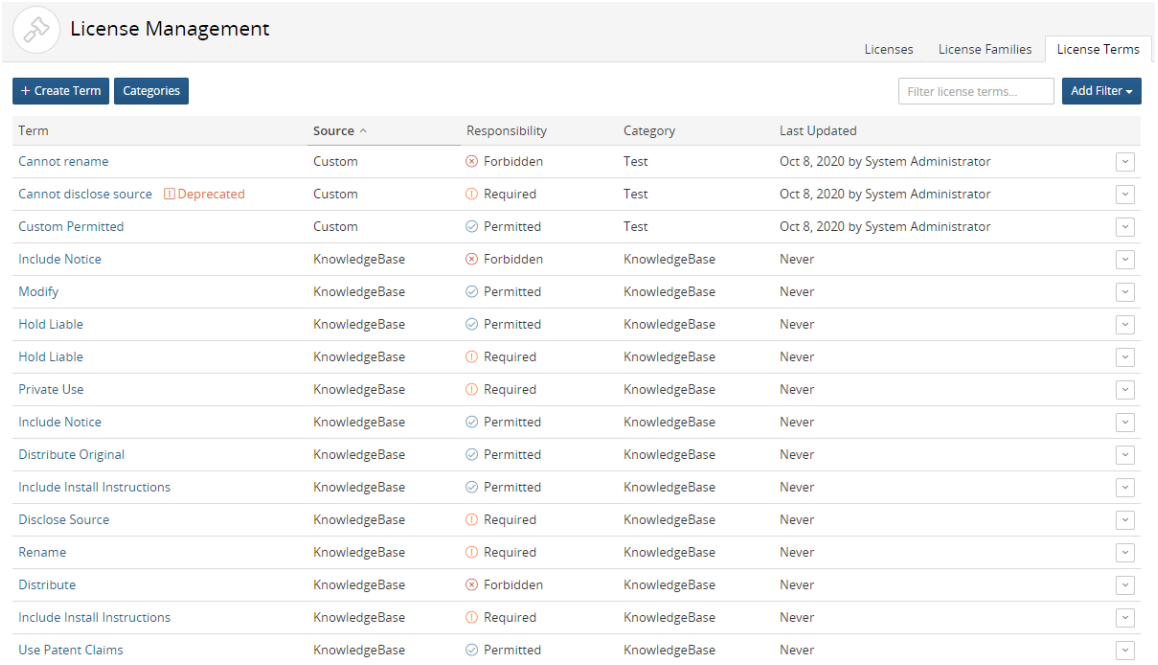
To deprecate a custom license term:

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 > **Licenses**.

   The License Management page appears.

   Select the **License Terms** tab to display all license terms.



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 

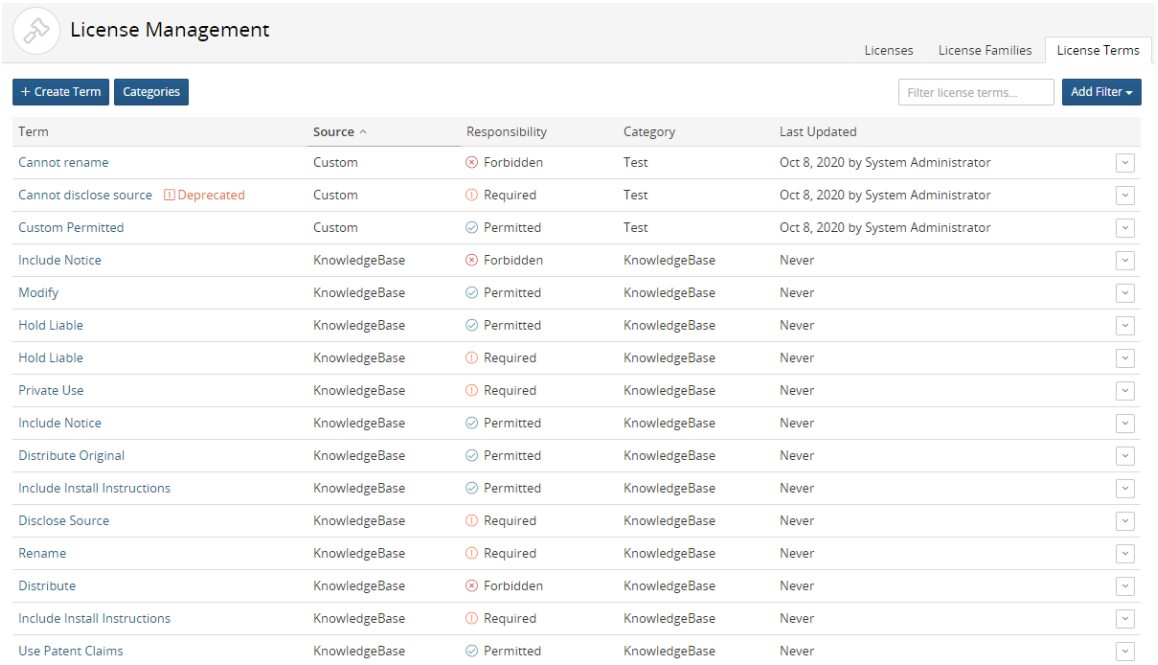
label does not appear to the BOM manager for any licenses that have this term associated to it.

To undo the deprecation status of a custom license term:

1. Log in to Black Duck with the License Manager role.
2. Click Manage > **Licenses**.

   The License Management page appears.

   Select the **License Terms** tab to display all license terms.



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 

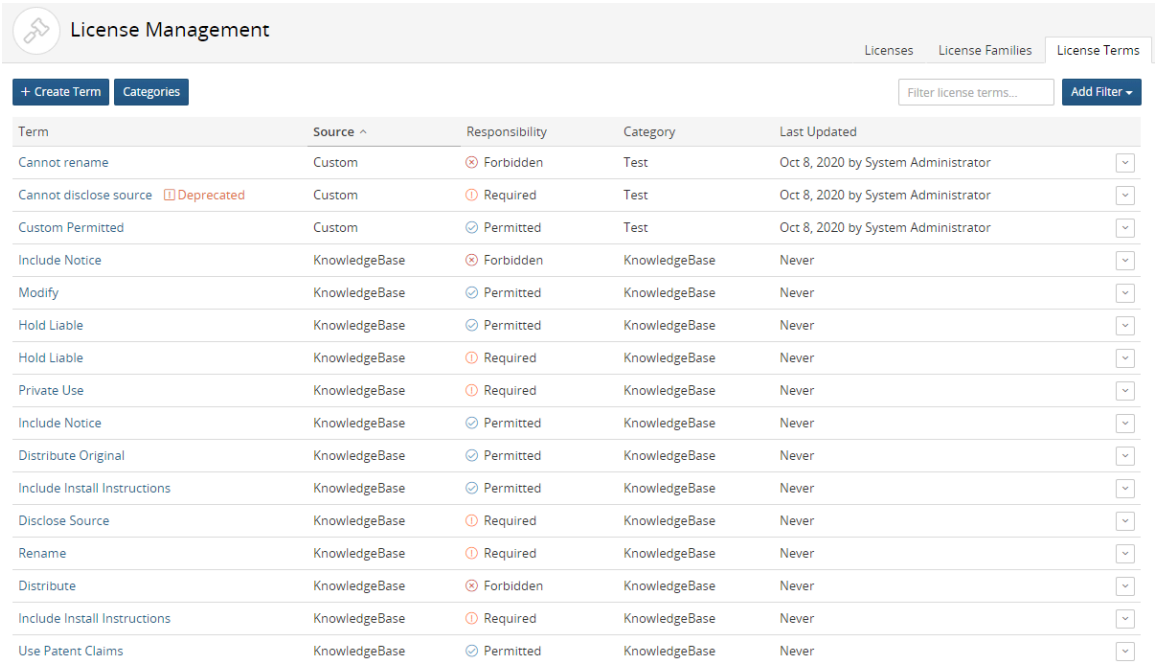
label is removed from the license term.

Removing a license term

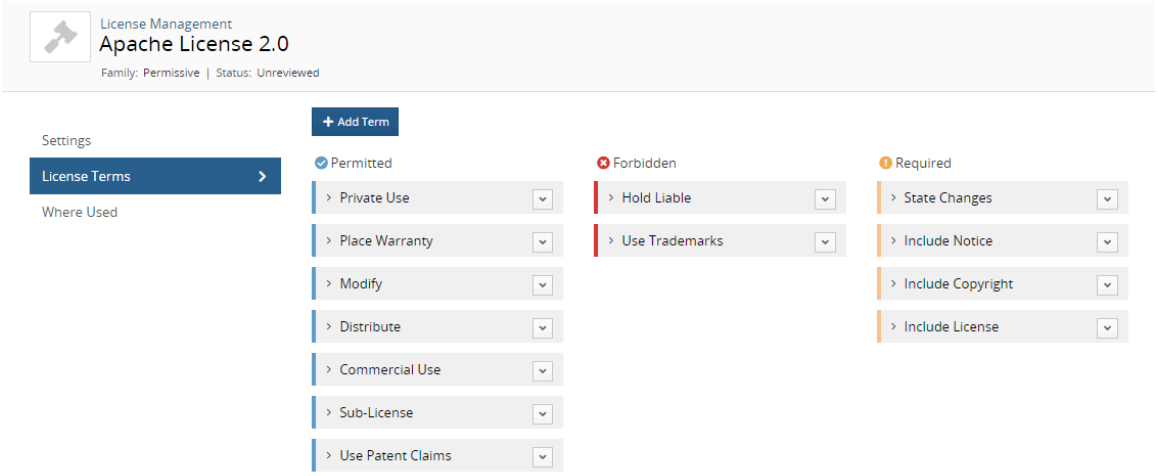
Use these procedures to remove a license term that you associated to a custom license or a KnowledgeBase license. When you remove a license term from a license, the term no longer appears to users viewing license terms, for example when BOM Managers view license information in the BOM.

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



- Using the **License Terms** tab for an individual license



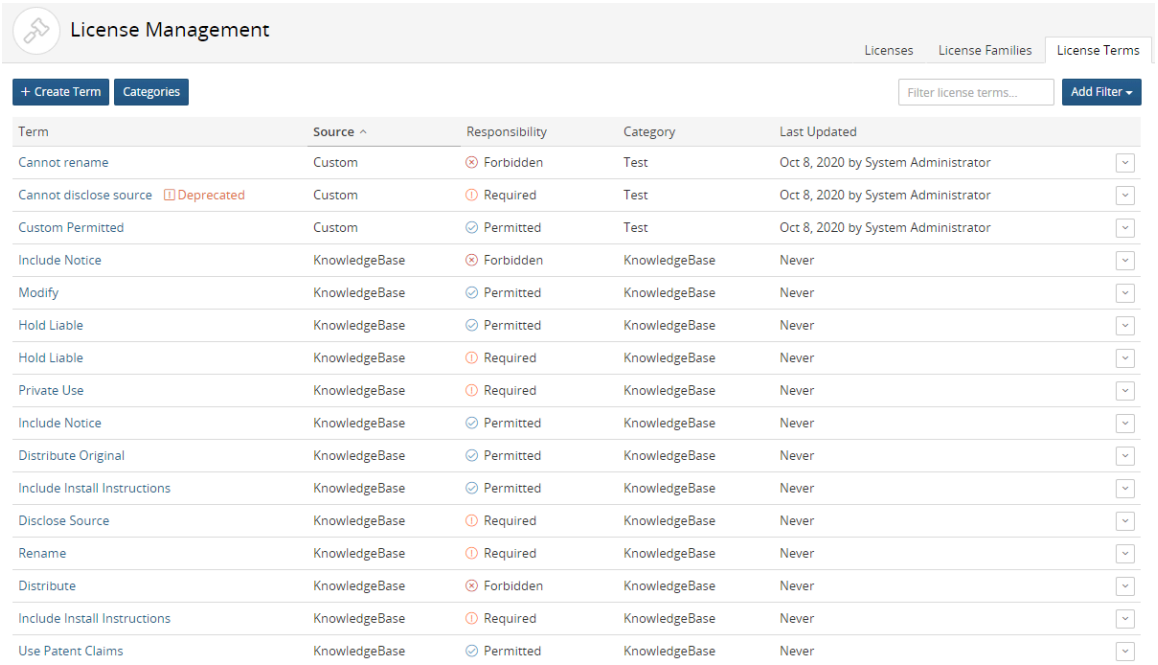
To remove a license term from one or more licenses :

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 Manage > **Licenses**.

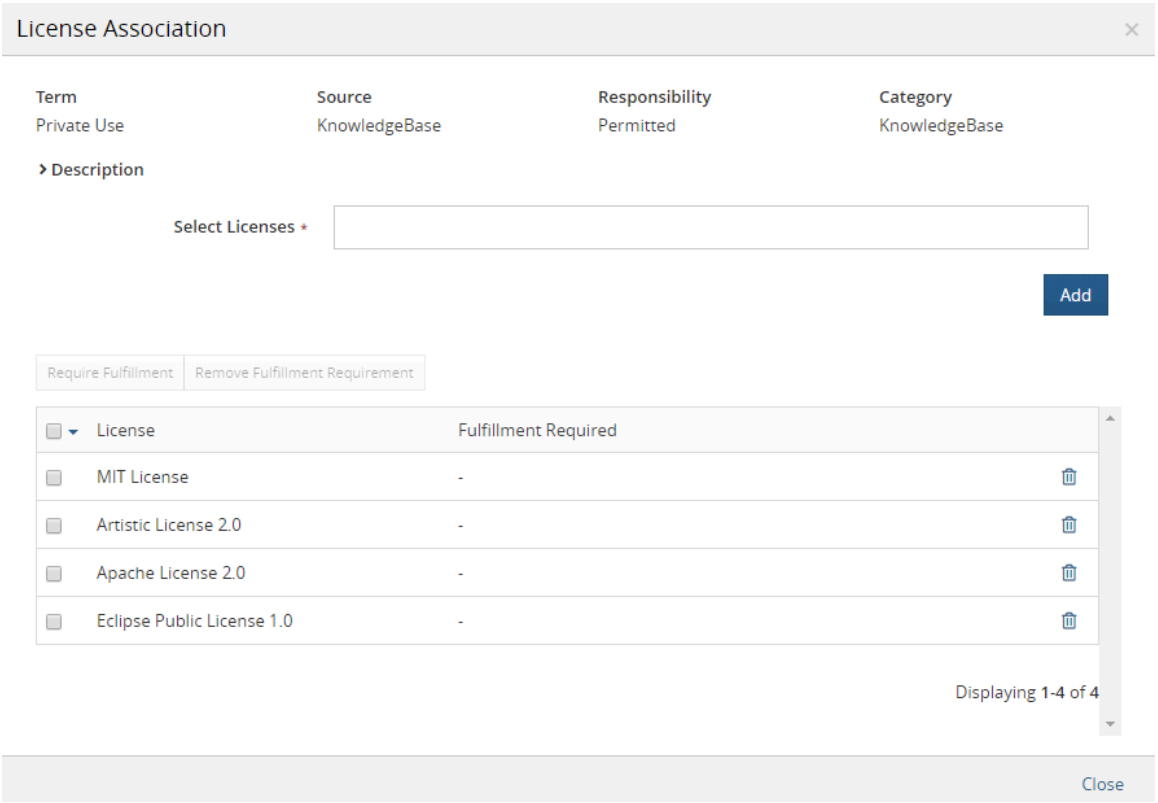
   The License Management page appears.

   Select the **License Terms** tab to display all license terms.



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.

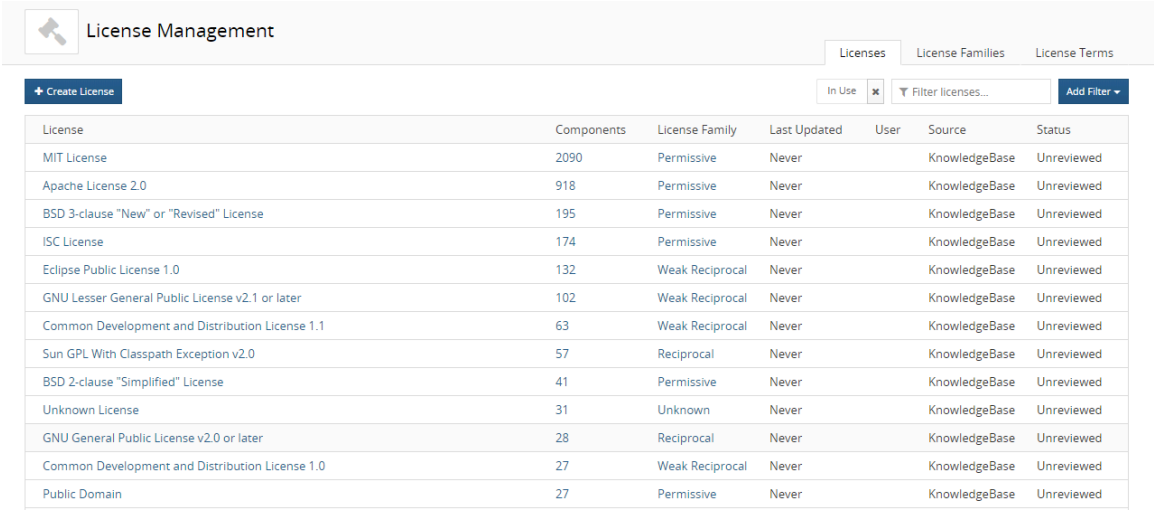
   The term is removed from the license.

To remove a license term from a single license:

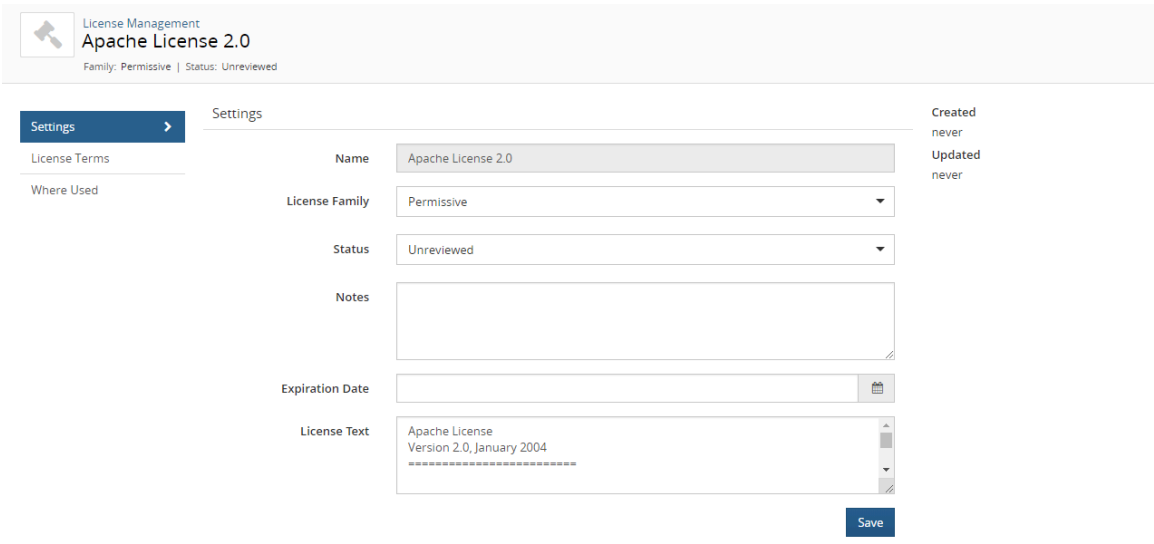
Use this method to remove a license term when viewing all terms for a single license.

1. Log in to Black Duck with the License Manager role.
2. Click Manage > **Licenses**.

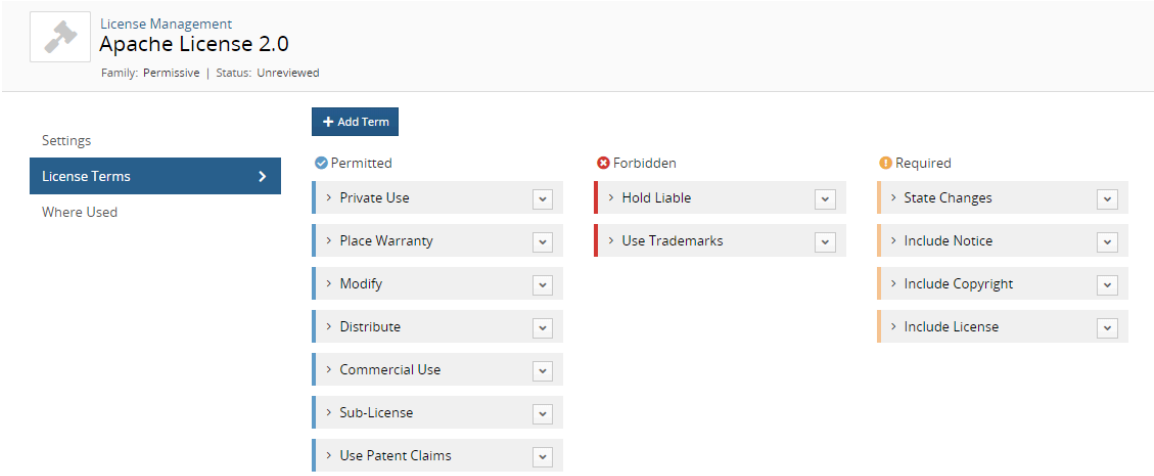
   The License Management page appears.



3. In the **Licenses** tab, select the license name to display the *License Name* **Settings** tab.



4. Select the **License Terms** tab to view the terms associated with this tab.



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.

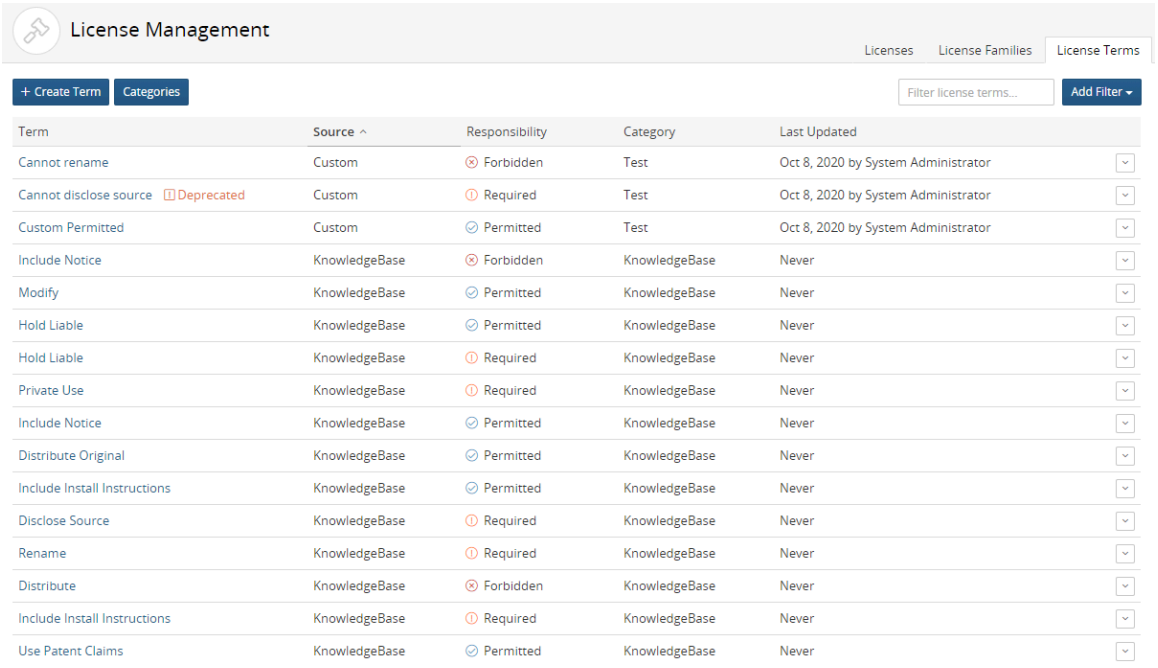
Deactivating a KnowledgeBase term

You may decide not to show your users specific license terms that are defined by Black Duck KnowledgeBase.

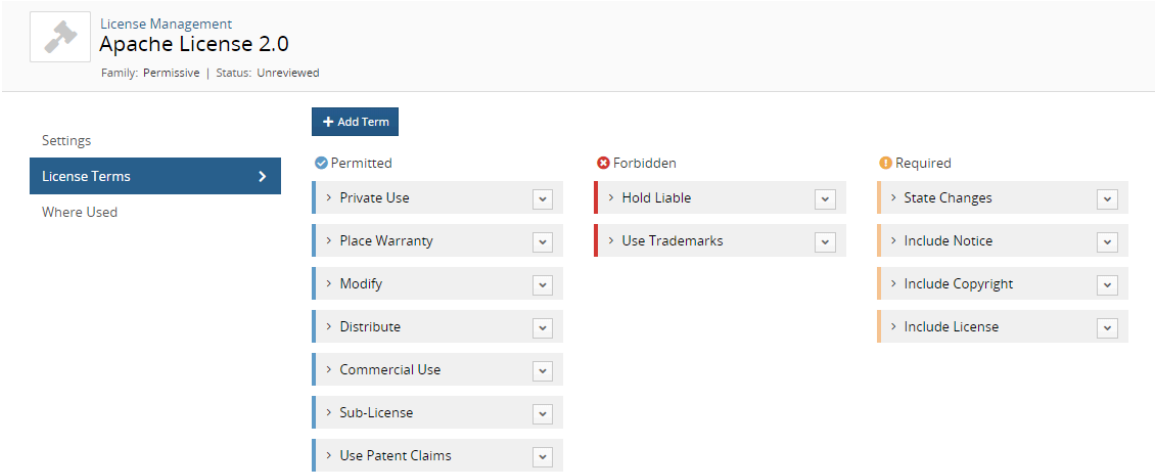
When a term is deactivated, it does not appear when users view the terms for a KnowledgeBase license; for example, when BOM Managers view the license terms in the BOM.

There are two methods you can use to deactivate a KnowledgeBase license term:

- Using the **License Terms** tab which lists all license terms



- Using the **License Terms** tab for an individual license



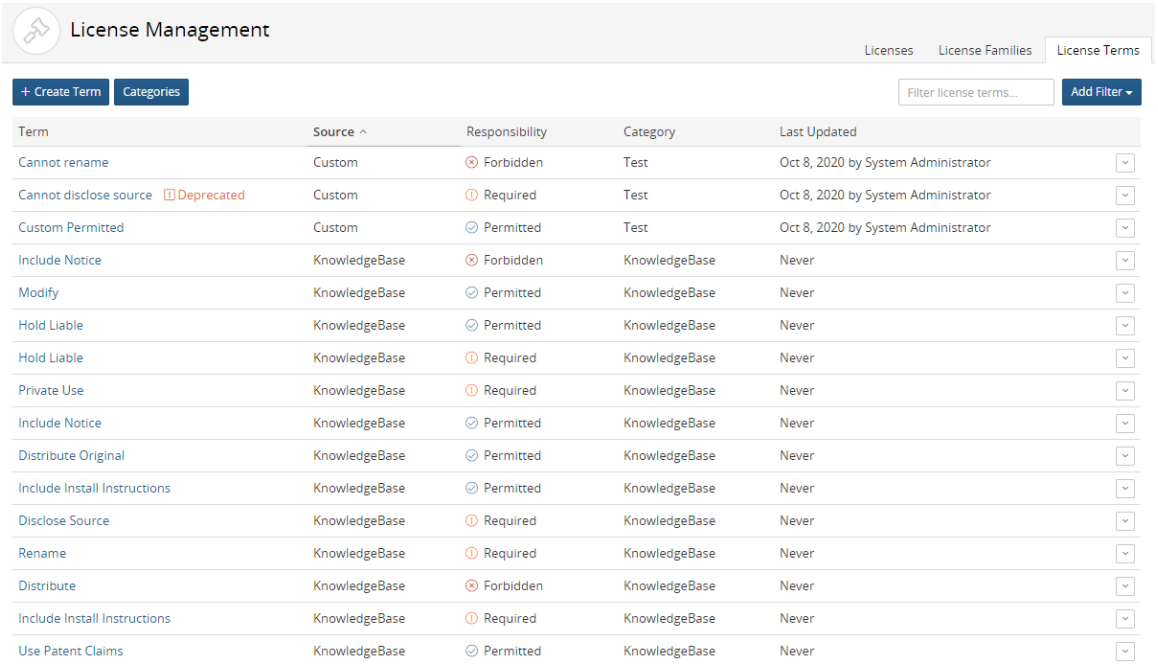
Deactivated KnowledgeBase license terms can be restored.

To deactivate a KnowledgeBase license term when viewing all terms :

1. Log in to Black Duck with the License Manager role.
2. Click Manage > **Licenses**.

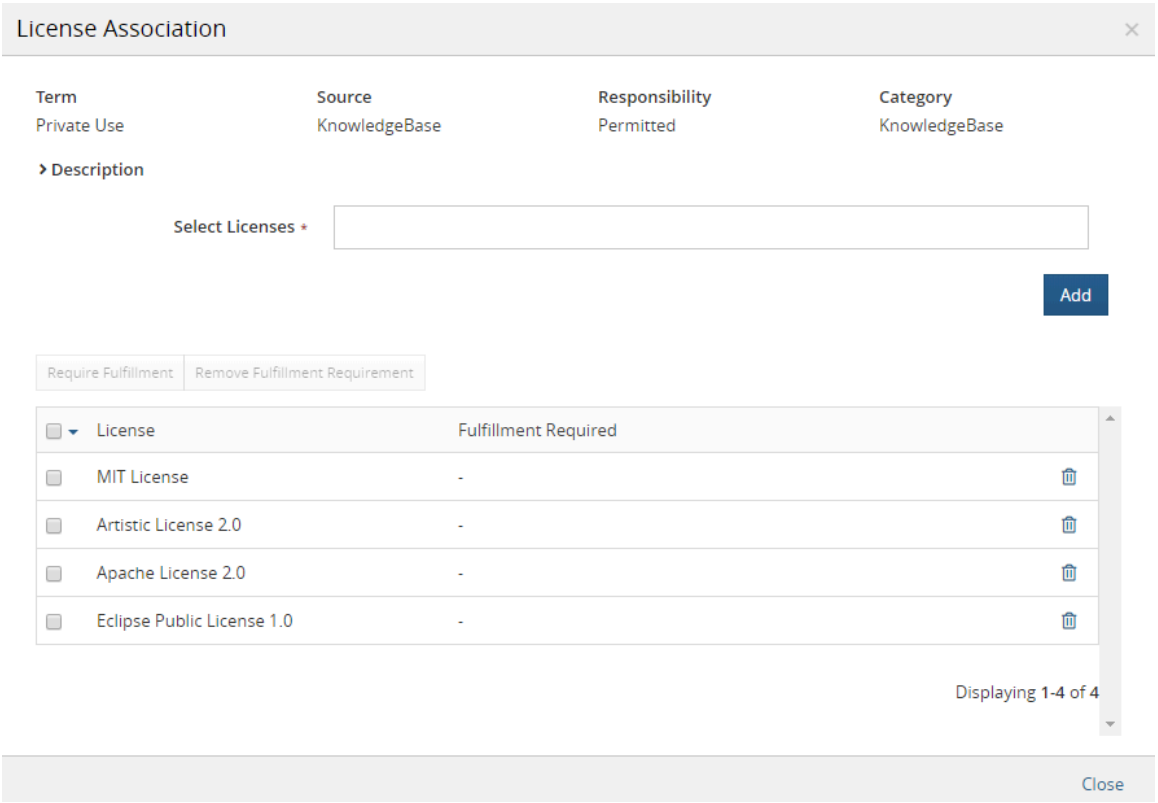
   The License Management page appears.

   Select the **License Terms** tab to display all license terms.

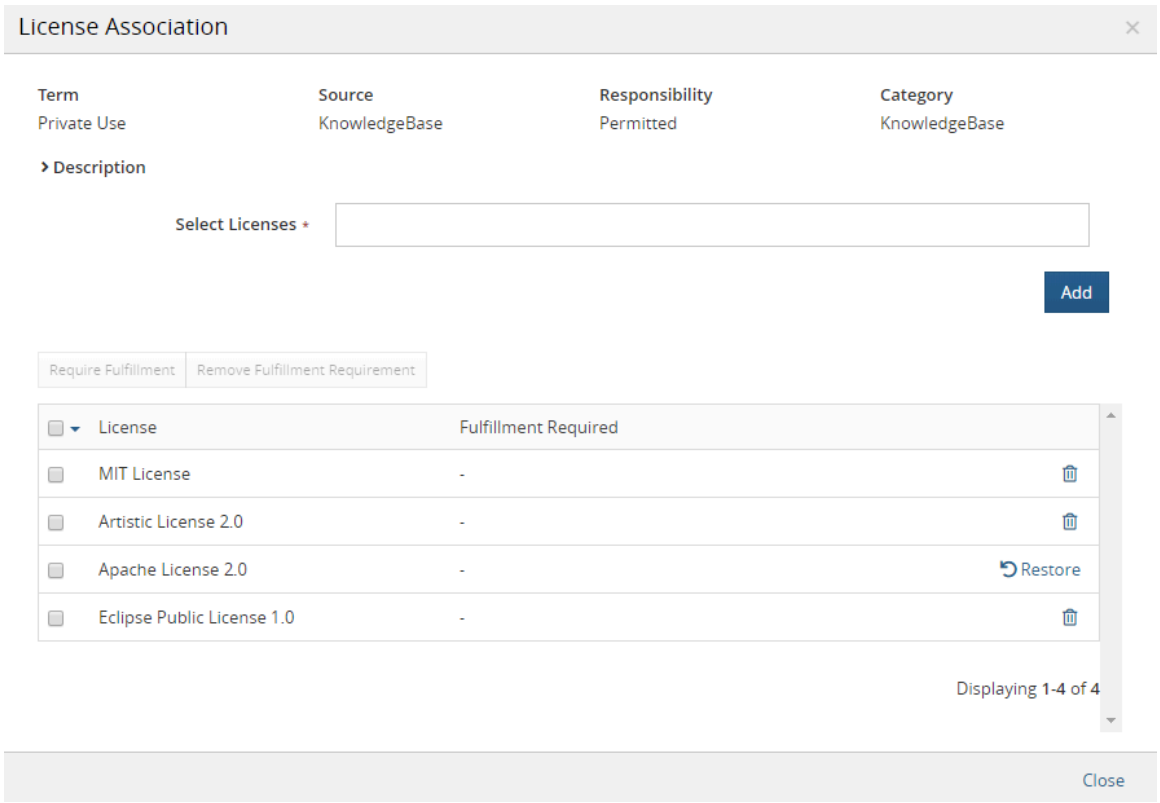


3. Click in the row of the license term and select **License Association**.

   The License Association dialog box appears showing all licenses that have this license terms associated to it.



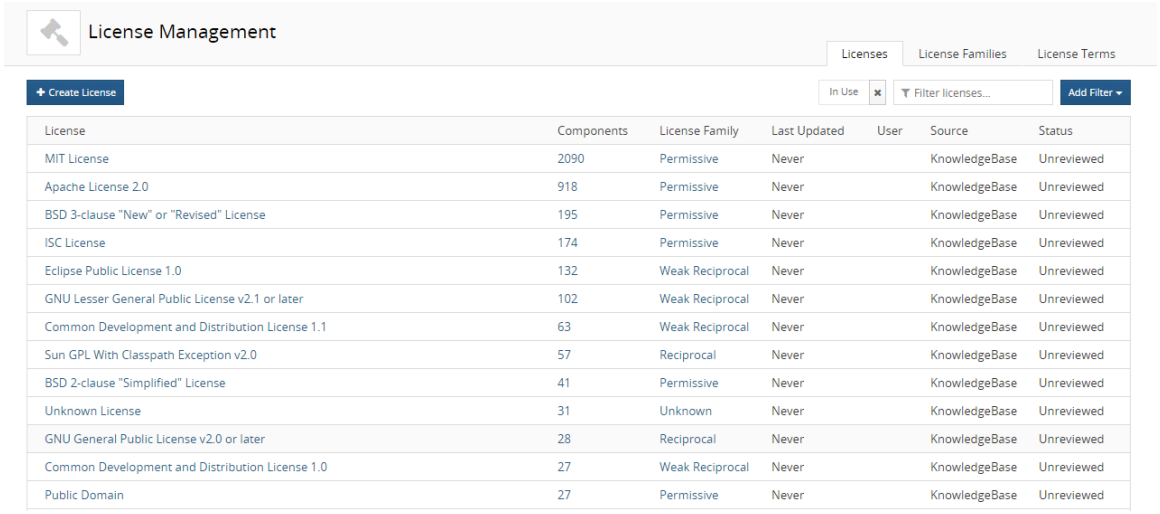
4. Click in the row of the license you want to disassociate to this license term.
5. Select **Deactivate** to confirm. The license term is no longer associated to that license.



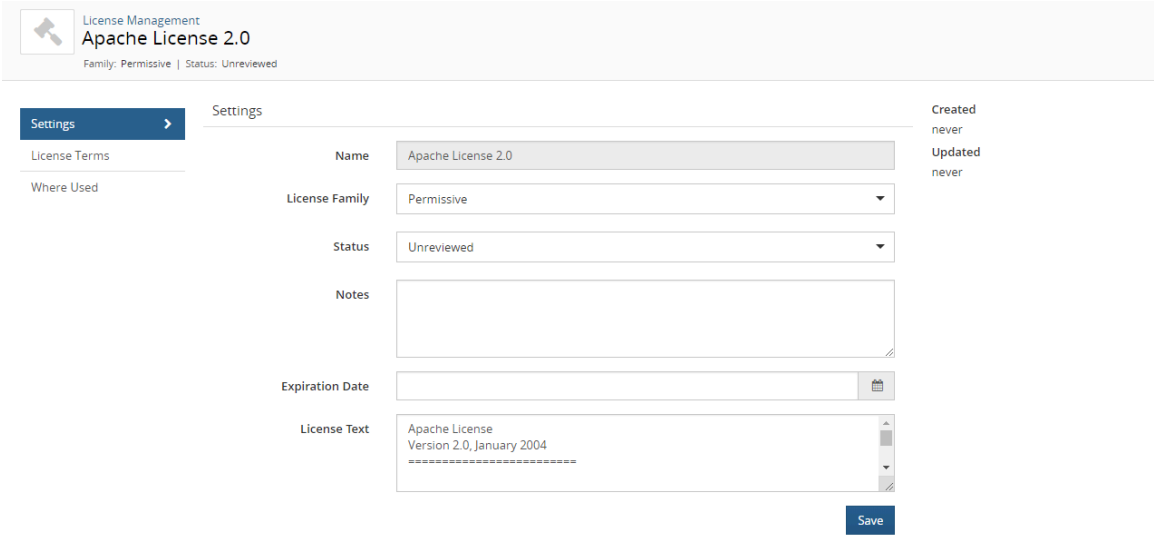
To deactivate a KnowledgeBase license term when viewing a license:

1. Log in to Black Duck with the License Manager role.
2. Click Manage > **Licenses**.

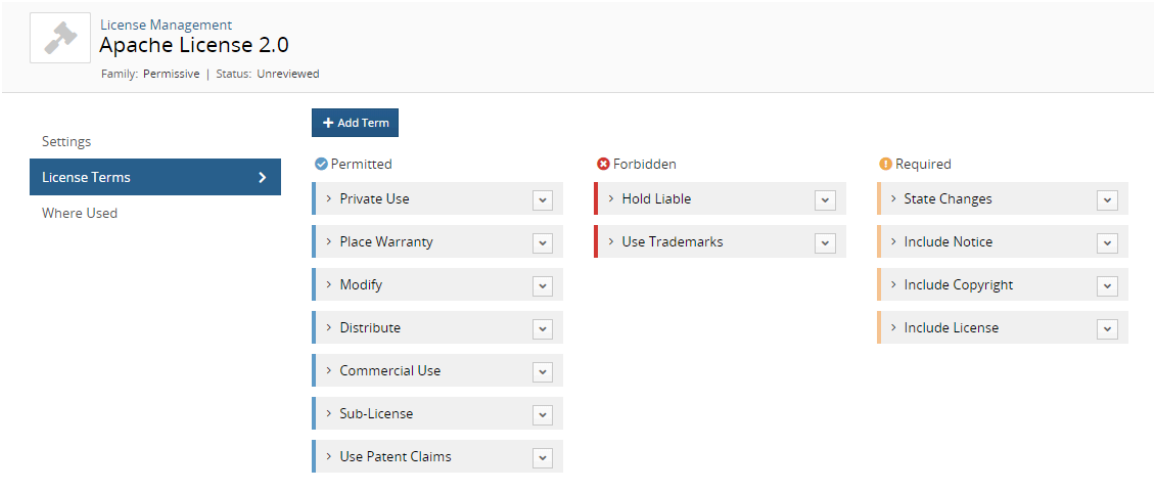
   The License Management page appears.



3. In the **Licenses** tab, select the license name to display the *License Name* **Settings** tab.



4. Select the **License Terms** tab to view the terms associated with this tab.

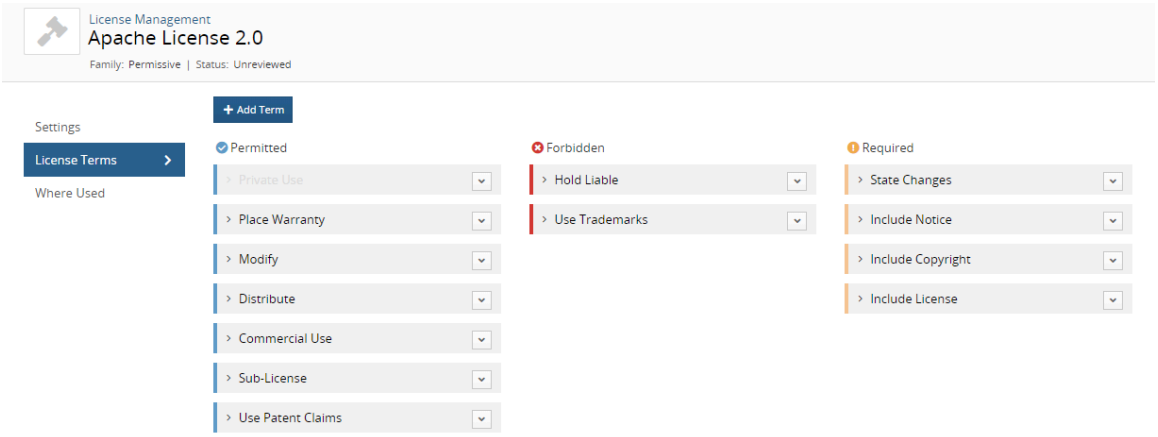


5. Click next to the KnowledgeBase license term you wish to deactivate and select **Deactivate**.

   The Deactivate Term dialog box appears.

6. Click **Deactivate** to confirm.

   The **License Terms** tab displays the term as deactivated.

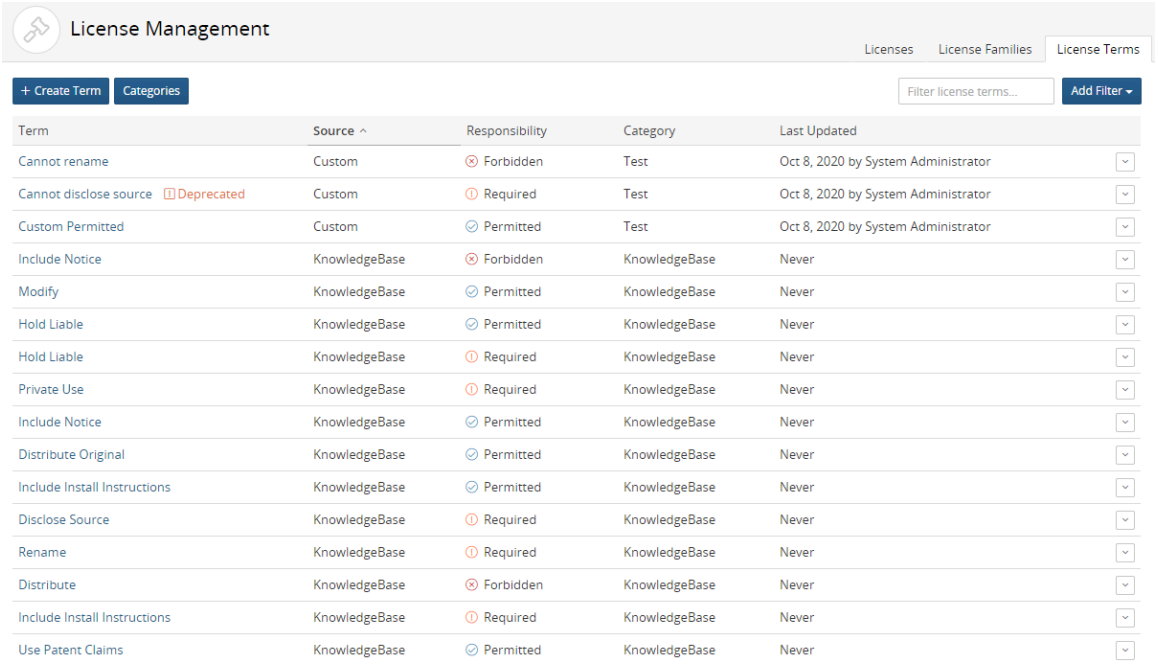


Restoring a KnowledgeBase license term

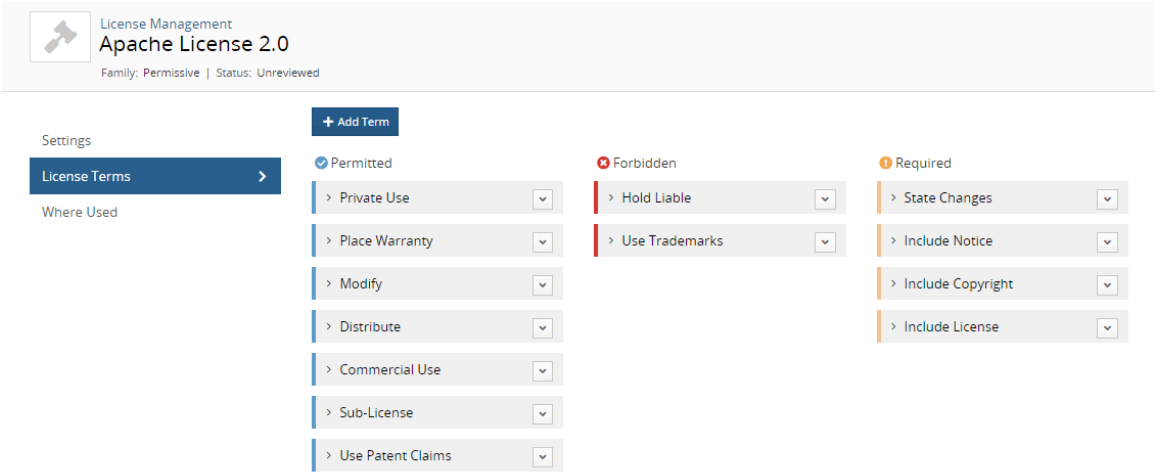
Use these procedures to restore a KnowledgeBase license term that you previously deactivated.

There are two methods you can use to restore a KnowledgeBase license term:

- Using the **License Terms** tab which lists all license terms



- Using the **License Terms** tab for an individual license

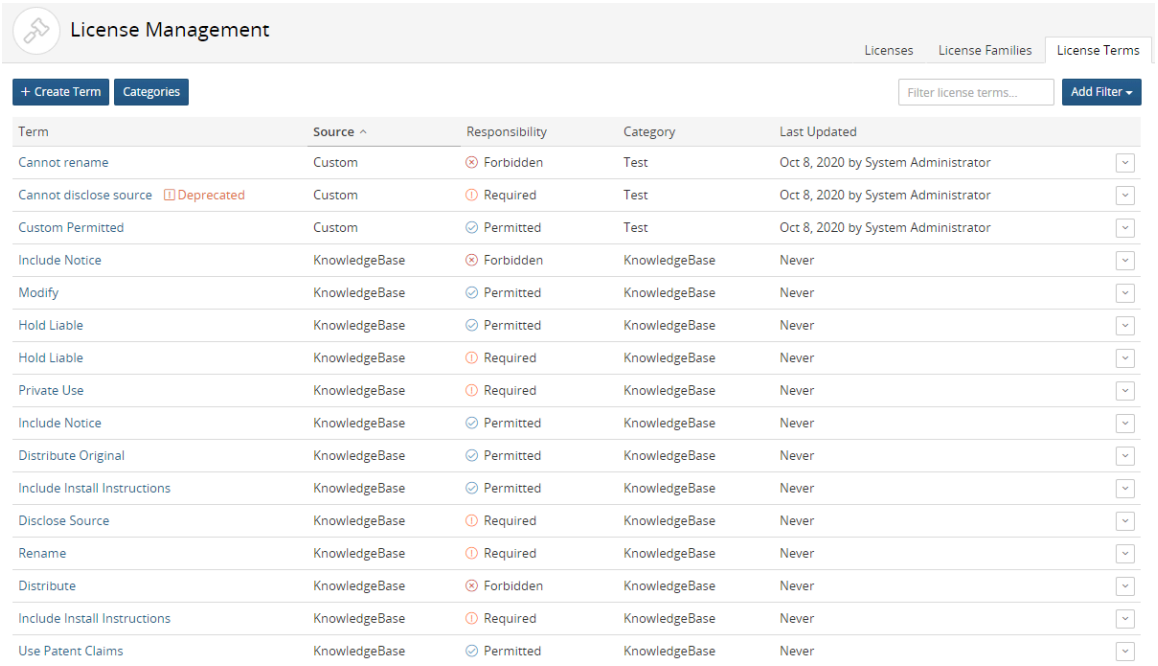


To restore a KnowledgeBase license term when viewing all terms:

1. Log in to Black Duck with the License Manager role.
2. Click Manage > **Licenses**.

   The License Management page appears.

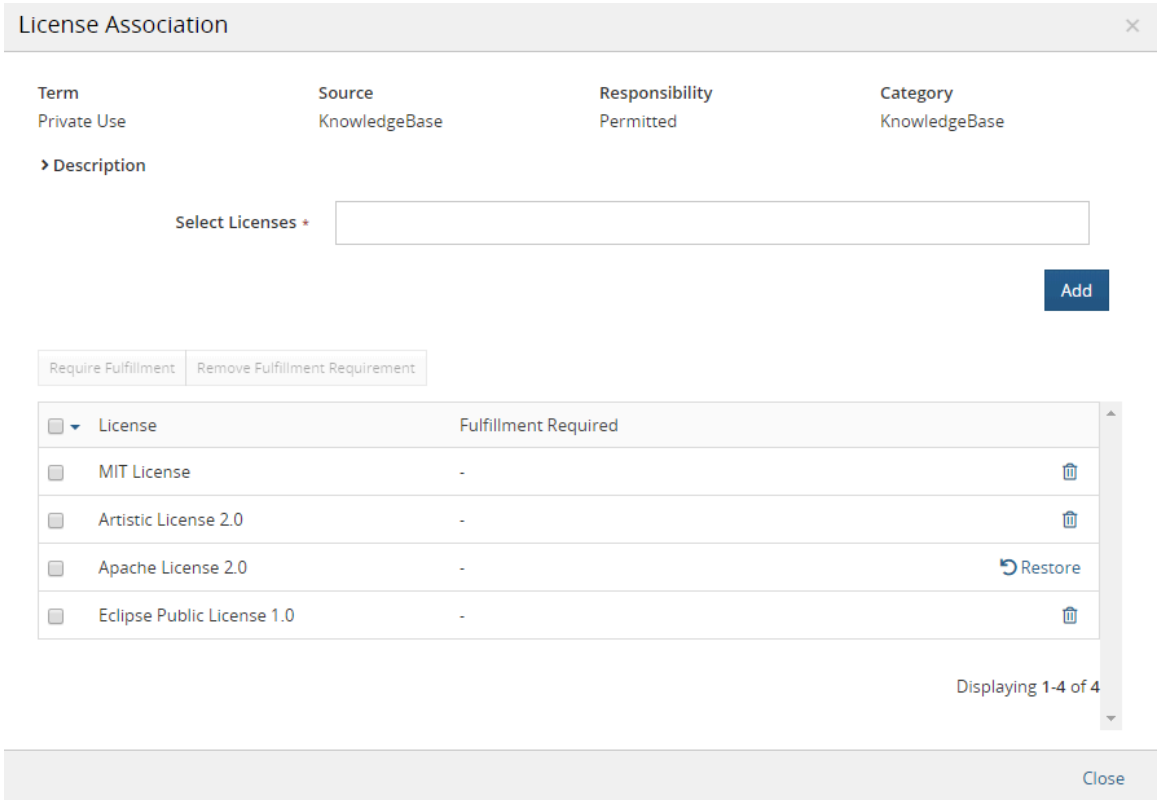
   Select the **License Terms** tab to display all license terms.



Term	Source	Responsibility	Category	Last Updated
Cannot rename	Custom	Forbidden	Test	Oct 8, 2020 by System Administrator
Cannot disclose source Deprecated	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

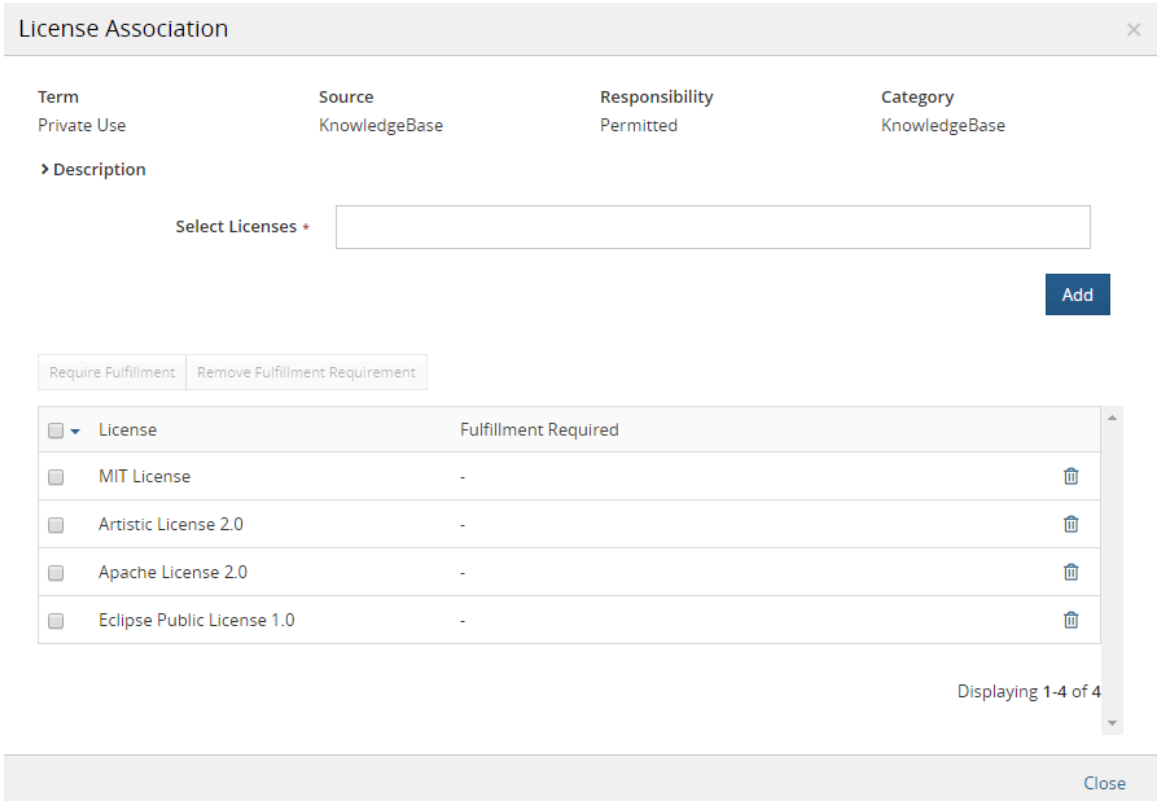
3. Click in the row of the KnowledgeBase license term and select **License Association**.

   The License Association dialog box appears showing all licenses that have this license terms associated to it.

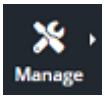


4. Click **Restore** in the row of the license(s) you want to restore.

   The license term is enabled for this license.

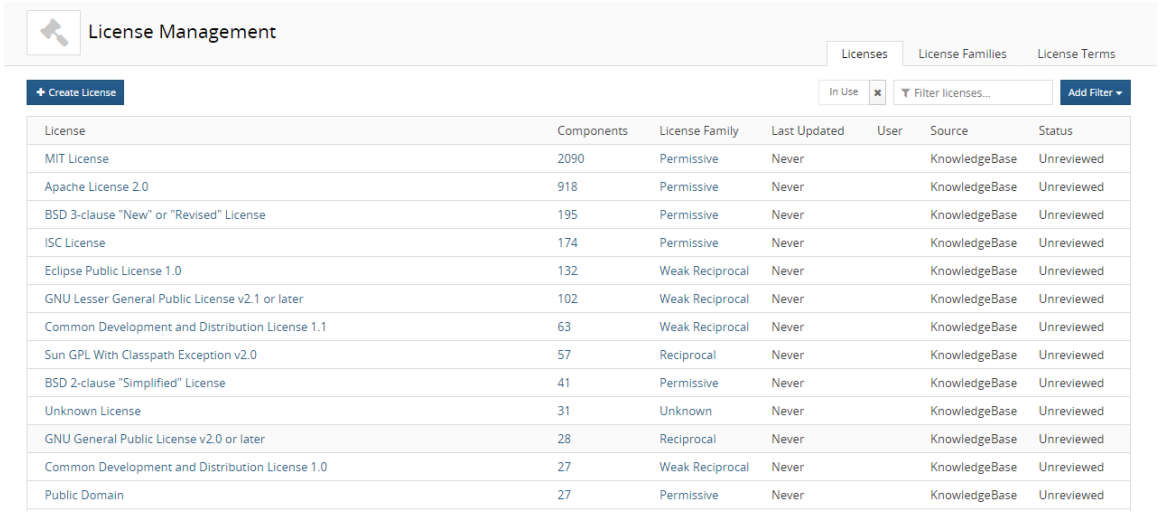


To restore a KnowledgeBase license term when viewing a license:

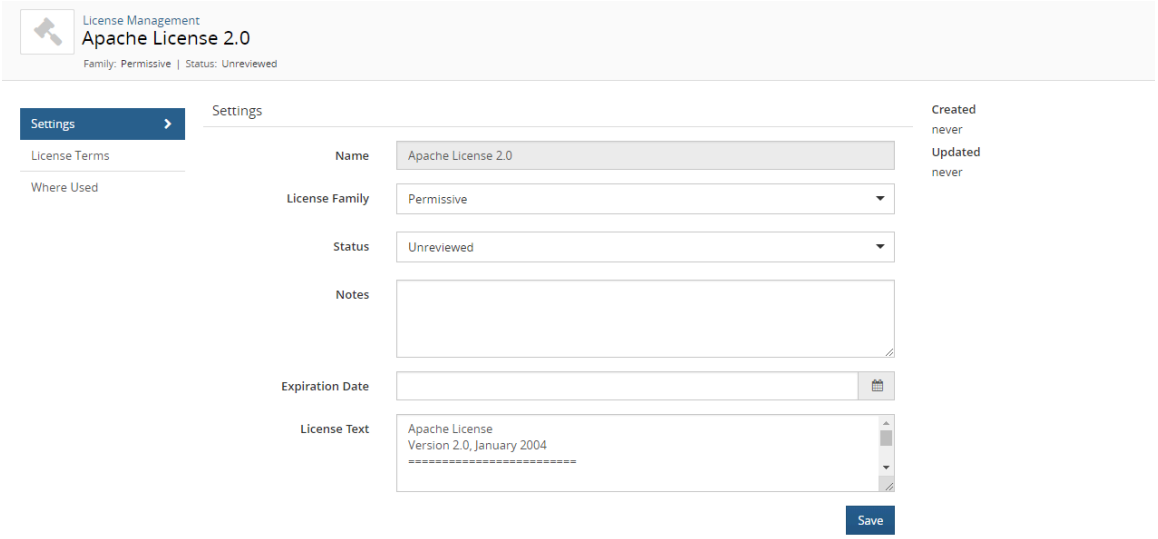
1. Log in to Black Duck with the License Manager role.
2. Click 

> **Licenses**.

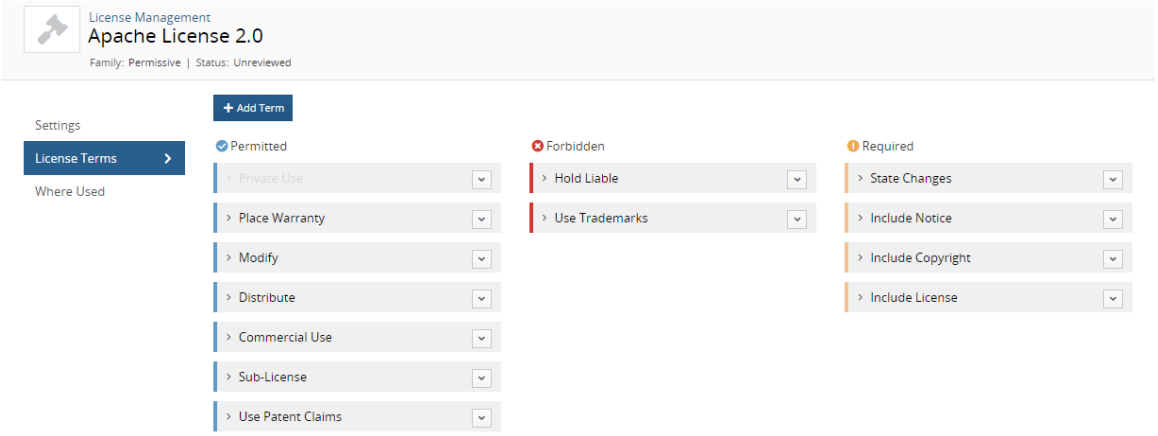
   The License Management page appears.



3. In the **Licenses** tab, select the license name to display the *License Name* **Settings** tab.

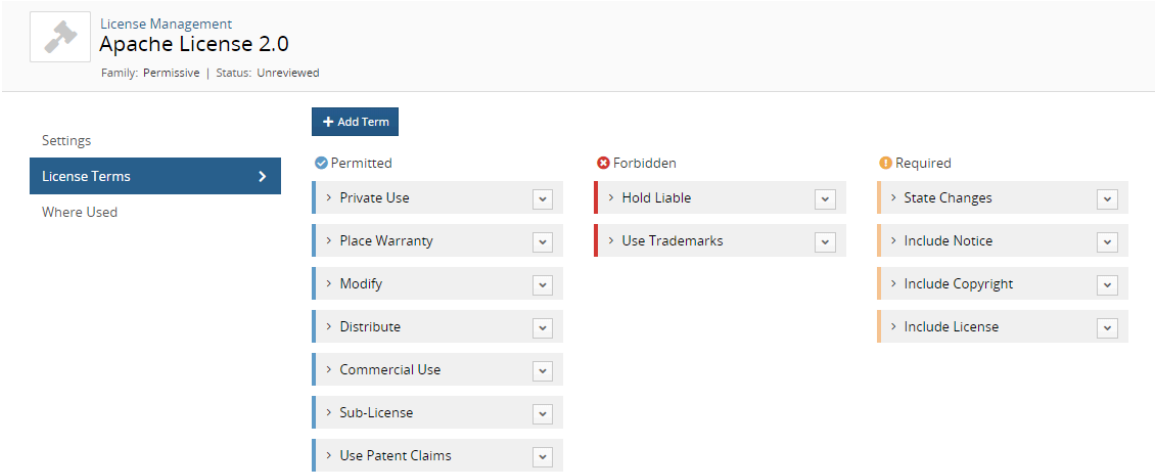


4. Select the **License Terms** tab to view the terms associated with this tab.



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.



KnowledgeBase licenses

Editing a KnowledgeBase license

KnowledgeBase licenses can be edited by users with the License Manager role and by users with the BOM Manager, 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 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:

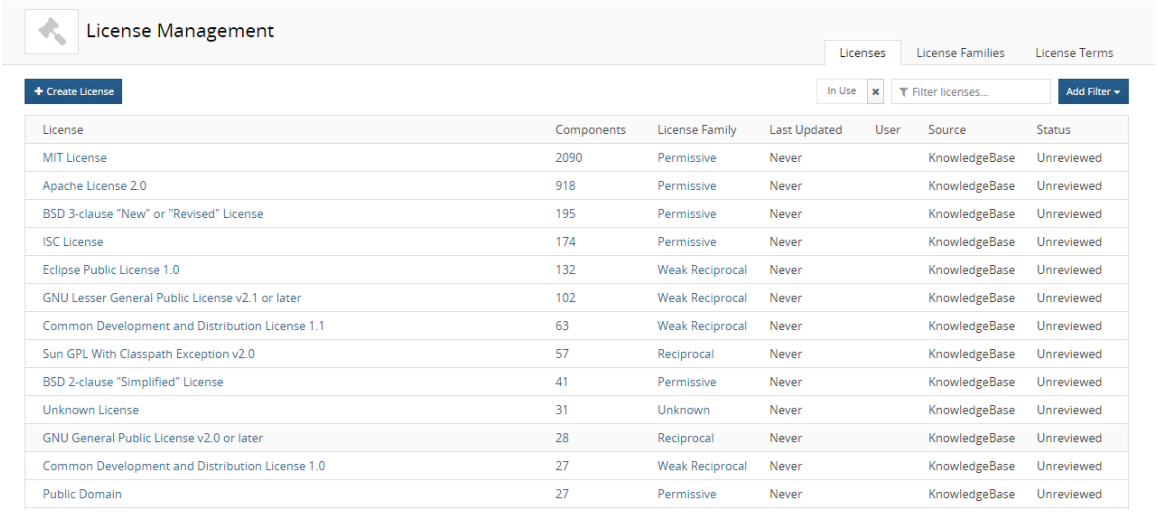
- Edits to the license family and license terms are always propagated to the KnowledgeBase licenses used in BOMs.
- Edits to the license text *may or may not* be propagated to the KnowledgeBase licenses used in BOMs:
  - If the BOM Manager/Project Manager *edited the license text*, the edits made by the License Manager *are not* propagated to the version of the KnowledgeBase license used in the BOM.
  - If the BOM Manager/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).

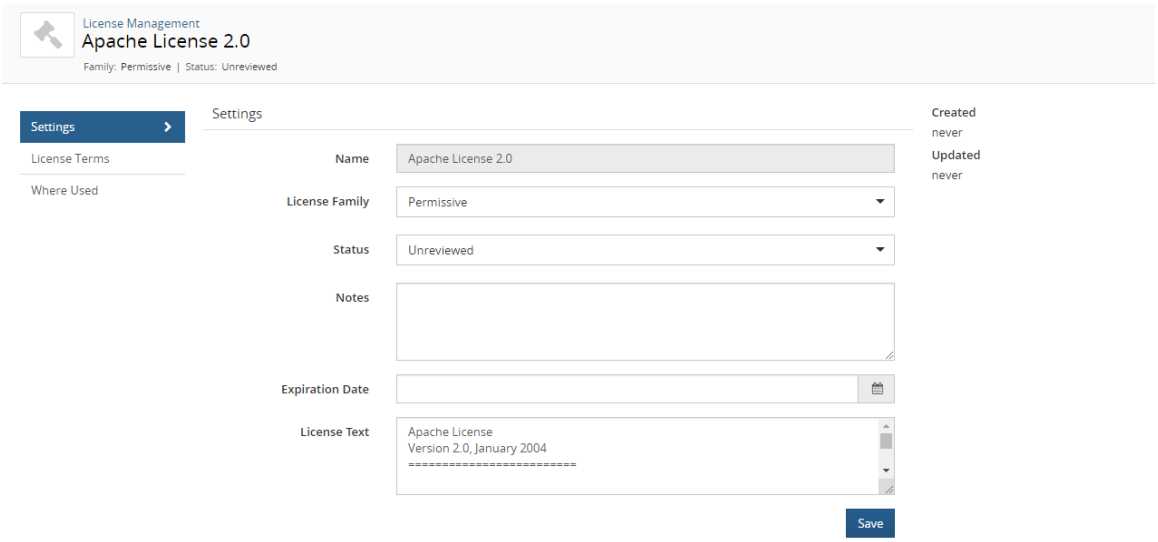
1. Log in to Black Duck with the License Manager role.
2. Click 

> **Licenses**.

The License Management page appears.



3. Select the KnowledgeBase license name to display the *License Name* **Settings** tab.



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**.

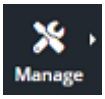
In the License Management page, the source for this license changes to **Modified KnowledgeBase** with the username of the user who edited this license listed in the **User** column and the time the license was modified listed in the **Last Updated** column.

KnowledgeBase licenses can be restored to their original values.

Restoring the original text and family of a KnowledgeBase license

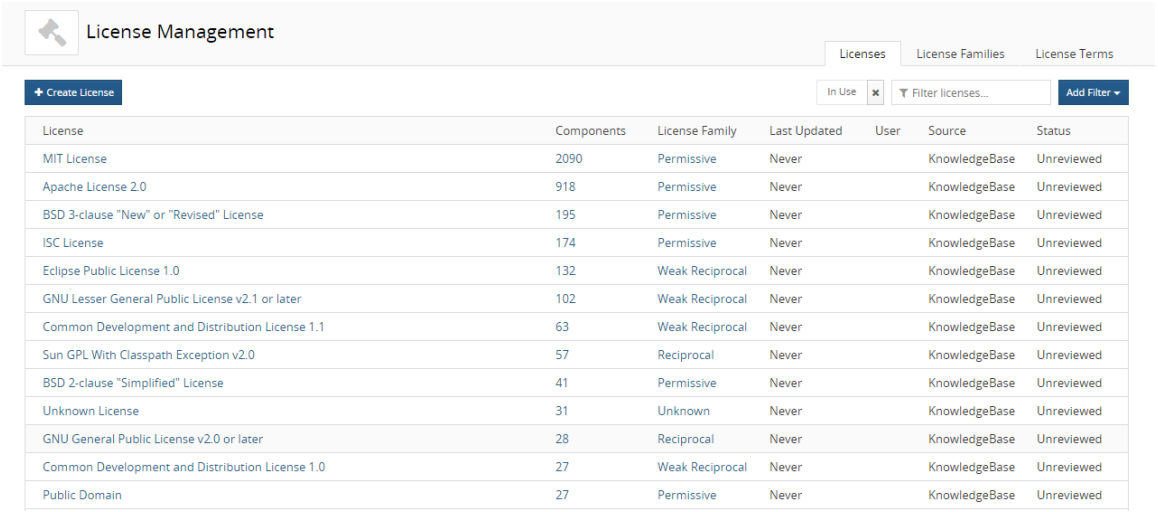
If a user with the License Manager role has modified the text or license family of a KnowledgeBase license, they can restore that license to its original values, as defined by the Black Duck KnowledgeBase.

To restore a KnowledgeBase license:

1. Log in to Black Duck with the License Manager role.
2. Click 

> **Licenses**.

   The License Management page appears.



License	Components	License Family	Last Updated	User	Source	Status
MIT License	2090	Permissive	Never		KnowledgeBase	Unreviewed
Apache License 2.0	918	Permissive	Never		KnowledgeBase	Unreviewed
BSD 3-clause "New" or "Revised" License	195	Permissive	Never		KnowledgeBase	Unreviewed
ISC License	174	Permissive	Never		KnowledgeBase	Unreviewed
Eclipse Public License 1.0	132	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
GNU Lesser General Public License v2.1 or later	102	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.1	63	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Sun GPL With Classpath Exception v2.0	57	Reciprocal	Never		KnowledgeBase	Unreviewed
BSD 2-clause "Simplified" License	41	Permissive	Never		KnowledgeBase	Unreviewed
Unknown License	31	Unknown	Never		KnowledgeBase	Unreviewed
GNU General Public License v2.0 or later	28	Reciprocal	Never		KnowledgeBase	Unreviewed
Common Development and Distribution License 1.0	27	Weak Reciprocal	Never		KnowledgeBase	Unreviewed
Public Domain	27	Permissive	Never		KnowledgeBase	Unreviewed

3. 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.

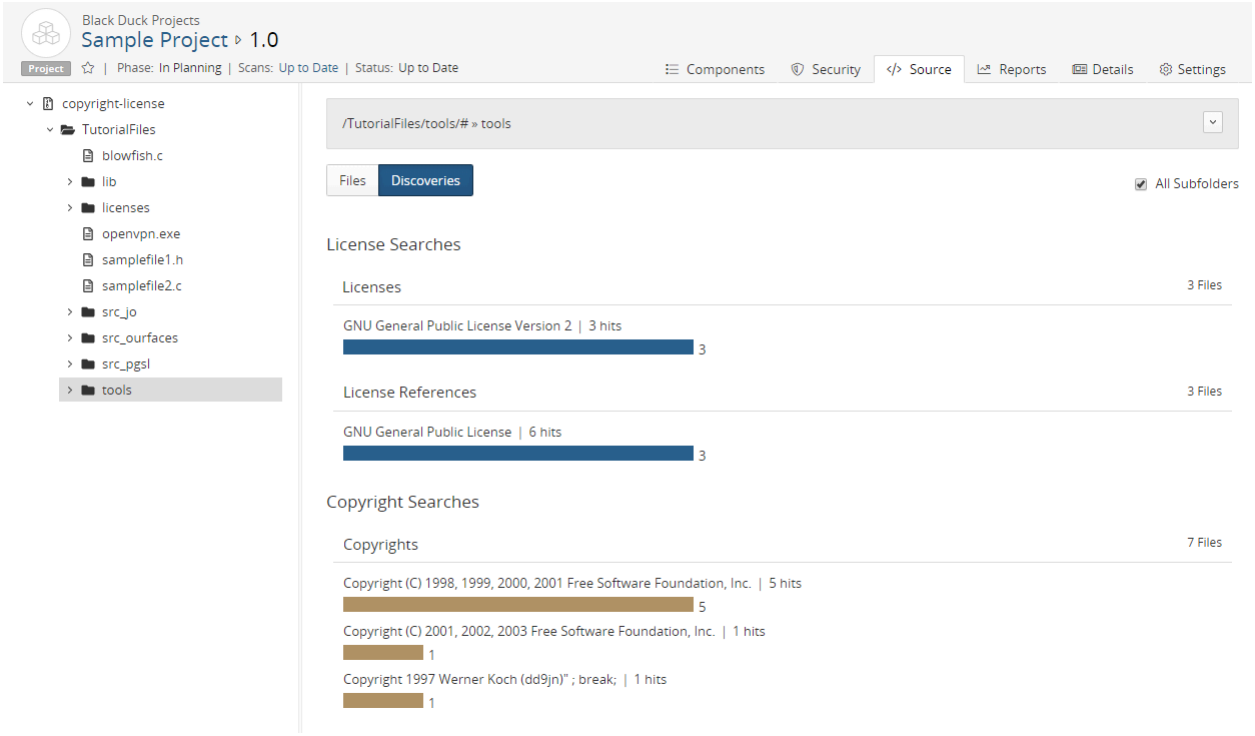
Note: This procedure does not restore the KnowledgeBase *license terms* to their original values. Click here for more information on restoring KnowledgeBase license terms.

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.



Black Duck groups the detected copyright statements into the **Copyright Searches** section.

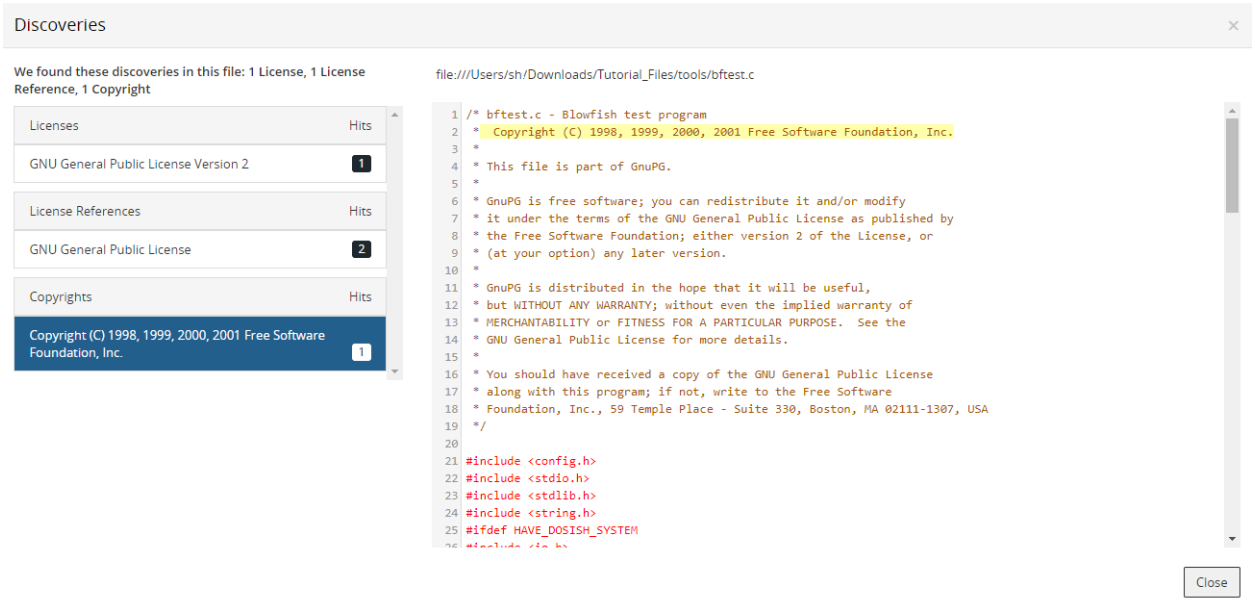
For the copyright text found, Black Duck displays the number of:

- "Hits". The number of instances that copyright text was found in all files.
- Files where these "hits" were found.

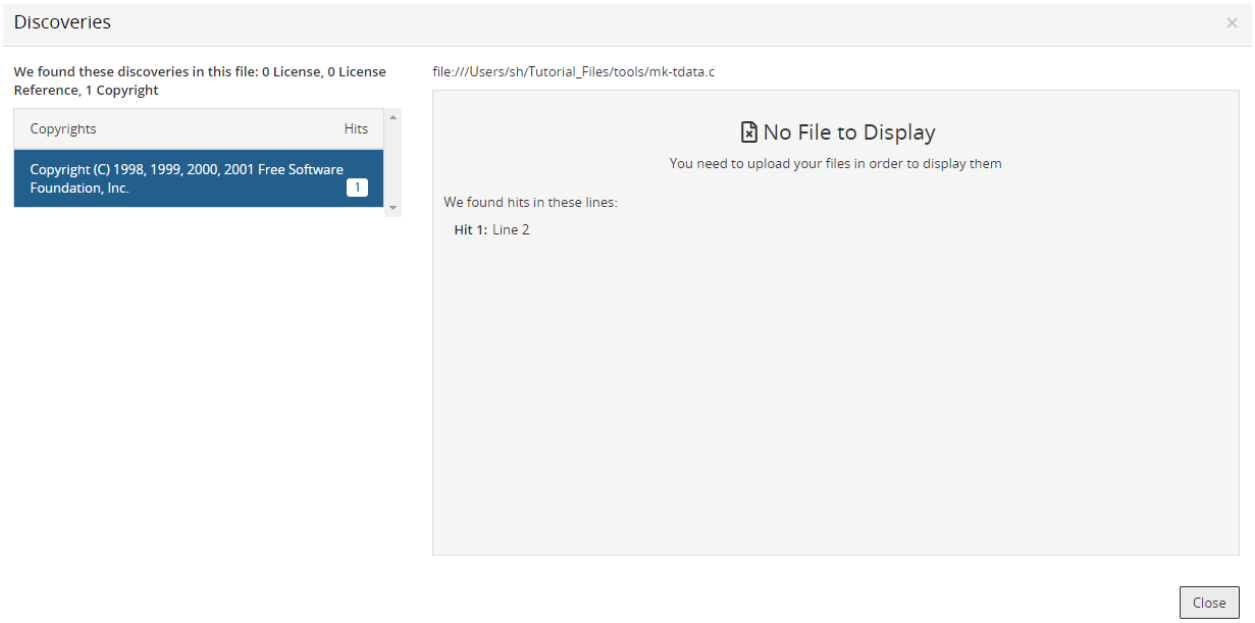
In the example shown above, there were three instances of copyright text found in seven files.

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.



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:



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:

- bdsl

- 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. The following scanning methods have an option to enable copyright string search:
   - Signature Scanner command line: Use the `-copyright-search` parameter.
   - Black Duck Detect Desktop: Enable the Signature Scanner Copyright Search option in **Scan Settings**.
   - Black Duck Detect: Use the `--detect.blackduck.signature.scanner.copyright.search=true` parameter.
2. Review the copyright text.

   Black Duck displays the location of these copyright statements in your code tree.

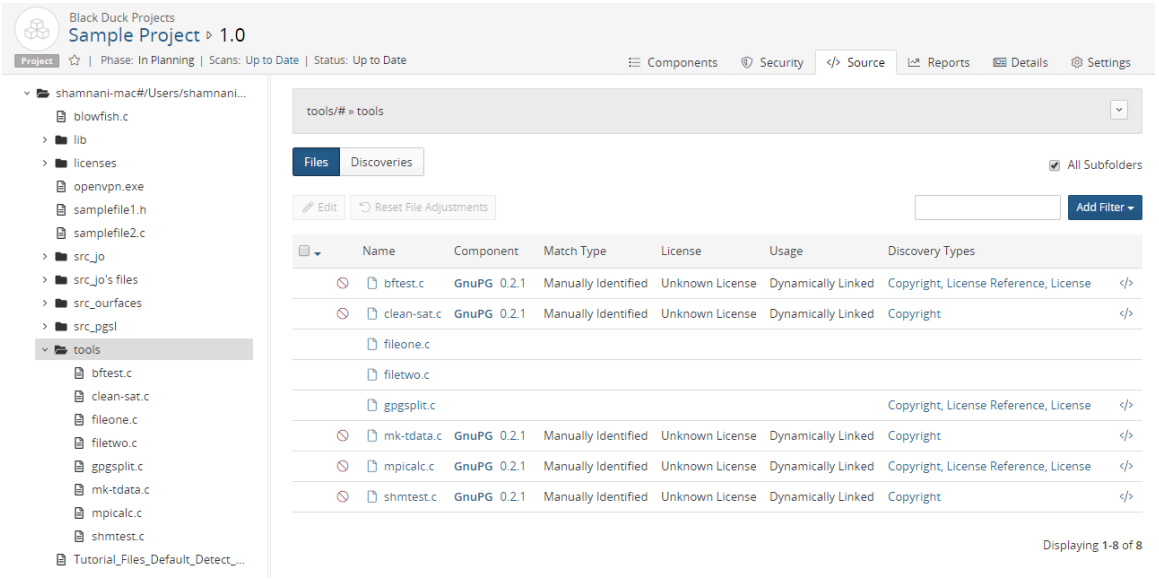
   To review copyright text:

   a. After enabling copyright text search, select the **Source** tab from your project version BOM page.

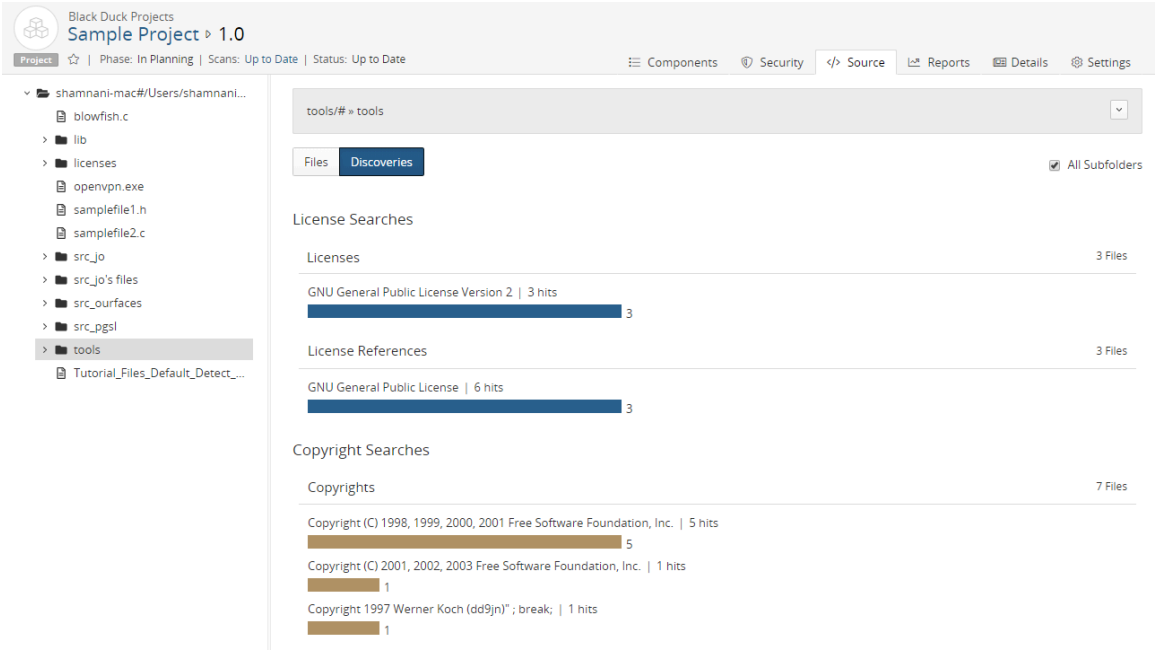
   b. Select a folder in the code tree that you want to determine if there is copyright text.

      Optionally, select **All Subfolders** to view information for all subfolders.

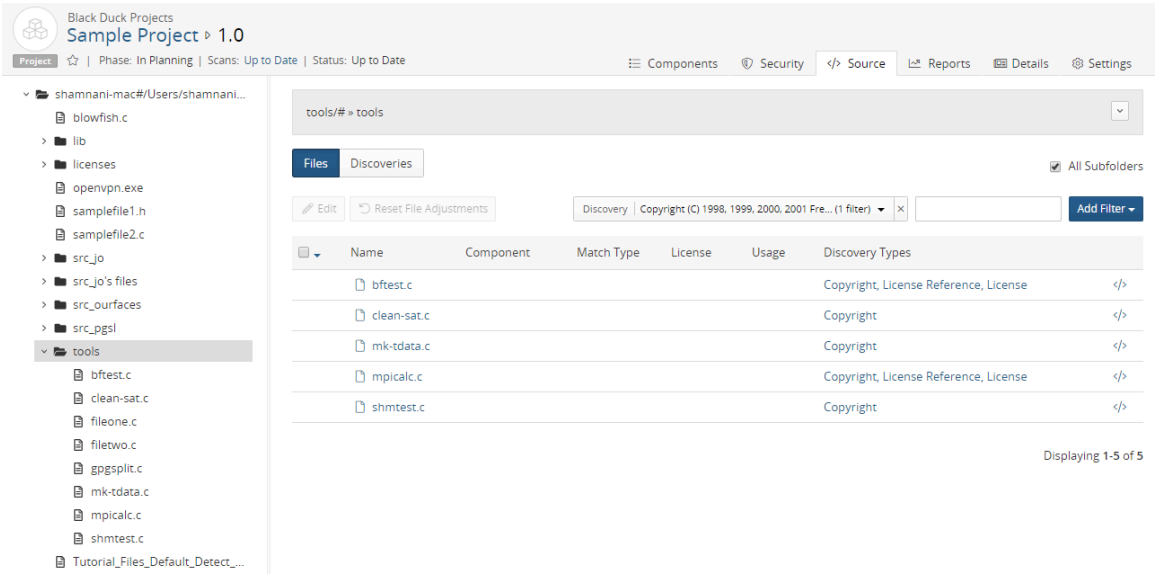
      The table displays information in the table for the selected location. By default the **Files** option is selected.



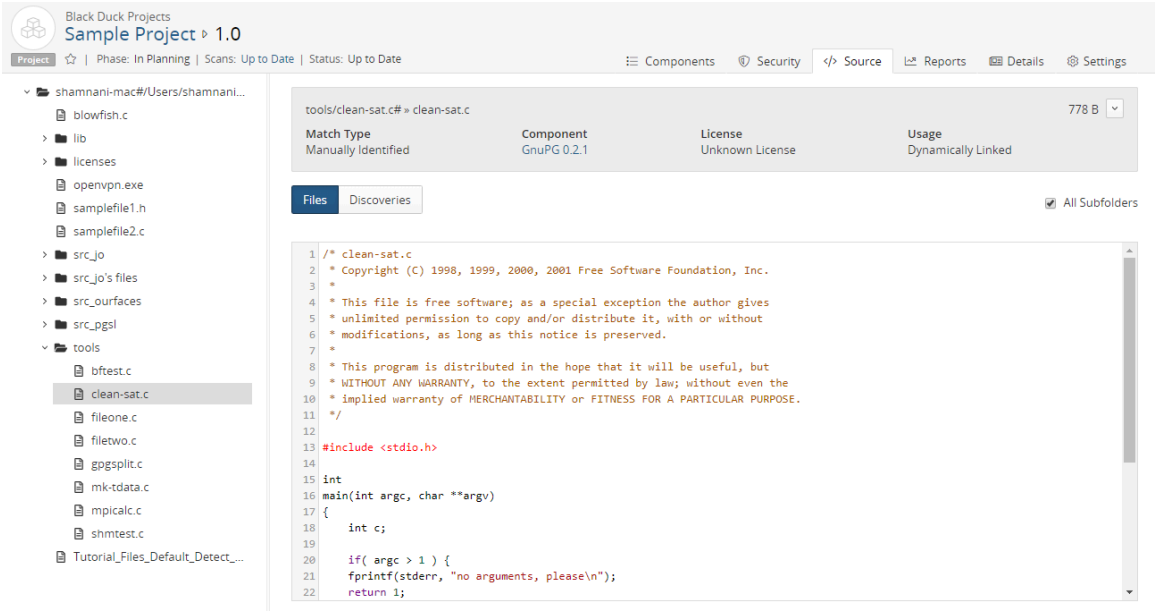
c. Select **Discoveries** to view the list of copyright text, shown in the **Copyright Searches** section.



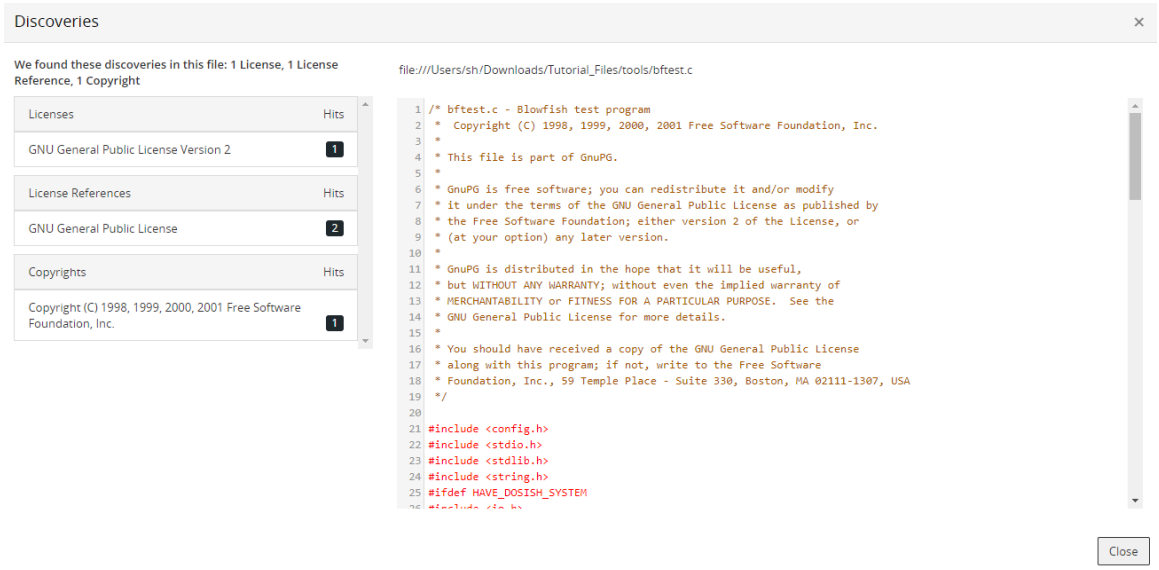
d. Select a copyright statement to view the **Source** tab filtered to display the files that contain the selected copyright text.



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.



e. Select **Copyright** from the **Discovery Type** column to open the Discoveries dialog box.

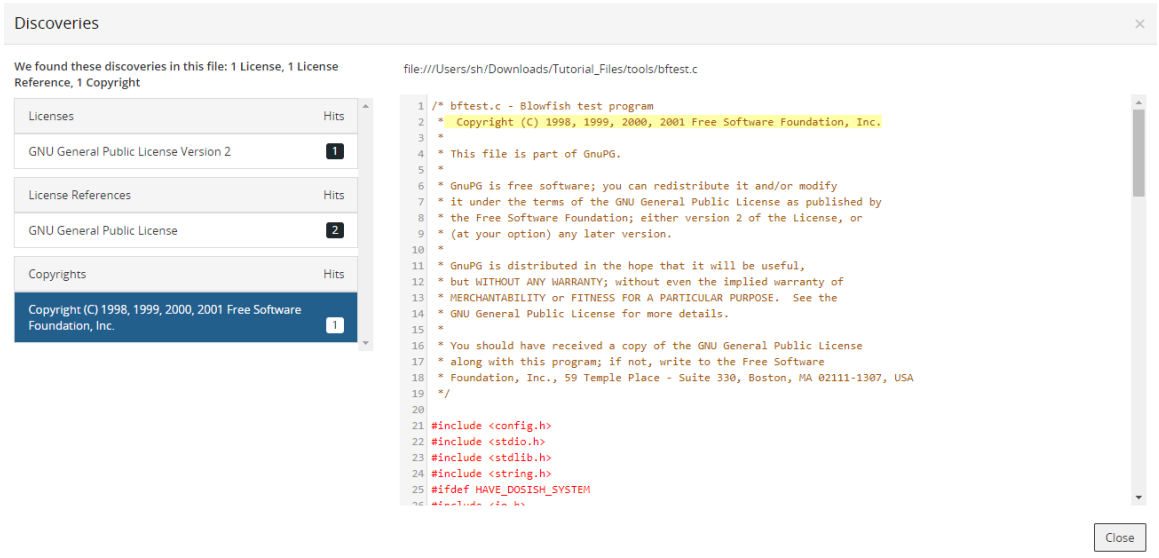


The Discoveries dialog box shows all copyright text found for the selected file. If embedded licenses and license references were also found, that text is also shown.

The information that appears here depends on whether you uploaded source files.

In the example shown above, source files were uploaded in the scan.

f. Select the copyright text to view the highlighted text.



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:

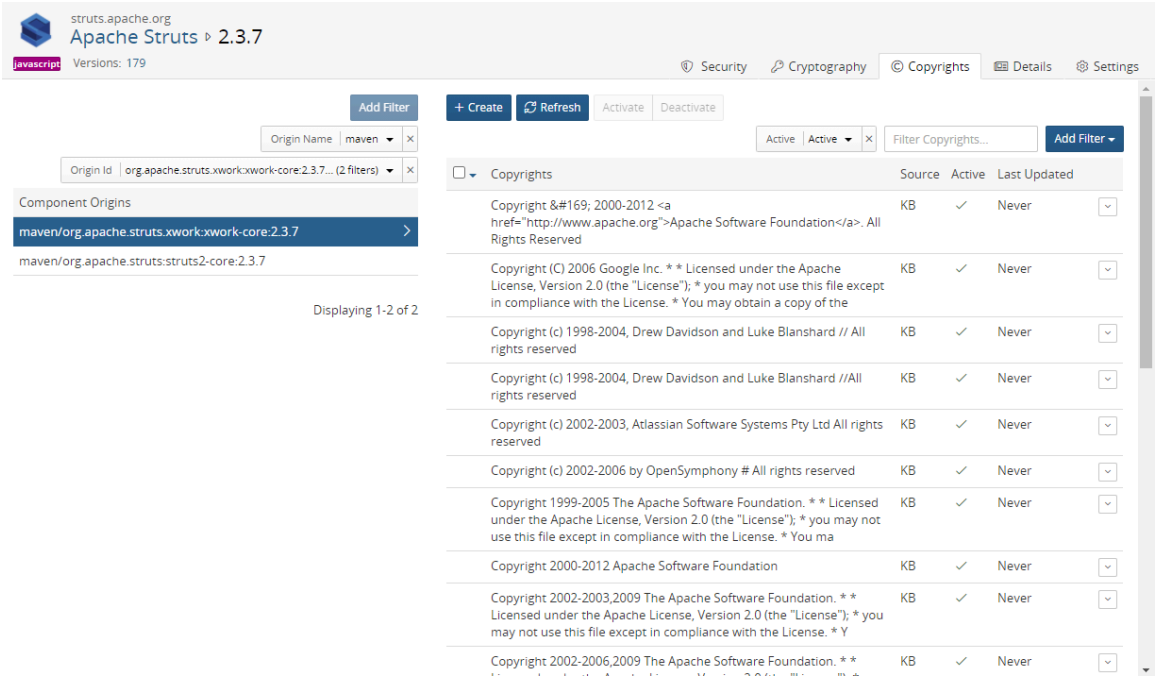
1. Review the existing Black Duck KnowledgeBase copyright statements.
2. If necessary, edit the existing KnowledgeBase copyright statements and/or create custom copyright statements.
3. Deactivate any copyright statements that do not apply.
4. Create the Notices file report and select the **Copyright 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:

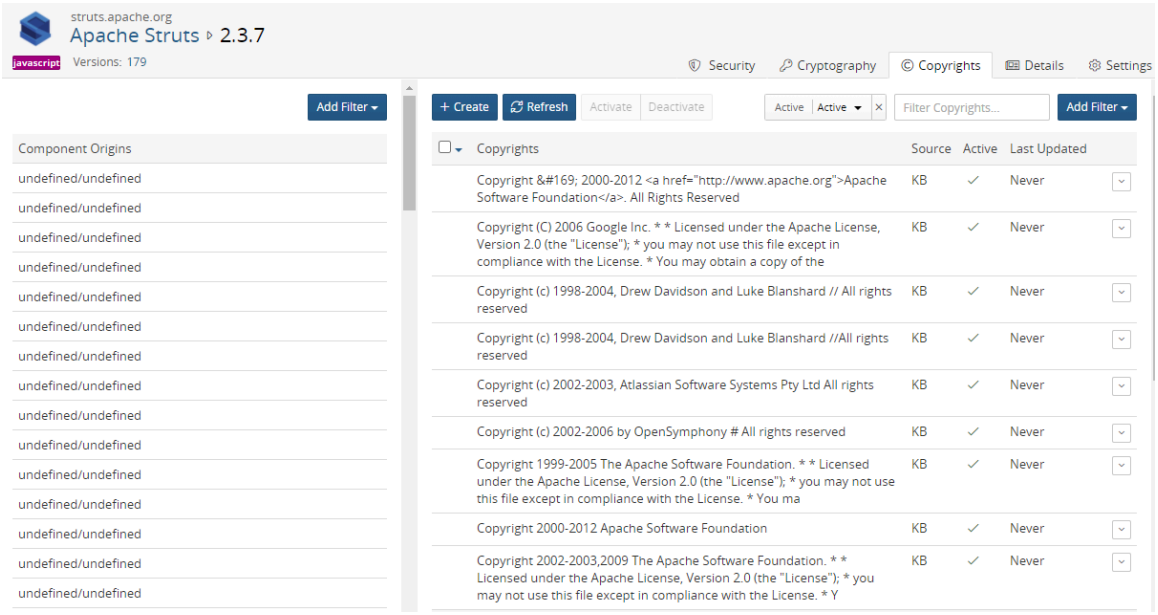
- In the project version BOM, click in the row of the component version you wish to view copyright statements and select **Copyrights**.

  The component version **Copyrights** tab appears filtered to display the copyright statements for the origin used in your BOM.



  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.



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:

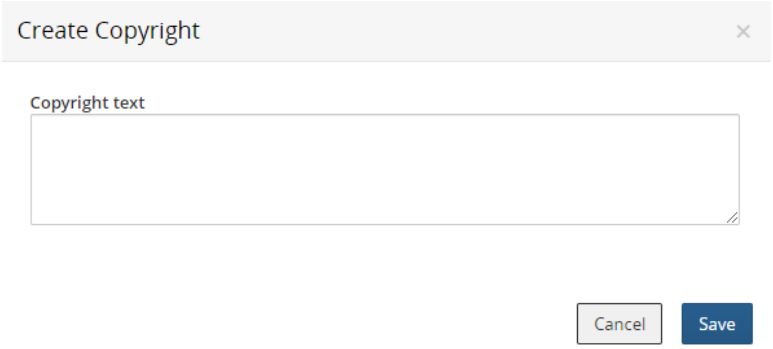
Column	Description
Copyrights	Copyright text.
Source	Source for this copyright statement. Possible values are: • **KB**. An unmodified, active copyright statement from Black Duck KnowledgeBase. • **KB Modfied**. A copyright statement from Black Duck KnowledgeBase that has been edited, deactivated, or reactivated. • **Custom**. Copyright statement created by a user with the Copyright Editor role.
Active	One of the following icons appears: • ✓ 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: • **Never** indicates that the statement from Black Duck KnowledgeBase has never been modified. • Date and username. • 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

Column	Description
	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:

1. As copyright statements are based by component origin, select the origin for this copyright statement from the **Component Origins** section.
2. Click **Create**. The Create Copyright dialog box appears.



3. Enter the copyright text and click **Save**.

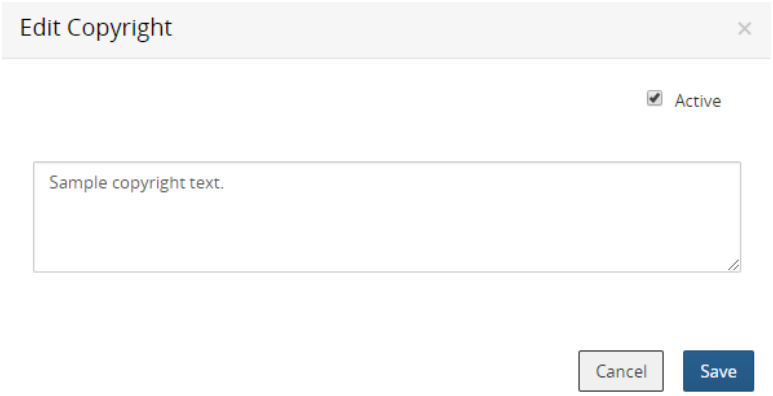
Copyright statements are active by default. See below to deactivate this statement.

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**.

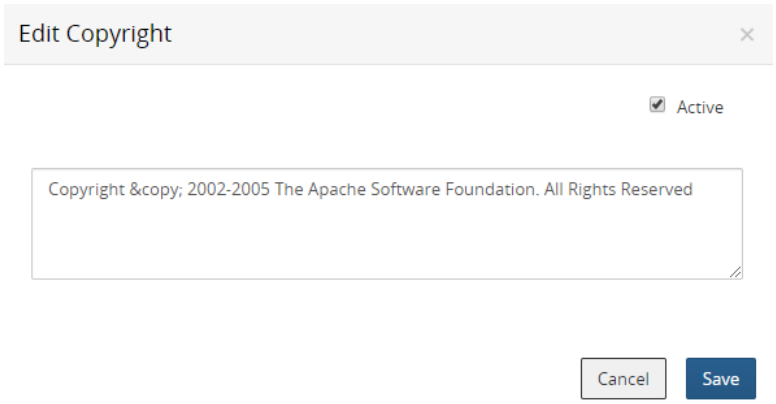
   You can also activate a copyright statement when editing it.

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**.

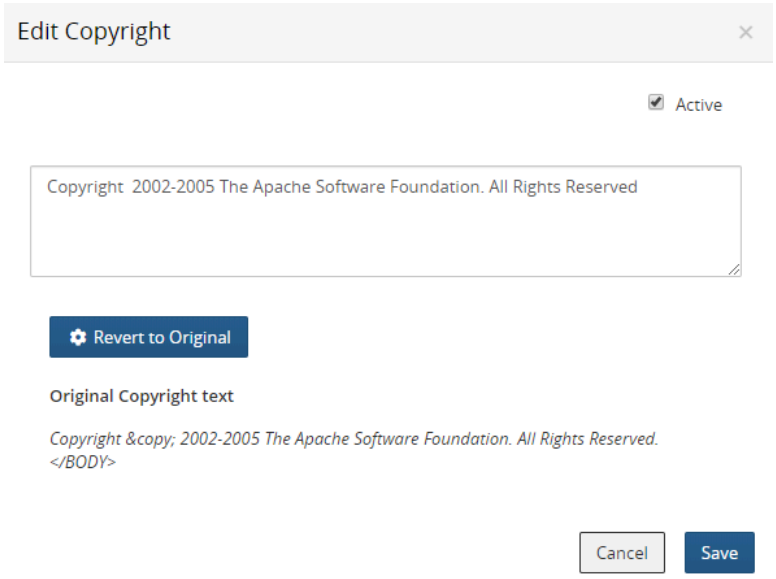
Reverting KnowledgeBase copyright statements

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:

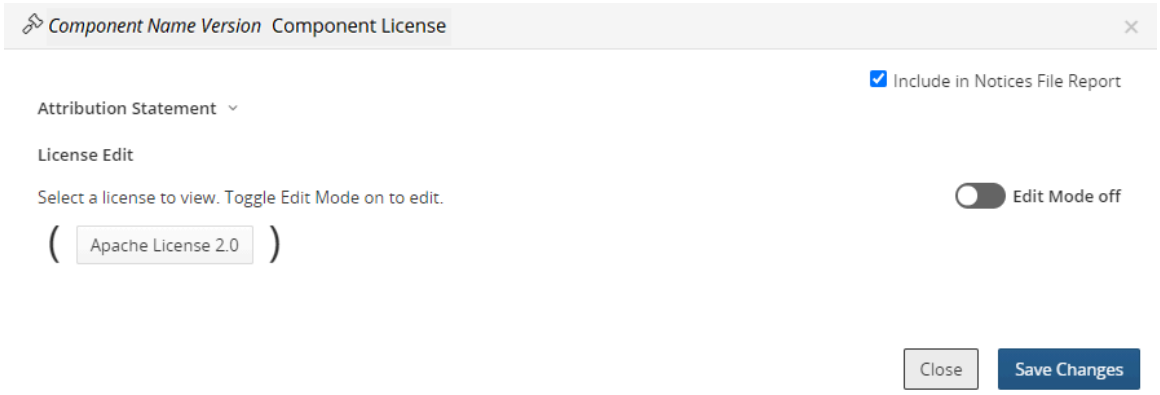
1. Open the **Copyrights** tab, as described previously.
2. Select a component origin.
3. Click **Refresh**.

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 SCA.
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.

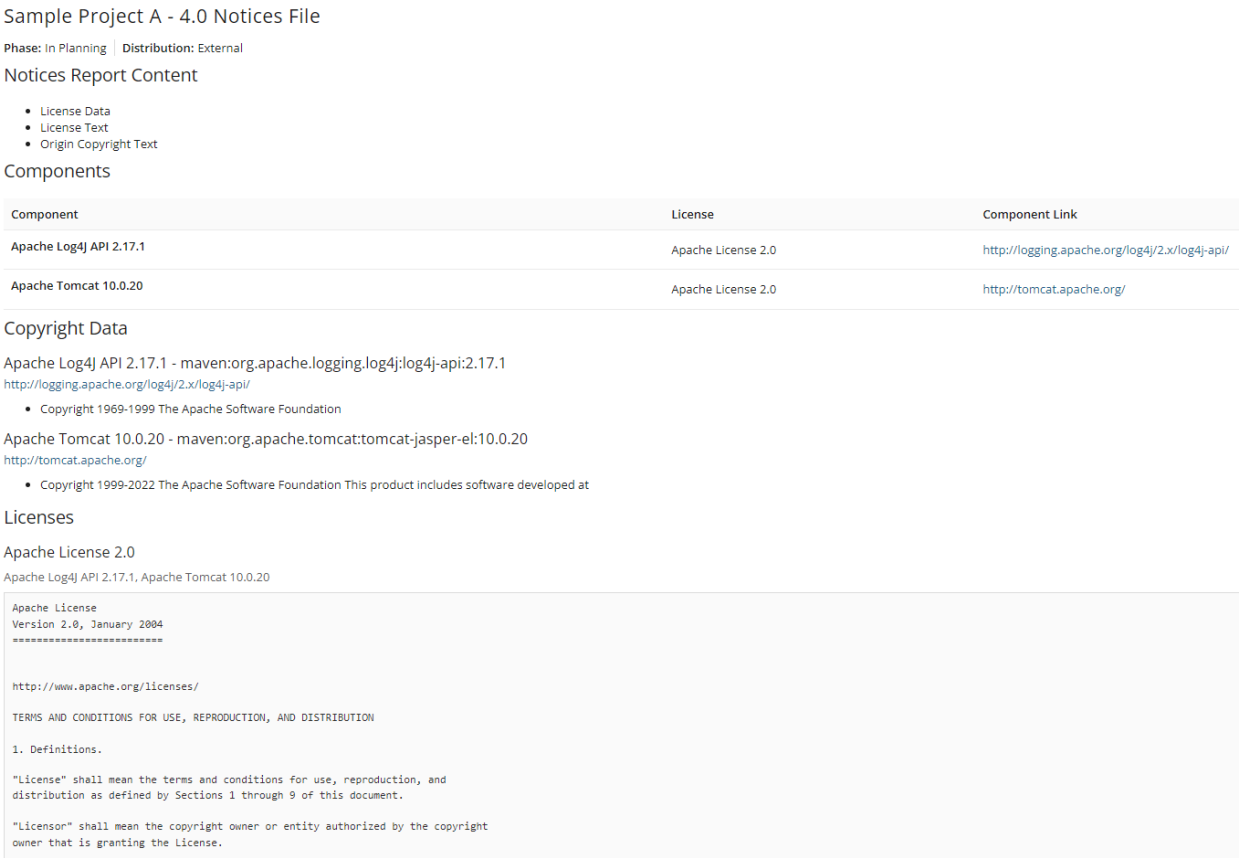


5. Click > to open the **Attribution Statement** field and enter the text.

   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 A - 4.0 Notices File

Phase: In Planning | **Distribution:** External

Notices Report Content

- License Data
- License Text
- Origin Copyright Text

Components

Component	License	Component Link
Apache Log4J API 2.17.1	Apache License 2.0	http://logging.apache.org/log4j/2.x/log4j-api/
Apache Tomcat 10.0.20	Apache License 2.0	http://tomcat.apache.org/

Copyright Data

Apache Log4J API 2.17.1 - maven:org.apache.logging.log4j:log4j-api:2.17.1
http://logging.apache.org/log4j/2.x/log4j-api/

- Copyright 1969-1999 The Apache Software Foundation

Apache Tomcat 10.0.20 - maven:org.apache.tomcat:tomcat-jasper-el:10.0.20
http://tomcat.apache.org/

- Copyright 1999-2022 The Apache Software Foundation This product includes software developed at

Licenses

Apache License 2.0

Apache Log4J API 2.17.1, Apache Tomcat 10.0.20

```
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

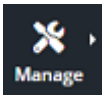
You can define incompatible terms for your custom license terms with a forbidden or required responsibility, including deprecated custom license terms.

- 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

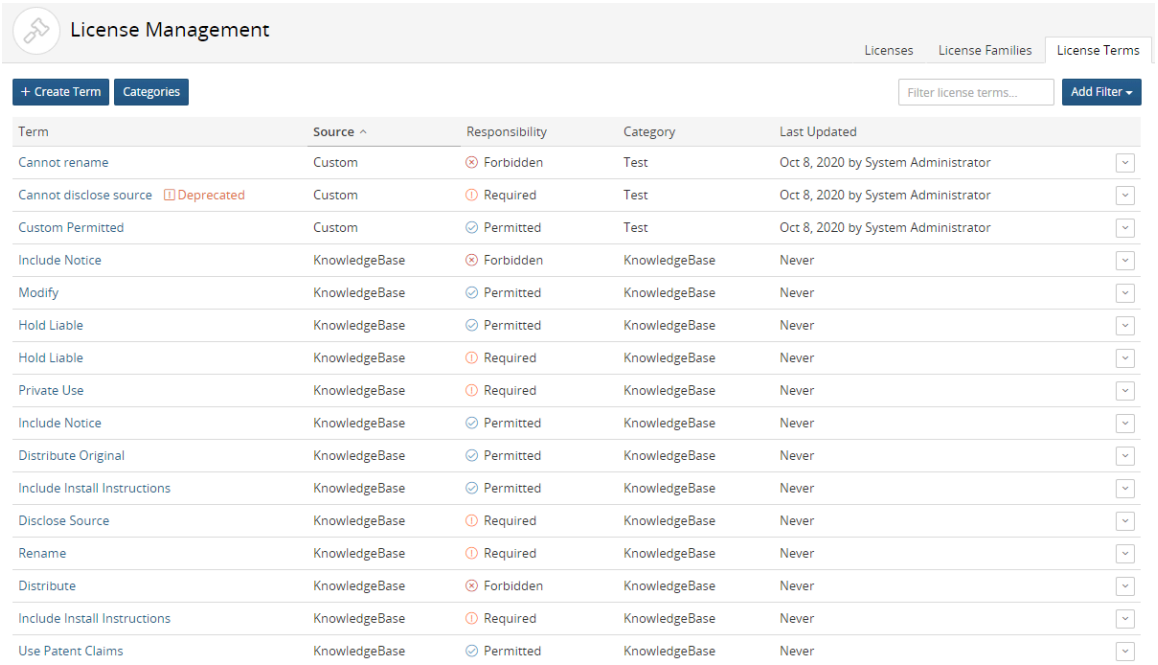
To define an incompatible term:

1. Log in to Black Duck with the License Manager role.
2. Click 

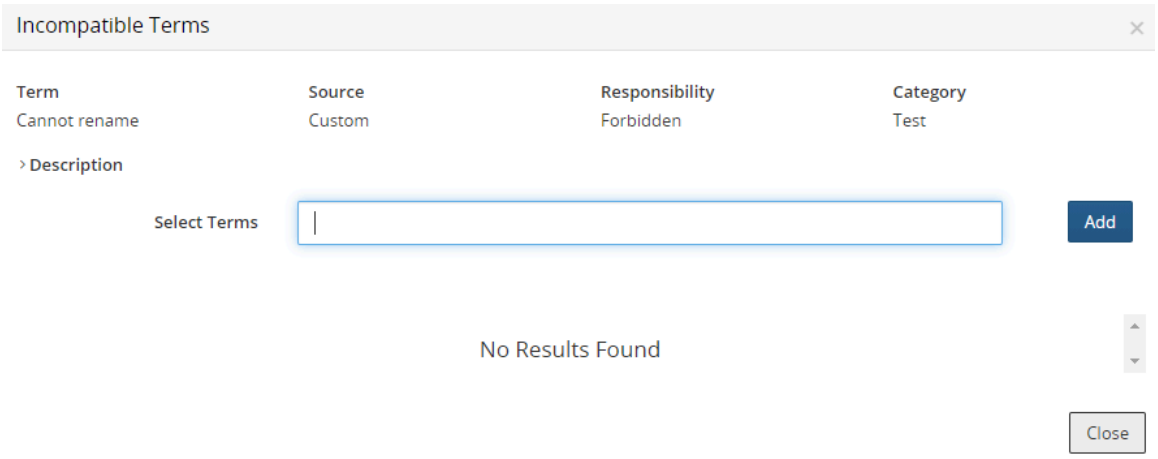
> **Licenses**.

   The License Management page appears.

   Select the **License Terms** tab to display all license terms.



3. Click in the row of the custom license term and select **Incompatible Terms** to open the Incompatible Terms dialog box.



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

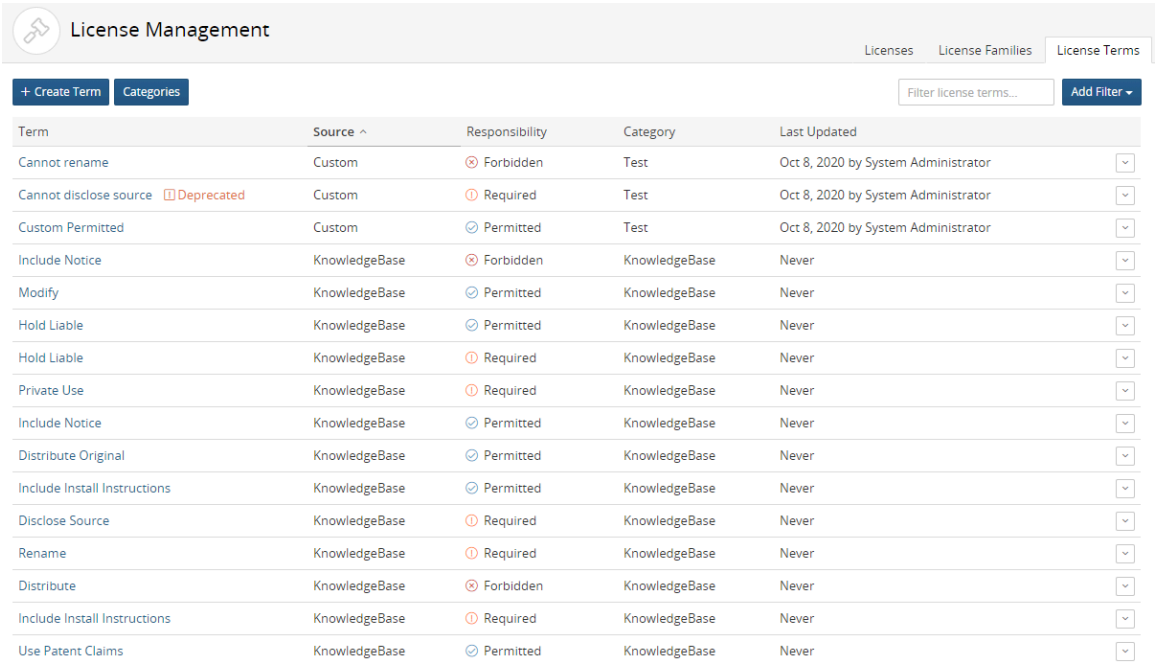
1. Log in to Black Duck with the License Manager role.

2. Click 

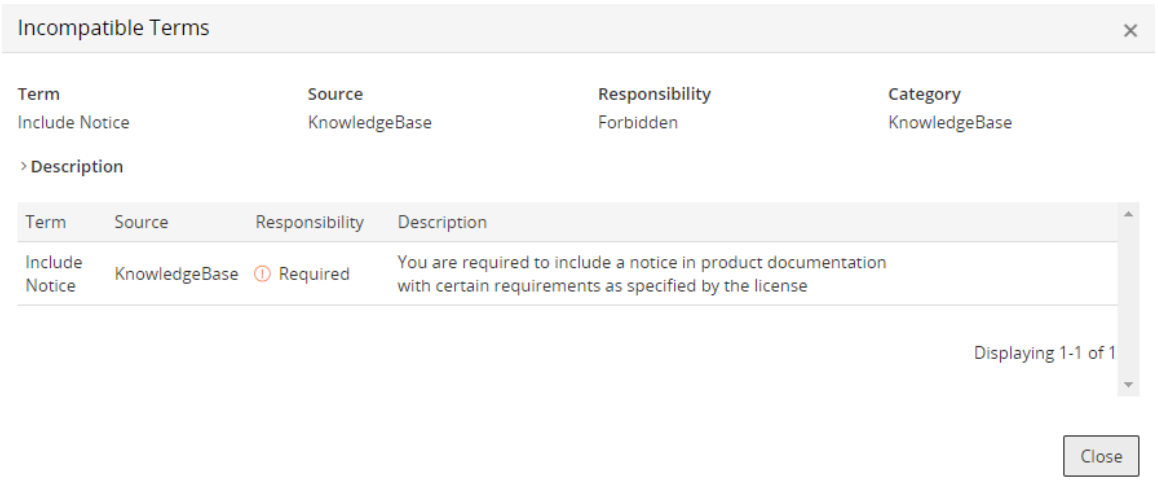
> **Licenses**.

The License Management page appears.

Select the **License Terms** tab to display all license terms.



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.



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.

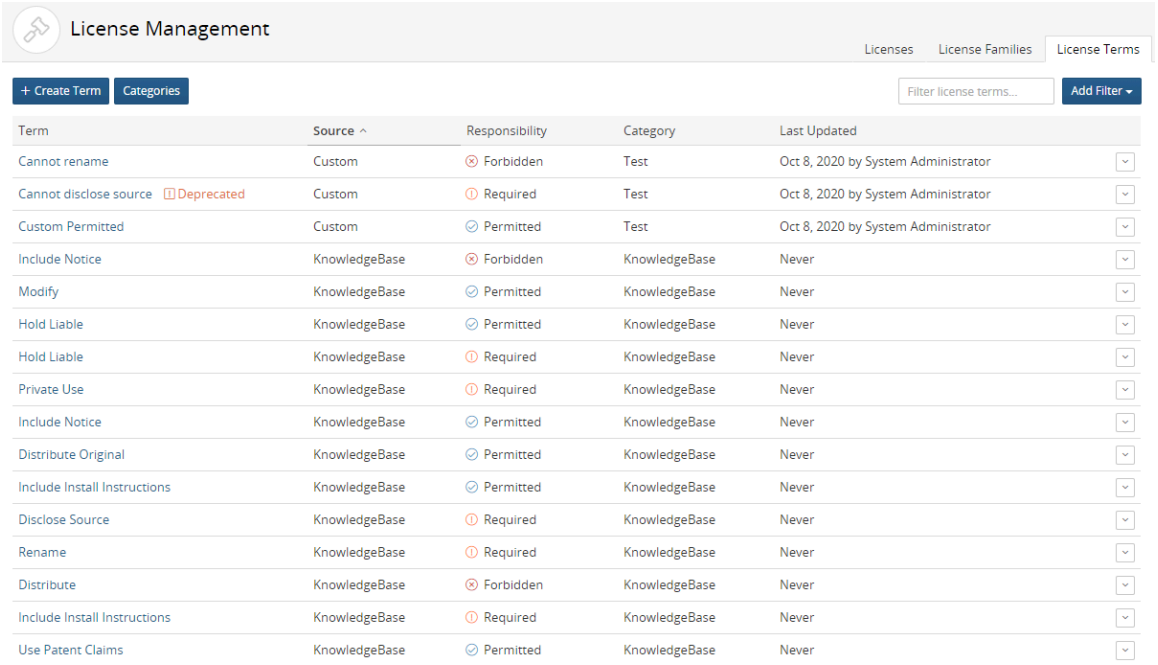
Deleting incompatible license terms

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 > **Licenses**.

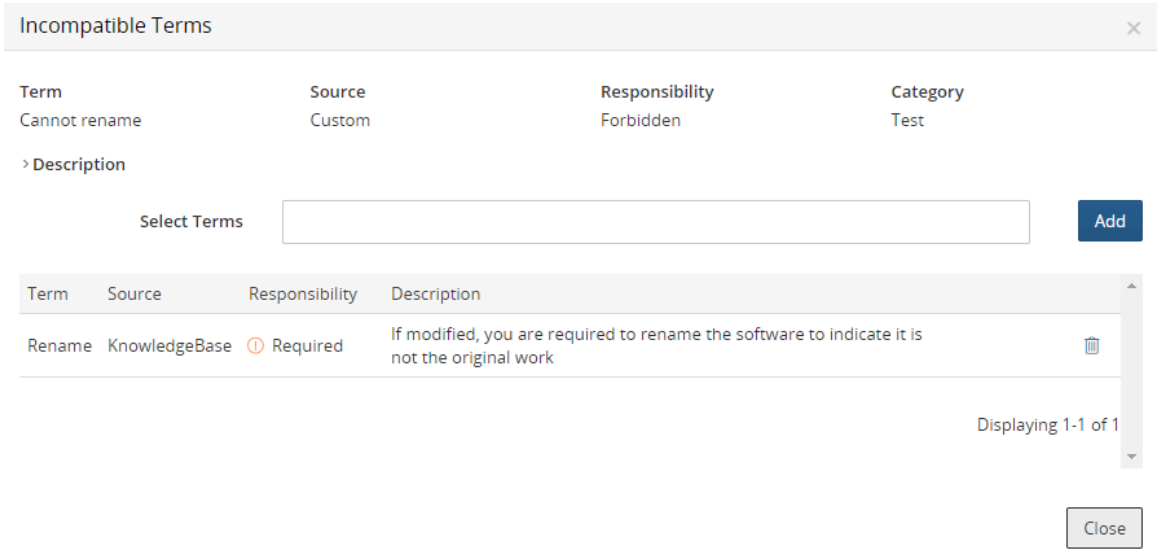
   The License Management page appears.

   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 Deprecated	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.



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

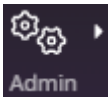
BOM Managers, and other users with the appropriate role, can view conflicts for a license term using the **License Conflicts** tab in the *Project Version's* **Legal** tab.

By default, these tabs are disabled. To enable this feature:

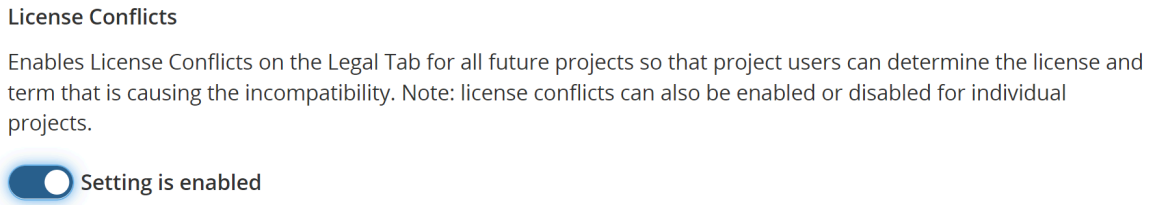
1. Use the System Setting page to enable the feature for all *future* projects.
2. Once future projects are enabled, use a project's **Setting** tab to enable the feature for a *current* project.

To enable license conflicts for future projects:

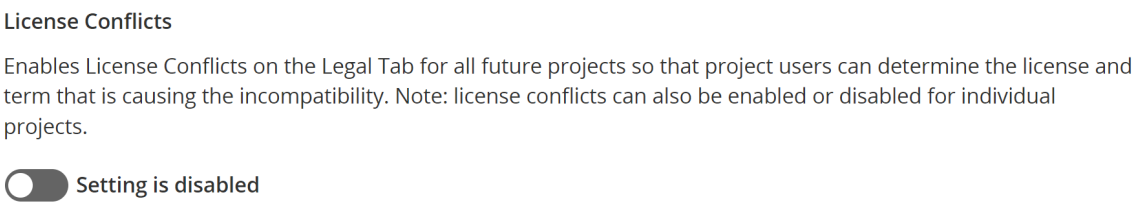
Use the System Settings page to enable the **Legal** and **License Conflicts** tabs for all future projects.

1. Log into Black Duck with the System Administrator role.
2. Click 

.
3. Select **System Settings**.
4. Click **Legal**.
5. Set the toggle located in the **License Conflicts** section to *Setting is enabled* to display the **Legal** and **License Conflicts** tabs. Enabling the setting will not take effect immediately for existing projects.



Set the toggle located in the **License Conflicts** section to *Setting is disabled* 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.

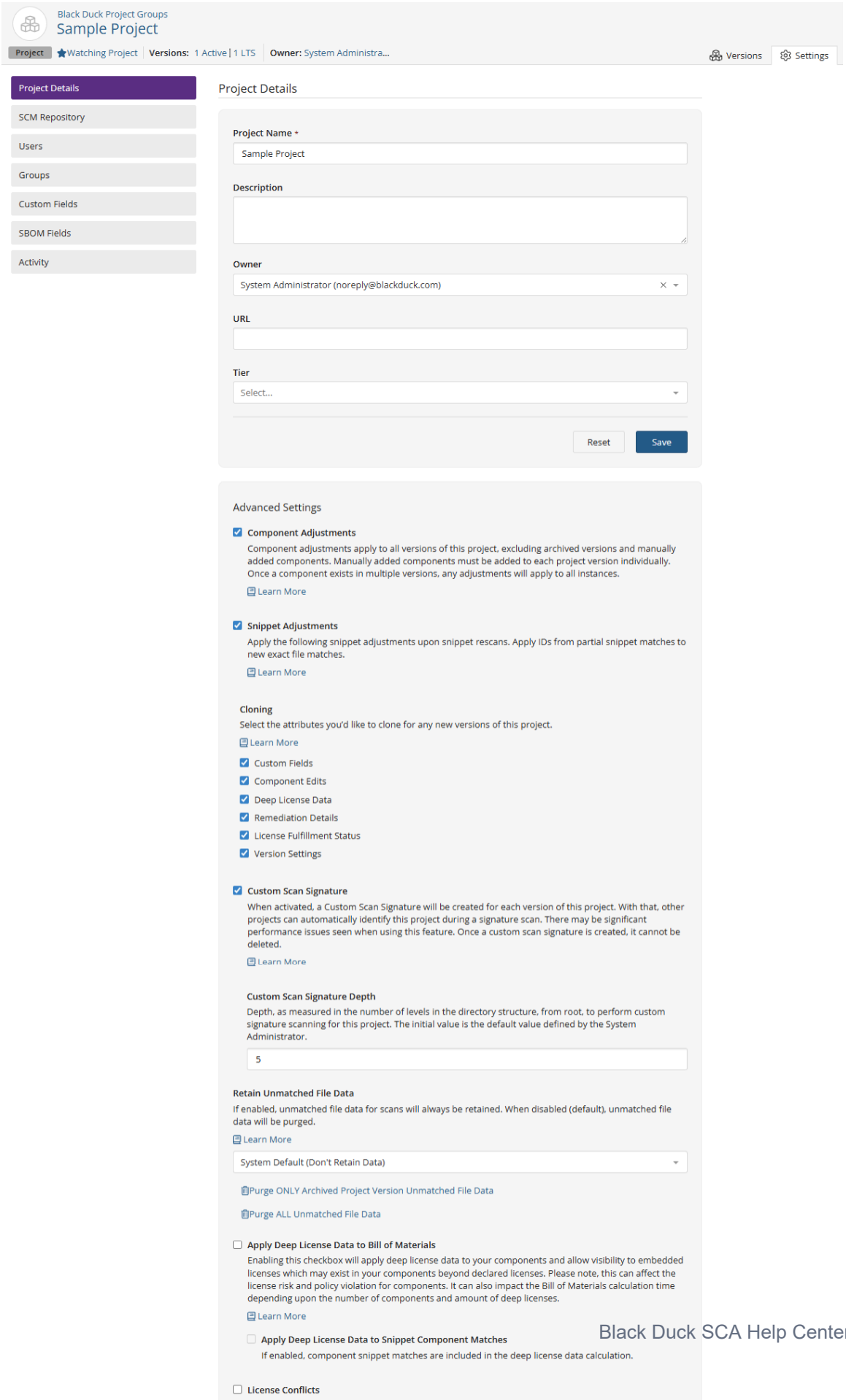


Enabling or disabling license conflicts for a specific project

If the system setting is not enabled prior to the creation of the Black Duck project, then the setting must be set at the project level.

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.


Black Duck Project Groups
Sample Project
Project
Watching Project
Versions: 1 Active | 1 LTS
Owner: System Administra...
Versions
Settings
Project Details
SCM Repository
Users
Groups
Custom Fields
SBOM Fields
Activity
Project Details
Project Name *
Sample Project
Description
Owner
System Administrator (noreply@blackduck.com)
URL
Tier
Select...
Reset
Save
Advanced Settings
Component Adjustments
Component adjustments apply to all versions of this project, excluding archived versions and manually added components. Manually added components must be added to each project version individually. Once a component exists in multiple versions, any adjustments will apply to all instances.
Learn More
Snippet Adjustments
Apply the following snippet adjustments upon snippet rescans. Apply IDs from partial snippet matches to new exact file matches.
Learn More
Cloning
Select the attributes you'd like to clone for any new versions of this project.
Learn More
Custom Fields
Component Edits
Deep License Data
Remediation Details
License Fulfillment Status
Version Settings
Custom Scan Signature
When activated, a Custom Scan Signature will be created for each version of this project. With that, other projects can automatically identify this project during a signature scan. There may be significant performance issues seen when using this feature. Once a custom scan signature is created, it cannot be deleted.
Learn More
Custom Scan Signature Depth
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.
5
Retain Unmatched File Data
If enabled, unmatched file data for scans will always be retained. When disabled (default), unmatched file data will be purged.
Learn More
System Default (Don't Retain Data)
Purge ONLY Archived Project Version Unmatched File Data
Purge ALL Unmatched File Data
Apply Deep License Data to Bill of Materials
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.
Learn More
Apply Deep License Data to Snippet Component Matches
If enabled, component snippet matches are included in the deep license data calculation.
License Conflicts

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 or subproject 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.

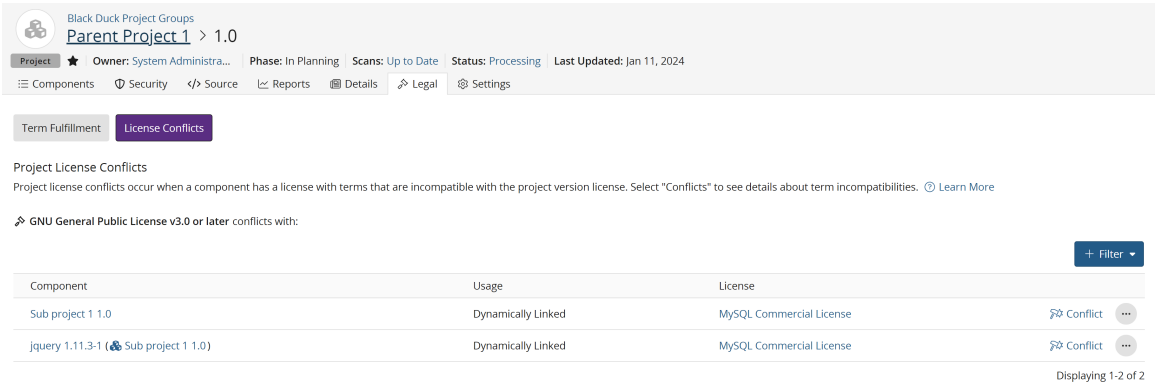
Note the following:

- 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 subprojects and 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.

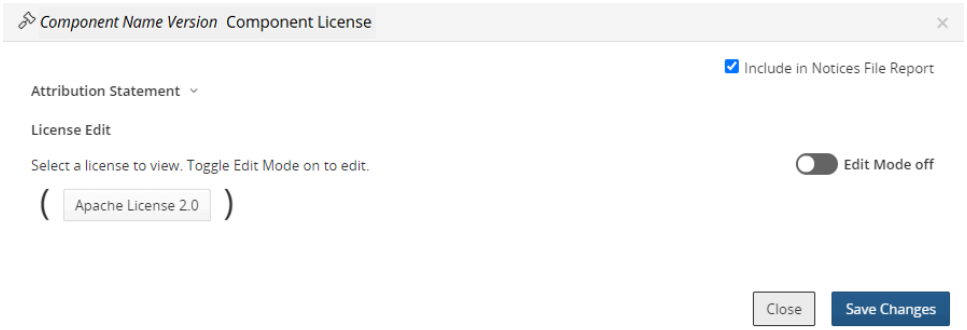
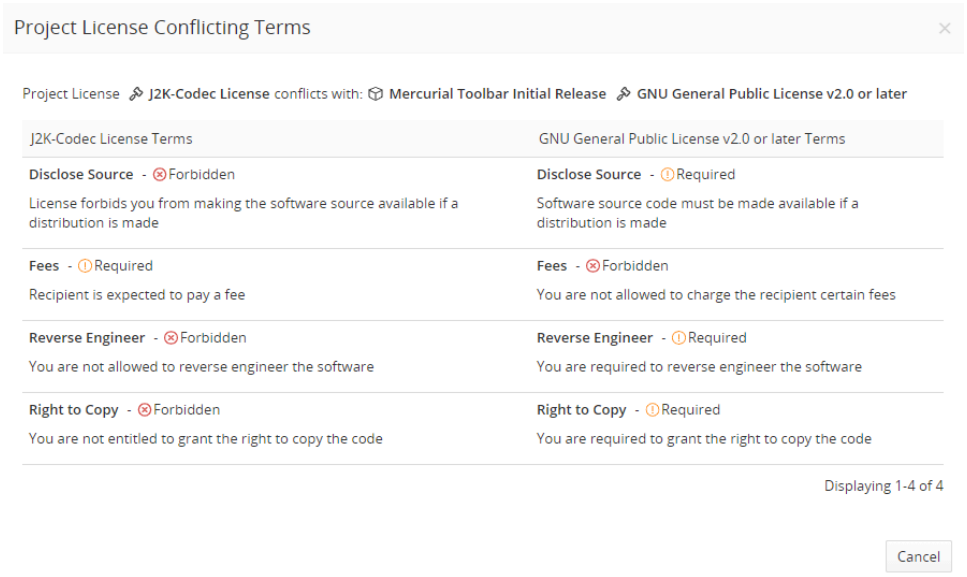
  Manual edits to a BOM, including changing the usage for a component or the license of the project version using the **License Conflict** or **Components** tab will trigger a recalculation of the license conflict.
- License conflicts for snippets are not shown until the snippet is confirmed.
- You can create a policy rule that is triggered when a component's license conflicts with the license for a project version.

Viewing project license conflicts

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.

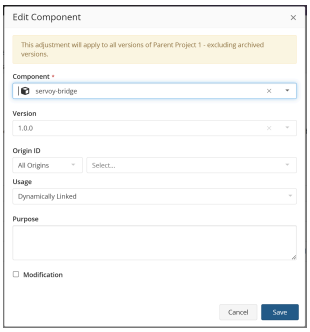


The table displays the following information:

Column	Description
⦸	Policy violation. 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 or subproject name. If a license conflict is detected in a subproject of this project, the component name will be displayed along with the subproject in parentheses.
Usage	Indicates the usage of this component.
License	The license for this component. Select the license name to open the *Component Name Version* Component License dialog box.  Use this dialog box to edit the existing license(s), view obligations, and view/edit the license text.
Conflict	Indicates there is a conflict. Select **Conflict** to open the Project License Conflicting Terms dialog box.  Use this dialog box to view the list of project license terms and conflicting component version license terms.

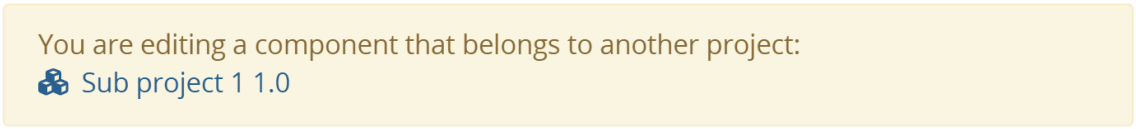
Editing a component or subproject with license conflicts

You can edit the component or subproject by clicking in the row of the component and select **Edit** to open the Edit Component modal.



If the component or subproject is found in the parent project itself, a warning at the top of the Edie Component modal notifying you that any changes made to the Component details will be reflected in all versions of the project it is located in. Once you have made the desired updates, click the **Save** button to accept your changes.

If you are editing a component found in a subproject, a warning will appear at the top of the modal notifying you that you are editing a component belonging to another project along with the project's name and version. Clicking the project name will take you to that project version's Details page.



Adding comments to a component or subproject with license conflicts

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.

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.

Policies in Black Duck SCA are essential for ensuring that your projects adhere to defined standards regarding components, licenses, and vulnerabilities. This document outlines how to manage policies effectively across different scanning scenarios.

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.

Policy applicability by scan mode and deployment

The policy engine behaves the same in both Hosted (SaaS) and On-Premises deployments. There is no difference in how policies are evaluated based on the deployment model.

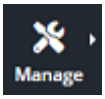
How policies are evaluated depends on the scan mode:

Scan Mode	Policy Evaluation Behavior
Detect Full Scan (Online)	All enabled Full Scan policies are evaluated after the scan is processed by the Black Duck SCA server. Policy violations are stored in the project version, the BOM is updated, and results are available in reports, dashboards, notifications, and the UI.
Detect Offline Scan	Policies are *not* evaluated while the scan is generated offline. Evaluation occurs only after the scan data is uploaded to a connected Black Duck SCA server and processed — at which point violations are calculated the same as for an online Full Scan.
Detect Rapid Scan	Only policies enabled for Rapid Scan are evaluated. Rapid Scan returns immediate policy results to the user or CI/CD pipeline but does not create or update a project BOM or project version. Workflows that

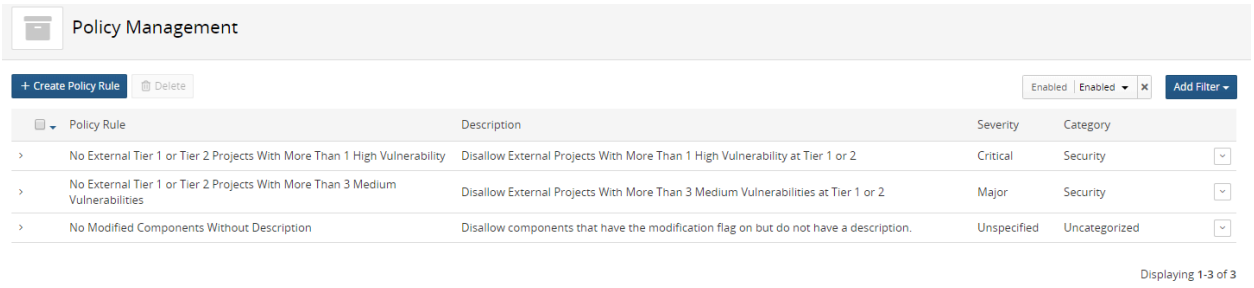
Scan Mode	Policy Evaluation Behavior
	rely on a persisted BOM (reporting, notifications, dashboards, project history) do not apply.

Tip: When creating a policy, ensure that it is enabled for the appropriate scan mode (Full Scan, Rapid Scan, or both), depending on where you want the policy to be evaluated.

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 

and selecting **Policies**:



- 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

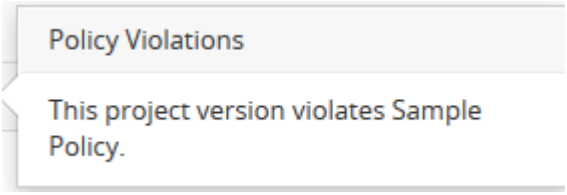
When a component is in violation of a policy rule, the Policy Violation icon (⦸) appears in the UI on the following pages:

- 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 Tree 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.

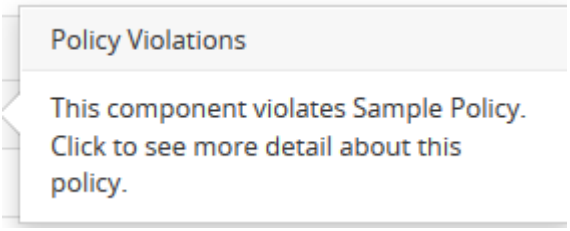
Hover over the icon to view to view more information:

- 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, Global Project Administrator, Global Project Manager, Project Manager, and Policy Violation Reviewer roles:

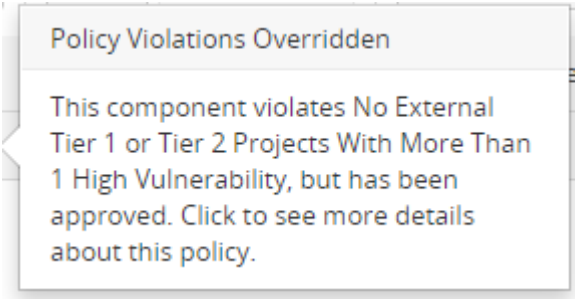


Clicking the icon (when viewing the BOM using the List view) displays the Policy Violations dialog box from which you can override the policy violation.

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 Tree View, ⓘ 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

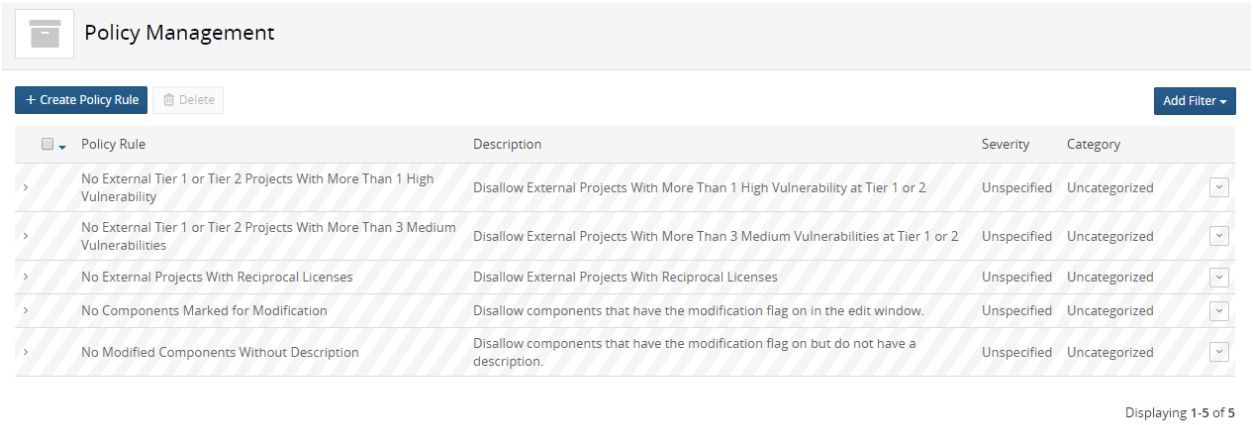
If a violation of a policy should not have been overridden, you can remove the override.

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 

> **Policies** and selecting to view disabled rules:



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

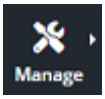
Creating a policy rule

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.

Creating a policy

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

and select **Policies**.
3. Click **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 SCA 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 a subset of 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 project condition: ensure you have the **A Subset of Projects, filtered by...** option is enabled, select an attribute from the Project Conditions list, select an operator, and specify a value.

   Click **+ Project Condition** to specify additional project conditions.
6. 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.
7. 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.
8. To remove a condition, click in the row of the condition you wish to remove.
9. 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.

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 Conditions filters

The Project Conditions filter is displayed when the **A Subset of Projects, filtered by...** option is enabled.

Project Condition filters are divided into these categories:

- Properties
- Project Custom Fields
- Project Version Custom Fields

Table 15: Properties

Project Filters	Value
Project Name	Begin typing to view possible values.
Project Group 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: • Deprecated • In Development • In Planning • Pre-Release • Released
Project Distribution Type	Select one of the following values: • External

Project Filters	Value
	• Internal • Open Source • SaaS

Table 16: 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.

Table 17: 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

Table 18: 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: • Dev. Tool / Excluded • Dynamically Linked • Source Code • Statically Linked • Separate Work • Implementation of Standard • Prerequisite • Merely Aggregated • Unspecified
Review Status	Select one of the following values:

Component Condition	Value
	• Not Reviewed • Reviewed
Match Type	Select one of the following values: • Files Added/Deleted • File Dependency • Direct Dependency • Transitive Dependency • Exact Directory • Exact File • Files Modified • Manually Added • Manually Identified • Partial • Snippet • Binary • Direct Dependency Binary • Transitive Dependency 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: • Unreviewed • In Review • Reviewed • Approved • Limited Approval • Rejected • Deprecated
Component Version Approval Status	Select one of the following values: • 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.

Component Condition	Value
Unconfirmed Snippets	Select **True** or **False**. If you select **True**, any snippet that has not been reviewed will trigger a policy violation.

Table 19: Operational

Component Condition	Value
Component Release Date	Select a date.
Component Release Age	Enter a number value in days.
Newer Versions Count	Enter a number.
Commits in the past year	Enter a number.
Contributors in the past year	Enter a number.

Table 20: 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.

Table 21: Licenses

Component Condition	Value
Unfulfilled License Terms	Select **True** or **False**. If you select **True**, any component that has unfulfilled license terms will trigger a policy violation. **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.
License Conflict with Project Version	Select **True** or **False**. If you select **True**, a policy violation is triggered when a component's license conflicts with the license for a project version.
License Risk	Select one of the following values: • None • Low • Medium

Component Condition	Value
	• 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)	Select one of the following values: • Unreviewed • In Review • Reviewed • Approved • Limited Approval • Rejected • Deprecated
License Expiration Date (Declared)	Select an expiration date for the declared license.
License Expiration Date Comparison (Declared)	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 *add to* 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. • 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. The following are examples using 'before.' The project version release date is the 10th. • 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. The following are examples using 'after.' The project version release date is the 10th. • 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.

Component Condition	Value
	• 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)	Select one of the following values for the deep (embedded) license: • Unreviewed • In Review • Reviewed • 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)	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 *add to* 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. • 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. The following are examples using 'before.' The project version release date is the 10th. • 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. The following are examples using 'after.' The project version release date is the 10th.

Component Condition	Value
	• 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.

Table 22: 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.

Table 23: 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.

Table 24: 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

When configuring policy rules with vulnerability conditions, it is important to understand how remediation status affects policy evaluation.

Vulnerability Score and Remediation

Policy rule vulnerability conditions do not automatically exclude remediated vulnerabilities from results. Vulnerabilities retain their original overall score even after remediation actions have been taken. This means that if a policy is based on vulnerability score, the policy condition will not be automatically cleared when a vulnerability is remediated.

Remediation Status in Component Vulnerability Conditions

Unlike Vulnerability Conditions, **Component Conditions** automatically exclude vulnerabilities with remediated statuses from evaluation. For example, when using the Component Condition **Highest Vulnerability Score**, remediated vulnerabilities are not considered in the score calculation.

This behavior makes Component Conditions a preferred option for customers who want policy rules that automatically account for remediation without requiring additional expressions. For instance, customers may choose to use the Component Condition **Highest Vulnerability Score** instead of combining the Vulnerability Condition **Overall Score** with a **Remediation Status** expression.

Best Practice: Include Remediation Status

To ensure that your policy rules accurately reflect the current state of vulnerabilities in your project:

- Consider including an expression for **Remediation Status** in your policy conditions. This allows you to create more precise policies that account for remediation actions and avoid false positives from vulnerabilities that have already been addressed.
- **Use Vulnerability Conditions** when you need granular control over vulnerability evaluation and want to explicitly define how remediation status should be handled.
- **Use Component Conditions** when you want remediated vulnerabilities to be automatically excluded from policy evaluation.

Vulnerability condition	Value
Overall Score	Enter a number from 0 to 10.
Vulnerability IDs	Enter a specific vulnerability ID (CVE or BDSA).
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.
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: • Duplicate • Ignored • Known Affected • Known Affected: Mitigation • Known Affected: No fix planned • Known Affected: None available • Known Affected: Vendor fix • Known Affected: Workaround • Known Not Affected • Known Not Affected: Component not present • Known Not Affected: Vulnerable code not present • Known Not Affected: Vulnerable code cannot be controlled by adversary

Vulnerability condition	Value
	• Known Not Affected: Vulnerable code not in execute path • Known Not Affected: Inline mitigations already exist • Mitigated • Needs Review • New • Patched • Remediation Complete • Remediation Required
Vulnerability Tags	Select one or more of the following values: • AI Assisted • Zero-click Remote Code Execution • Malicious Code Identified • Embargoed Vulnerability Details • Unconfirmed Vulnerability • Automated Security Advisory • CISA Known Exploited Vulnerability • Rapid Review
Published Age	Allows you to define a policy based on how recently a vulnerability was published. Enter a number value in days.

Operators in Black Duck SCA Policies

In Black Duck SCA, operators are essential for defining rules and conditions within policy management. They allow users to create precise and effective policies for managing open source components, vulnerabilities, and compliance requirements.

This section outlines the different types of operators available, including logical, comparison, membership, string, and existence operators. Understanding how to use these operators is crucial for formulating policies that align with your organization's security and compliance objectives.

Types of Operators

- **Comparison Operators**: Compare values to establish conditions for policy enforcement.
  - **Equals**: Checks if two values are equal.
  - **Not Equal To**: Checks if two values are not equal.
  - **Less Than**: Checks if a value is less than another value.
  - **Greater Than**: Checks if a value is greater than another value.
  - **Less Than or Equal To**: Checks if a value is less than or equal to another value.
  - **Greater Than or Equal To**: Checks if a value is greater than or equal to another value.
- **Membership Operators**: Determine whether a value exists within a specified set.
  - **In**: Checks if a value exists within a specified list or set.
  - **Not in**: Checks if a value does not exist within a specified list or set.
- **Time Operators**: Used to filter and assess vulnerabilities or components based on their timestamps.
  - **Before**: Checks if a specified date or time occurs before another date or time.

  - **After**: Determines if a specified date or time occurs after another date or time.
  - **Older than**: Compares a date or time against a specified duration, checking if a record is older than the given time frame (e.g., older than 30 days).
  - **Younger than**: Compares a date or time against a specified duration, checking if a record is younger than the given time frame (e.g., younger than 30 days).
- **Other Operators**
  - **Included**: Checks if a component is part of a defined list or collection based on the criteria specified in the policy. It can be used to determine whether certain components are explicitly included in a project or BOM (Bill of Materials) for the purposes of vulnerability assessment, licensing compliance, or other related evaluations.
  - **Matches**: Used to determine if the value matches a specified pattern.

Creating policy rules for approved or barred items

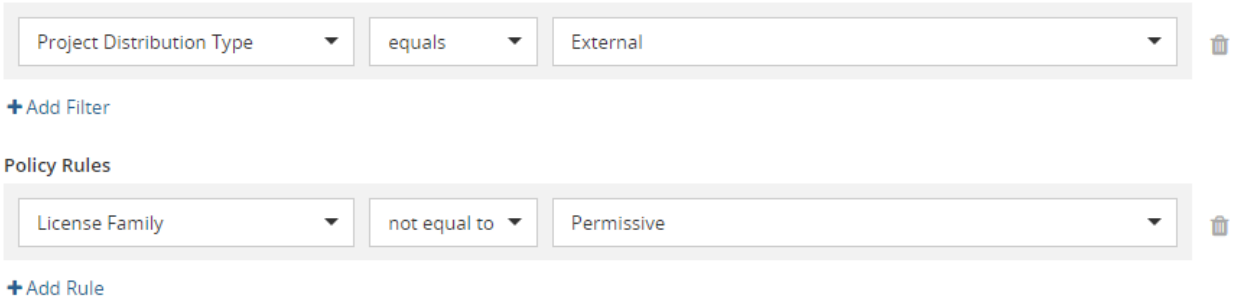
You can create policy rules that enforce your company's policy of approved or barred items. For example, you can create a policy rule to:

- 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

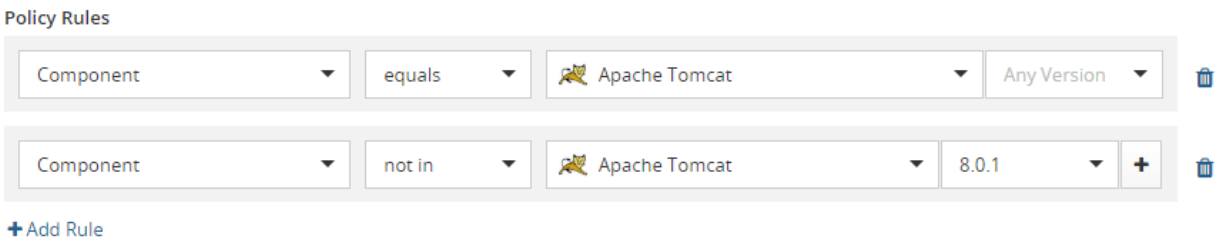
Suppose you want to create a policy rule whereby externally distributed projects with permissive licenses are pre-approved: any component versions that have non-permissive licenses will trigger a policy violation.

To create this policy rule, follow the instructions for creating a policy rule, and set these conditions:



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.

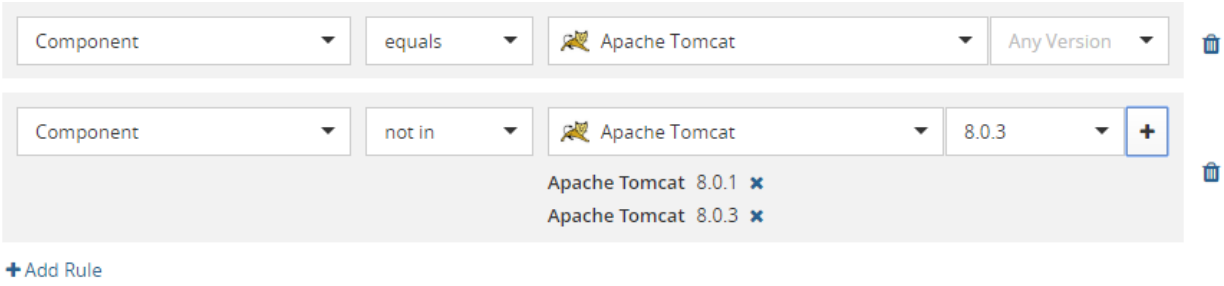
To create this policy rule, follow the instructions for creating a policy rule, and set these conditions:



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:



In this example, a policy violation is triggered when the Apache Tomcat version is not 8.0.1 or 8.0.3.

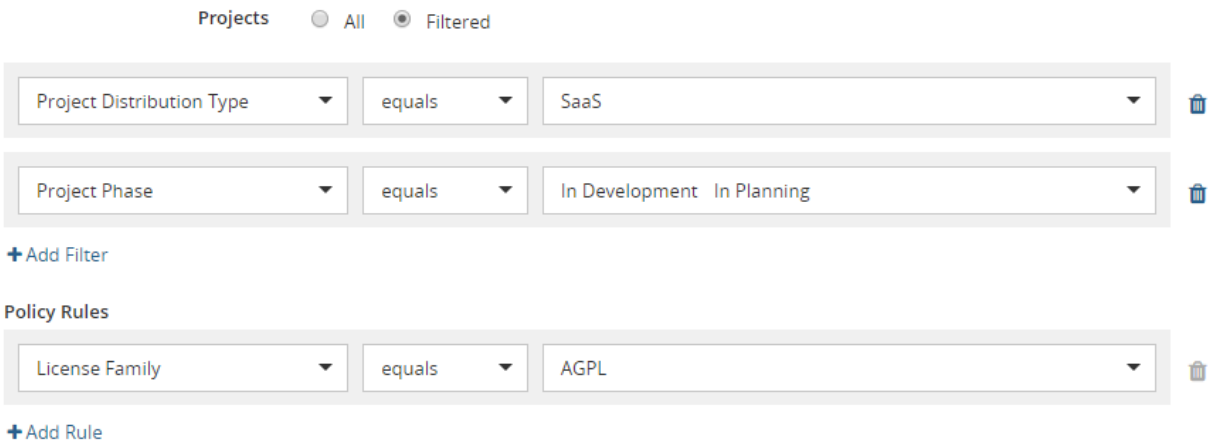
To create this condition:

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:



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 

> **Policies**.

   The Policy Management page appears.

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 

> **Policies**.

   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 

> **Policies**.

   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 > **Policies**.

   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.

     If you enter a date, the override will expire at that date. When it expires it will return to a violation state.
   - 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 or date 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.

Managing Project Groups

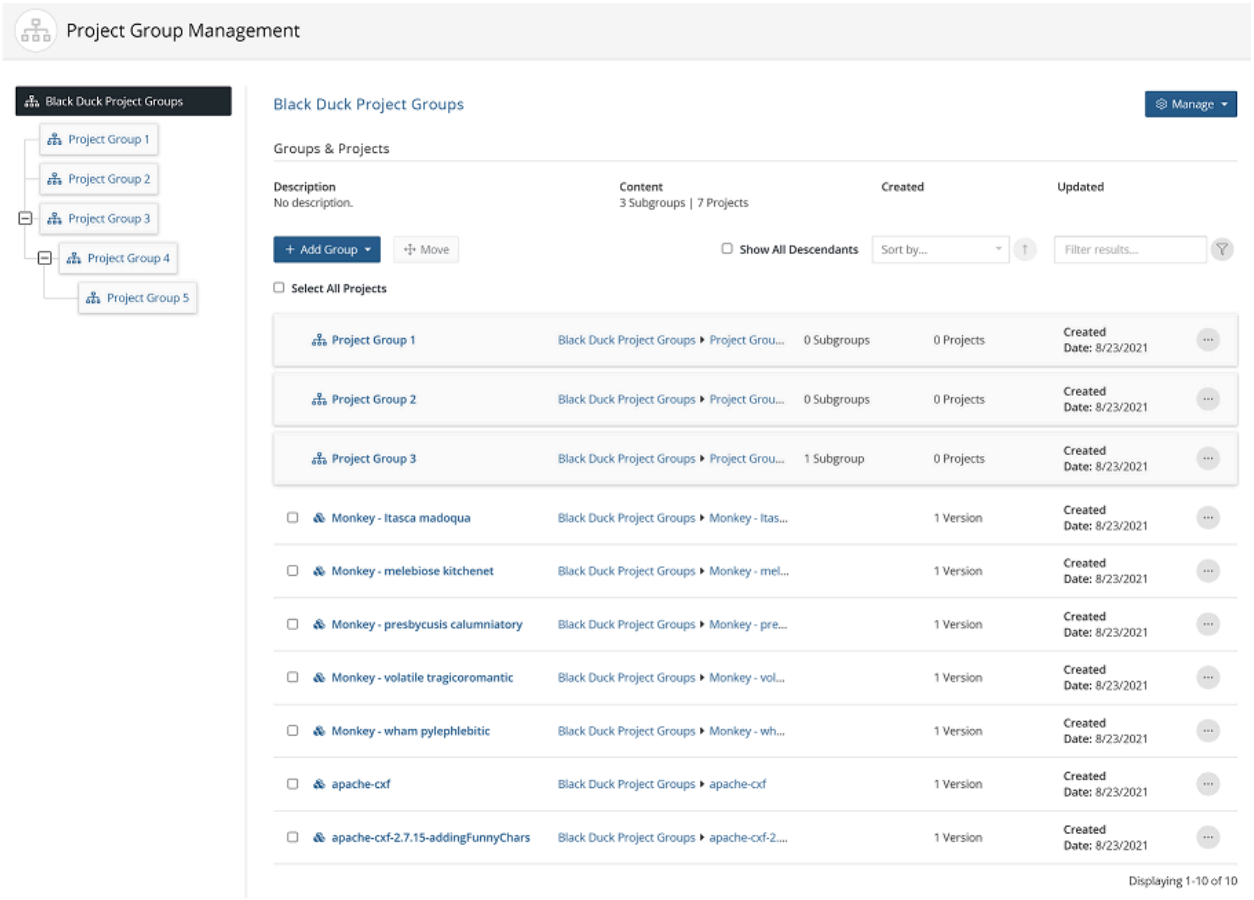
Black Duck provides the ability to logically group all your projects in the Hub, allowing you to organize which projects belong to which business unit making it easier for you to view risk across the organization. Project groups can contain both projects and other project groups to provide a multi-level hierarchy.

To manage project groups:

1. Log in to Black Duck SCA.
2. Click 

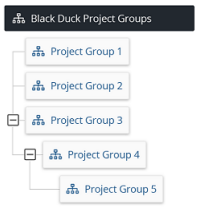
and select **Project Groups**.

From the Project Group Management page, you will see the root project group for all projects and groups.



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.



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.

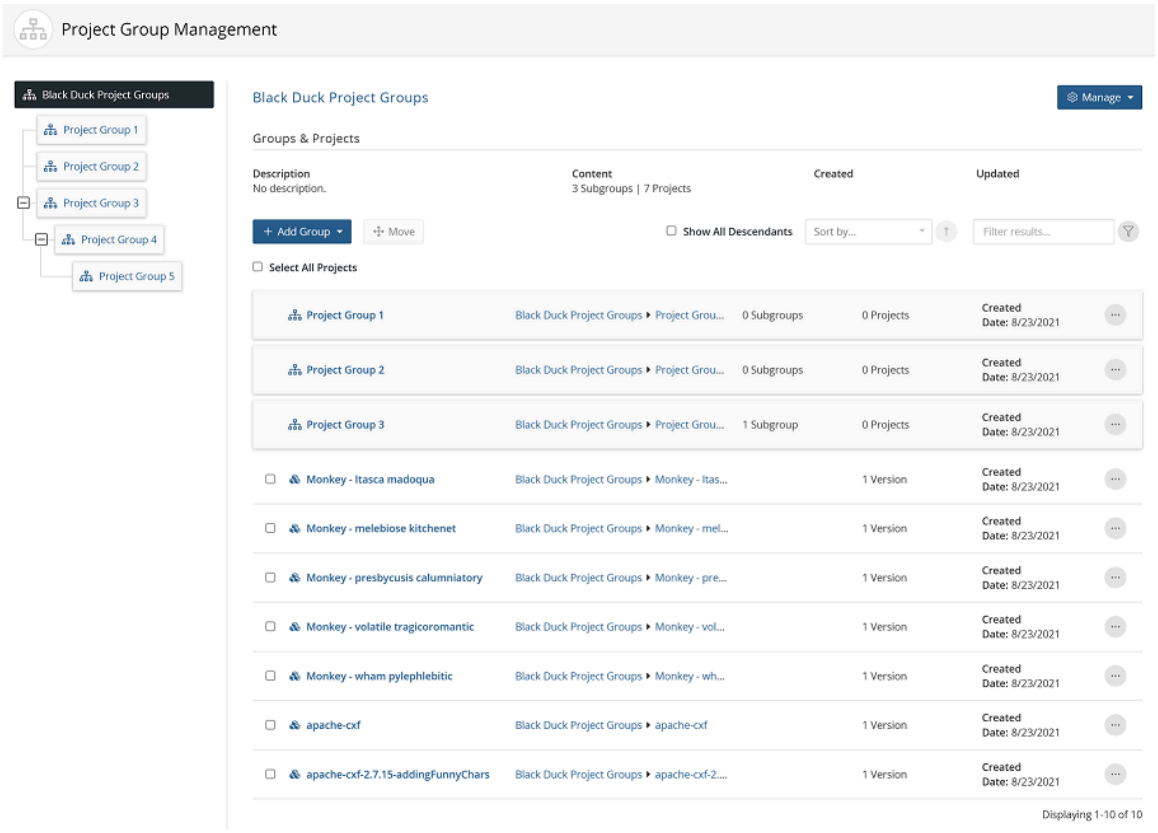


As a result, users Bom and Copyright now have Indirect Access to Project 4 as seen above. This extends to all children of Project 3. They will hold the same role for all child projects and project groups of Project 3.

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 Groups** to display the Project Group Management page.



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:

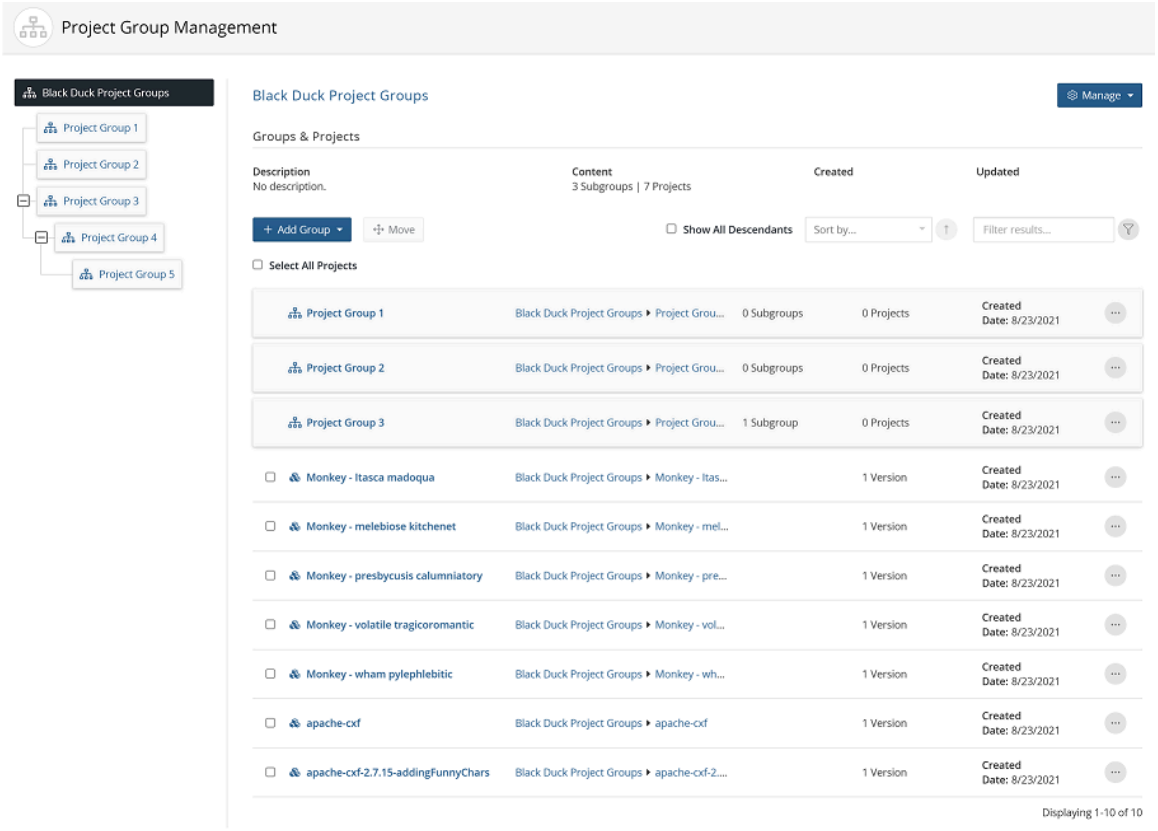
- Add members and user groups to the project group.
- Validate the generation of SBOM reports against policies.

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 as well as set the option to validate the generation of SBOM reports against policies for projects belonging to specific project groups.

To do so, follow the steps listed below:

1. Log in to Black Duck.
2. Click Manage.
3. Select **Project Groups** to display the Project Group Management page.



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 + Add Group.
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 Manage and select **Users** from the dropdown menu.
3. Click + Add Member.
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 Manage and select **Users** 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 Manage and select **Users** 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 Manage 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 Manage and select **User Groups** from the dropdown menu.
3. Click ⋯.
4. Select **Delete Direct Access**.
5. Click **Delete** from the confirmation dialog box.

Enabling or disabling SBOM report validation

When setting is enabled, the ability to generate SBOM reports will be disabled if the project has policy violations.

1. Select the desired project group from the project group tree in the left-hand panel.
2. Click Manage and select **Settings** from the dropdown menu.
3. Scroll to the **Reports** section.

4. Check or uncheck the **Don't generate SBOM reports for projects with policy violations** checkbox.
5. Select either:
   - **Apply setting to all projects in this group only**: Selecting this option will make it so that only this specific project group will have validation enabled when generating a SBOM report.
   - **Apply setting to all projects in this group and child groups**: Selecting this option will make it so this project group and all its child groups will have validation enabled generating a SBOM report.

Editing SBOM and VEX fields

These are additional fields that can be included in the SBOM report.

1. Select the desired project group from the project group tree in the left-hand panel.
2. Click Manage and select **SBOM & VEX fields** from the dropdown menu.
3. Edit any of the following fields:
   - **Creator**: Replaces default creator information with the person(s) or organization(s) that created the SBOM file.
   - **Namespace**: The Namespace is a URL that is under control of your organization and can be used as a globally unique identifier for the VEX document issuing party. E.g., "https://www.mycompanyname.com"
   - **Creator Comments**: An optional field for creators of the SBOM file to provide general comments about the creation of the SPDX file or any other relevant comment not included in the other fields.
   - **VEX Legal Disclaimer**: An optional field to define your organization's legal disclaimer for VEX documents. If populated, this text will be automatically included in all generated VEX reports and cannot be excluded.
   - **Propagate field values to all child groups**: When enabled, all project groups under this group will inherit the field values, but they can be overriden in each group.

Administering Black Duck

Creating system announcements

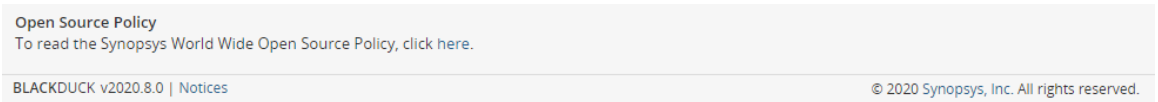
System Administrators can create custom announcements or messages to be displayed in a variety of locations in Black Duck.

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 – Black Duck 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.

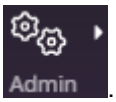


- Welcome. A message that appears after the user logs in to Black Duck.

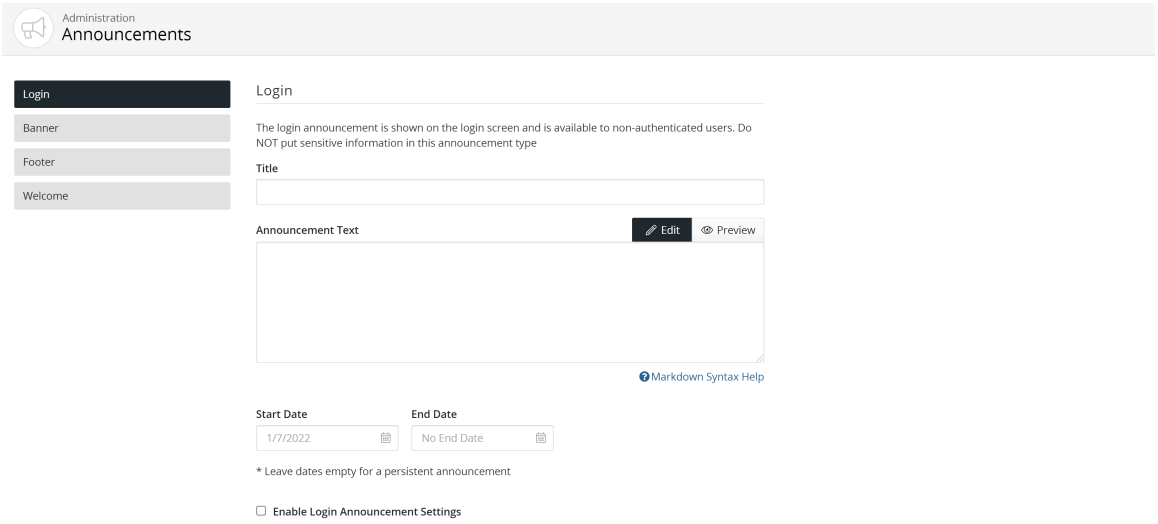
  Unlike other announcements, you can provide an option so that users can suppress this message: users will not see this message again unless you edit the message.

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 

.
3. Select **Announcements** to display the **Announcements** page.



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 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
Auto Remediate Unmapped	Auto remediate CVEs with unmapped related BDSAs.
BDIO Data Transfer	Processes scan data and prepares it for the matching process.
BDIO Storage Migration Check	Checks if there are any BDIO files to migrate to the storage service.
BOM Event Cleanup	Cleans up BOM events based on the retention policy.
BOM Vulnerability Recomputation Check	Checks if BOM computations are required when certain settings change and starts the necessary jobs.
Check Sigma Tool Version	Checks if the Sigma tool version is up to date and schedules the work if it is not.
Checks Need for Hierarchical BOM Calculation	Checks if hierarchical BOM computations are required and starts the necessary jobs to process them
Hierarchical Version BOM	Creates and updates the hierarchical version BOM.
Job History Statistics	Calculates statistics from the job history.
Journal Partition Maintenance	Creates new database partitions for the project audit trails and drops old partitions. The Journal table is partitioned by month. The first partition is special and contains all existing journal events. Journal events older than 5 years will be purged.
KnowledgeBase Update Check	Initiates updates received from the KnowledgeBase. See Understanding the Knowledge Base Update Job for more information.
License Term Check	Checks if license fulfillment processing is required and starts the necessary jobs.
LTS Transition Scheduler	Schedules LTS conversion jobs based on data retention settings.
Notification Purge Check	Checks if there are notifications that need cleanup and starts the necessary jobs.
Policy Rule Bulk Propagation	Publish BOM messages for policy rules that need propagation
Populate License Terms	Updates Black Duck with the latest Black Duck KB license term data.
Purge API Token Check	Determines if the access tokens auto purge job needs to run.

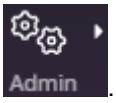
Job Name	Description
Purge API Token	Deletes inactive access tokens based on configured settings.
Purge Deleted Storage Objects	Removes deleted storage objects from the system and cleans out orphaned records.
Purge Notifications	Manages data retention for existing notifications.
Purge Orphan BOMs	Deletes any BOM data not associated with a project version.
Purge Orphan BOMs Check	Checks to see if any BOM data is not associated with a project version and starts the necessary jobs.
Purge Reports	Manages data retention for existing reports and purges expired reports.
Purge Scan Data - Delete Abandoned Scans	Deletes scans that were started but never completed.
Purge Scan Data - Delete Expired Scans	Deletes scans that are older than the expiration threshold.
Purge Scan Data - Delete Orphaned Scan Identifiers	Deletes orphaned snippet scan identifiers.
Purge Scan Data - Delete Orphaned Scans	Deletes orphaned scans.
Purge Scan Data - Delete Stale Projects	Deletes stale projects.
Purge Scan Data - Delete Stale Releases	Deletes stale releases.
Purge Scan Data - Delete Unmapped Scans	Deletes unmapped scans.
Purge Scan Data - Purge Component Mapping Audit Events	Purges component mapping audit events.
Purge Scan Data - Purge Deleted Scans	Deletes scans that were previously queued for deletion.
Purge Scan Data - Purge Scan Archives	Deletes scan archives that are eligible for deletion.
Purge Scan Data - Purge Unmatched Files	Deletes unmatched files from scans that are eligible for deletion.
Purge Scan Data Scheduler	Schedules job that removes old scan data.
Purge Scan Statistics	Removes old scan statistics.
Report Storage Migration Check	Checks whether there are reports that need to be migrated into the storage service.
Reporting Database Transfer	Migrates Black Duck data to the Black Duck reporting warehouse.
Reporting Database Transfer Scheduler	Schedules job that transfers reporting data to the reporting schemas for customers.
SBOM Report	Generates the SBOM report for a Project Version.
Scan Auto BOM Calculation	Calculates a BOM from a scan.

Job Name	Description
Scan Statistics	Collects scan statistics shown on the **usage: scan completion** section on the System Information page.
Scheduled Policy Rule Changes Check	Finds policy violation overrides with scheduled expirations.
Schema Difference Report	Calculates the differences between the current database schema and the ideal as determined at release.
SCM Onboarding daily auto scanning	Schedules nightly job that performs auto scanning of previously onboarded SCM repositories.
SCM Onboarding daily cleanup	Schedules nightly job that cleans up from SCM Onboarding.
Search Dashboard Refresh	Updates the information shown on the Projects and Components Dashboards.
Search Dashboard Refresh Check	Checks if it is time to refresh the Projects and Components Dashboards and starts the necessary job.
Snippet BOM Calculation	Calculates a BOM from a snippet scan.
Storage Migration Check	Checks to see if there is migration work to perform.
Storage Pruning Check	Checks if the object storage system has items to prune.
System Maintenance and Reporting	Processes system statistics and registers them with the KnowledgeBase.
Test Performance Dispatch	System diagnostic that tests the performance of job dispatch.
Update KnowledgeBase Data - BDSA Vulnerability Update	Updates BDSA vulnerability information received from the KnowledgeBase.
Update KnowledgeBase Data - Component Update	Updates component information received from the KnowledgeBase.
Update KnowledgeBase Data - Component Version Security Update	Processes component version updates received from the KnowledgeBase.
Update KnowledgeBase Data - License Update	Updates license information received from the KnowledgeBase.
Update KnowledgeBase Data - NVD Vulnerability Update	Updates NVD vulnerability information received from the KnowledgeBase.
Update KnowledgeBase Data - Summary	Issues a summary report about the most recent KnowledgeBase update.
Update Origin Copyrights	Updates origin copyrights.
Version BOM Notification	Notifies on various BOM-related events.
Version BOM Notification Check	Checks to see if any version BOM notifications need processing.

Job Name	Description
Version Bom Computation Check	Checks if any version BOMs need computations and schedules the work.
Version License Report	Creates the Notices File report.
Version Report	Creates the Project Version report.
Version Vulnerability Remediation Report	Creates the Project Version Vulnerability Remediation report.
Version Vulnerability Status Report	Creates the Project Version Vulnerability Status report.
Version Vulnerability Update Report	Creates the Project Version Vulnerability Update report.
Vulnerability Remediation Report	Creates the Vulnerability Remediation report.
Vulnerability Status Report	Creates the Vulnerability Status report.
Vulnerability Update Report	Creates the Vulnerability Update report.
Watchdog	Monitors the job subsystem for errors and reports or fixes issues as they arise.

Viewing jobs

To view a list of jobs and their current statuses:

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

.
3. Click **Jobs**.
4. Click the **Jobs** tab to display the Jobs page.

Filtering the Jobs table

You can refine the jobs displayed in the table by selecting one of the following options:

- **Finished**: Displays all finished jobs.
- **Scheduled**: Displays all jobs set to run in your environment.
- **Processing**: Displays all jobs currently processing.

Finished jobs table

Clicking the Finished button on the Jobs page displays a table of all completed jobs listed in chronological order. The table of jobs is composed of the following columns:

- **Status**: Displays the current state of the job. The statuses are as follows:
  - Success
  - Error
- **Job name**: The name of the job run.

- **Attempts**: How many times the job was run.
- **Scheduler type**: The scheduler type of the job.
  - Periodic: Jobs that are permanently stored with one or more repeating triggers.
  - On Demand: Non-durable jobs that are triggered from a specific or periodic event and are auto-deleted after the triggering event.
- **Started**: When the job was started in your environment.
- **End Time**: When the job was finished in your environment.
- **Duration**: The amount of time the job took to complete its run.

Filtering the Finished jobs table

You can refine the list of jobs displayed in the table by clicking the **+ Filter** button and selecting one of the following options:

- **End Time**: Selecting this filter displays a date picker where you can choose a start and end date. Once selected, the table will show all jobs finished within the specified time frame.
- **Error Status**: Selecting this filter will show all jobs that ended with an error.
- **Job Name**: Selecting this filter displays a list of all available jobs. Once a job is selected, the table will show all completed entries for this job. Multiple jobs can be selected.
- **Scheduler Type**: Selecting this filter display a list of scheduler types. Once a scheduler type is selected, the table will show all entries of the selected scheduler type.

Scheduled jobs table

Clicking the Scheduled button displays all jobs set to run in your environment. The table of jobs is composed of the following columns:

- **Job Name**: The name of the job.
- **Scheduler type**: The scheduler type of the job.
  - Periodic: Jobs that are permanently stored with one or more repeating triggers.
  - On Demand: Non-durable jobs that are triggered from a specific or periodic event and are auto-deleted after the triggering event.
- **Job Frequency Type**: The job's type of trigger and its recurrence.
- **Scheduled Time**: The next time the job is scheduled to run.
- **Enabled**: Whether or not the job is enabled to run in your environment.

Enabling or disabling jobs

You can enable or disable a particular job by clicking the button at the end of its row and selecting the desired option.

Filtering the Scheduled table

You can refine the list of jobs displayed in the table by clicking the **+ Filter** button and selecting one of the following options:

- **Enable**: Select either Enabled or Disabled to display all jobs of the desired type.

- **Job Frequency Type**: Select any from Run Once, Cron Pattern, or Periodic Interval to display all jobs of the desired type. Multiple options can be selected simultaneously.
- **Job Name**: Selecting this filter displays a list of all available jobs. Once a job is selected, the table will show all completed entries for this job. Multiple jobs can be selected.
- **Scheduler Type**: Selecting this filter display a list of scheduler types. Once a scheduler type is selected, the table will show all entries of the selected scheduler type.

Processing jobs table

Clicking the Processing button displays all jobs currently processing in your environment. The table of jobs is composed of the following columns:

- **Job Name**: The name of the job.
- **Scheduler type**: The scheduler type of the job.
  - Periodic: Jobs that are permanently stored with one or more repeating triggers.
  - On Demand: Non-durable jobs that are triggered from a specific or periodic event and are auto-deleted after the triggering event.
- **Job Frequency**: How often the job is set to run.
- **Started**: When the job was started in your environment.
- **Elapsed Time**: How long the job has been running. A icon appearing next to the job's elapsed time indicates that the job is running longer than normal. Mousing over the icon will display the typical amount of time this job takes to complete.

Filtering the Processing table

You can refine the list of jobs displayed in the table by clicking the **+ Filter** button and selecting one of the following options:

- **Job Frequency Type**: Select any from Run Once, Cron Pattern, or Periodic Interval to display all jobs of the desired type. Multiple options can be selected simultaneously.
- **Job Name**: Selecting this filter displays a list of all available jobs. Once a job is selected, the table will show all completed entries for this job. Multiple jobs can be selected.
- **Long Running**: Selecting this filter displays a list of jobs that are currently running longer than normal.
- **Scheduler Type**: Selecting this filter displays a list of scheduler types. Once a scheduler type is selected, the table will show all entries of the selected scheduler type.

Downloading log files and heatmap data

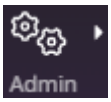
You may need to troubleshoot an issue or provide log files to Customer Support. Users with the System Administrator role can download a zipped file that contains the current log files or the heatmap data for your system.

Please note that it may take a few minutes to prepare the log or heatmap files. Refer to the installation guide for more information on obtaining logs and configuring heatmap data.

Accessing log files

To download the log files from the Black Duck UI:

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

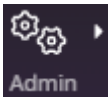
2. Click 

→ **Jobs**.
3. Select the **System Information** tab.
4. Click either **Logs for Last 2 Days (.zip)** or **Logs for Last 14 Days (.zip)**.

Accessing heatmap data

You can review and analyze terminal scan trends by downloading the heatmap as a compressed CSV and create the heatmap as a pivot in a spreadsheet program.

To download the heatmap data for your system:

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

→ **Jobs**.
3. Select the **System Information** tab.
4. Click **Download Heatmap (.zip)** in the Heatmap data section.

Creating heatmap data from terminal-data endpoint

The data contained in the heatmap download link from the **System Information** page is in a different format (ISO 861 UTC) as opposed to what is contained in the **debug** folder. This means that a heatmap can't be generated easily from that data set. To overcome this, follow these steps:

1. Open the heatmap-scan-terminal csv. You will see something like this - note the hour column is a combined date/time column with a time stamp:

hour	codeLocationId	codeLocationName	versionName	projectName	scanCount	s
2023-02-10T17:00:00Z	709b30b2-4cd0-400c-b220-b3bb99c3f04f	OI_Blackduck-Actions-IAC-terragoat_Comp/4 bdio			1	P
2023-02-20T16:00:00Z	85e82dbd-3062-4215-a18f-6f0f87164b52	blackduck-upload-cache/upload-cache/master signature	master	upload-cache	1	S
2023-02-20T16:00:00Z	85e82dbd-3062-4215-a18f-6f0f87164b52	blackduck-upload-cache/upload-cache/master signature	master	upload-cache	1	S
2023-02-20T16:00:00Z	bf2ce281-5022-4ffa-8d96-b0570f718272	WebGoat/webgoat/master signature	master	webgoat	1	S
2023-02-20T16:00:00Z	bf2ce281-5022-4ffa-8d96-b0570f718272	WebGoat/webgoat/master signature	master	webgoat	1	S
2023-02-20T16:00:00Z	bf2ce281-5022-4ffa-8d96-b0570f718272	WebGoat/webgoat/master signature	master	webgoat	2	S
2023-02-20T16:00:00Z	bf2ce281-5022-4ffa-8d96-b0570f718272	WebGoat/webgoat/master signature	master	webgoat	1	S
2023-02-20T16:00:00Z	b082dffc-146c-4b2d-ac3d-e6b091cb9dc3	seamonkey-2.53.13.source.tar/seamonkey/master signature	master	seamonkey	1	S
2023-02-20T16:00:00Z	b082dffc-146c-4b2d-ac3d-e6b091cb9dc3	seamonkey-2.53.13.source.tar/seamonkey/master signature	master	seamonkey	1	S
2023-02-20T17:00:00Z	bf2ce281-5022-4ffa-8d96-b0570f718272	WebGoat/webgoat/master signature	master	webgoat	1	S
2023-02-20T17:00:00Z	bf2ce281-5022-4ffa-8d96-b0570f718272	WebGoat/webgoat/master signature	master	webgoat	1	S
2023-02-20T17:00:00Z	bf2ce281-5022-4ffa-8d96-b0570f718272	WebGoat/webgoat/master signature	master	webgoat	1	S
2023-02-20T17:00:00Z	bf2ce281-5022-4ffa-8d96-b0570f718272	WebGoat/webgoat/master signature	master	webgoat	1	S
2023-02-21T13:00:00Z	69831964-54e5-4a3a-a648-a00e50f69319	springpom/webgoat/master signature	master	webgoat	1	S
2023-02-21T13:00:00Z	69831964-54e5-4a3a-a648-a00e50f69319	springpom/webgoat/master signature	master	webgoat	1	S
2023-02-21T13:00:00Z	e4da7f40-8c44-4c18-81a4-2a32aba31b98	springboot/master bdio	master	springboot	1	P
2023-02-21T14:00:00Z	5b6033da-1594-4d38-9a23-f5693d89e55c	art/Default Detect Version Black Duck I/O Export	Default Detect Version	art	1	P
2023-02-21T17:00:00Z	0e5df8ff-2408-456a-8216-e7a782011b52	zipper_jobexecutionstats_service signature	zipper_jobexecutionstats_service	FloCx	1	S
2023-02-21T17:00:00Z	0e5df8ff-2408-456a-8216-e7a782011b52	zipper_jobexecutionstats_service signature	zipper_jobexecutionstats_service	FloCx	1	S
2023-02-21T17:00:00Z	ffc9e884-2a69-4dd1-8e8e-48d4c6dba866	synopsysctl-linux-amd64.tar.gz/test/Default Detect Version binary	Default Detect Version	test	1	B
2023-02-21T17:00:00Z	0e5df8ff-2408-456a-8216-e7a782011b52	zipper_jobexecutionstats_service signature	zipper_jobexecutionstats_service	FloCx	1	S
2023-02-21T18:00:00Z	0e5df8ff-2408-456a-8216-e7a782011b52	zipper_jobexecutionstats_service signature	zipper_jobexecutionstats_service	FloCx	4	S
2023-02-21T18:00:00Z	ffc9e884-2a69-4dd1-8e8e-48d4c6dba866	synopsysctl-linux-amd64.tar.gz/test/Default Detect Version binary	Default Detect Version	test	1	B
2023-02-23T11:00:00Z	7f9b4c22-3538-4f50-8caa-02fdaefbcd48		Rapid-Policy	Dell	2	R
2023-02-23T12:00:00Z	7f9b4c22-3538-4f50-8caa-02fdaefbcd48		Rapid-Policy	Dell	1	R
2023-02-24T10:00:00Z	30306c0b-fa98-4f9c-b16d-6179009c575a	src/canon/snippet signature	snippet	canon	1	S
2023-02-24T10:00:00Z	30306c0b-fa98-4f9c-b16d-6179009c575a	src/canon/snippet signature	snippet	canon	1	S
2023-02-21T18:00:00Z	a6b07380-0e26-45df-9950-35d8137939a4	h2-image-master/h2-image-master/Default Detect Version signature	Default Detect Version	h2-image-master	1	S
2023-02-21T18:00:00Z	837def61-34bc-4d8a-b007-96a7b7208aed	zipper_jobexecutionstats_service bdio	zipper_jobexecutionstats_service	FloCx	2	P
2023-02-21T18:00:00Z	0e5df8ff-2408-456a-8216-e7a782011b52	zipper_jobexecutionstats_service signature	zipper_jobexecutionstats_service	FloCx	5	S
2023-02-21T18:00:00Z	0e5df8ff-2408-456a-8216-e7a782011b52	zipper_jobexecutionstats_service signature	zipper_jobexecutionstats_service	FloCx	9	S
2023-02-24T15:00:00Z	d8879f2e-1513-416a-ba3b-e2cfcbb5096d	AWSGoat/AWSGoat/master iac	master	awsgoat	1	II
2023-02-27T14:00:00Z	d295d15f-d4d9-4ebe-8ff9-8e6016ba2b80	Unknown-2023-02-27T14:02:59.703611Z[GMT]	develop-SNAPSHOT	IVRA__ssa-insights-lambda-notification	1	P
2023-02-21T18:00:00Z	8d3436c1-d622-4d0c-95d8-9100cef2c437	zipper_jobexecutionstats_service binary	zipper_jobexecutionstats_service	FloCx	4	B
2023-02-21T18:00:00Z	0e5df8ff-2408-456a-8216-e7a782011b52	zipper_jobexecutionstats_service signature	zipper_jobexecutionstats_service	FloCx	1	S
2023-02-23T11:00:00Z	9ae1218d-9ff2-4350-9558-43e6cda68bcf	Dell/Rapid-Policy bdio	Rapid-Policy	Dell	1	P
2023-02-27T14:00:00Z	d5c26ca1-575a-4921-ae4c-bef3152b975f	python/master bdio	master	python	1	P
2023-02-27T14:00:00Z	d5c26ca1-575a-4921-ae4c-bef3152b975f	python/master bdio	master	python	2	P
2023-02-28T14:00:00Z	c8aa80fc-181c-4745-9962-d24252bb72cd	maven_test/8 bdio	8	maven_test	1	P

2. Create new columns to facilitate the data required:

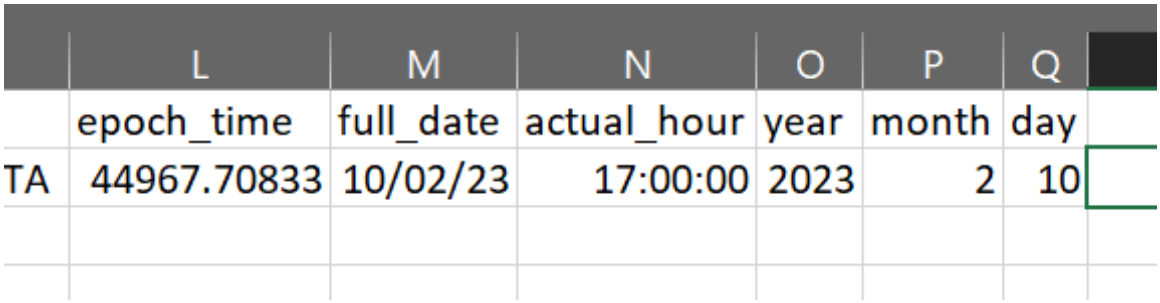
Cell number	Value
L1	epoch_time
M1	full_date

Cell number	Value
N1	actual_hour
O1	year
P1	month
Q1	day

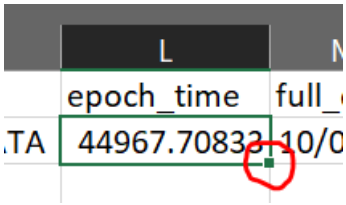
3. Add the following formulae into these new rows to get the data points desired:

Column name	Formula
epoch_time	=DATEVALUE(MID(A2,1,10))+TIMEVALUE(MID(A2,12,8))
full_date	=INT(L2)
actual_hour	=L2-M2
year	=YEAR(L2)
month	=MONTH(L2)
day	=DAY(L2)

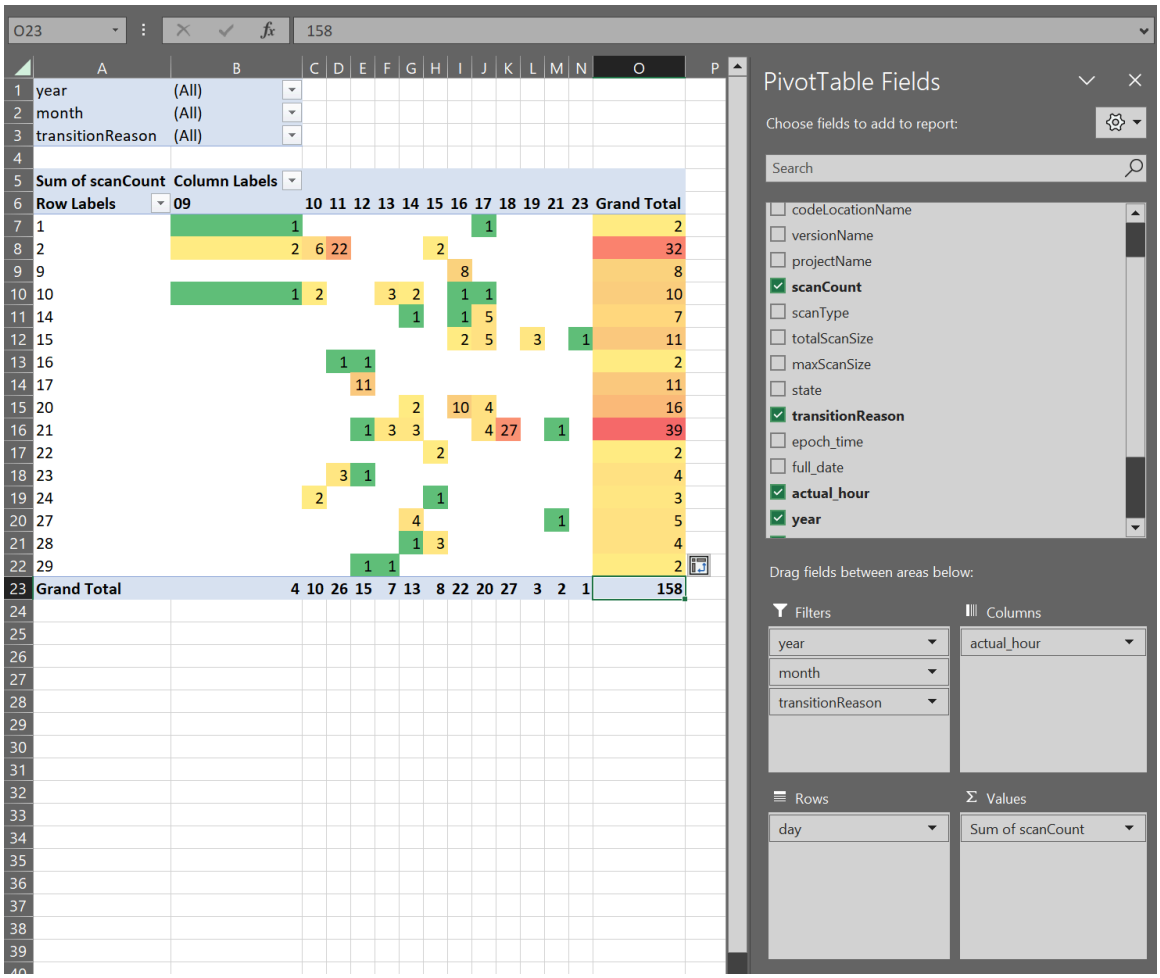
The end result should end up looking something like this:



4. Propagate the data into the rest of the rows by double-clicking on the autofill handle (the little green square at the bottom right of the highlighted cell - your cursor will turn into a cross when you're over it):



5. Create a heatmap as usual including the new columns.



Creating a Heatmap using Excel

Creating a heatmap in Excel on Mac

Users with the System Administrator role can download a zipped file that contains the current log files.

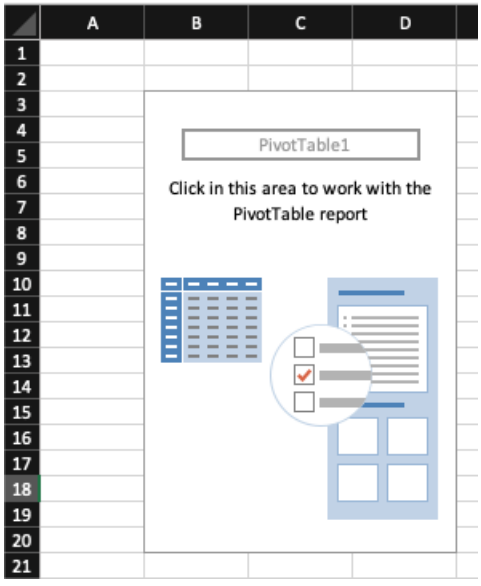
To create a heatmap in Excel:

1. Log in to Black Duck with the System Administrator role.
2. Download the log files.
3. Extract the logs to a folder.
4. Open a new and blank workbook using Microsoft Excel. Using other spreadsheet/workbook programs may yield different results or differ in the steps below.
5. On the first top empty cell of an empty worksheet:
   a. Click the Data tab > Get Data > From text
   b. Navigate to the location of your download logs folder/debug folder
   c. Select the Heatmap CSV file (scan-heatmap-YYYY-MM-##########.csv)
   d. Click Import
6. From the Text Import Wizard perform following:

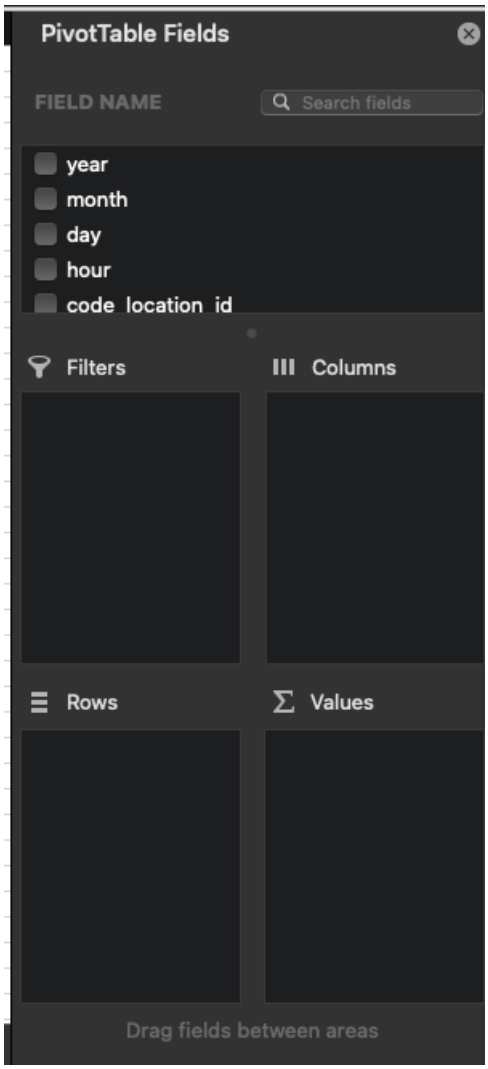
a. Enable the Delimited radio button
b. Start import at row <1>
c. Click the Next button
d. Uncheck the Tab (as delimiter) box
e. Check the Comma (as delimiter) box
f. Click the Next button
g. Click the Finish button

7. In the Import Data modal:
   a. Select either the Existing sheet or New Sheet from 'Where do you want to put data'
8. Select the entire data that has been imported
9. Click the Insert tab
10. Click the Table icon
11. In the Create Table modal that appears, click OK. If an alert pop-up appears, select Yes. The imported data should now be a table with the first row as filters. This is the raw data for the Heatmap. Note the Table name in the top left corner of Excel (i.e. Table1).
12. In a new worksheet:
    a. Click on any cell
    b. Click the Insert tab
    c. Click the Pivot icon
13. In the Create PivotTable dialog box, do the following:
    a. Type the table name (in this case 'Table1') in the *Select a Table or Range* text field
    b. Select Existing worksheet in the *Choose where to place the PivotTable* section
    c. Click the OK button

    Your worksheet should look like this:



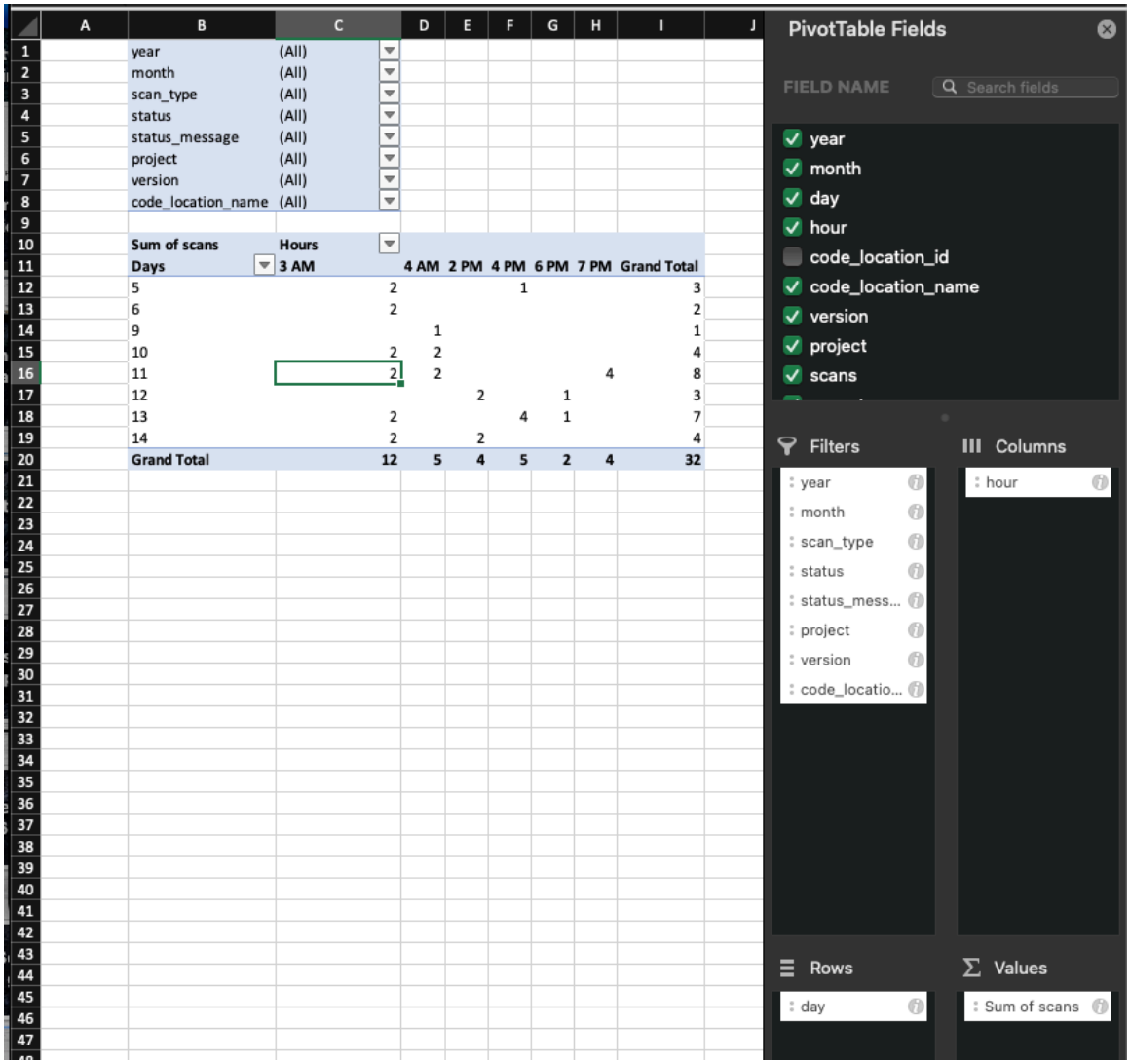
14. Click in the Blank Pivot created, this should open the Pivot column panel.



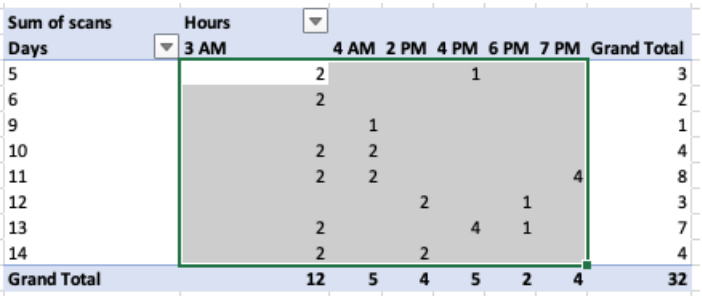
15. Drag the following to filter section in this order:
    a. Year
    b. Month
    c. Scan_Type
    d. Status
    e. Status_Message
    f. Project, Version
    g. Code_Location_Name
16. Drag the Hour field to the Columns section.
17. Drag the Day field to the Rows section.
18. Drag the Scans field to the Values section. It should show as ‘Sum of Scans’.
19. Rename Column header from Row Labels to Days

20. Rename ‘Column Labels’ Filter in second column to Hours

    The presentation in Excel should now look like this:

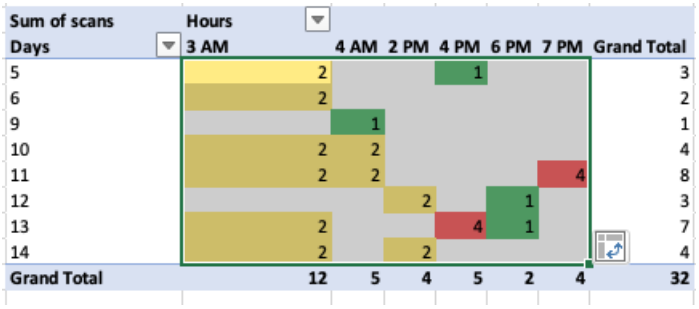


21. Select all the cells that fall within the range between the first to last Hours column (do not include filter top row, ‘Grand Total’ bottom row or last column (Grand Total)



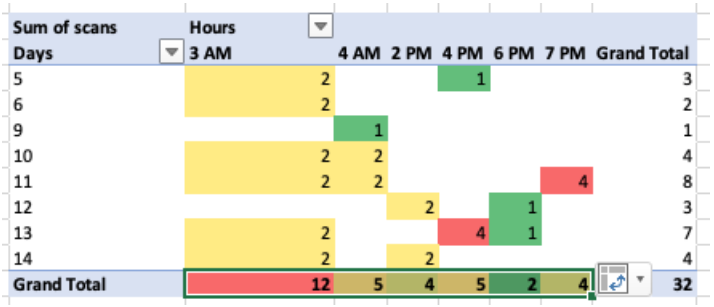
22. Click the Home menu item

23. From the Conditional Formatting > Color Scales, select Red - Yellow - Green scale



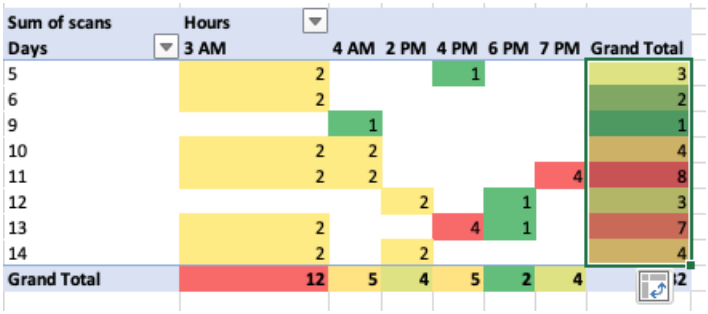
Sum of scans	Hours						
Days	3 AM	4 AM	2 PM	4 PM	6 PM	7 PM	Grand Total
5	2			1			3
6	2						2
9		1					1
10	2	2					4
11	2	2				4	8
12			2		1		3
13	2			4	1		7
14	2		2				4
Grand Total	12	5	4	5	2	4	32

24. Select the cells in bottom Grand Total row and apply the same color scale (excluding first and last column. Only Hours data cells).



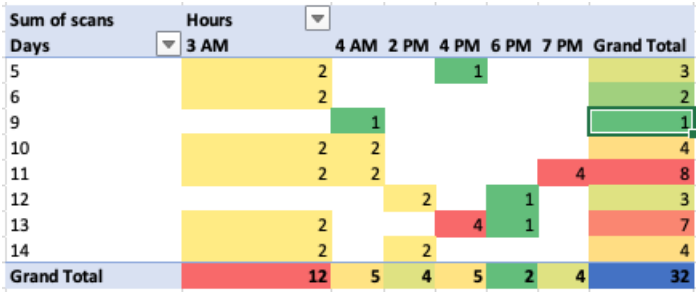
Sum of scans	Hours						
Days	3 AM	4 AM	2 PM	4 PM	6 PM	7 PM	Grand Total
5	2			1			3
6	2						2
9		1					1
10	2	2					4
11	2	2				4	8
12			2		1		3
13	2			4	1		7
14	2		2				4
Grand Total	12	5	4	5	2	4	32

25. Select the cells in the rightmost Grand Total column and apply the same color scale.



Sum of scans	Hours						
Days	3 AM	4 AM	2 PM	4 PM	6 PM	7 PM	Grand Total
5	2			1			3
6	2						2
9		1					1
10	2	2					4
11	2	2				4	8
12			2		1		3
13	2			4	1		7
14	2		2				4
Grand Total	12	5	4	5	2	4	32

26. Select the bottom right corner cell alone and provide it with a blue background.



Sum of scans	Hours						
Days	3 AM	4 AM	2 PM	4 PM	6 PM	7 PM	Grand Total
5	2			1			3
6	2						2
9		1					1
10	2	2					4
11	2	2				4	8
12			2		1		3
13	2			4	1		7
14	2		2				4
Grand Total	12	5	4	5	2	4	32

27. To analyze the data behind any cell, double click on it.

Maximum Scan Size Heatmap

1. Add a blank new sheet to the workbook.
2. In the new worksheet, click on any cell, click on Insert tab, click on Pivot icon.

3. Click on the Blank Pivot created, this should open the Pivot column panel.
4. Drag the following to filter section in this order:
    a. Year
    b. Month
    c. Scan_Type
    d. Status
    e. Status_Message
    f. Project
    g. Version
    h. Code_Location_Name
5. Drag the Hour field to the Columns section.
6. Drag the Day field to the Rows section.
7. Drag avg_scan_size_in_gb field to the Values section
8. Change to show Maximum such that it should display ‘Max of avg_scan_size_in_gb’:
    a. Click on the field in the values section. This will launch the PivotTable field.
    b. Choose Maximum instead of Sum.
    c. Click the OK button.
9. Select all the cells that fall within the range between the first to last Hours column (do not include filter top row, ‘Grand Total’ bottom row or last column (Grand Total)
10. Click the Home tab, Conditional Formatting > Color Scales > Yellow - Green color scale.
11. Select the cells in bottom Grand Total row and apply the same color scale.
12. Select the cells in the rightmost Grand Total column and apply the same color scale.
13. Select the bottom right corner cell alone and provide it with a blue background.

Scan Weight Heatmap

1. Add a blank new sheet to the workbook.
2. In the new worksheet, click on any cell, click on Insert tab, click on Pivot icon.
3. Click on the Blank Pivot created, this should open the Pivot column panel.
4. Drag the following to filter section in this order:
    a. Year
    b. Month
    c. Scan_Type
    d. Status
    e. Status_Message
    f. Project
    g. Version
    h. Code_Location_Name

5. Drag the Hour field to the Columns section.
6. Drag scan_weight field to the Values section and change to show Maximum such that it should display ‘Avg of scan_weight’. This can be set by clicking on the field in the values section and then choosing Average instead of Sum.
7. Drag the Scans field to the Values section. It should show as ‘Sum of Scans’.
8. Select all the cells that fall within the range between the first to last Hours column (do not include filter top row, ‘Grand Total’ bottom row or last column (Grand Total)
9. Click the Home tab, Conditional Formatting > Color Scales > Red - White color scale.
10. Select the cells in bottom Grand Total row and apply the same color scale.
11. Select the cells in the rightmost Grand Total column and apply the same color scale.

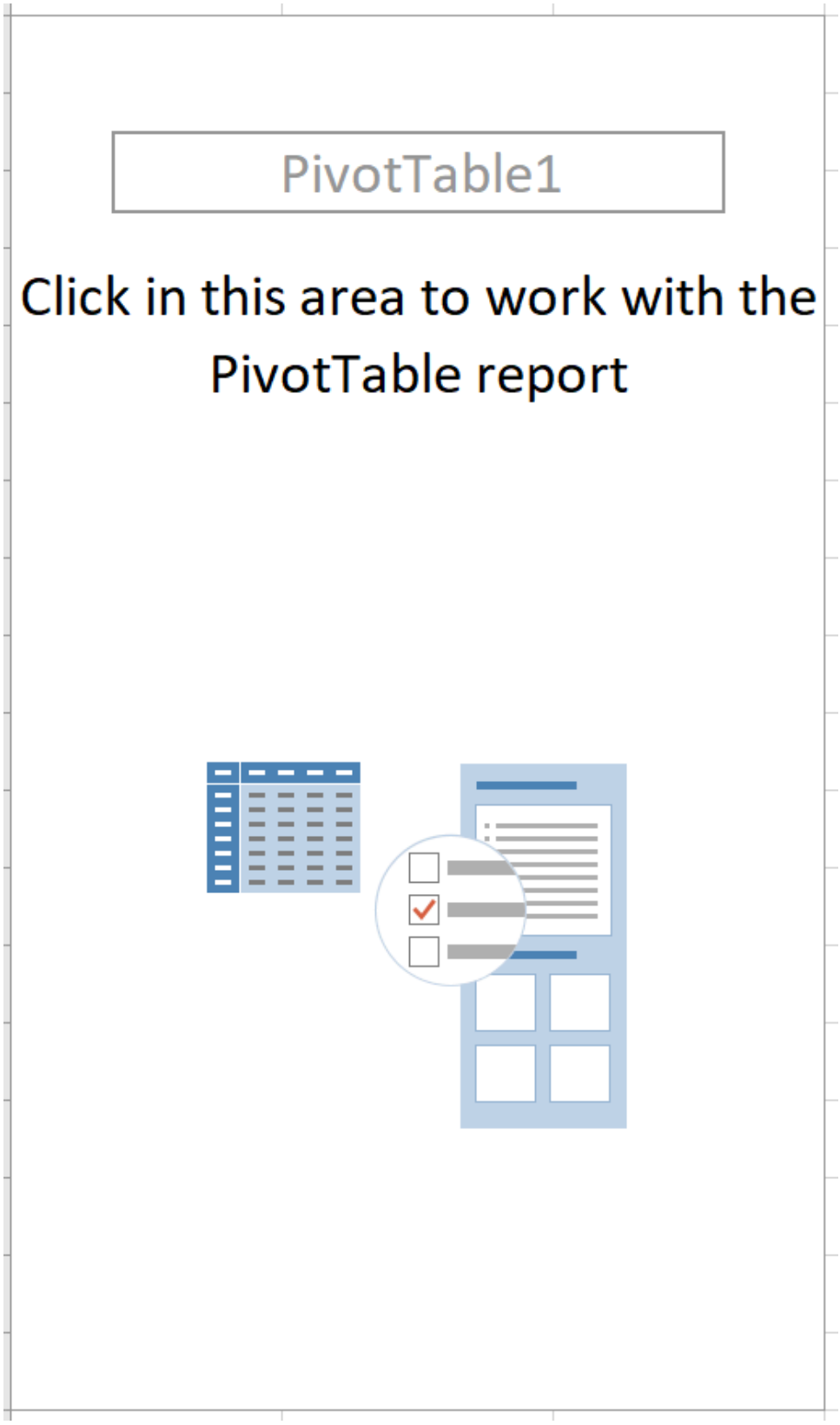
Creating a heatmap in Excel on Windows

Users with the System Administrator role can download a zipped file that contains the current log files.

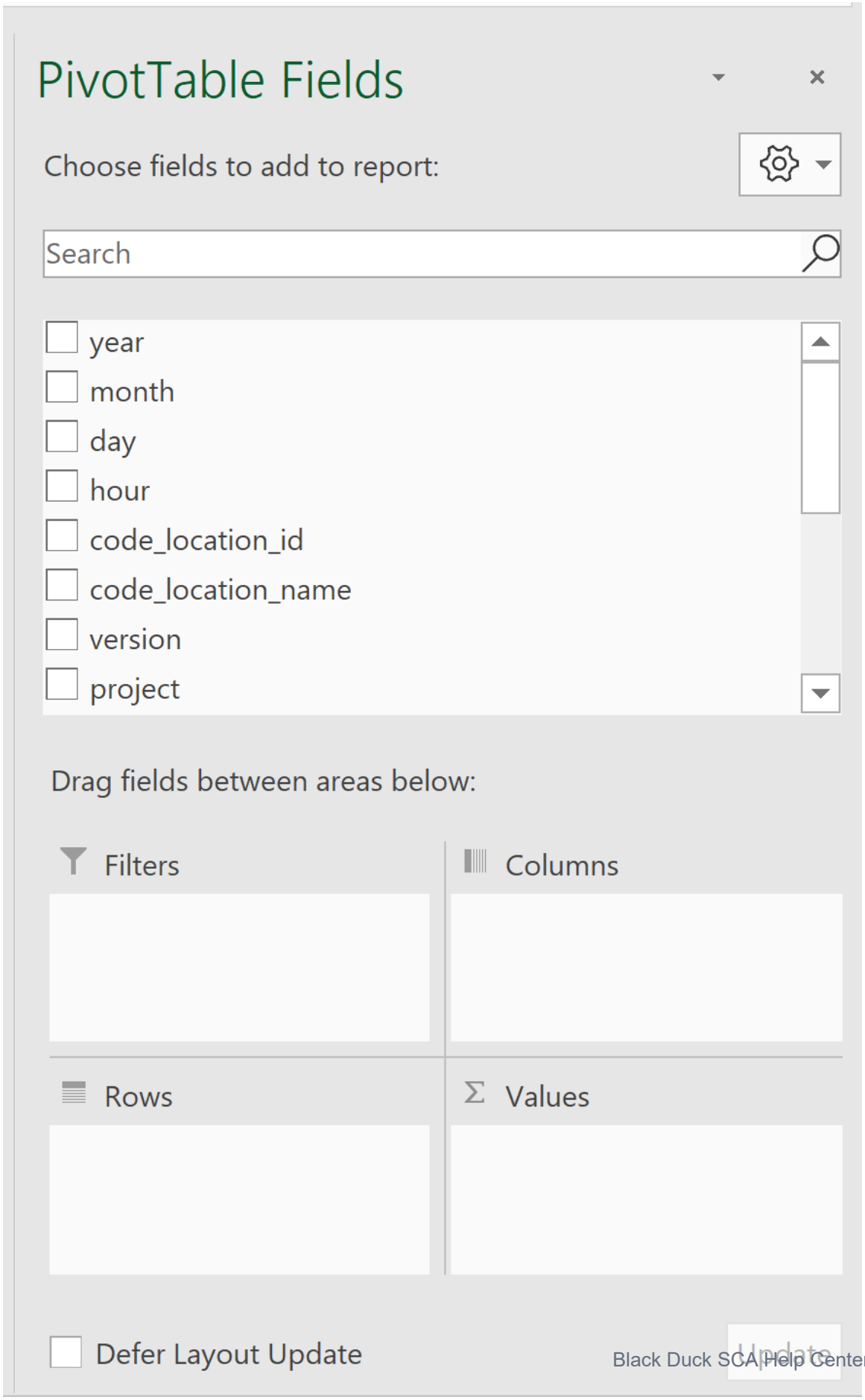
To create a heatmap in Excel:

1. Log in to Black Duck with the System Administrator role.
2. Extract the logs to a folder.
3. Open a new and blank workbook using Microsoft Excel. Using other spreadsheet/workbook programs may yield different results or differ in the steps below.
4. On the first top empty cell of an empty worksheet:
   a. Click the Data tab > Get Data > From File > From Text/CSV
   b. Navigate to the location of your download logs folder/debug folder
   c. Select the Heatmap CSV file (scan-heatmap-YYYY-MM-##########.csv)
   d. Click Import
5. From the Import Wizard perform following:
   a. Ensure that the Delimiter dropdown is set to Comma
   b. Click the Load button
6. Select the entire data that has been imported
7. Click the Insert tab
8. Click the PivotTable icon
9. In the Create PivotTable modal that appears:
   a. Ensure the *Select a table or range* radio button is selected with the appropriate selected field
   b. In the *Choose where you want the PivotTable report to be placed* section, select the New Worksheet
   c. Click the OK button.

      Your worksheet should look like this:

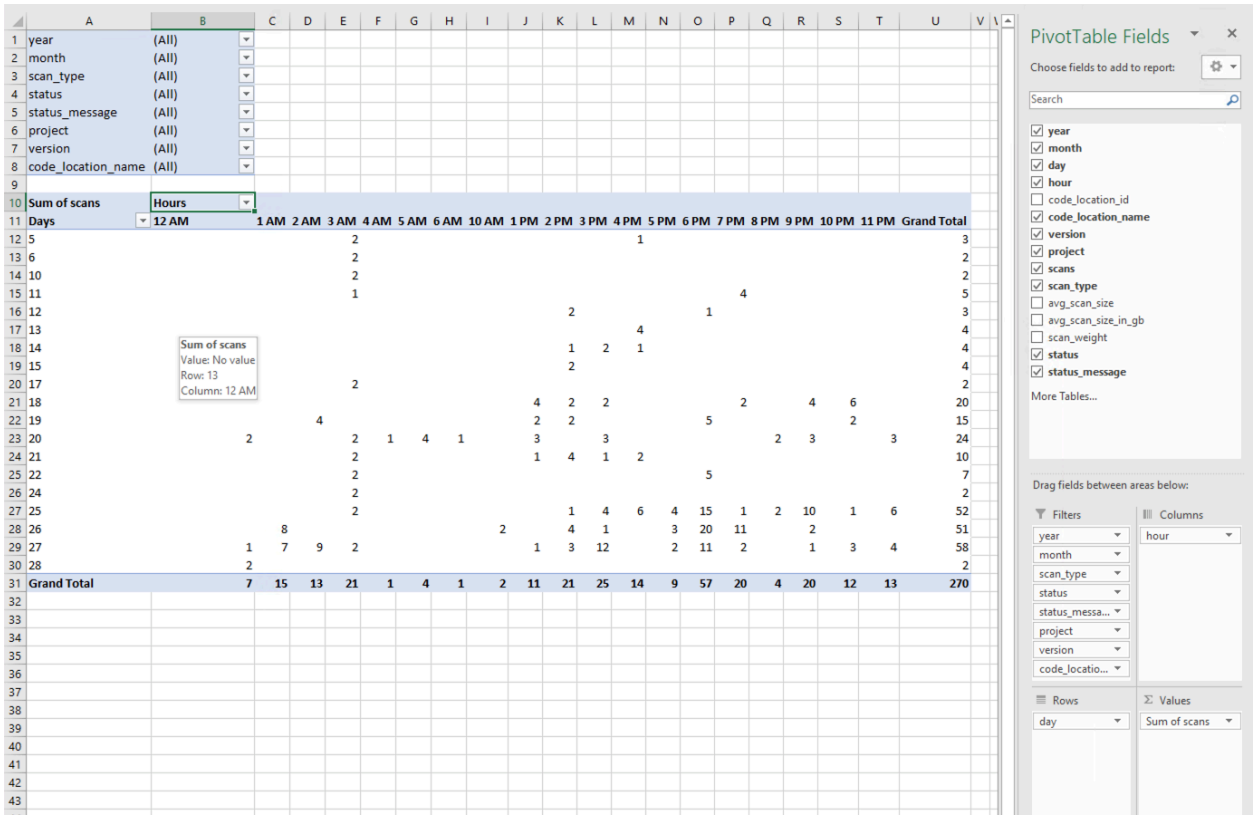


10. Click in the Blank Pivot created, this should open the PivotTable Fields column panel.

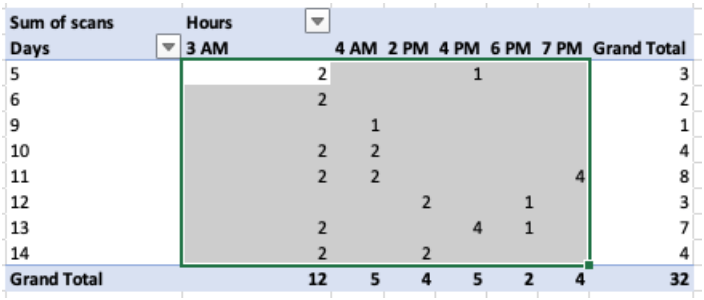

PivotTable Fields
Choose fields to add to report:
Search
year
month
day
hour
code_location_id
code_location_name
version
project
Drag fields between areas below:
Filters
Columns
Rows
Values
Defer Layout Update
Update

11. Drag the following to the Filters section in this order:
    a. Year
    b. Month
    c. Scan_Type
    d. Status
    e. Status_Message
    f. Project, Version
    g. Code_Location_Name
12. Drag the Hour field to the Columns section.
13. Drag the Day field to the Rows section.
14. Drag the Scans field to the Values section. It should show as ‘Sum of Scans’.
15. Rename the column header from 'Row Labels' to Days
16. Rename the 'Column Labels' filter in second column to Hours

    The presentation in Excel should now look like this:

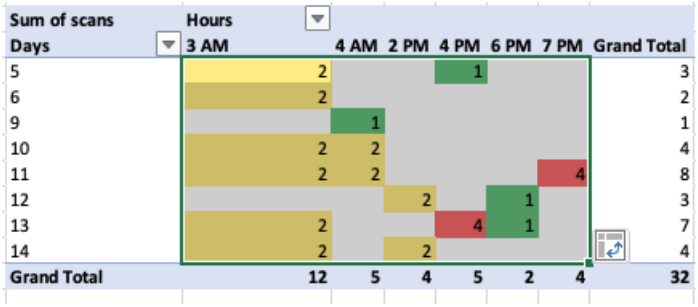


17. Select all the cells that fall within the range between the first to last Hours column (do not include filter top row, ‘Grand Total’ bottom row or last column (Grand Total)



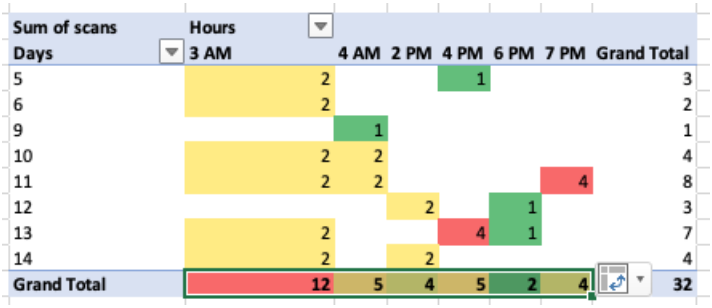
Sum of scans	Hours						
Days	3 AM	4 AM	2 PM	4 PM	6 PM	7 PM	Grand Total
5	2			1			3
6	2						2
9		1					1
10	2	2					4
11	2	2				4	8
12			2		1		3
13	2			4	1		7
14	2		2				4
Grand Total	12	5	4	5	2	4	32

18. Click the Home menu item
19. From the Conditional Formatting > Color Scales, select Red - Yellow - Green scale



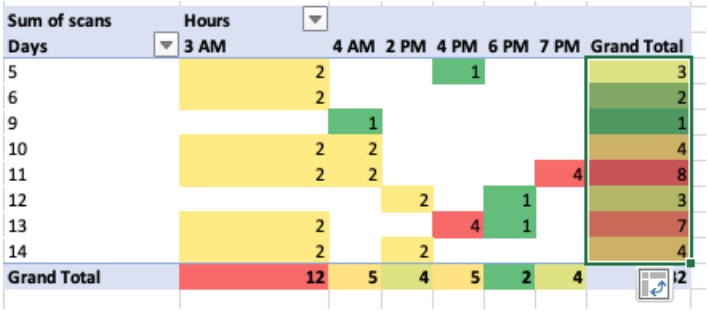
Sum of scans	Hours						
Days	3 AM	4 AM	2 PM	4 PM	6 PM	7 PM	Grand Total
5	2			1			3
6	2						2
9		1					1
10	2	2					4
11	2	2				4	8
12			2		1		3
13	2			4	1		7
14	2		2				4
Grand Total	12	5	4	5	2	4	32

20. Select the cells in bottom Grand Total row and apply the same color scale (excluding first and last column. Only Hours data cells).



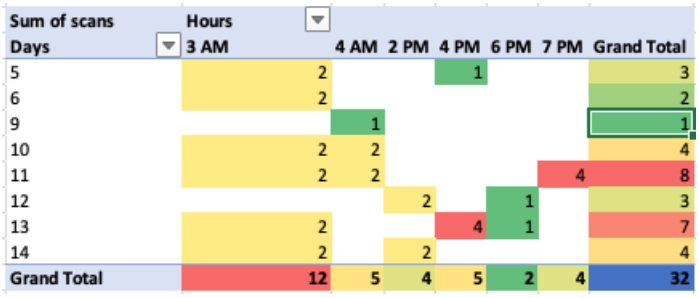
Sum of scans	Hours						
Days	3 AM	4 AM	2 PM	4 PM	6 PM	7 PM	Grand Total
5	2			1			3
6	2						2
9		1					1
10	2	2					4
11	2	2				4	8
12			2		1		3
13	2			4	1		7
14	2		2				4
Grand Total	12	5	4	5	2	4	32

21. Select the cells in the rightmost Grand Total column and apply the same color scale.



Sum of scans	Hours						
Days	3 AM	4 AM	2 PM	4 PM	6 PM	7 PM	Grand Total
5	2			1			3
6	2						2
9		1					1
10	2	2					4
11	2	2				4	8
12			2		1		3
13	2			4	1		7
14	2		2				4
Grand Total	12	5	4	5	2	4	32

22. Select the bottom right corner cell alone and provide it with a blue background.



Sum of scans	Hours						
Days	3 AM	4 AM	2 PM	4 PM	6 PM	7 PM	Grand Total
5	2			1			3
6	2						2
9		1					1
10	2	2					4
11	2	2				4	8
12			2		1		3
13	2			4	1		7
14	2		2				4
Grand Total	12	5	4	5	2	4	32

23. To analyze the data behind any cell, double click on it.

Maximum Scan Size Heatmap

1. Add a blank new sheet to the workbook.
2. In the new worksheet, click on any cell, click on Insert tab, click on Pivot icon.
3. Click on the Blank Pivot created, this should open the Pivot column panel.
4. Drag the following to filter section in this order:
   a. Year
   b. Month
   c. Scan_Type
   d. Status
   e. Status_Message
   f. Project
   g. Version
   h. Code_Location_Name
5. Drag the Hour field to the Columns section.
6. Drag the Day field to the Rows section.
7. Drag avg_scan_size_in_gb field to the Values section
8. Change to show Maximum such that it should display ‘Max of avg_scan_size_in_gb’:
   a. Click on the field in the values section. This will launch the PivotTable field.
   b. Choose Maximum instead of Sum.
   c. Click the OK button.
9. Select all the cells that fall within the range between the first to last Hours column (do not include filter top row, ‘Grand Total’ bottom row or last column (Grand Total)
10. Click the Home tab, Conditional Formatting > Color Scales > Yellow - Green color scale.
11. Select the cells in bottom Grand Total row and apply the same color scale.
12. Select the cells in the rightmost Grand Total column and apply the same color scale.
13. Select the bottom right corner cell alone and provide it with a blue background.

Scan Weight Heatmap

1. Add a blank new sheet to the workbook.

2. In the new worksheet, click on any cell, click on Insert tab, click on Pivot icon.
3. Click on the Blank Pivot created, this should open the Pivot column panel.
4. Drag the following to filter section in this order:
   a. Year
   b. Month
   c. Scan_Type
   d. Status
   e. Status_Message
   f. Project
   g. Version
   h. Code_Location_Name
5. Drag the Hour field to the Columns section.
6. Drag scan_weight field to the Values section and change to show Maximum such that it should display 'Avg of scan_weight'. This can be set by clicking on the field in the values section and then choosing Average instead of Sum.
7. Drag the Scans field to the Values section. It should show as 'Sum of Scans'.
8. Select all the cells that fall within the range between the first to last Hours column (do not include filter top row, 'Grand Total' bottom row or last column (Grand Total)
9. Click the Home tab, Conditional Formatting > Color Scales > Red - White color scale.
10. Select the cells in bottom Grand Total row and apply the same color scale.
11. Select the cells in the rightmost Grand Total column and apply the same color scale.

Viewing heatmap data

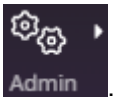
The Black Duck Heatmap provides an intuitive and powerful solution to capture, present, analyze, distribute, and automate data analysis, problem detection, problem identification, and applying limited solutions that are mapped to known problems. Statistical data from Black Duck is represented in a matrix with hour of day as one axis and day (or date) of month as the other axis.

The heatmaps are color coded to display the minimum and maximum value in the dataset and to calculate percentages for each value relative to the minimum and maximum. The smallest number is 0% (green color) and the largest is 100% (red color). We then use this percentage to determine the exact color value to use in the heatmap.



Note: Red values in the heatmap does not indicate an error. It is the maximum boundary when determining the amount of scans conducted in the last 30 days.

To view a UI representation of the heatmap:

1. Log in to Black Duck with one of the following roles:
   - Global Code Scanner
   - Global Notification Viewer
   - Global Project Administrator
   - Global Project Manager
   - Global Project Viewer
2. Click 

.
3. Select **Heatmaps** in the **Diagnostics** section of the Admin menu.

Note: Heatmap data is populated when the scan is completed and becomes read-only afterwards. Changes made to the projects are not synced with heatmap data.

Filtering the heatmap

You can filter the data displayed in the heatmap by clicking the blue **+ Filter** button on the top right of the page and then using any of the following options. Multiple filters can be added simultaneously to further refine the results:

- **Code Location ID**: Select one or many code location IDs.
- **Code Location Name**: Select one or many code location names.
- **Project Name**: Select one or many project names.
- **Scan Date**: Select a start and end date.
- **Scan Status**: Select from Success, Started, and/or Failure.
- **Scan Type**: Select from the available scan types.
- **Version Name**: Select any version names.

You can edit filters by clicking on an existing filter and adding additional criteria. You can remove an active filter by clicking the × button to the right of the filter.

Administering user accounts

There are two ways to administer user accounts in Black Duck:

1. Administering user accounts manually. A user with the User Administrator role can:
   - Navigating the Users & Groups page
   - Add a new user account
   - Inactivate a user account
   - Change user account information
   - Change a user's password
   - 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.

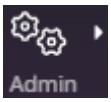
Note that you can also create external user accounts.

Users with the System Administrator role can also configure the password requirements for user accounts.

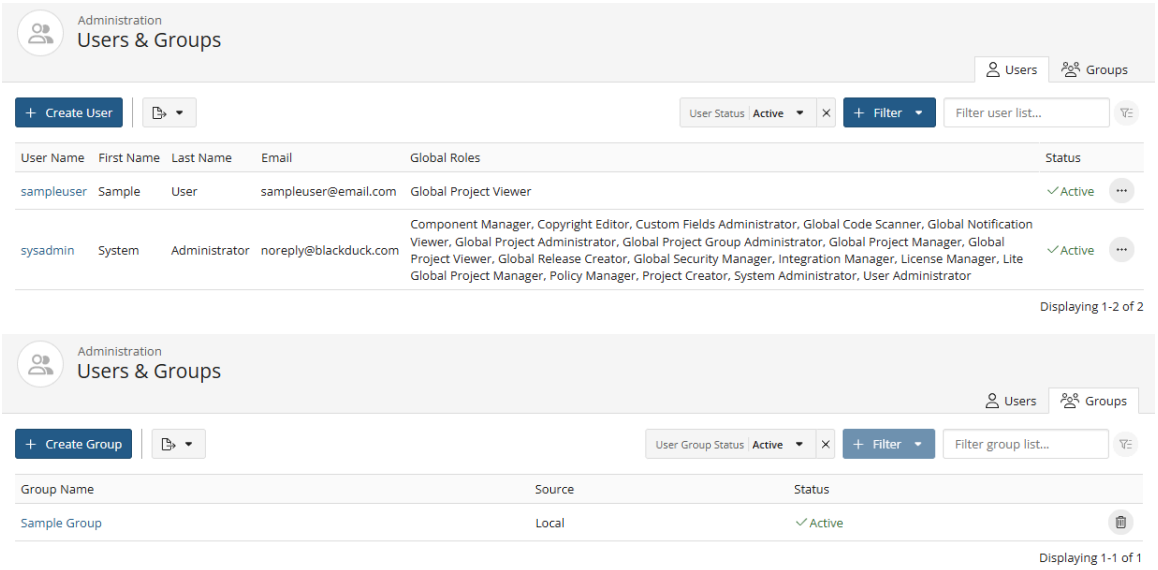
Navigating the Users & Groups page

The **Users & Groups** page provides a comprehensive view of user details, including roles, status, and group memberships. This allows administrators to quickly review access permissions and ensure users have the appropriate privileges within Black Duck.

To view the Users and Groups page:

1. Log in to Black Duck SCA.
2. Click 

→ **Users** to view the list of users or → **Groups** to view the list of groups.



Understanding the Users & Groups tables

The **Users & Groups** page contains two tabs—**Users** and **Groups**— each displaying a table providing an overview of users or groups within Black Duck. These tables help administrators efficiently manage access and permissions in Black Duck.

The **Users** tab table lists all registered users and their key attributes, including:

- **Username**: The unique identifier for the user.
- **First Name** / **Last Name**: The user's full name.
- **Email**: The email address associated with the account.

- **Global Roles**: Displays the user's global roles. Project-level roles are not shown in this column.
- **Status**: Indicates if the user accounts is active or inactive.

The **Groups** tab table displays all user groups and their details, including:

- **Group Name**: The name of the user group.
- **Source**: Indicates where the group originates from:
  - **Local**: A group manually created within Black Duck.
  - **External**: A group synchronized from an external authentication provider (LDAP or SAML).
- **Status**: Indicates if the user group is active or inactive.

Managing users and groups

The **Users & Groups** page allows administrators to manage individual users and user groups directly from their respective tabs.

In the **Users** tab, you can click the username or the button to manage the following:

- **User Details**: View and edit the user's full profile, including email, status, and authentication source. You can also inactivate a user from here.
- **Global Roles**: Review and modify the user's assigned global roles (project-level roles are not displayed here).
- **Project Groups**: The Projects Groups page lists the project groups to which this user belongs. In this section, you can select a group name to view the *Group Name* page from which you can manage group information, group roles, and group membership by adding or removing this user from one or more groups.
- **Project**: View a list of projects where the user has assigned roles.
- **User Groups**: The User Groups page lists the user groups to which this user belongs. In this section, you can select a group name to view the *Group Name* page from which you can manage group information, group roles and group membership.

  Users can view the user groups that they belong to by using the Profile page.

In the **Groups** tab, administrators can view and manage groups, including:

- **Edit group**: Click the group name to view and edit the project group details, associated roles, related projects and groups, and group membership by adding or removing users.
- **Delete group**: Delete the desired user group.

Exporting to CSV

You can export the list of users or groups to CSV which converts the individual rows to tabular data. To do so, click either the **Users** or **Groups** tab on the top right, click the button, and select CSV.

Creating a user account

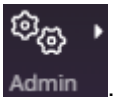
You can create a Black Duck user account for a local user (an internal user account) for an external user (such as a user managed by an external source, such as LDAP).

If you have enabled LDAP, you can create users on your LDAP server instead of in Black Duck SCA. 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.

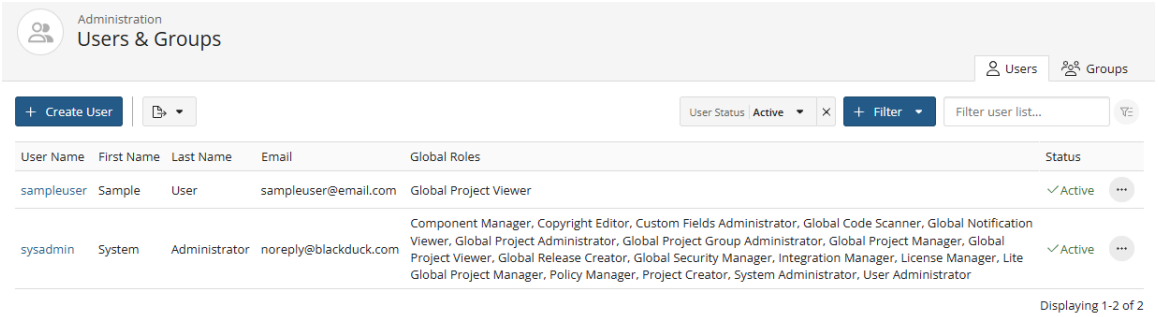
Note that with external user accounts:

- 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.

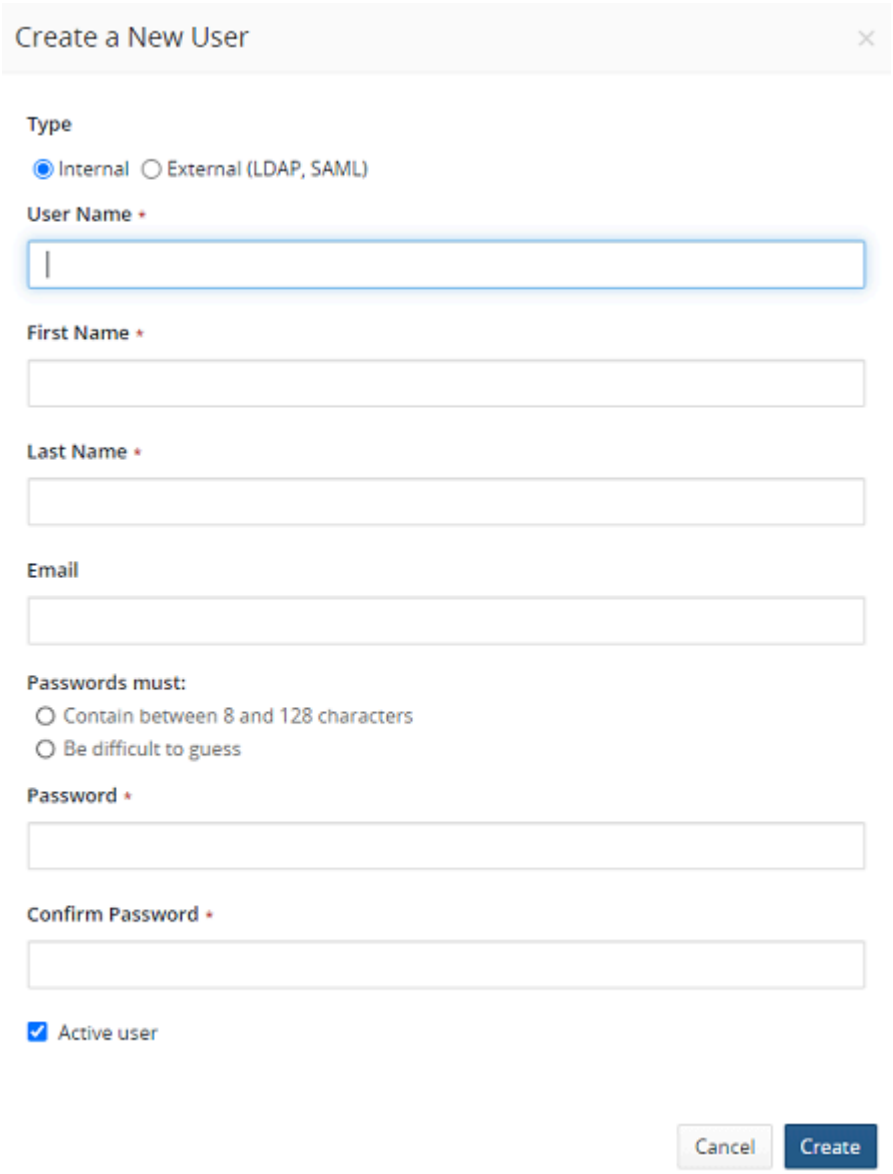
To create a user account:

1. Log in to Black Duck SCA.
2. Click 

.
3. Select **Users** to display the **Users & Groups** page.



4. Click **+ Create User**. The Create a New User dialog box appears.



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.

Understanding roles

Black Duck SCA uses roles to control what users can view and manage. Roles define the actions users can perform and the information they can access. Users can be assigned roles individually or through group membership. Permissions from multiple roles are combined.

If no roles are assigned, users have read-only access to Black Duck and to the projects to which they are assigned.

Role types

Black Duck provides three types of roles:

- **Global roles**

  Global roles grant permissions across the entire Black Duck SCA environment. These roles are typically assigned to users who perform administrative, security, compliance, reporting, or system-wide management tasks.

  Common global roles include:

Role	Purpose
System Administrator	Configure and administer Black Duck
User Administrator	Manage users, groups, and access
Global Project Administrator	Manage projects and project versions across the system

Role	Purpose
Global Project Manager	Manage projects, project versions, and BOMs across the system
Global Project Viewer	View all projects and BOMs
Global Code Scanner	Manage scans across all projects
Component Manager	Manage custom components
License Manager	Manage licenses and license data
Policy Manager	Manage policy rules
Global Security Manager	Manage remediation statuses
Integration Manager	Manage integrations

For detailed role capabilities, see **Global roles reference**

- **Project roles**

  Project roles apply only to the projects to which a user has been assigned. Use project roles when users need access to specific projects without receiving broader system-wide permissions.

  Common project roles include:

Role	Purpose
Project Administrator	Manage project settings and membership
Project Manager	Manage project versions, BOMs, and project membership
BOM Manager	Manage BOM data
BOM Annotator	Add comments and update BOM component custom fields
Project Code Scanner	Manage scans for assigned projects
Security Manager	Remediate vulnerabilities
Policy Violation Reviewer	Manage policy overrides
Project Viewer	View project information and reports

  For detailed role capabilities, see **Project roles reference**.

- **Project group roles**

  Project group roles provide the same capabilities as their project-role counterparts but apply to every project within an assigned project group. They help administrators manage permissions consistently across multiple related projects.

  Examples include:

  - Project Group Administrator
  - Project Manager
  - Project Code Scanner
  - BOM Manager

- BOM Annotator
- Security Manager
- Policy Violation Reviewer
- Project Viewer

For detailed role capabilities, see **Project group roles reference**.

Assigning roles

Roles can be assigned directly to users or to groups. When a role is assigned to a group, all group members inherit that role and its permissions.

Users can also receive project-level permissions when they are assigned a role within a project or project group.

How permissions are combined

Users can have multiple roles. A user's effective permissions are the combination of all assigned global, project, and project group roles.

For example, a user might:

- Have a global role that allows project administration.
- Have a project role that allows BOM management.
- Have a project group role that provides access to multiple projects.

Together, these roles determine the user's available actions within Black Duck SCA.

Direct and indirect access

Project groups introduce two methods of access:

Direct access

The user is assigned directly to a project.

Indirect access

The user gains access through a project group assignment or through membership in a user group that is associated with a project group.

Project group roles allow permissions to be applied consistently across multiple projects without assigning users to each project individually.

Choosing the appropriate role type

If the user needs to...	Assign...
Administer Black Duck SCA or manage resources across all projects	A global role
Work within specific projects only	A project role
Work across multiple projects in the same project group	A project group role

Related information

- Global roles reference
- Project roles reference
- Project group roles reference
- Black Duck SCA user role matrix

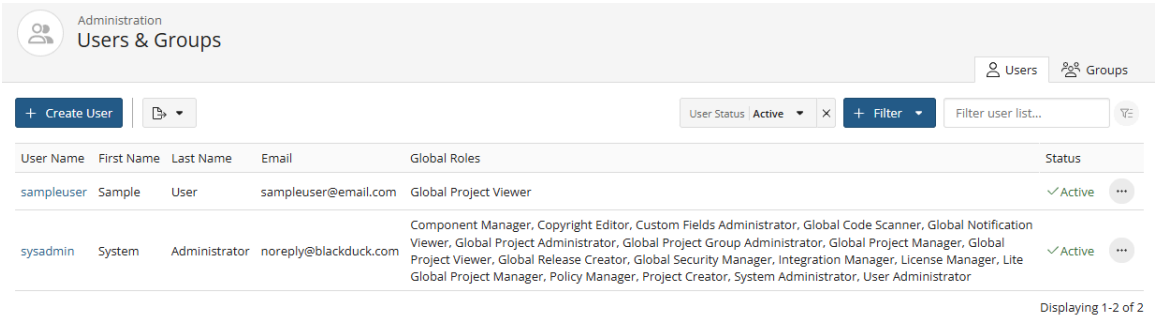
Managing user roles

Once you have created a user account, you can add global roles to the user account. These global 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.

To assign a global role to a user:

1. Click Admin → **Users**.



   The roles assigned to each user appears on the page.
2. Find the user to whom you want to assign a role:
   - Add the **Inactive** option to the **User Status** filter to include inactive users.
   - 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.
3. Select the username to display the *Username* page.
4. In the **Global 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 **Global Roles** section; instead the **User Groups** section lists roles for each user group.

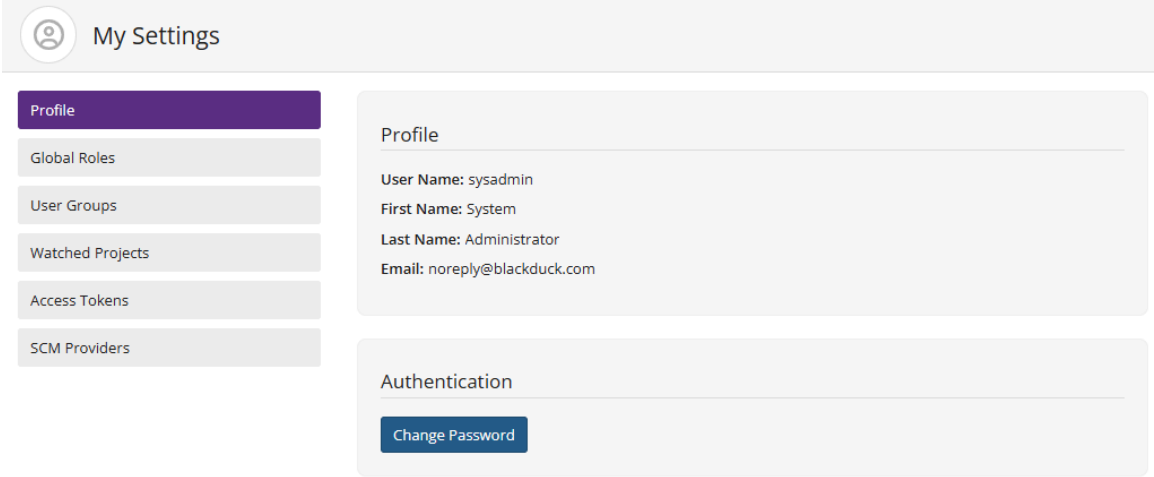
Viewing your roles

Use the Profile page to view the roles assigned to your user account.

To view your roles:

1. Log in to Black Duck SCA.
2. From the user menu located on the top navigation bar, select **Profile**.

   The Profile page appears.



   Select **Global 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 User Administrator role can view the roles assigned to a user account by selecting the username in the Users & Groups page.

Black Duck SCA 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.

Note: Each project has exactly one **Project Owner**. The Project Owner is assigned by default when a project is created and possesses the standard capabilities of the **Project Manager** role for that project.

Global roles by task

Task	Roles (details or restrictions)
Manage code scans/Protex BOM files: • Scan code • Upload scans to Black Duck SCA. • Map or unmap scans to projects • Delete scans	• Global Project Administrator (Map/unmap/delete scans only) • Global Project Manager (Map/unmap/delete scans only) • Global Code Scanner
Create, edit, delete projects	• Global Project Administrator • Global Project Manager • Project Creator (edit/delete solely the project created by the corresponding user; automatically designated as the Project Owner)
Add or remove users from a project	• Global Project Administrator (add users with a defined role)

Task	Roles (details or restrictions)
	• Global Project Manager (Add users but cannot define their roles. Users added to projects by a global project manager will have read only access to the projects and will not be able to edit or modify the BOM.) • User Administrator (add users with a defined role) • Global Project Group Administrator (add users with a defined role) **Note:** If the appropriate Role setting is not enabled, additional restrictions may apply to role assignment.
Manage projects versions: • Create, edit, delete project versions • Edit project or version settings, including tags	• Global Project Administrator • Global Project Manager • Global Release Creator (Create permission only) • Project Creator (See Project Manager role for permissions obtained when creating a project)
Manage custom components	• Component Manager • Global Project Administrator • Global Project Manager
Manage licenses: • Create, edit, delete custom licenses • Manage KnowledgeBase licenses • Create, edit, delete custom license families • Manage KB and custom license terms	• License Manager
View BOMs: • View BOM • Add/edit/view comments • Print BOM • Compare BOMs	• Global Project Administrator (Cannot edit comments created by other users) • Global Project Manager • Global Project Viewer (View all projects only) **Important:** Users with only the **Global Project Viewer** role cannot view SCM project BOMs. To create, view, and manage SCM projects, users must have either the **Lite Global Project Manager** role or the **Integration Manager** role. • Any other user assigned to the project
Manage BOMs: • Manually add components; delete manually added components • Ignore components • Review components • Remediate security vulnerabilities • Override policy violations • 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	• Global Project Manager

Task	Roles (details or restrictions)
• Indicate license term fulfillment status • Manage deep license data • View license conflicts	
Manage policy rules	• Policy Manager (can create global policy rules, as well as edit and delete group-scoped policy rules) • Global Project Admin + Policy Manager (can create global or project group scoped policy rules for any project groups) • Project Group Admin + Global Policy Manager (can create global or project group-scoped policy rules for their project groups)
Update Additional (Custom, SBOM) fields	• Component Manager (Can only update Component and Component Version custom fields) • Custom Fields Administrator (Can only create, edit, delete Custom Fields) • Global Project Administrator (Update custom field values for project, project version, and BOM)
Create, edit, or delete global remediation statuses	• Global Project Administrator • Global Project Manager • Global Security Manager (Must be assigned to a project to view data)
Run project vulnerability reports from the Reports menu	The following roles can create a project vulnerability report for any project: • Global Project Manager • Global Project Administrator • Global Project Viewer The following roles must be assigned to a project to create a project vulnerability report: • Component Manager • Copyright Editor • Custom Fields Administrator • Global Code Scanner • Global Project Group Administrator • Global Release Creator • Global Security Manager • License Manager • Policy Manager • Project Creator • System Administrator • User Administrator
Create and modify copyright statements	• Copyright Editor (Must be assigned to a project to view data)
Run Project version reports: • Version Details report	The following roles can create a project version report for any project:

Task	Roles (details or restrictions)
• Vulnerability report • Notices File report • Software Bill of Materials (SBOM) report	• Global Project Manager • Global Project Administrator The following roles must be assigned to a project to create a project version report: • Component Manager • Copyright Editor • Custom Fields Administrator • Global Code Scanner • Global Project Group Administrator • Global Project Viewer • Global Release Creator • Global Security Manager • License Manager • Policy Manager • Project Creator • System Administrator • User Administrator
Delete Project version reports	The following roles can delete a project version report for any project: • Global Project Manager • Global Project Administrator The following roles can only delete project version reports created by themselves: • Component Manager • Copyright Editor • Custom Fields Administrator • Global Code Scanner • Global Project Group Administrator • Global Project Viewer • Global Release Creator • Global Security Manager • License Manager • Policy Manager • Project Creator • System Administrator • User Administrator
View information in Dashboard pages	The following roles can view any project from the Dashboard page: • Global Project Manager • Global Project Administrator The following roles can only view any projects to which they are associated on the Dashboard page: • Component Manager • Copyright Editor • Custom Fields Administrator • Global Code Scanner

Task	Roles (details or restrictions)
	• Global Project Group Administrator • Global Project Viewer • Global Release Creator • Global Security Manager • License Manager • Policy Manager • Project Creator • System Administrator • User Administrator
Access the Tools page: • Download the scanner • Access links to the Community and Customer Education	All roles
Use the Search function	All roles
Administer Black Duck SCA. Use the Admin menu to: • View jobs • Register Black Duck SCA. • Configure LDAP • Configure SAML • Manage system settings • Manage system announcements • Configure password requirements	• System Administrator
Administer users and groups. Use the Admin menu to: • Manage users, including resetting passwords • Manage groups	• User Administrator
Manage snippets	• Global Project Manager • Global Project Administrator
View issues	• Global Project Manager • Global Project Administrator
Manage project groups: • Create/Edit/Delete project groups • Add/Remove members and user groups from project groups	• Global Project Group Administrator
Manage Access Tokens	• User Administrator
View notifications	• Global Notification Viewer (View notifications for all projects and receives all system notifications regardless of user preferences)

Task	Roles (details or restrictions)
Download the heatmap CSV report	• System Administrator
View the scan heatmap	• Global Project Administrator • Global Project Manager • Global Notification Viewer • Global Project Viewer • Global Code Scanner
Manage integration servers	• Integration Manager
Manage lightweight BOMs	• Lite Global Project Manager
Convert project versions to LTS	• Global Project Administrator • Global Project Manager
Enable/disable Multi-Factor Authentication (MFA)	• User Administrator

Project and Project Group roles

Task	Roles (details or restrictions)
Manage project groups: • Create/Edit/Delete project groups • Add/Remove members and user groups from project groups	• Project Group Administrator (below parent group)
Manage code scans/Protex BOM files: • Scan code • Upload scans to Black Duck SCA. • Map or unmap scans to projects • Delete scans	• Project Manager / Project Owner (Can unmap/delete scans from their projects) • Project Group Manager (Can unmap/delete scans from their projects) • Project Code Scanner (Can map/unmap/delete a code scan to/from projects for which they have access) • Project Group Code Scanner
Create, edit, delete projects	• Project Administrator (Delete/Edit only) • Project Group Administrator (must already have access to these projects) • Project Manager / Project Owner (cannot create projects but can delete projects to which they are associated; each project has exactly one Project Owner)
Manage projects: • Create, edit, delete project versions • Edit project or version settings, including tags	• Project Administrator • Project Manager / Project Owner (Only projects they manage or own) • Project Group Administrator (must already have access to these projects) • Project Code Scanner (Can only create project versions)

Task	Roles (details or restrictions)
Add or remove users or groups to projects	• Project Administrator (add users with a defined role on projects they administer) • Project Manager / Project Owner (Add users but cannot define their roles on projects they administer. Users added to projects by a project manager or project owner default to read-only access [Project Viewer] and will not be able to edit or modify the BOM.) • Project Group Administrator (must already have access to these projects) **Note:** Project Owners and Project Managers cannot administer or assign user roles to project members. Users added to a project by a Project Owner or Project Manager default to read-only access (the Project Viewer role). To grant users the ability to assign user roles on projects, assign them a project-group-scoped **Project Administrator** or **Project Group Administrator** role. If the appropriate Role setting is not enabled, Project Managers and Project Group Administrators cannot assign the Security Manager role at the project level.
Manage custom licenses: • Create, edit, delete custom licenses	• BOM Manager
View BOMs: • View BOM • View notifications • Add/edit/view comments • Print BOM • Compare BOMs	• All roles
Manage BOMs: • 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	• Project Administrator (Update Project and Project Version custom fields, view BOM custom fields) • Project Manager / Project Owner • Project Group Administrator (must already have access to these projects, Update Project and Project Version custom fields, view BOM custom fields). • BOM Manager
Manage policy violations: • Override policy violations • Remove override of policy violations	• Project Manager / Project Owner (Can only manage policy violations if enabled by the system administrator) • Policy Violation Reviewer

Task	Roles (details or restrictions)
Remediate security vulnerabilities	• Project Manager / Project Owner (Can only remediate security vulnerabilities if enabled by the system administrator) • Security Manager (Can only modify remediation for vulnerabilities associated with components)
Update custom field values	• Project Manager / Project Owner (Can only update BOM Component, Project, and Project Version custom fields) • BOM Annotator (Can only update BOM Component custom field) • BOM Manager (Can only update BOM Component custom field)
Manage policy rules: • Create, edit, or delete policy rules	• No project or project group level role can perform this task
Run project vulnerability reports from the Report menu: • Vulnerability Remediation Report • Vulnerability Status Report • Vulnerability Update Report	• All roles
Run Project version reports: • Version Details report • Vulnerability report • Notices File report • Software Bill of Materials (SBOM) report	• All roles
Delete project version reports	The following roles can delete all reports: • Project Administrator • Project Manager / Project Owner • Project Group Administrator (must already have access to these projects) The following roles can delete reports generated by themselves: • All roles
View information in Dashboard pages	• All roles
Access the Tools page from which user can: • Download the scanner • Access API documentation	• All roles
Search	• All roles
Manage snippets	• Project Administrator • Project Manager / Project Owner • BOM Manager

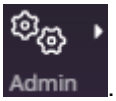
Task	Roles (details or restrictions)
Convert project versions to LTS	• Project Administrator • Project Manager / Project Owner

Managing the Project Manager and Project Group Manager roles

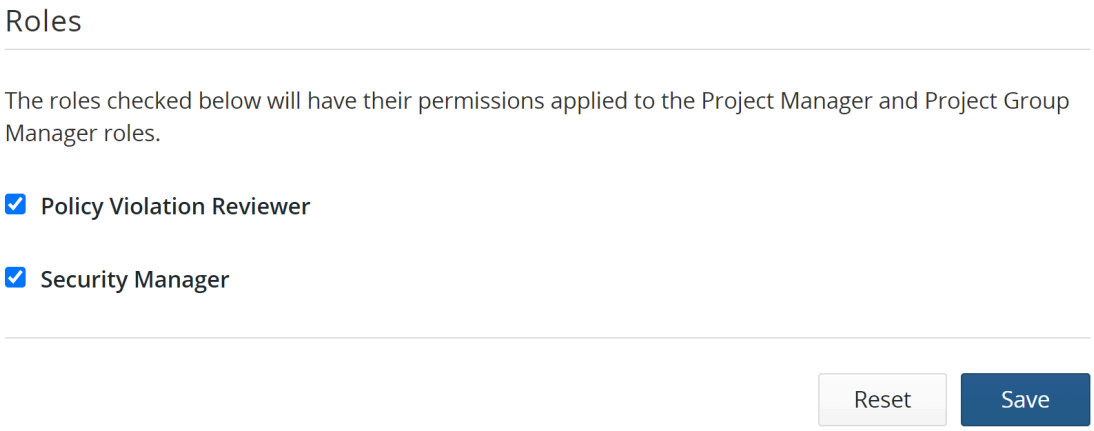
System administrators can define whether users with the Project Manager or Project Group Manager role can manage policy violations (override policy violations or remove overrides) or remediate security vulnerabilities for a project.

Note: This is a global setting: all users with the Project Manager or Project Group Manager role are affected by any changes you make to the role.

To modify the Project Manager or Project Group Manager role:

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

.
3. Select **System Settings**.
4. Click **Roles**.
5. In the **Roles** section, select or clear the **Policy Violation Reviewer** and/or **Security Manager** options.



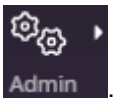
6. Click **Save**.

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.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 SCA.
2. Click 

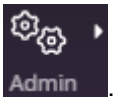
.
3. Select **Users** to display the **Users & Groups** page.
4. Find the user whose information you want to change:
   - Filter the users that appear on the page.
   - Sort the list of users by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.
   - 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*'s User Details page.
6. Select the **User Details** tab.
7. Enter the updated information in the **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.
8. Click **Update**.

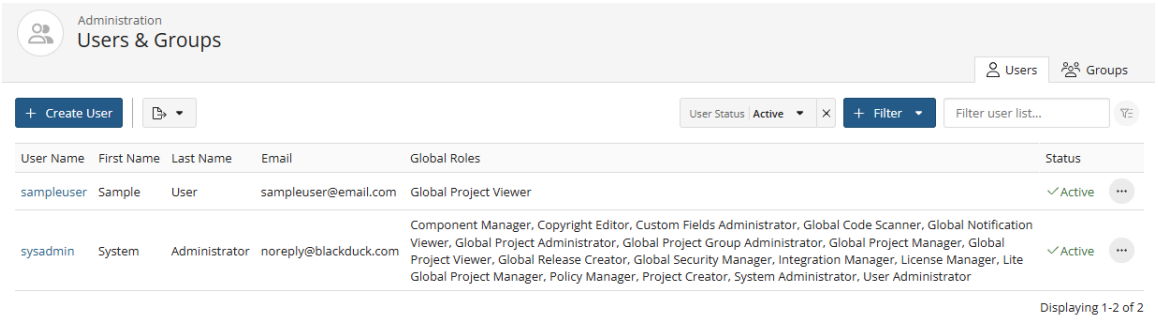
Viewing a user's projects and groups

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 SCA.
2. Click 

.
3. Select **Users** to display the **Users & Groups** page.



4. Find the user you want to find:
   - Filter the users that appear on the page.

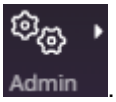
- Sort the list of users by selecting any of the column names. An arrow next to the column name indicates the direction the list is sorted.
- Use the pagination bar at the bottom of the list to go to the appropriate page if there are more users than are listed on this page.

5. Select the user to display the *Username* page.
6. Click **Projects** in the left-hand menu.
7. The **Projects** page 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.

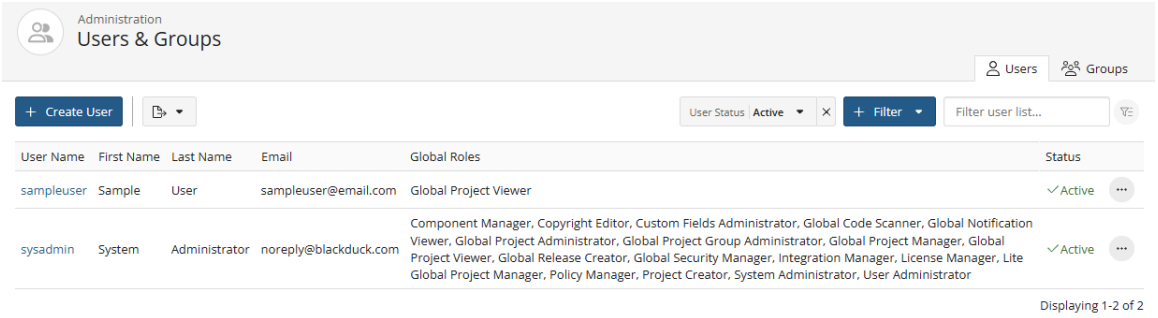
Inactivating a user

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 inactivate a user account:

1. Log in to Black Duck SCA.
2. Click 

.
3. Select **Users** to display the **Users & Groups** page.



4. Find the user you want to inactivate:
   - 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*'s User Details page.

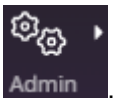
6. Clear the **Active user** check box in the **Internal** or **External User Details** section and click **Save**.

Converting a user account

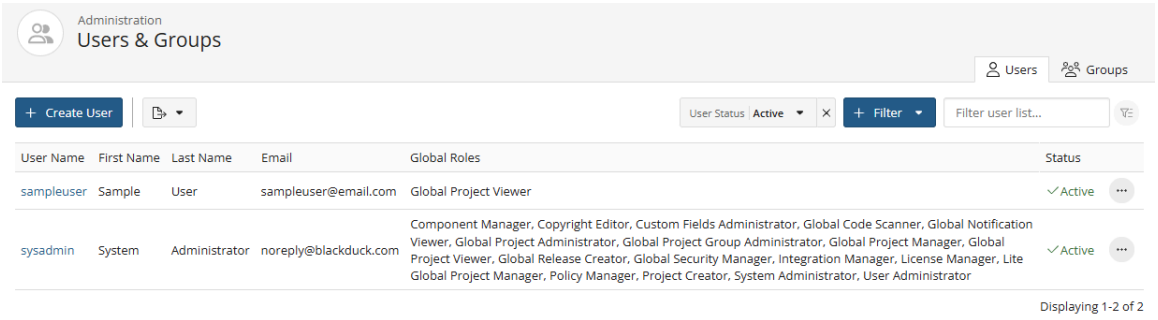
You can convert an internal account to an external account or an external account to an internal account.

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 SCA.
2. Click 

.
3. Select **Users** to display the **Users & Groups** page.



4. Select the username of the account you wish to convert. The *Username*'s User Details page appears.

   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**.

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.

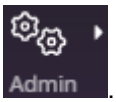
  Note that when using the Black Duck APIs, users with a password that does not meet your requirements will receive a 412 response code which will include the reason why the current password does not meet requirements.

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 

.
3. Select **System Settings**.
4. Click**Local Authentication**.

5. Add or remove the check in the **Enable Password Settings** checkbox to enable or disable password settings.
6. 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. Click **Save**.

Changing your Black Duck password

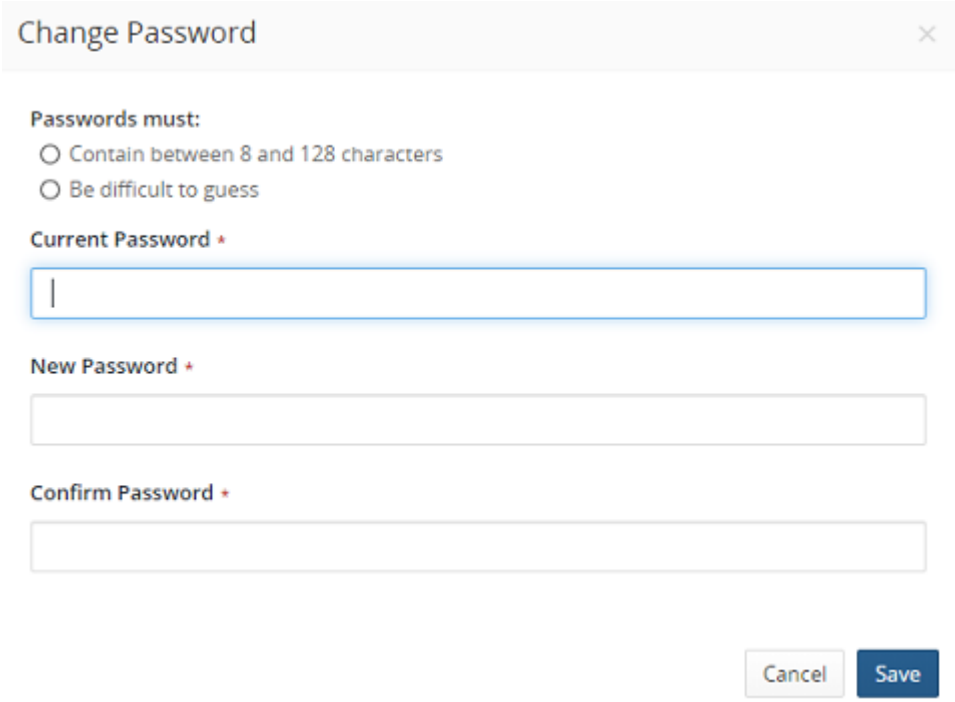
Note: If your system administrator has enabled LDAP on the Black Duck server, user account information and passwords are managed by LDAP. You cannot change your password in Black Duck.

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 User Administrator role can change it for you.

To change your password:

1. Log in to Black Duck SCA.
2. From the user menu located on the top navigation bar, select **Profile**.
3. Click **Change Password** to display the Change Password modal.



4. Type your current password in the **Current Password** field.

5. Type your new password in the **New Password** field.

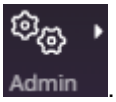
   If there are password requirements, those requirements are listed in the dialog box. Black Duck notes when each requirement is met as you type your new password. You will not be able to save this password if it does not meet *all* requirements.

6. Type the same new password in the **Confirm Password** field.
7. Click **Save**.

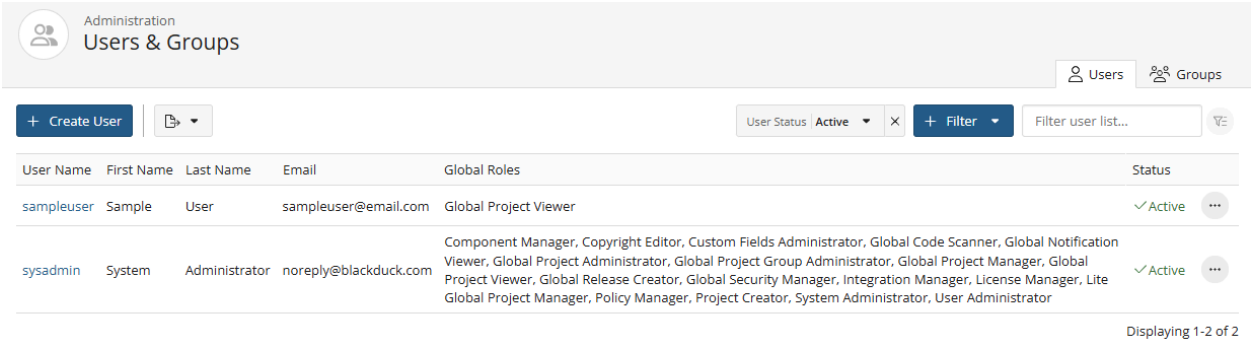
Changing a user's password

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 SCA.
2. Click 

.
3. Select **Users** to display the **Users & Groups** page.



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. 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**.

Resetting your or a user's MFA configuration

If the user needs to reset their Multi-Factor Authentication (MFA) configuration, they can easily do so from their Profile page. Alternatively, a user administrator user can initiate the reset process for any user through the Users page. Resetting the MFA will require the user to go through the initial MFA setup process again, including scanning the QR code and linking the account with an authenticator app.

To reset your MFA configuration as a user:

1. Go to your Profile page by clicking your *name* user button on the top right of the page and then select **Profile**.
2. Under the **Profile** tab, click the **Reconfigure MFA** button found in the **Authentication** section.

   Warning: This action cannot be undone.

To reset a MFA configuration as a user administrator:

1. Click **Admin** → **Users**.
2. Select the desired user from the list displayed.
3. Click the **Reset User MFA** button in the **Local Authentication** section. The user will then be asked to reconfigure MFA upon their next login.

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.

Administering user groups

You can use user groups in Black Duck to manage global 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 user group
- Manage user group information such as the group name or status
- Manage user group global roles
- Add or remove a member from a user group
- Delete a user group

Note: If you are using an external LDAP directory server to authenticate users and have enabled LDAP group synchronization, the Users & Groups 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 user groups

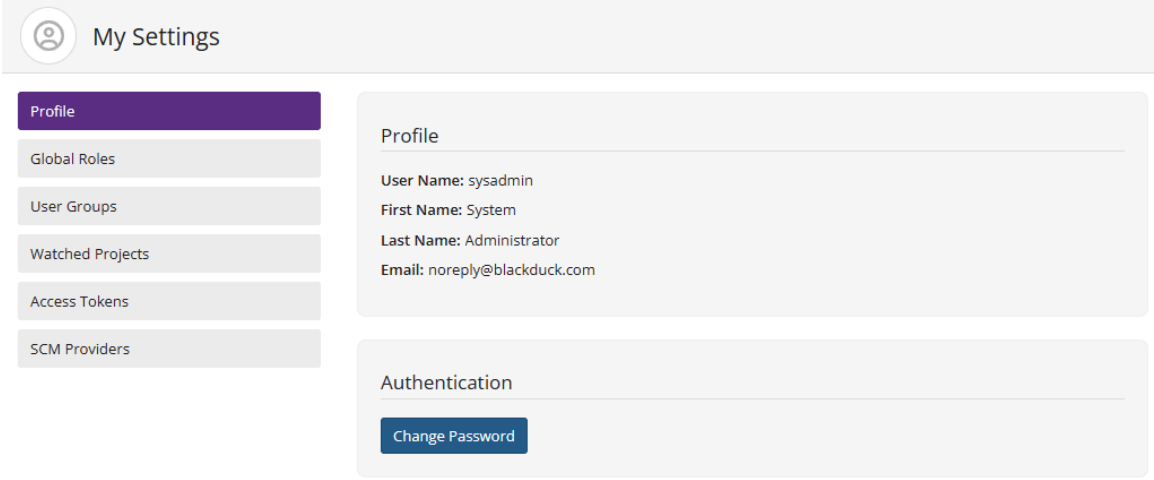
You can view the user groups you belong to, and the source, status, and roles associated with each user group.

To view your user groups:

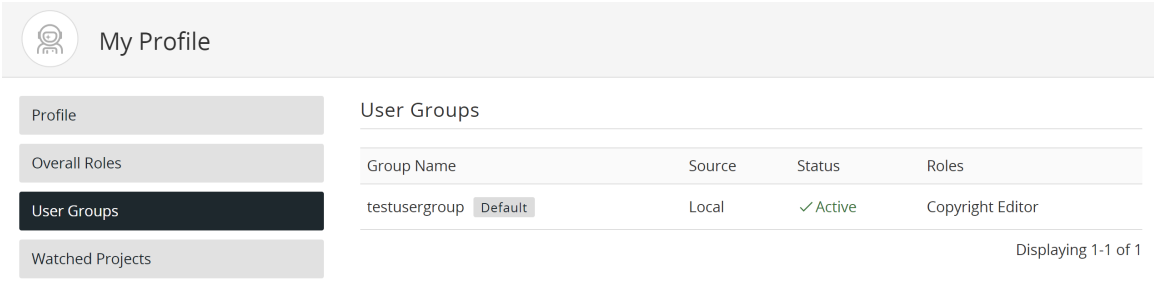
1. Log in to Black Duck SCA.

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

   The Profile page appears.



3. Select **User Groups** to view the group in which you are a member.



You can view the user groups associated with a particular user.

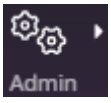
Creating user groups

You can create and configure a group with specific roles that will be granted to all members of the group.

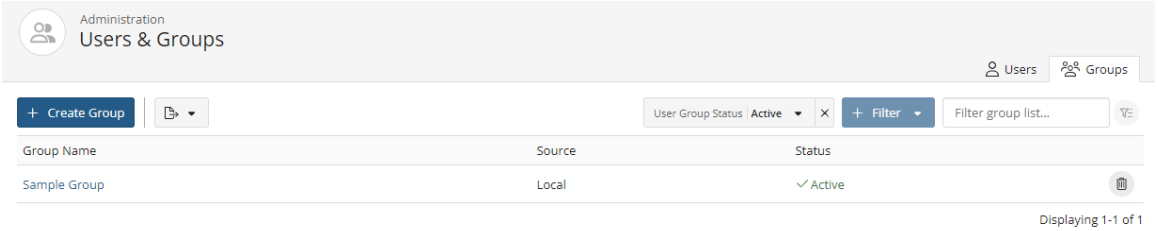
If you create a default group, subsequent new users are automatically added to this group and are granted all roles and access to all projects configured for this group. Note that:

- You can have more than one default group.
- Default groups have a status of *Status* - Default.

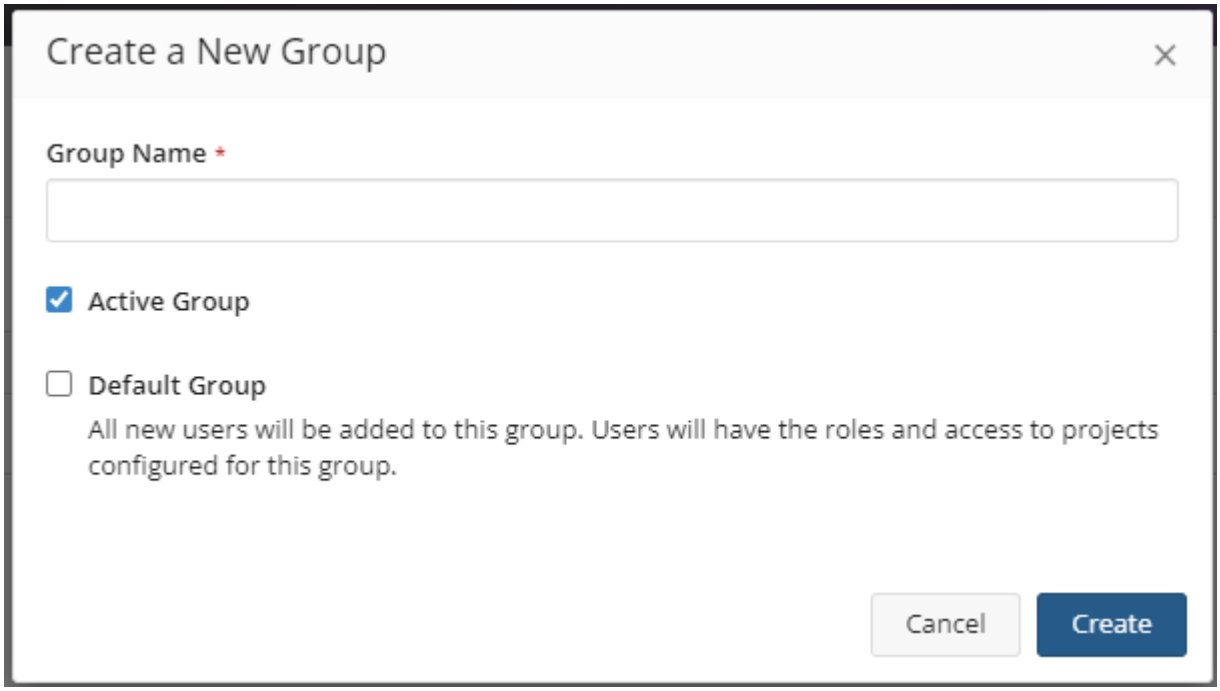
To create a group:

1. Log in to Black Duck SCA.
2. Click 

→ **Groups**.



3. Click **+ Create Group** to display the Create a New Group dialog box.



4. In the Create a New Group dialog box:
   a. 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:

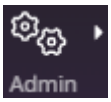
- Add members to the group.
- Assign roles to the group.

Managing user group information

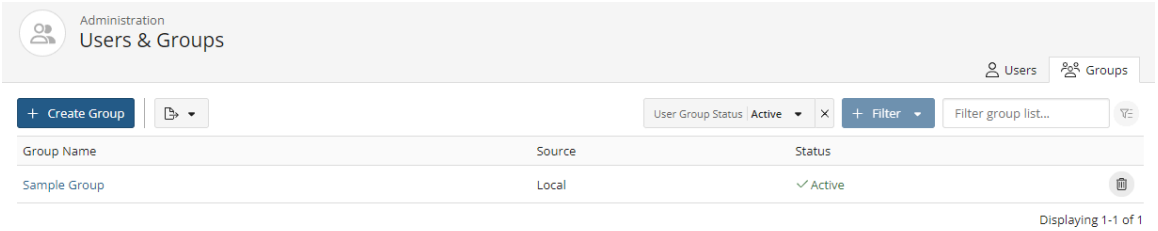
After you have created a user group, you can change the user group name, status (active/inactive), and/or whether this is a default group.

To manage user group information:

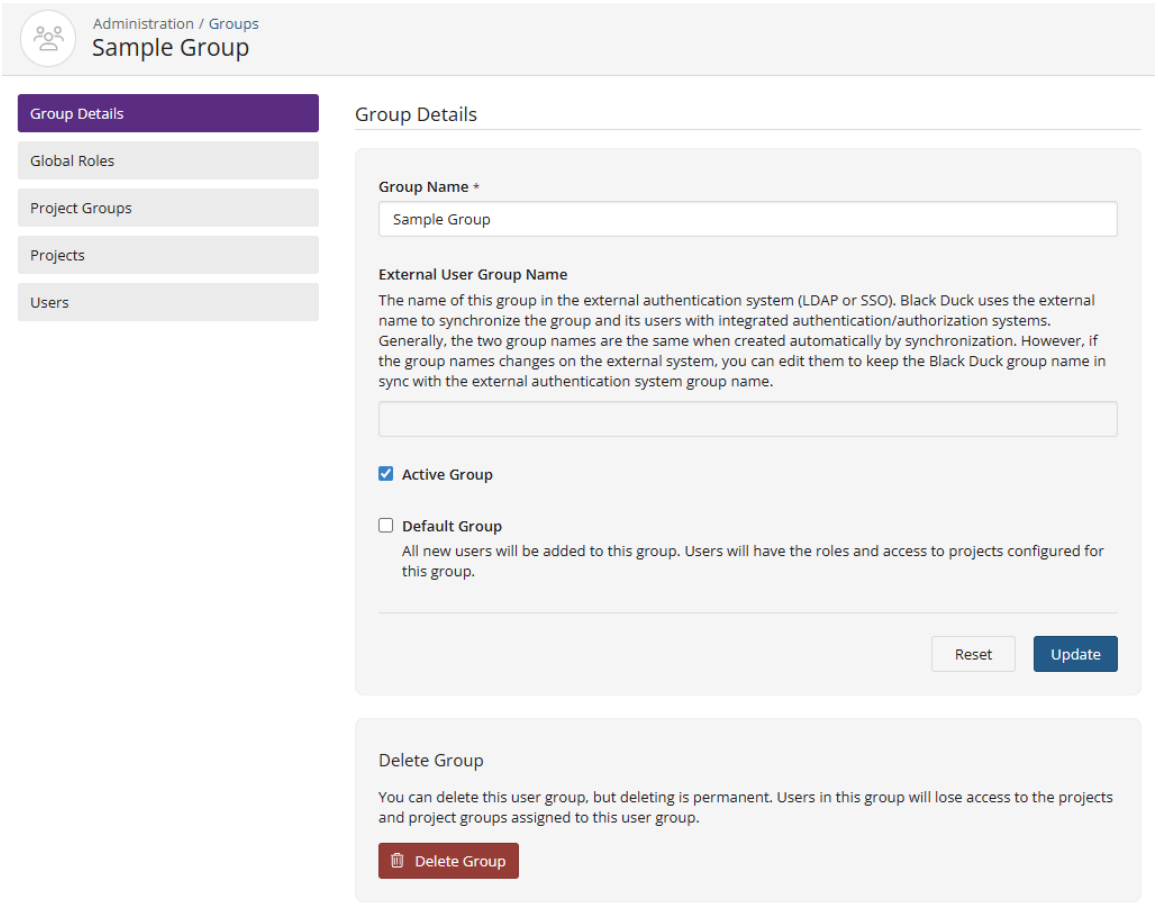
1. Log in to Black Duck SCA.

2. Click 

→ **Groups**.



3. Find the name of the group whose name you want to modify:
   - Add the **Inactive** option to the **User Groups Status** filter to include inactive groups.
   - 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.
4. Select the group name you want to edit to display the *Group Name* page.



5. On the **Group Details** page, type the new group name, change the status, or change whether this is a default group.

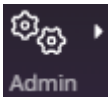
Note that if you enabled group synchronization when configuring LDAP or SAML, the name of this group in the external authentication system (LDAP or SSO) appears in the **External Group Name** field. Black Duck uses the external name to synchronize the group and its members with integrated authentication/ 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.

6. Click **Save** to save the changed information.
7. Use the other sections on this page to:
   - Manage user group roles.
   - Add or remove user group members.
   - Add or remove projects.

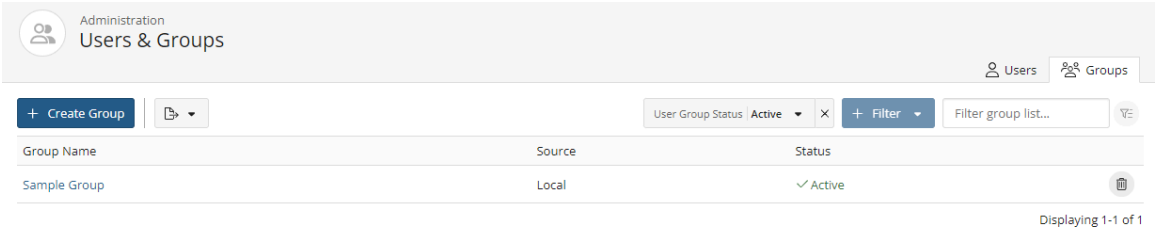
Managing user group projects

You can manage the projects assigned to a user group using the *Group Name* page.

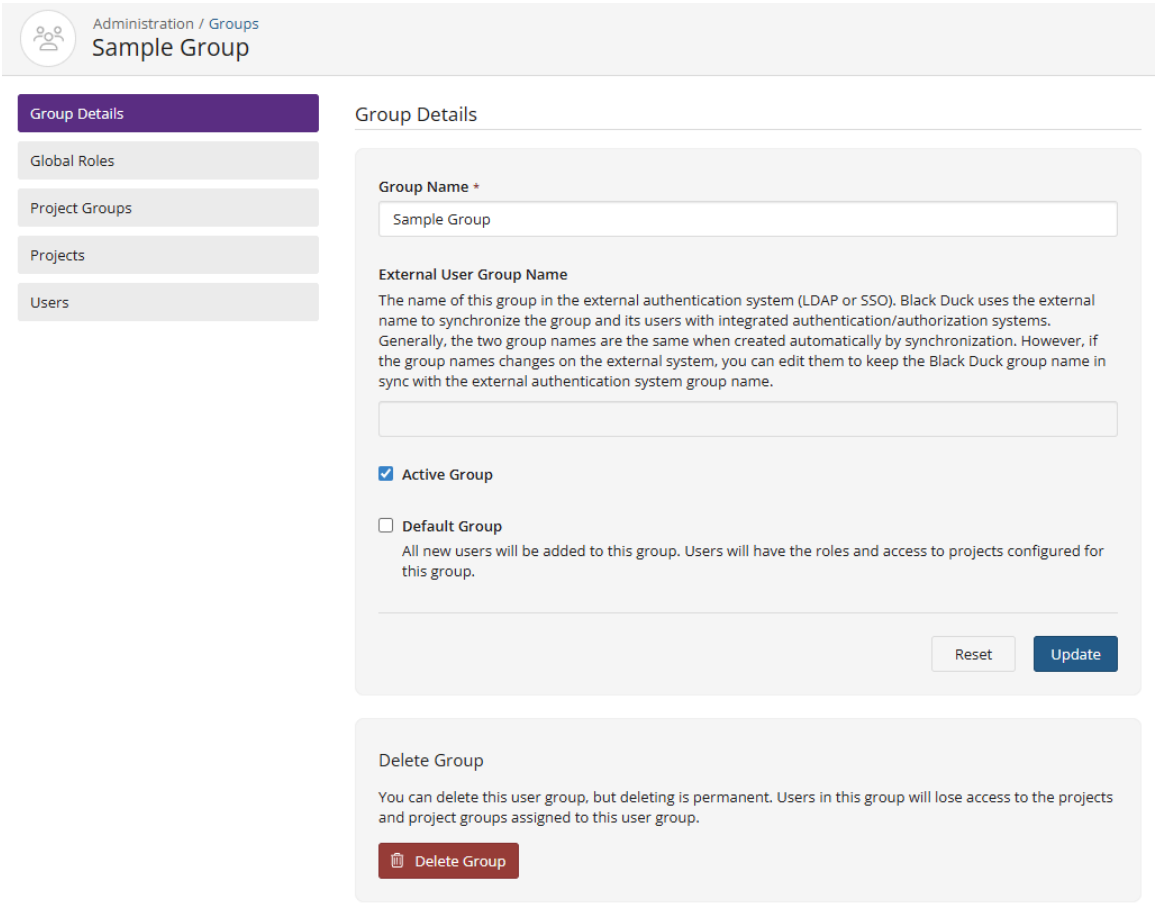
To assign or remove a project from a user group:

1. Log in to Black Duck SCA.
2. Click 

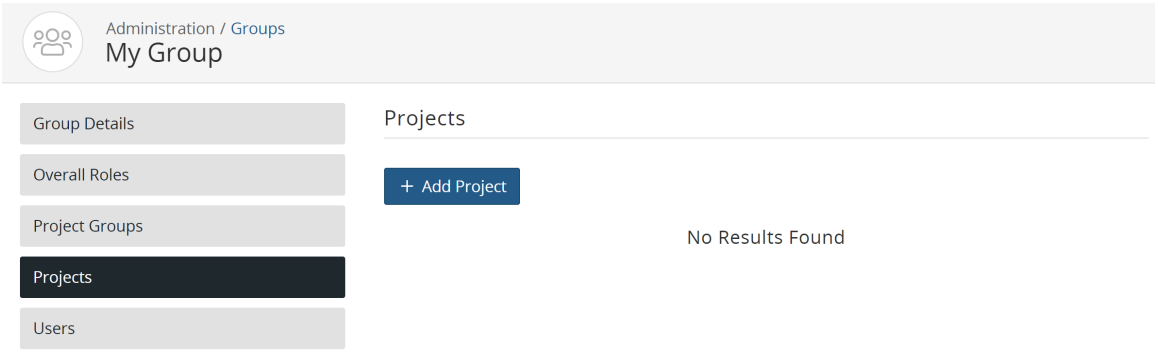
→ **Groups**.



3. Select the name of the user group to display the *Group Name* page.



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



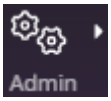
5. To add a project to a user group:
   - Click **Add Project** to display the Add Project dialog box.
   - Enter one or more projects and click **Add**.

   Alternatively, to remove a project from a user group:
   - Click in the row of the project you want to remove from the user group.
   - Click **Delete Direct Access** to remove the project.

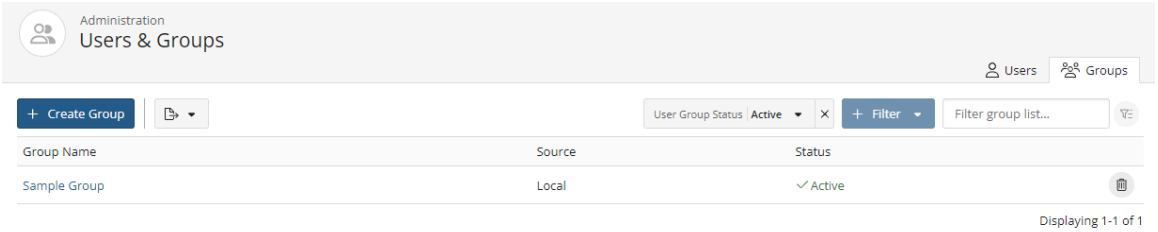
Managing user group roles

Once you have added global roles to a user group, you can add users to the user group, then assign that user group to one or more projects. These users will have the global roles assigned to the user group and will be members of all project teams to which the user group has been added.

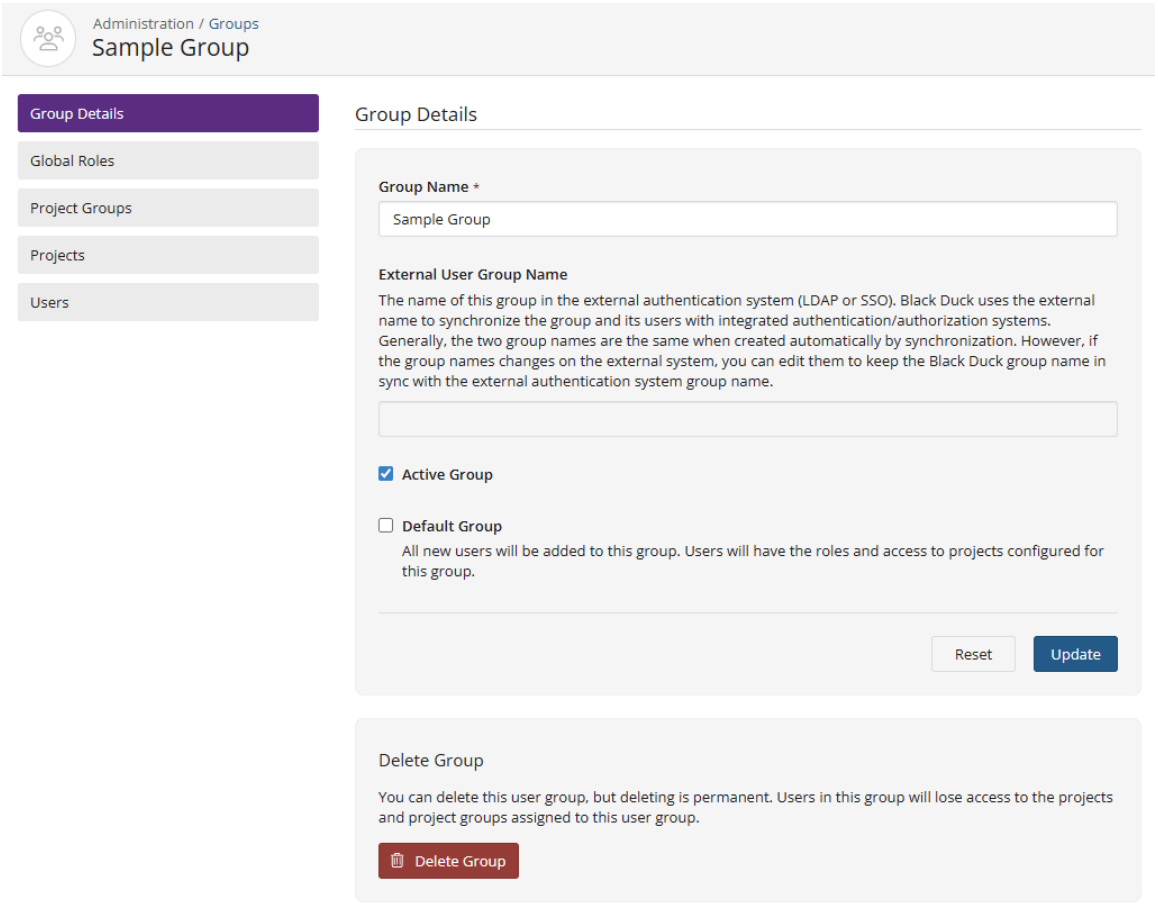
To manage user group roles:

1. Click 

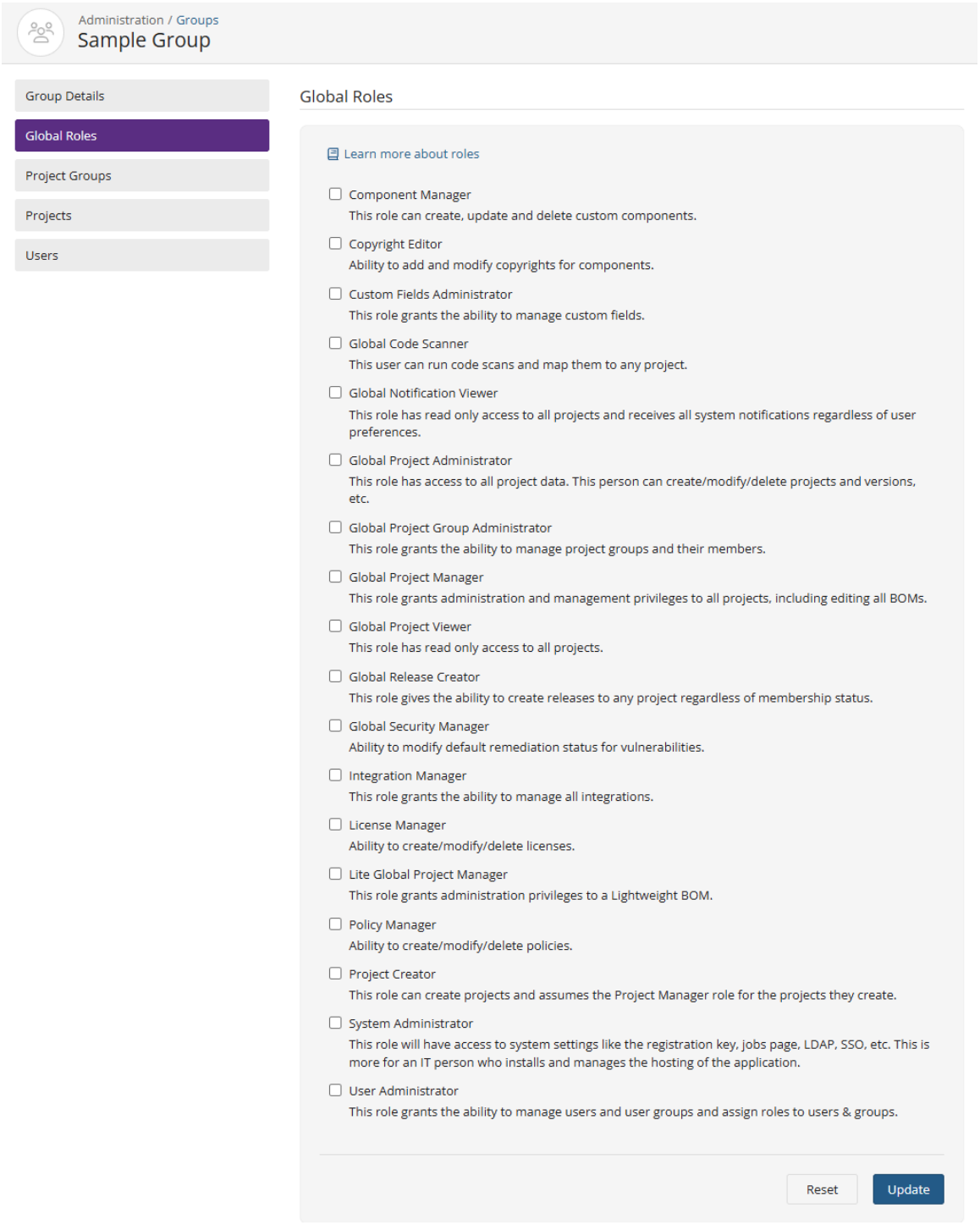
→ **Groups**.



2. Find the name of the user group for which you want to manage roles to display the *Group Name* page:
   - Filter the user groups that appear on the page.
   - Sort the list of user 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 user groups than are listed on this page.
3. Select the name of a user group to display the *Group Name* page.



4. Click **Global Roles** in the left-hand menu.



5. Select the roles that you want to assign to all members of this user group. Deselect any roles that you want to remove from this user group.
   The role is automatically assigned to the user group. You do not have to save your configuration information.

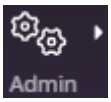
Adding or removing members from a user group

You can add or remove members from a user group by:

- Managing a user group and adding or removing members
- Managing a user and adding or removing the user from user groups

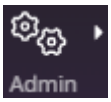
Note that subsequent users are automatically added to default groups.

Adding or removing members by managing a user group

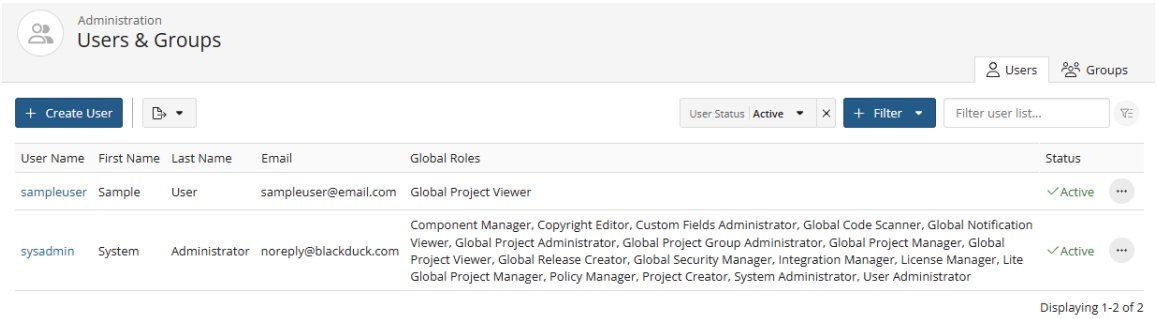
1. Log in to Black Duck SCA.
2. Click 

→ **Groups**.
3. Find the name of the group for which you want to manage membership:
   - Add the **Inactive** option to the **User Group Status** filter to include inactive groups.
   - 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 a group to display the *Group Name* page.
5. Click **Users** in the left-hand menu.
6. To add a user:
   - Click **+ Add User** to display the Add Users to Group dialog box.
   - Add a user or multiple users by:
     - 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.
     - Click the **Users** dropdown box to see a list of users.
     - Select any number of users to add to the project group.
7. Click **Add**.

Adding or removing a member from a project group by managing a user

1. Log in to Black Duck SCA.
2. Click 

→ **Users**.



3. Find the desired user:

- Add the **Inactive** option to the **User Status** filter to include inactive users.
- 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 user to display the *Username* page.
5. Click **User Groups**.
6. To add the user:
   - Click **+ Add Group**.
   - 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 groups you want this user to join.
   - Click **Save**.

     Note that the roles assigned to this user are determined by the group.

   To remove the user:

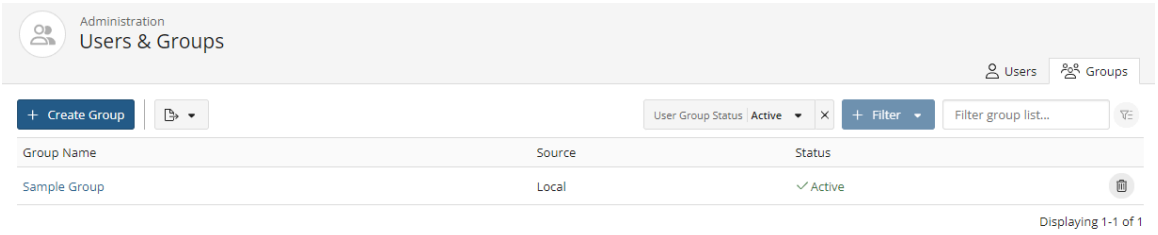
   - Click in the row of the user group you want to remove.
   - In the Remove User from Group dialog box, click **Remove**.

Deleting user 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 SCA.
2. Click Admin → **Groups**.



3. Find the name of the user group you want to delete:
   - Filter the user groups that appear on the page.
   - Sort the list of user 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 user groups than are listed on this page.

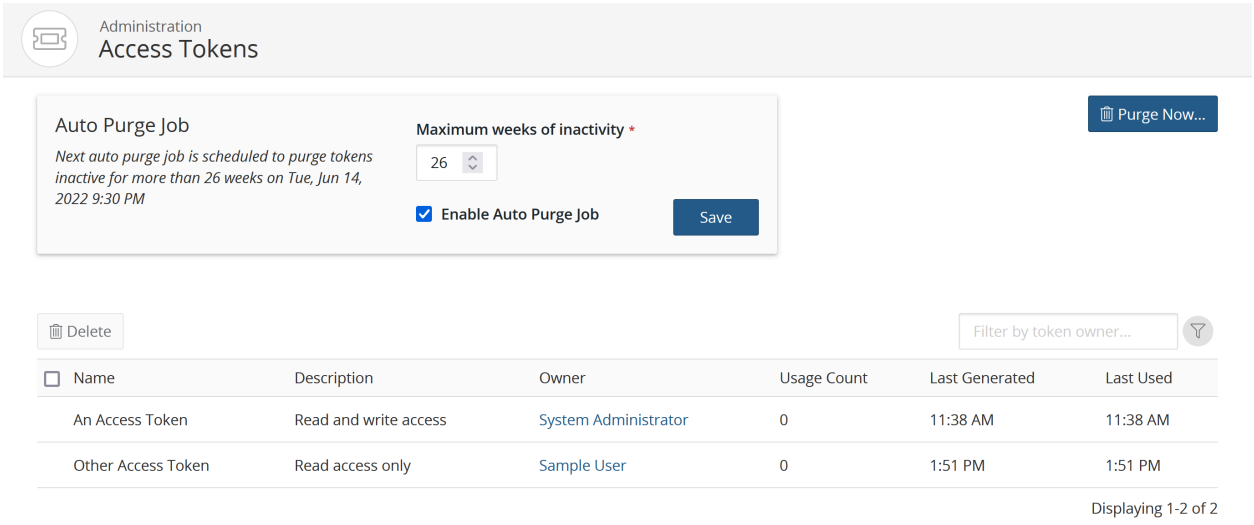
4. Click in the row of the group that you want to delete.
5. In the Delete Group dialog box, click **Delete**.
   The group is deleted from Black Duck. Users who were assigned to the deleted group no longer have any global roles that were associated with belonging to that group and no longer have membership on project teams granted through that group.

Access Tokens

User Administrators of the Black Duck system need a mechanism to maintain and control access to Black Duck via access tokens. User access is often controlled via Single Sign-On integration, but access tokens are managed independently by Black Duck. Administrators need to ensure security of the system and therefore need the tools to revoke or reset access when required. This page allows the User Administrator to manage all access tokens by either curating the list manually or by setting up an automated purging schedule.

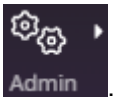
The list of access tokens is composed of the following:

- **Name**: The name of the access token
- **Description**: The description given to the access token.
- **Owner**: The name of the user who created the access token.
- **Usage Count**: The number of times the access token was used.
- **Last Generated**: The date or time the access token was created or regenerated.
- **Last Used**: The date or time the access token was last used.



Manual access token deletion

To delete access tokens from the list manually:

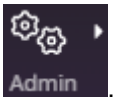
1. Log in to Black Duck with the User Administrator role.
2. Click 

.
3. Click **Access Tokens**.

4. Check the box next to any number of access tokens.
5. Click the **Delete** button. A **Delete Token** dialog box will appear.
6. Confirm the access token deletion by clicking the **Delete** button in the dialog box.

Setting up the automated access token purge job

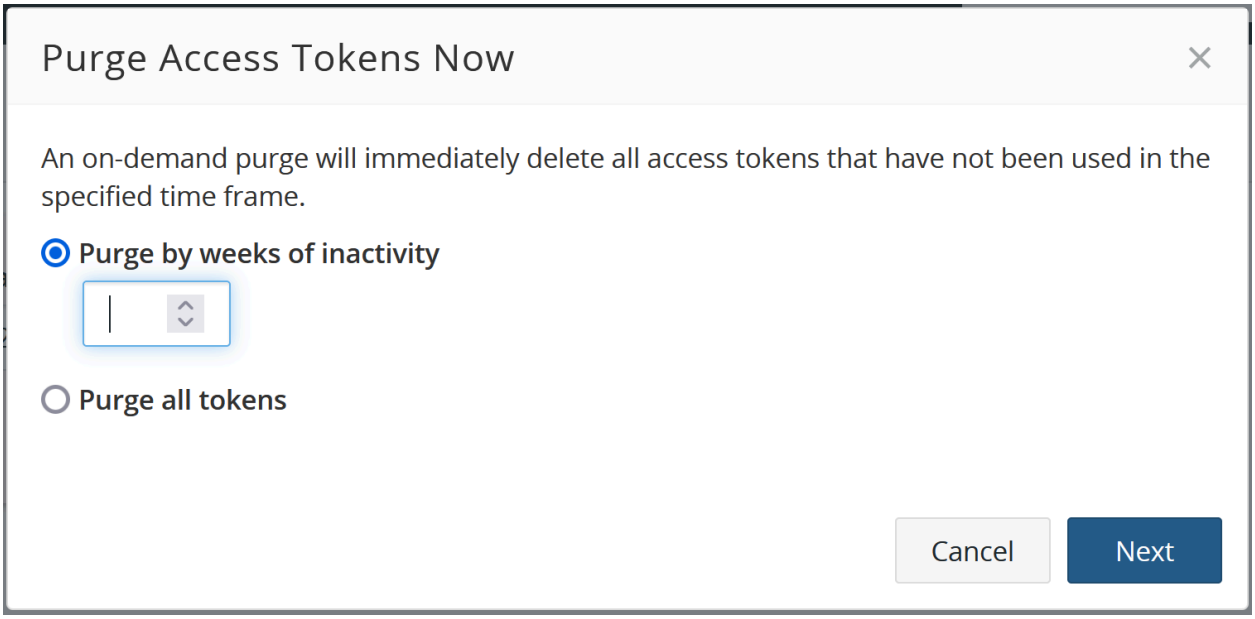
To change the access token purge job setting:

1. Log in to Black Duck with the User Administrator role.
2. Click 

.
3. Click **Access Tokens**.
4. Set the desired period of time for the **Maximum weeks of inactivity**.
5. Check the **Enable Auto Purge Job** checkbox to activate the feature or remove the check to disable it.
6. Click either the **Save** button.

On-demand access token purge

You can also initiate an on-demand purge immediately by clicking the **Purge Now...** button. Clicking this button will open the Purge Access Tokens Now dialog box.



Select either of the following options:

- **Purge by weeks of inactivity**: Set the time frame for the number of weeks of inactivity.
- **Purge all tokens**: Will delete all created access tokens.

Click the **Next** button to continue the action.

The job responsible for conducting the access token purging is ApiTokenPurgeCheckJob.

Managing user access tokens

Black Duck SCA provides the ability for you to generate one or more “tokens” for accessing Black Duck SCA 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.

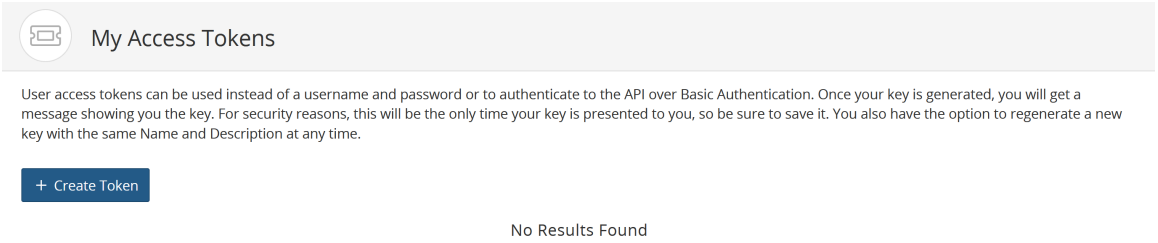
Note the following:

- 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, but can be purged after a set period of inactivity.
- If a user is inactivated, their tokens are invalidated.

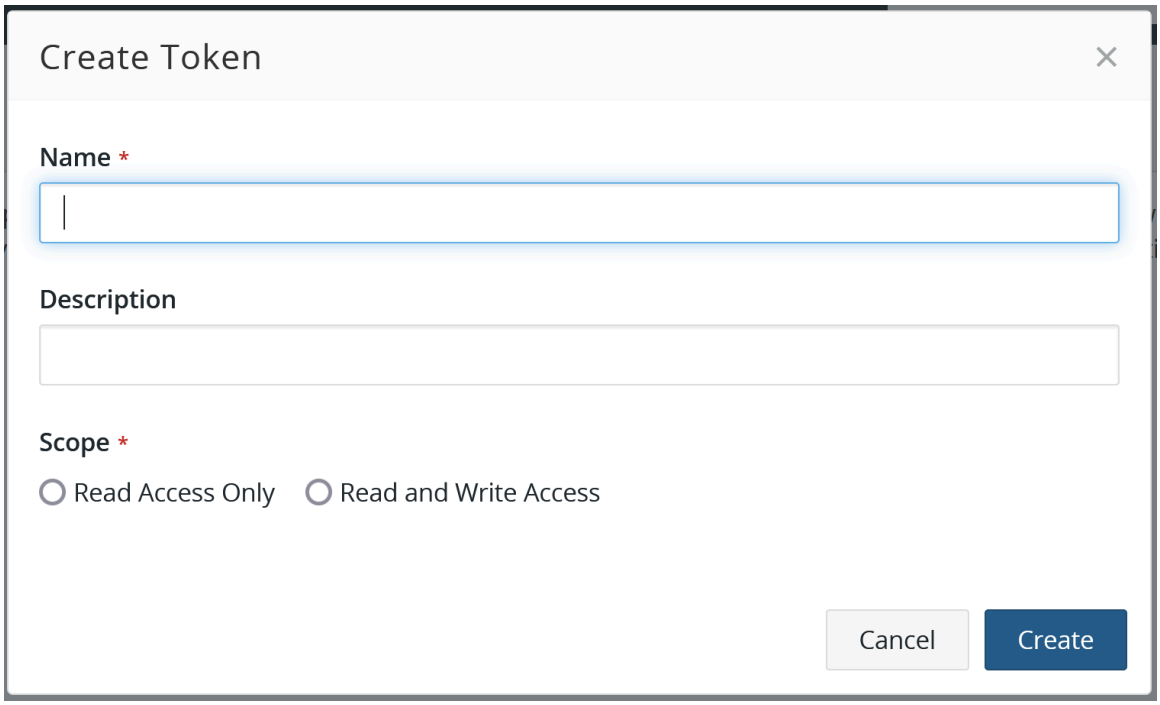
Generating an access token

1. Log into Black Duck SCA.
2. From the user menu located on the top navigation bar, select **My Access Tokens**.

   The My Access Tokens page appears.



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**.

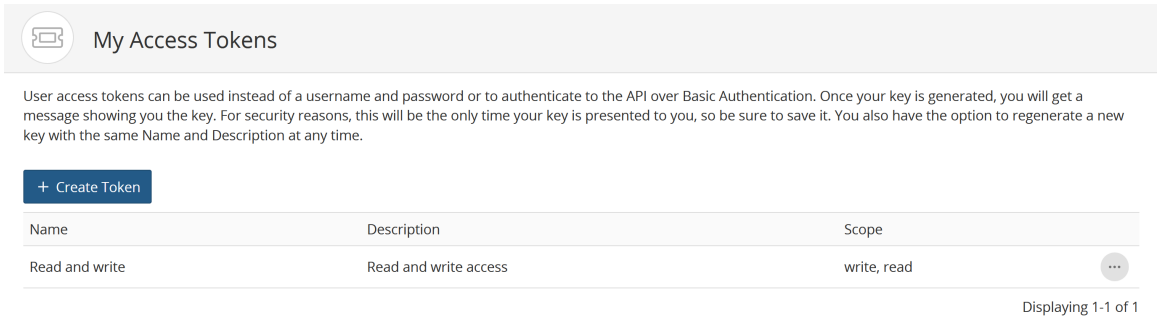
   The *Access Token Name* dialog box appears with the access token.
6. Copy the access token shown in the dialog box. This token can only be viewed here at this time. Once you close the dialog box, you cannot view the value of this token.
7. Click **Close**.

Regenerating an access token

You can regenerate a new access token which provides a different key for the same name, description, and access.

1. Log into Black Duck SCA.
2. From the user menu located on the top navigation bar, select **My Access Tokens**.

   The My Access Tokens page appears.



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.

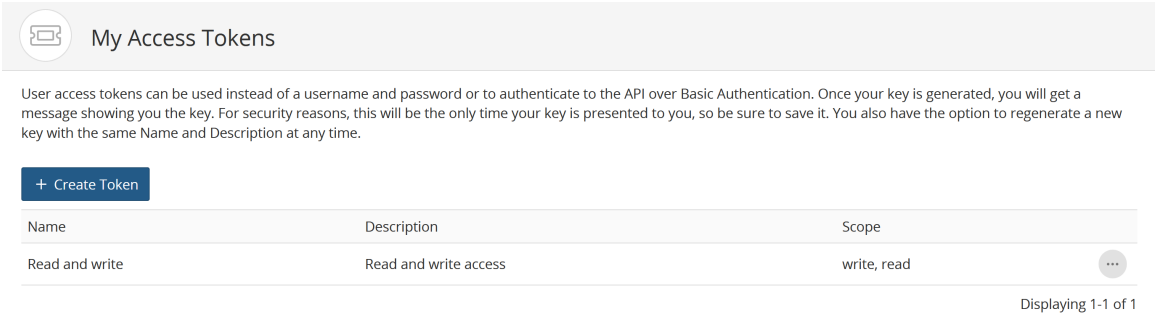
5. Copy the access token shown in the dialog box. This token can only be viewed here at this time. Once you close the dialog box, you cannot view the value of this token.

6. Click **Close**.

Deleting an access token

1. Log into Black Duck SCA.
2. From the user menu located on the top navigation bar, select **My Access Tokens**.

   The My Access Tokens page appears.



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.

Configuring Integrations

Setting Up SCM Providers

SCM integrations allow for direct communication with SCM providers, enabling automatic retrieval of repository and branch information during Detect scans on cloned Git repositories. This integration enhances usability and data accuracy by populating dropdowns and search boxes.

Supported SCM Integrations

Currently, the following SCM integrations are supported:

- **GitHub**
- **GitHub Enterprise**
- **GitLab Self-Managed**
- **GitLab SaaS**
- **Bitbucket**

- **Bitbucket Data Center**

Enabling SCM integration

SCM integration operates within a Kubernetes environment (either native or Kubernetes in Docker - KinD). Follow these steps to enable SCM integration:

- **Install Required Helm Charts:** Use the helm charts to install the necessary components.
- **Activate the Feature:**
  - This feature is not enabled by default. To activate it, add the feature to your Product Registration key.
  - Update your `values.yaml` file with:

```
enableIntegration: true
```

Note: Currently, self-signed certificates are not accepted for SCM integrations.

Creating an OAuth App

Before setting up a SCM provider in Black Duck SCA, you must first authenticate the project.

For GitHub and GitHub Enterprise, you must create an OAuth App:

1. Go to https://github.com/settings/developers and OAuth Apps and create a new app (or the corresponding URL for GitHub Enterprise).
2. Fill the following fields:
   - **Application Name**
   - **Homepage URL**: The URL of your Black Duck SCA Server
   - **Application Description**
   - **Authorization Callback URL**: `<Homepage URL>/api/scm/github/callback`
3. Click **Save**. This will generate the Client ID to be used in Black Duck SCA.
4. Click **Generate secret**. This will generate a secret string to be used in Black Duck SCA.

For GitLab Self-Managed:

1. Go to <gitlab_server_name>/-/profile/applications. You should see add new application.
2. Fill the following fields:
   - **Name**: provide any name.
   - **Redirect URI**: <bd_server_name>/api/scm/gitlab/callback
3. Uncheck the **Confidential** checkbox.
4. Enable API in the **Scopes** section.

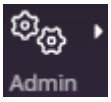
For BitBucket:

1. Go to <bitbucket_server_name>/plugins/servlet/applinks/listApplicationLinks
2. Click **Create Link**.
3. Select **External application**.
4. Select **Incoming** in the **Direction** dialog box and then click OK.

5. Fill the following fields:
   - **Name**: Provide a name.
   - **Redirect URI**: <bd_server_name>/api/scm/bitbucket/callback
6. Check the **Write** checkbox under **Repositories** in the **Application permissions** section.

Setting up a GitHub.com SCM integration

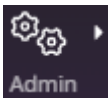
To set up a GitHub.com SCM integration:

1. Log into Black Duck SCA as a System Administrator.
2. Click 

and select **Integrations**.
3. Click **GitHub.com**.
4. Fill the following fields:
   - Check the **Enable Server** checkbox.
   - Enter the **Client ID** generated from the GitHub website.
   - Enter the **Secret** generated from the GitHub website.
5. Click **Save**.

Setting up a GitHub Enterprise SCM integration

To set up a GitHub Enterprise SCM integration:

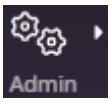
1. Log into Black Duck SCA as a System Administrator.
2. Click 

and select **Integrations**.
3. Click **GitHub Enterprise**.
4. Click **+ Add Server**.
5. Fill the following fields:
   - **Server Name**: Enter a name for your server.
   - **Server URL**: Enter your GitHub Enterprise server URL.
   - **Client ID**: Enter the **Client ID** generated from the GitHub website.
   - **Secret**: Enter the **Secret** generated from the GitHub website.
   - Check the **Enable Server** checkbox.
6. Click **Create**.

Setting up a GitLab Self-Managed SCM integration

To set up a GitLab Self-Managed SCM integration:

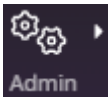
1. Log into Black Duck SCA as a System Administrator.

2. Click 

and select **Integrations**.
3. Click **GitLab Self-Managed**.
4. Click **+ Add Server**.
5. Fill the following fields:
   - **Server Name**: Enter a name for your server.
   - **Server URL**: Enter your GitLab Self-Managed server URL.
   - **Client ID**: Enter the **Client ID** generated from the GitLab website.
   - **Secret**: Enter the **Secret** generated from the GitLab website.
   - Check the **Enable Server** checkbox.
6. Click **Create**.

Setting up a GitLab SaaS SCM integration

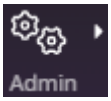
To set up a GitLab SaaS SCM integration:

1. Log into Black Duck SCA as a System Administrator.
2. Click 

and select **Integrations**.
3. Click **GitLab SaaS**.
4. Click **+ Add Server**.
5. Fill the following fields:
   - **Server Name**: Enter a name for your server.
   - **Client ID**: Enter the **Client ID** generated from the GitLab website.
   - **Secret**: Enter the **Secret** generated from the GitLab website.
   - Check the **Enable Server** checkbox.
6. Click **Create**.

Setting up a Bitbucket SCM integration

To set up a Bitbucket SCM integration:

1. Log into Black Duck SCA as a System Administrator.
2. Click 

and select **Integrations**.
3. Click **Bitbucket**.
4. Click **+ Add Server**.
5. Fill the following fields:
   - **Server Name**: Enter a name for your server.

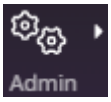
- **Client ID**: Enter the **Client ID** generated from the Bitbucket website.
- **Secret**: Enter the **Secret** generated from the Bitbucket website.
- Check the **Enable Server** checkbox.

6. Click **Create**.

Setting up a Bitbucket Data Center SCM integration

Note: Black Duck SCA supports Bitbucket Data Center version 8.19. Users should ensure they are using this version for optimal compatibility and functionality. For personal repositories not linked to a project, the Bitbucket Data Center integration only supports the default branch.

To set up a Bitbucket Data Center SCM integration:

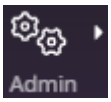
1. Log into Black Duck SCA as a System Administrator.
2. Click 

and select **Integrations**.
3. Click **Bitbucket Data Center**.
4. Click **+ Add Server**.
5. Fill the following fields:
   - **Server Name**: Enter a name for your server.
   - **Server URL**: Enter your Bitbucket Data Center server URL.
   - **Client ID**: Enter the **Client ID** generated from the Bitbucket Data Center website.
   - **Secret**: Enter the **Secret** generated from the Bitbucket Data Center website.
   - Check the **Enable Server** checkbox.
6. Click **Create**.

Managing SCM providers

Editing SCM integrations

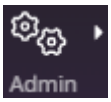
To edit the GitHub.com SCM integration:

1. Log into Black Duck SCA as a System Administrator.
2. Click 

and select **Integrations**.
3. Click **GitHub.com**.
4. Edit any of the fields as desired.
5. Click **Update**.

To edit a GitHub Enterprise, GitLab Self-Managed, GitLab SaaS, Bitbucket, or Bitbucket Data Center SCM server:

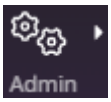
1. Log into Black Duck SCA as a System Administrator.

2. Click 

and select **Integrations**.
3. Click the applicable SCM Provider from the available options.
4. Click the icon for the server and select **Edit**.
5. Edit any of the fields as desired.
6. Click **Update**.

Deleting SCM integrations

To delete a GitHub Enterprise, GitLab Self-Managed, GitLab SaaS, Bitbucket, or Bitbucket Data Center SCM server:

1. Log into Black Duck SCA as a System Administrator.
2. Click 

and select **Integrations**.
3. Click the applicable SCM Provider from the available options.
4. Click the icon for the server and select **Delete**.
5. Click **Delete**.

Authenticating users for SCM providers

Black Duck users wishing to use SCM integration must authenticate with GitHub cloud, or if the system administrator has configured it, GitHub Enterprise. This can be done at the time the SCM integration is used, such as attempting to assign a repository, or directly in the my profile area of Black Duck. Based on the selection, users will be redirected to an appropriate Git landing page where they will log into their account. If authentication is successful, users will be redirected back to the Black Duck application along with an access token. This token will be stored in the database for use in future communications with Git.

To authenticate yourself with a SCM provider:

1. Click your username on the top right of any page.
2. Select **SCM Providers**.
3. Click **Authenticate** next to the SCM server name. This will redirect you to the SCM server where you will be prompted to confirm you want to authorize the OAuth app and what access it has.
4. Click **Yes/Ok** to return to Black Duck.

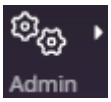
You can also authenticate yourself when creating a new project.

Configuring SCM repositories auto-scanning

SCM repository auto-scanning allows Black Duck to check daily for any changes such as commits, pushes, or merges in the repository branch mapped to your SCM projects and perform scans if changes were made.

To enable SCM repositories auto-scanning:

1. Log into Black Duck as a System Administrator.

2. Click 

and select **Jobs**.
3. Click the **Scheduled** button.
4. Find the **SCM Onboarding daily auto scanning** job in the table. Note that this job is disabled by default.
5. Click the at the of the **SCM Onboarding daily auto scanning** line.
6. Select **Enable**.

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 SCA user account is created.
- Centralized management of user account details. Each time a user logs in to Black Duck SCA, Black Duck SCA synchronizes with the directory server. If changes were made to mapped attributes, Black Duck SCA 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 SCA user account, as well as a Black Duck SCA group, is created. The group is populated with the new user.

Note: If the Black Duck SCA group already exists, the Black Duck SCA user account is created, and the group is populated.

Tip: All hosted customers should secure access to their Black Duck SCA application by leveraging out-of-the-box support for single sign-on (SSO) via SAML or LDAP. For information on enabling and configuring these security features, see the installation guides. Customers using a SAML SSO provider that offers two-factor authentication are also encouraged to enable that technology to further secure access to their Black Duck SCA application. For more information, see Configuring SAML for Single Sign-On.

Before starting

Contact your LDAP administrator and gather the following information:

- **LDAP server details**

  This is the information that Black Duck SCA uses to connect to the directory server.

  - **Server URL** (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:389`

    For more information, see Configuring secure LDAP.

  - **Authentication Type**: If credentials are required for LDAP access, the authentication type to use: Simple, None, or Digest-MD5.
  - **Manager DN** (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`

- **LDAP user attributes and LDAP attribute mappings**

  This is the information that Black Duck SCA uses to locate users in the directory server:

  - **User Search Base** (required): The absolute base DN under which users can be located.

    Example: `dc=example,dc=com`

  - **User Search Filter** (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}`

  - **User DN Pattern** (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`

  - **First Name, Last Name, Email** (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 SCA uses to locate user groups in the directory server:

  - **Group Search Base** (required): The absolute base DN under which groups can be located.

    Example: `ou=groups,dc=example,dc=com`

  - **Group Filter** (required): The attribute used to match a unique user member within a given group.

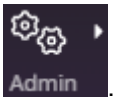
    Example: `uniquemember={0}`

  - **Group Name Attribute** (required): The attribute that identifies a specific, unique group name.

    Example: `cn`

Configuring LDAP

To configure LDAP:

1. Log in to Black Duck SCA as a system administrator.
2. Click 

.
3. Select **Integrations** → **External Authentication**.
4. Click **Lightweight Directory Access Protocol (LDAP)**.
5. Check the **Enable LDAP Configuration** checkbox.
6. In the **LDAP Server Details** section, enter the server connection and authentication details that Black Duck SCA is to use to connect to the directory server.
7. In the **LDAP User Attributes** section, enter the user attribute values that Black Duck SCA is to use to locate users.

   Optionally, clear the **Create user accounts automatically in Black Duck SCA** check box to turn off the automatic creation of users when they authenticate with LDAP. This check box is selected by default so

that users who do not exist in Black Duck SCA are created automatically when they log in using LDAP. This applies to new installs and upgrades.

8. (Optional) Enter the attributes that map to user-specific information in the **LDAP Attribute Mappings** section.
9. (Optional) Select **Synchronize LDAP groups** and enter the group attribute values that Black Duck SCA is to use to locate groups in the **LDAP Groups** section.
10. (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 LDAP group synchronization is enabled and configured, the user's first name, last name, email address, and LDAP groups are displayed for successful connections.
11. Click **Save**.

Configuring Secure LDAP

To connect Black Duck SCA to a secure LDAP server, you must establish a trust connection by importing the LDAP server certificate into the Black Duck SCA LDAP truststore. This is most commonly required when using a self-signed certificate.

Important: If you have configured custom CA certificates using `AUTH_CUSTOM_CA`, `HUB_PROXY_CERT_FILE`, or the `certAuthCACertSecretName` Helm secret, those certificates are imported into the proxy truststore only. They are **not** automatically imported into the LDAP truststore. To establish a trust connection for your LDAP or LDAPS server, you must import the server certificate separately using the Black Duck SCA UI, as described below.

Obtaining your LDAP information

Contact your LDAP administrator and gather the following information before proceeding.

LDAP Server Details

This is the information that Black Duck SCA 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:636`
- (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 User Attributes

This is the information that Black Duck SCA uses to locate users in the directory server.

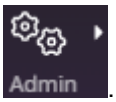
- (required) The absolute base DN under which users can be located.
  Example: `dc=example,dc=com`
- (required) The attribute used to match a specific, unique user. The value of this attribute personalizes the user profile icon with the name of the user.
  Example: `uid={0}`

Test Username and Password

- (required) The user credentials to test the connection to the directory server.

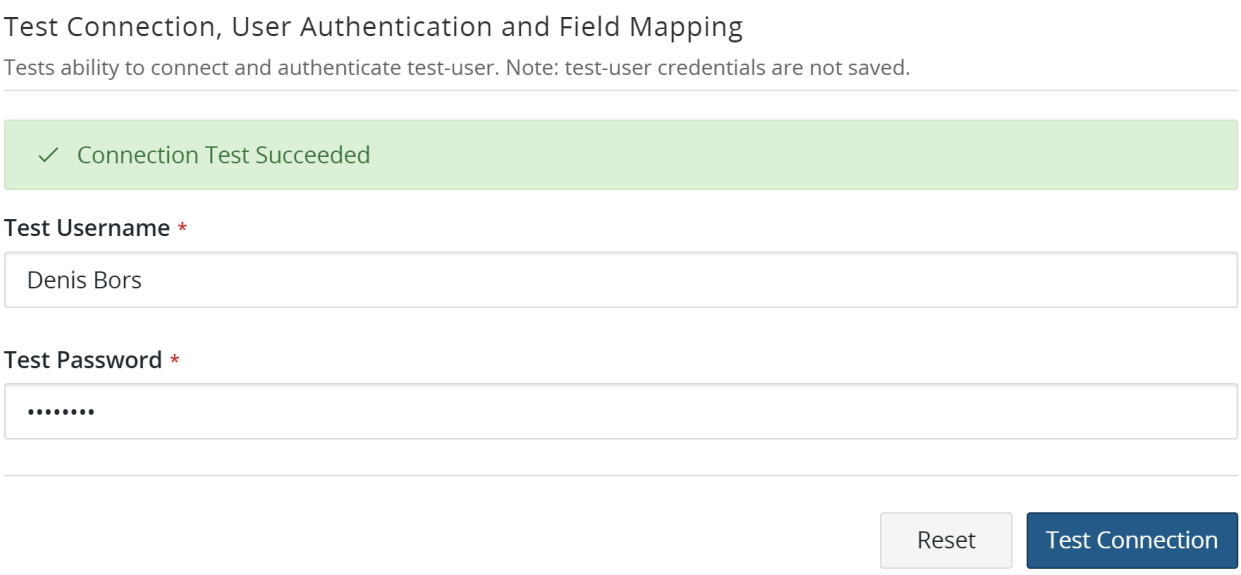
Importing the server certificate

To import the server certificate:

1. Log in to Black Duck SCA as a system administrator.
2. Click 

.
3. Select **Integrations** → **External Authentication**.
4. Click **Lightweight Directory Access Protocol (LDAP)**.
5. Check the **Enable LDAP Configuration** checkbox and complete the information in the **LDAP Server Details** section, as described above. In the **Server URL** field, ensure that the protocol scheme is `ldaps://`.
6. Complete the information in the **LDAP User Attributes** section, as described above.

   Optionally, clear the **Create user accounts automatically in Black Duck SCA** check box to turn off the automatic creation of users when they authenticate with LDAP. This check box is selected by default so that users who do not exist in Black Duck SCA are created automatically when they log in using LDAP. This applies to new installs and upgrades.
7. Enter the user credentials in the **Test Connection, User Authentication and Field Mapping** section and click **Test Connection**.
8. If there are no issues with the certificate, it is automatically imported and the "Connection Test Succeeded" message appears:



9. If there is an issue with the certificate, a dialog box listing details about the certificate will appear. Do one of the following:
   - Click **Cancel** to fix the certificate issues. Once fixed, retest the connection to verify that the issues have been resolved and the certificate has been imported. If successful, the "Connection Test Succeeded" message appears.

- Click **Save** to import the certificate as-is. Verify that the certificate has been imported by clicking **Test Connection**. If successful, the "Connection Test Succeeded" message appears.

Configuring SAML for Single Sign-On

Security Assertion Markup Language (SAML) is an XML-based, open-standard data format for exchanging authentication and authorization data between parties. For example, between an identity provider and a service provider. Black Duck SCA's SAML implementation provides single sign-on (SSO) functionality, enabling Black Duck SCA users to be automatically signed-in to Black Duck SCA when SAML is enabled. Enabling SAML applies to all your Black Duck SCA users and cannot be selectively applied to individual users.

Tip: All hosted customers should secure access to their Black Duck SCA application by leveraging out-of-the-box support for single sign-on (SSO) via SAML or LDAP. For information on enabling and configuring these security features, see the installation guides. Customers using a SAML SSO provider that offers two-factor authentication are also encouraged to enable that technology to further secure access to their Black Duck SCA application. For more information, see Configuring SAML for Single Sign-On.

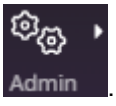
Note the following:

- It is not possible to configure both SAML and LDAP at the same time.
- To enable or disable SAML functionality, you must be a user with the system administrator role.
- Black Duck SCA is able to synchronize and obtain an external user's information (Name, FirstName, LastName and Email) if the information is provided in attribute statements. Note that the first and last name values are case-sensitive.

  Black Duck SCA is also able to synchronize an external user's group information if you enable group synchronization in Black Duck SCA.
- When logging in with SAML enabled, you are re-directed to your identity provider's login page, not Black Duck SCA's login page.
- When SSO users log out of Black Duck SCA, a logout page now appears notifying them that they successfully logged out of Black Duck SCA. This logout page includes a link to log back into Black Duck SCA; users may not need to provide their credentials to successfully log back in to Black Duck SCA.
- If there are issues with the SSO system and you need to disable the SSO configuration, you can enter the following URL to log in to Black Duck SCA:

```
<Black Duck servername>/ui/login
```

Enabling or disabling single sign-on using SAML

1. Click 

.
2. Select **Integrations** → **External Authentication**.
3. Click **Security Assertion Markup Language (SAML)**, complete the following:
   a. Select the **Enable SAML configuration** check box.
   b. **Service Provider Entity ID** field: Enter the information for the Black Duck SCA server in your environment in the format **https://*host*** where *host* is your Black Duck SCA server.
   c. Select one of the following **Identity Provider Metadata**:

- **URL** and enter the URL for your identity provider.
- **XML File** and either drop the file or click in the area shown to open a dialog box from which you can select the XML file.

d. **Service Provider Entity ID** field. Enter the information for the Black Duck SCA server in your environment in the format **https://*host*** where *host* is your Black Duck SCA server.

e. **External Black Duck SCA URL** field. The URL of the public URL of the Black Duck SCA server.

For example: https://blackduck-docker01.dc1.lan

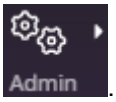
f. Optionally, select any of the following:

- **Send Signed Authentication Request**: If this option is enabled, it indicates the asserting party's preference that relying parties should sign the authentication request before sending.
- **Enable Group Synchronization**: If this option is enabled, upon login, groups from the Identity Provider (IDP) are created in Black Duck SCA and users will be assigned to those groups. Note that you must configure IDP to send groups in attribute statements with the attribute name of 'Groups'.
- **Enable Local Logout Support**: If this option is enabled, after logging out of Black Duck SCA, the IDP's login page would appear. When local logout support is enabled, SAML requests are sent with ForceAuthn="true". Check with the IDP to confirm that this is supported.
- **Create user accounts automatically in Black Duck SCA**: If a user logs in using the IdP and the user doesn't exist in Black Duck, we create a local user in Black Duck SCA's database if that option is selected.

4. Click **Save**.

After clicking **Save**, the **Black Duck SCA Metadata URL** field appears. You can copy the link or directly download the SAML XML configuration information.

To disable single sign-on using SAML

1. Click 

.
2. Select **Integrations → External Authentication**.
3. Click **Security Assertion Markup Language (SAML)**
4. Clear the **Enable SAML configuration** check box.
5. Click **Save**.

Additional information

- Assertion Consumer Service (ACS): **https://<host>/saml/SSO**
- Recommended Service Provider Entity ID: **https://<host>** where *host* is your Black Duck SCA server location.

Signed Authentication Certificates

Black Duck SCA supports SAML signed authentication certificates, enhancing security by validating authentication requests against a trusted certificate.

To create a signed authentication certificate:

1. Click **+ Certificate**.
2. Enter or select an **Expiration Date** in the **Create SAML Certificate** modal.
3. Click **Create**.

Note: A maximum of two certificates can be created.

To activate, download, or delete a signed authentication certificate:

1. Click for the desired certificate.
2. Select from the following options:
   - **Activate**. Specify this certificate to be used to validate SAML requests. Once activated, the certificate will be marked with the .
   - **Download**. Download the certificate as a CERT file.
   - **Delete**. Delete the certificate from the list of signed authentication certificates. Active certificates cannot be deleted.

You will be notified when a signed authentication certificate is nearing its expiration date. Notifications include a banner at the top of the screen, a pop up message, as well as an  icon on the certificate itself 30 days prior to the expiry date. Notifications are triggered well in advance to provide time for system administrators to update or replace the certificate.

Configuring System Settings

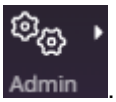
Customizing Your Branding

You can replace the logo that appears in the header of the user interface:



The dimensions will be constrained to a maximum height of 36px and a maximum width of 259px. The filesize cannot exceed 100Kb. The logo supports various image formats, including PNG, GIF, SVG, WEBP, JPEG, and AVIF.

To change the logo:

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

.
3. Select **System Settings**.
4. Click **Branding**.
5. Click **Upload New Logo** and select the file.

   The new logo will appear in the header.

   Note: To restore the original Black Duck logo, select **Restore to Original**. This option appears on the page after you customize the logo.

Managing custom fields

A custom field is a system-wide property that will apply to all BOM components, components, component versions, projects, or project versions which provides 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.

Users with the Custom Fields Administrator role can:

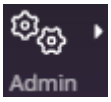
- Create, edit, or delete custom fields.
- Activate or deactivate a custom field. By default, a custom field is inactive and not shown to users.
- Determine the order of the custom fields as shown in the UI.

Recommended vs 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 recommended, 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.

When **Force Entry of Required Custom Fields** is enabled, users *must* enter values when editing objects which have required custom fields.

To enable **Force Entry of Required Custom Fields**:

1. Log into Black Duck as a system administrator user.
2. Click 

→ **System Settings**.
3. Click the **Custom Fields** tab.
4. Switch the **Force Entry of Required Custom Fields** toggle to **Enabled**.

When enabled, a red asterisk (*) will appear next to the custom field label on the related page's **Settings** tab, indicating it is a required custom field.

Note the following:

- 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 can create a policy rule for project or BOM component custom fields for any field type.

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.

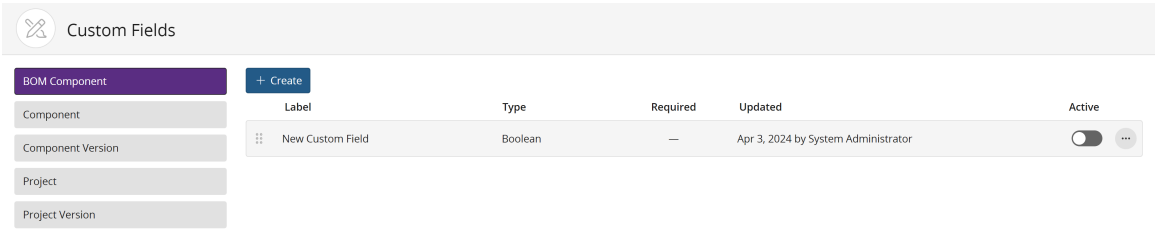
You must have the Custom Fields Administrator role to create and manage custom fields.

To create a custom field:

1. Click 

and then select **Custom Fields**.

   The Custom Fields page appears.



   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 page.

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:

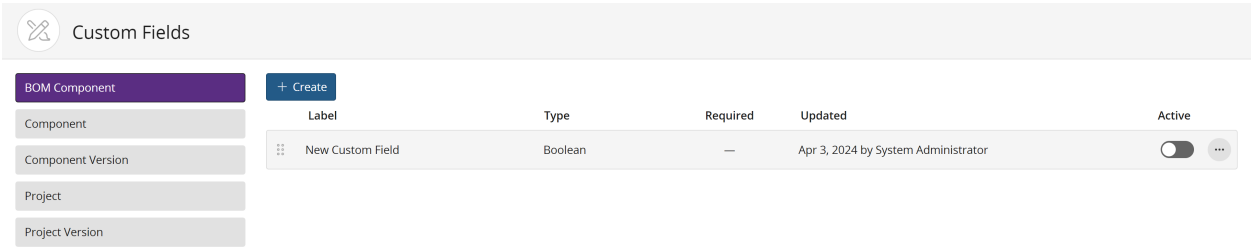
- rearrange the order of options
- edit existing options
- add new options

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 and then select **Custom Fields**.



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

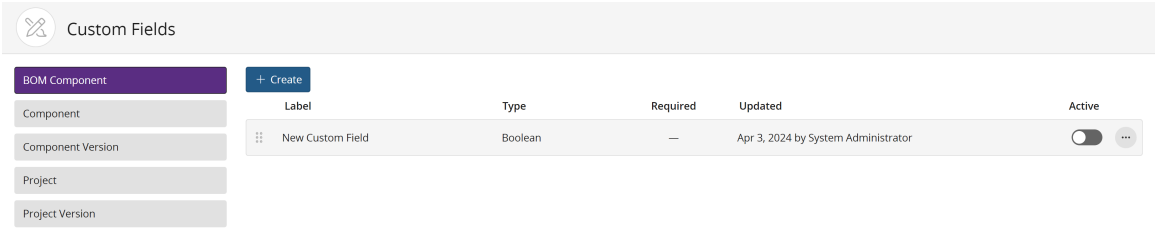
Deleting a custom field removes the custom field and all data associated with it.

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 and then select **Custom Fields**.



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**.

Activating or deactivating a custom field

By default, custom fields are deactivated. 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 Custom Fields Administrator role to activate or deactivate a custom field.

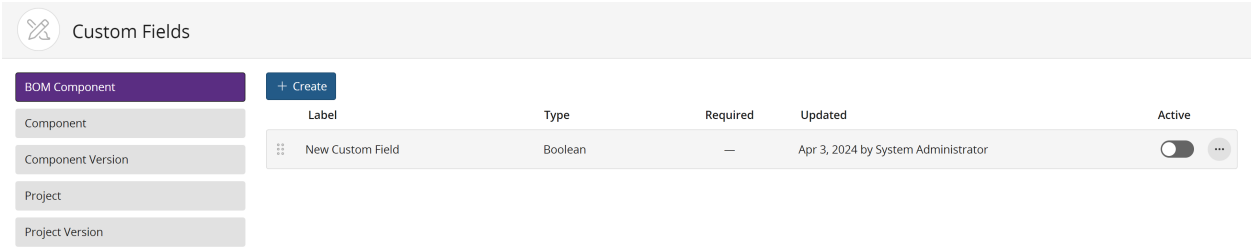
Note that you can deactivate a custom field at any time. If that custom field contained data (your users entered information for that custom field), it is retained; if you reactivate the field, the data for that custom field will reappear in the UI.

To activate or deactivate a custom field:

1. Click Manage.
2. Click **Custom Fields**.
3. Select the tab which contains the desired custom field.
4. Enable the **Active** switch in the row of the desired custom field:
   - 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 **Custom Fields** section in the project or project version **Settings** tab. This location is determined by the tables shown in the Custom Fields page – the order of the custom fields shown here defines the order of custom fields shown in the UI.



By default, when you create a new custom field, it appears on the top of the table on the Custom Fields 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.

Viewing custom fields

After creating or activating custom fields, you can find them in their relevant sections. See the sections below for the specific areas where the custom fields appear:

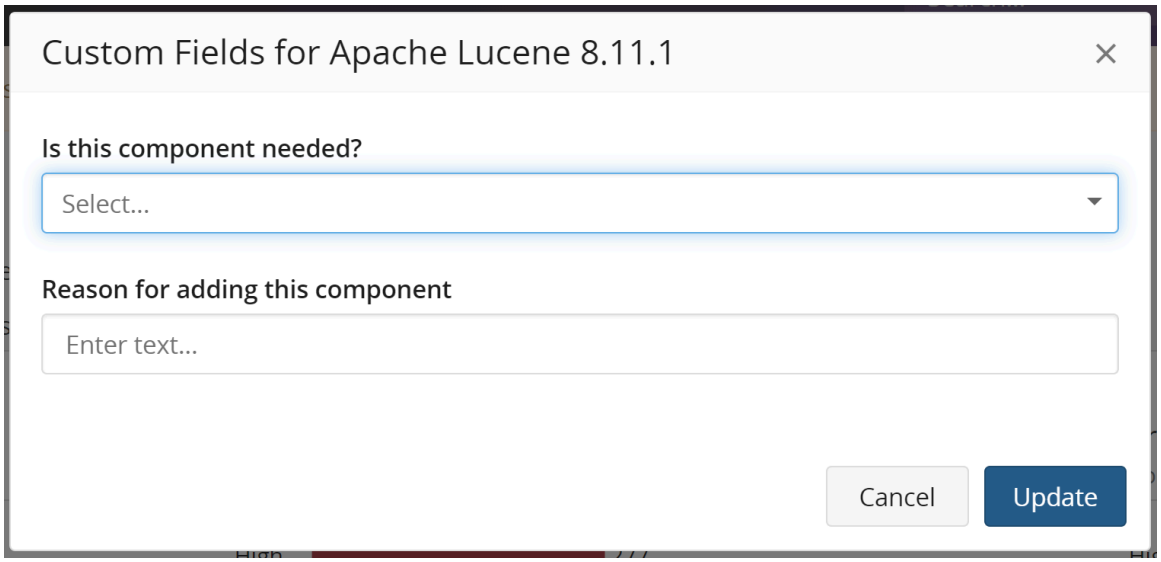
- BOM Component
- Component
- Component Version
- Project
- Project Version

BOM Component custom field information

BOM Component custom field information appears when viewing the details of a component version in the BOM.

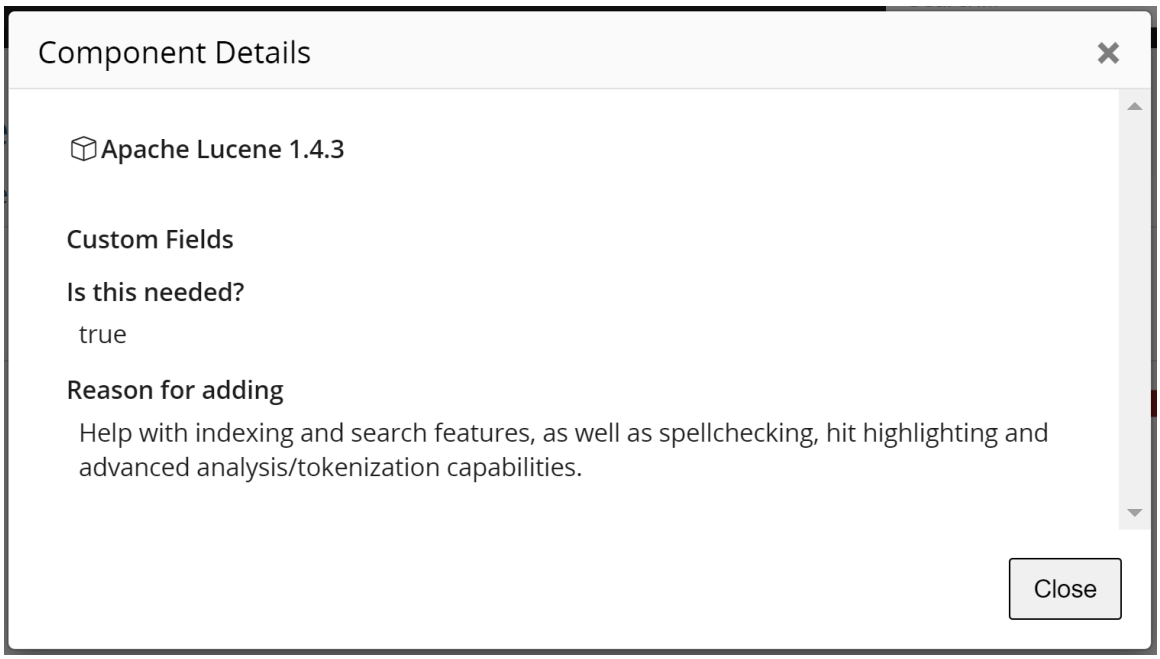
- The Information icon ( ) can indicate that there are custom fields for this component and information for the custom field has been added. Hover over the icon to see whether **Has Additional Fields** appears which indicates custom field information is available.
- Users with the BOM Manager, Global Project Administrator, Global Project Manager, or Project Manager (for the projects they are associated with) role can update or edit the values for a BOM Component custom field.
- To add information:

  1. Click and select **Custom Fields** and enter the information for the custom fields.



2. Click **Update**.

- Clicking the ⓘ opens the Component Details dialog box which displays the added information.

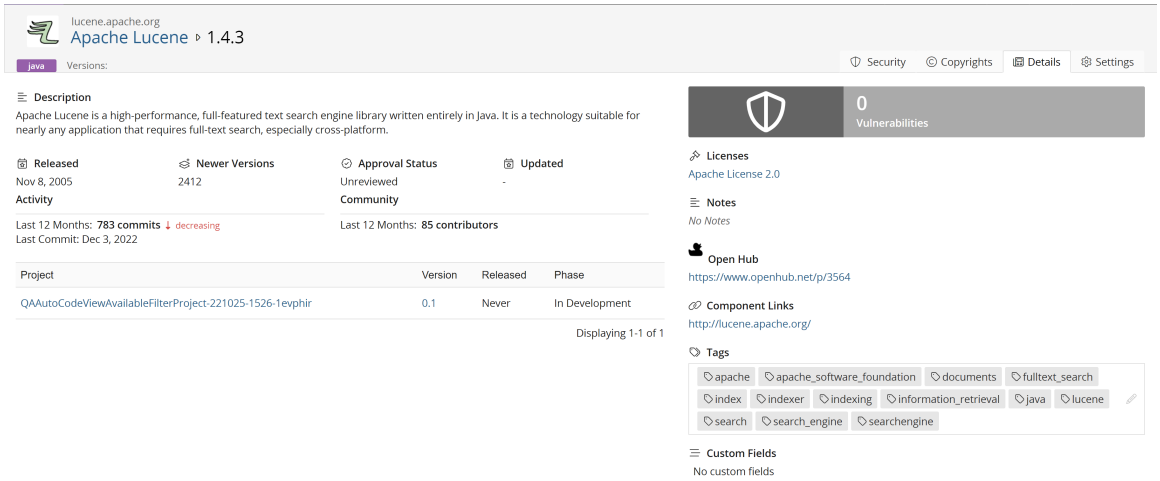


Component custom field information

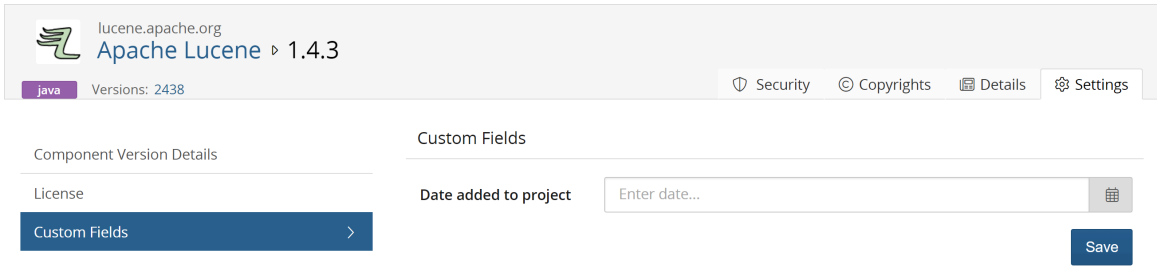
Component custom field information appears when viewing the details of a component in the BOM. Users with the Component Manager role can update or edit the values for a Component custom field.

To view and add information:

1. Click the component in your project. This will take you to the component version page.
2. Click the component name.



3. Click the **Settings** tab on the top right.
4. Click the **Custom Fields** tab in the lefthand menu.



Component Version custom field information

Component Version custom field information appears when viewing the details of a component version. Users with the Component Manager role can update or edit the values for a Component Version custom field.

To view and add information:

1. Click the component in your project. This takes you to the component version page.
2. Click the **Settings** tab on the top right.
3. Click the **Custom Fields** tab in the lefthand menu.

Project custom field information

Project custom field information is shown in the **Custom Fields** section of the *Project Name* **Settings** tab. Users with the Global Project Administrator, Global Project Manager, or Project Manager (for the projects they are associated with) role can update or edit the values for a Project custom field.

To view and add information:

1. Click the *Project Name* on your Dashboard.
2. Click the Settings tab on the top right.
3. Scroll to the Custom Fields section of the *Project Name* page.

Project Version custom field information

Project Version custom field information is shown in the **Custom Fields** section of the *Project Name*'s version **Settings** tab. Users with the Global Project Administrator, Global Project Manager, or Project Manager (for the projects they are associated with) role can update or edit the values for a Project Version custom field.

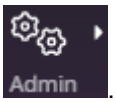
To view and add information:

1. Click the *Project Name* on your Dashboard.
2. Click the desired version of your project.
3. Click the Settings tab on the top right.
4. Scroll to the Custom Fields section in the **Version Detail** tab of the *Project Name* page.

Defining the default security risk calculation

Users with the system administrator role can update the preferred security ranking by selecting from the CVSS versions and source options below. These rankings will be used to determine and calculate risk in each of your projects.

To configure the default security risk calculation:

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

.
3. Select **System Settings**.
4. Click **Security Risk Ranking**.
5. Select the desired **CVSS Version** to assess the risk scores within your projects.

   By default Black Duck defines security risk initially using CVSS v4.x scores.

   For more information on the Common Vulnerability Scoring System, see the 3.1 and 4.0 specification documents.
6. Select the desired **Record Type** that will be used to calculate risk categories within your projects. Black Duck uses BDSA and NVD to calculate risk.

   Note: You must have BDSA enabled on your product registration key to take advantage of this feature.
7. Click **Save**.

   A confirmation dialog box appears. Do one of the following:

   - Click **Confirm**.

     The VulnerabilitySummaryFetchJob starts 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.

Note the following:

- Changing 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.
- If a CVE record has a related BDSA record (or vice versa), it cannot be remediated unless that vulnerability record type is prioritized in the Security Risk Ranking. This is due to the fact that the non-prioritized vulnerability record is not being used as a determinant and is not used to calculate security risk.

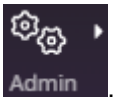
Enabling license term fulfillment

BOM Managers, and other users with the appropriate role, manage the fulfillment status for a license term using the **Term Fulfillment** tab in the *Project Version's* **Legal** tab.

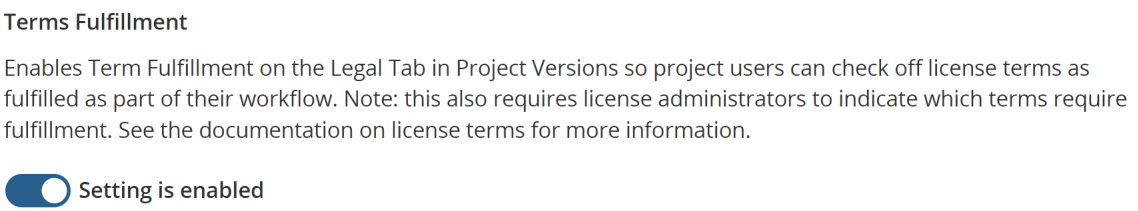
By default, these tabs are disabled. System Administrators must enable these tabs for them to appear to these users.

Note: Enabling the **Legal** tab is a global setting. Once enabled, all project versions will display the **Term Fulfillment** tab in the **Legal** tab.

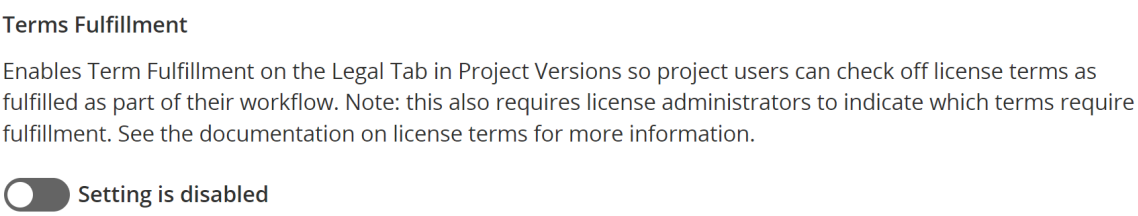
To display the **Legal** and **Term Fulfillment** tabs:

1. Log into Black Duck with the System Administrator role.
2. Click 

.
3. Select **System Settings**.
4. Click **Legal**.
5. Set the toggle located in the **Term Fulfillment** section to *Setting is enabled* to display the **Legal** and **Term Fulfillment** tabs. Enabling the setting will not take effect immediately for existing projects.

   Set the toggle located in the **Term Fulfillment** section to *Setting is disabled* to remove the **Legal** and **Term Fulfillment** tabs. Note that if you select to enable license conflicts, the **Legal** tab will still appear.

Configuring Multi-Factor Authentication (MFA)

To enhance the security of your account, Multi-Factor Authentication (MFA) can be enabled in Black Duck. MFA adds an extra layer of protection by requiring not only a username and password but also a time-based, one-time password (TOTP) generated through a QA code scan. This ensures that even if your password is compromised, access to your account will still require a verification code from your authenticator app.

Please note, MFA is not compatible with Single Sign-On (SSO) or Security Assertion Markup Language (SAML) based authentication methods. However, SAML can be configured after MFA is enabled. In such cases, MFA will apply only when local users log in, as SSO and SAML rely on external identity providers for authentication, bypassing MFA in Black Duck.

Enabling or disabling MFA

To enable or disable MFA:

1. Log into Black Duck as a system administrator user.
2. Click **Admin** → **System Settings** → **Local Authentication**.
3. Switch the **Multi-Factor Authentication** toggle to **Enabled** or **Disabled**.

Note: Enabling and disabling MFA applies to all users in the system. MFA cannot be enabled or disabled for individual users. Once activated or deactivated, this setting applies universally across the platform.

Configuring MFA secret time-to-live

Black Duck enforces a time-to-live (TTL) period for MFA secrets, ensuring they expire after a set duration or when a new secret is requested. This expiration time is configurable, allowing administrators to adjust the validity period to align with their security policies. The default TTL period is two (2) minutes.

To configure MFA secret expiration:

1. Locate and open the Black Duck configuration file where system-wide settings are managed (e.g., blackduck-config.env or Helm values file if deployed in Kubernetes).
2. Add or update the following parameter to define the desired expiration time (in minutes):

```
MFA_SECRET_TIME_TO_LIVE_MINUTE=<desired value in minutes>
```

3. If running Black Duck in a containerized environment, restart the necessary services to apply the changes.

Scanning the QR code to set up MFA

Once Multi-Factor Authentication (MFA) has been enabled, the next step is to configure an authenticator app to generate time-based, one-time passwords (TOTPs).

1. **Choose an authenticator app**

   You will need an authenticator app installed on your mobile device. If you don't already have one, you can download a free app from your app store. Popular options include:

   - Google Authenticator (available on Android and iOS)
   - Microsoft Authenticator (available on Android and iOS)
   - Okta Verify (available on Android and iOS)

2. **Open the app**

After installing the authenticator app, open it and prepare to scan a QR code. Each app may have slightly different navigation, but typically you will find the option to Add Account or Scan QR Code from the main menu.

3. **Scan the QR code**

   You can find the QR code in two places:

   - After logging into Black Duck using your username and password. This is where you will typically configure your MFA for the first time.
   - On your Profile page by clicking your *name* user button on the top right of the page and then selecting the **Profile**tab. If you choose to reset your MFA configuration, a new QR code will be generated from here.

   Using your authenticator app, scan the QR code shown on your screen. This links the app with your account and generates a unique TOTP for future logins.

   Tip: Ensure your phone's camera has access to focus properly on the QR code. Most apps will automatically recognize and scan the code in a few seconds.

   If you are unable to scan the QR code, many apps offer the option to manually enter the key by selecting an option like "Enter Key Manually" in the authenticator app.

4. **Verify the code**

   After scanning the QR code, the authenticator app will start generating 6-digit codes that refresh every 30 seconds. Input the current code in the **Passcode** field.

5. **Complete setup**

   Once you've entered the correct code, the setup process is complete. You will now be prompted to enter a verification code from the authenticator app each time you log into the product.

Resetting your or a user's MFA configuration

If the user needs to reset their Multi-Factor Authentication (MFA) configuration, they can easily do so from their Profile page. Alternatively, a user administrator user can initiate the reset process for any user through the Users page. Resetting the MFA will require the user to go through the initial MFA setup process again, including scanning the QR code and linking the account with an authenticator app.

To reset your MFA configuration as a user:

1. Go to your Profile page by clicking your *name* user button on the top right of the page and then select **Profile**.
2. Under the **Profile** tab, click the **Reconfigure MFA** button found in the **Authentication** section.

   Warning: This action cannot be undone.

To reset a MFA configuration as a user administrator:

1. Click **Admin** → **Users**.
2. Select the desired user from the list displayed.
3. Click the **Reset User MFA** button in the **Local Authentication** section. The user will then be asked to reconfigure MFA upon their next login.

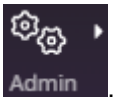
Account locking

If a user enters an incorrect Multi-Factor Authentication (MFA) code or password ten times consecutively, their account will be temporarily locked for 10 minutes. If you find yourself locked out, please reach out to your user administrator for assistance.

Changing notification subscriptions

As a System Administrator, you can select what notifications are sent to the users in your organization. Notification types that are available to users can be globally enabled or disabled. Reducing the enabled notification types can improve overall system performance. Please note that users with the Global Notification Viewer role will still receive all notifications on the system.

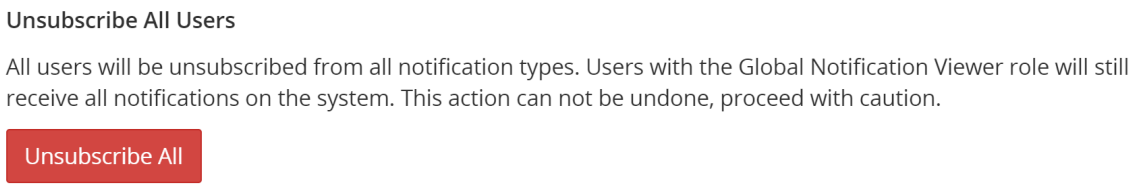
To change notification settings:

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

.
3. Select **System Settings**.
4. Click **Notifications**.

Unsubscribing all users

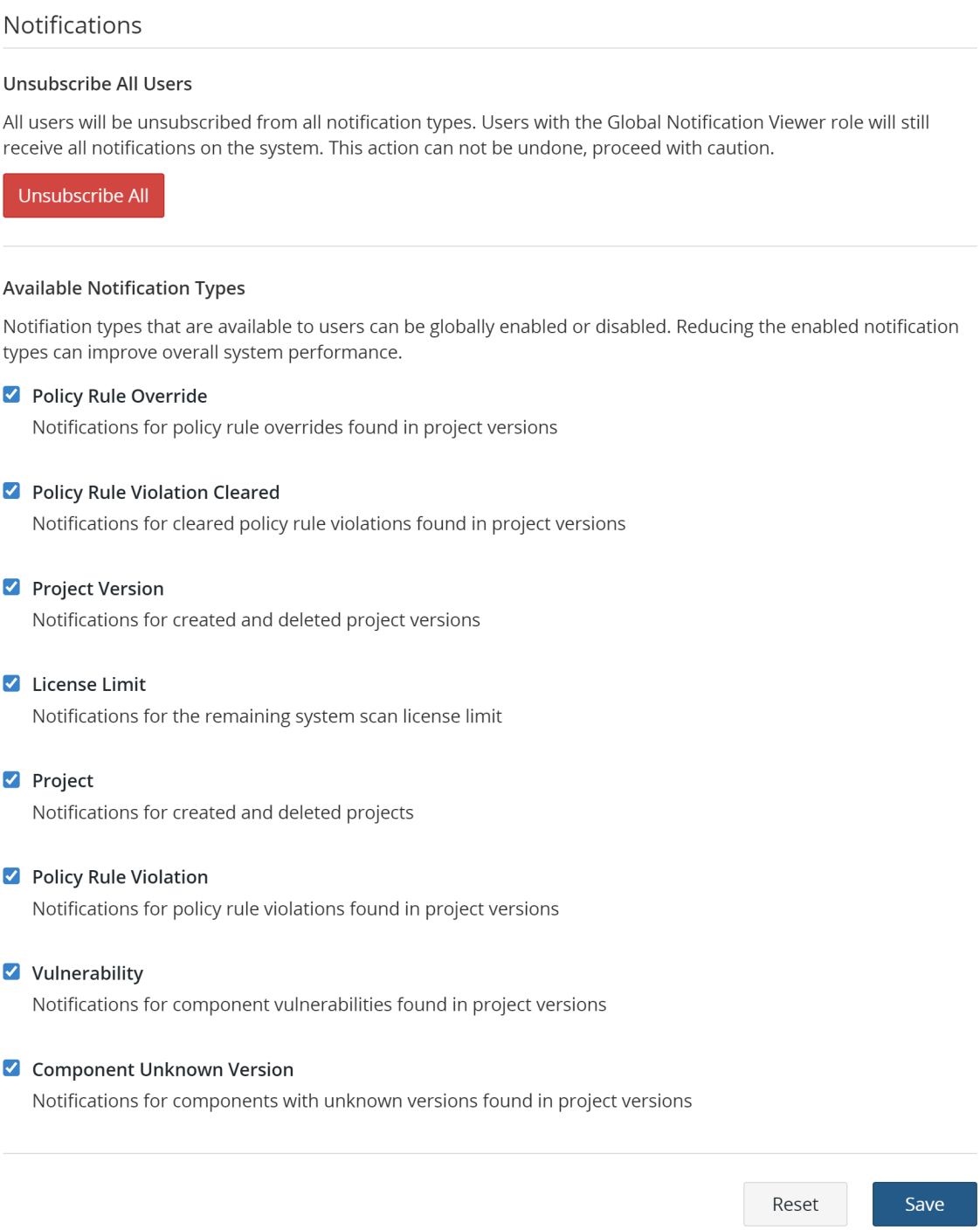
You have the ability to unsubscribe all users from all notifications. You can do so by clicking the **Unsubscribe All** button. This action can not be undone, proceed with caution.



Selecting available notification types

You can also select which notification types are globally enabled or disabled for your users. Reducing the enabled notification types can improve overall system performance.

To enable or disable a particular notification type, add or remove the checkmark associated to the desired notification.



Hosting location for Black Duck Detect

Managing and updating the versions of Detect across various pipeline jobs in Black Duck can be a challenge. When incompatible versions of Detect and Black Duck are used, it can take a lot of time and effort to update all jobs. Additionally, it's not always clear which version of Detect is being used or which versions are available for a given Black Duck version.

Black Duck offers two means to connect with Black Duck Detect to better suit your needs; Internally Hosted and Black Duck Hosted.

How does it work?

When Detect is invoked to scan source files, it first determines the configuration set in Black Duck (see below) and then validates the version set in Black Duck. This information is then communicated with Detect on client side.

If there is a difference between the client Detect version and the configuration set in Black Duck, Detect will pull the proper Detect version as configured in Black Duck and scan the source with the newly pulled archive instead.

- Enabling the **Internally Hosted** setting provides the option to host the Detect Binary JAR file directly on your own Artifactory to be pulled with the specific version specified for all users. If set to Internally Hosted, Detect will pull and use the version dictated in the Detect URL field.

  Using this option allows integration with Code Sight and Detect, but internally hosting the Detect JAR file does not provide a complete Detect installation; it will not include any inspector scripts or inspector tools like in a full air-gap mode installation and is not meant as an alternative to deploying Detect via air-gap mode.

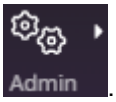
  Additionally, scan host machines still require access to the Internet for full functionality.

  ⚠ **Warning:** Black Duck does not validate the JAR file obtained from the provided internally hosted URL. Ensure that a valid version of the Detect JAR is available for downloaded in the hosting location.
- Using the **Black Duck Hosted** setting allows the option to use our Black Duck repository to download the Detect version set based on the system setting configured.. If set to Black Duck Hosted, it will pull from our repository directly from client side.
- Code Sight users can find Detect configuration instructions at **Setting Up Black Duck® SCA for Code Sight**.

Internally hosted Black Duck Detect

Users with limited external connectivity can define the internal hosting location of Black Duck 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.

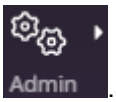
To specify the hosting location of Black Duck Detect:

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

.
3. Select **System Settings.**
4. Click **Black Duck Detect** in the left-hand menu.
5. Click the **Internally Hosted** box.
6. In the **Hosting location for Black Duck Detect** section, enter the valid URI for your internal instance of Black Duck Detect.
7. Click **Save**.

Black Duck hosted Detect

Non-airgapped users who want Black Duck to manage the version of Detect to use can select the Black Duck Hosted option:

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

.
3. Select **System Settings.**
4. Click **Black Duck Detect** in the left-hand menu.
5. Click the **Black Duck Hosted** box.
6. Select the desired version of Black Duck Detect from the **Black Duck Detect Version** dropdown menu.
7. Optionally, check the **Force newer versions of Detect to downgrade** box if you want to ensure users cannot perform scans with newer versions of Detect. If enabled, Black Duck Detect will downgrade to the selected version.

   Note: Black Duck Detect does not support downgrading itself to versions prior to 8.9.0 because such a downgrade will lose the ability to self-update again.
8. Click **Save**.

Viewing product registration information

The Product Registration page allows system administrators to manage and configure the licensing and registration details for your Black Duck SCA instance. This page is accessible from the Administration section of Black Duck SCA.

Product Registration

The Product Registration section displays the current registration key associated with your Black Duck SCA instance. This section includes the following information:

- **Registration ID**: The unique key used to activate and register your Black Duck SCA instance, unlocking the features and modules included in your license.

  The **Refresh** button manually contacts the Black Duck SCA master registration server (`updates.suite.blackducksoftware.com`) to download the latest registration data for your instance. During a refresh, the Black Duck SCA instance uploads usage metrics and receives back the full registration payload, including:

  - License properties
  - Feature flags
  - Expiration dates
  - Resource limits

  This process occurs automatically approximately every 6 hours. However, the **Refresh** button allows an administrator to trigger it on demand — for example, immediately after a license change has been made by customer support, eliminating the need to wait for the next automatic cycle.

  Note: The Refresh button is disabled for air-gapped users.
- **Registration State:** An indicator showing whether the current registration key is valid, expired, or invalid.
- **Expiration Date:** The date on which the current registration key and associated license will expire.

- **Reset** button: Re-activates (re-claims) the registration key with the Black Duck SCA master registration server. This performs a full round-trip activation, which is useful if registration data has become corrupted or needs to be re-established from scratch.

  Note: The Reset button is disabled for air-gapped users.

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

- **AI Model Scanning**

  AI Model Scanning enables identification of AI models in scanning. AI models will appear within a project version's Bill of Materials and provide associated metadata.

- **Artifactory Integration**

  Artifactory Integration is a service that enables scanning of artifacts within a set of configured repositories to identify open source components, using data gathered by Component Scanning.

- **Black Duck Binary Analysis**

  Black Duck Binary Analysis (BDBA) is a service that provides enhanced interrogation of binaries to surface the open source components within open source. It also supports expanded file type support including various firmware formats, filesystem/disk images, installation formats and various compression and archive formats.

- **Black Duck Secure Container (BDSC)**

  Black Duck Secure Container scanning provides capabilities to identify components within container images, their layers and base images.

- **Black Duck Security Advisory**

  The Black Duck Security Advisory (BDSA) is a Black Duck-exclusive vulnerability data feed, sourced and curated by our Security Research team which is part of the Black Duck COSRI (Centre of Open Source Research & Innovation). A BDSA offers deeper coverage for a wider set of vulnerabilities than is available through the NVD (National Vulnerability Database), providing detailed vulnerability insight including severity, impact, exploitability metrics and actionable remediation guidance. The BDSA data for new vulnerabilities is reported an average of three weeks earlier than the NVD's reported data.

- **Component Scanning**

  Component Scanning automates the discovery and identification of OSS components in a scan to provide metadata such as license type, security vulnerabilities, and OSS project health for those components. Component scans can be linked to internal project versions to automatically generate BOMs.

- **Cryptography**

Cryptography Management enables the display of cryptographic algorithms, and additional metadata, that are contained in open source components. This data is used to support compliance to security standards and legal export regulations.

- **License Management**

  License Management is the feature which allows Black Duck users to edit and maintain the data which can be used to create accurate and compliant open source notice files/reports at a Project/Release level.

- **Match as a Service**

  Match as a Service (MaaS) is a service that identifies OSS components, using data gathered by Component Scanning.

- **Notifications**

  Black Duck notifications alert your teams when vulnerabilities change or there are policy violations. Your organization can integrate with other platforms using the Notification API.

- **OSS Notices Report**

  The Notices Reports feature allows users to create notices (or attribution) reports for their projects. The notice files can then be included with the distribution or incorporated into documentation to satisfy the attribution obligation that exists in the vast majority of open source licenses.

- **Policy Management**

  The Policy Management feature enables companies to define rules to govern their use of open source components. With these rules, open source usage can be managed on an exception basis, i.e, as long as open source components meet the policy requirements their usage is allowed, thereby speeding time to market and freeing developers from a cumbersome approval process. Any open source components/versions that fail to meet policy are flagged, enabling a review process to determine if the use of the component should be allowed in the particular application.

- **Risk Management**

  Risk Management enables the identification, notification, and remediation of the security, license, and operational risks associated with the OSS components used in your internal projects.

- **SCM Integration**

  SCM Integration enables the configuration and authentication of Source Code Management (SCM) providers. It allows mapping, scanning, and management of SCM repositories in projects. The SCM integration feature must be enabled on your registration key for this to appear on the Product Registration list.

- **Snippets**

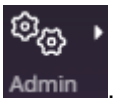
  This feature enables the option to invoke optimized scans of source files which find OSS usage at the file or code snippet level. These scans work in conjunction with a component scan to produce the best possible Bill of Materials (BOM) for a project. Once discovered, matches can be reviewed and added to a project BOM which pulls in the associated metadata for the component.

- **Vulnerability Exploitability eXchange (VEX)**

  Vulnerability Exploitability eXchange (VEX) documents are intended to communicate whether specific vulnerabilities in software components are exploitable in given product(s). This helps organizations prioritize their remediation efforts and gives better visibility of security risk for the associated products to customers.

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 

.
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**.

Data Retention

Project Version Auto Actions

Project Version Auto Actions allows you to automate the management of project versions based on defined criteria. This functionality allows users to efficiently handle version accumulation, particularly in environments with version limits, disk space constraints, or database performance issues.

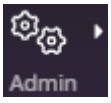
Key capabilities include:

- **Automatic Progression to LTS**: Users can configure project versions to automatically transition to Long-Term Support (LTS) status, ensuring ongoing maintenance and security.
- **Automatic Deletion of Obsolete Versions**: Users can enable the automatic deletion of project versions that are no longer needed, helping to maintain a clean and efficient workspace while preventing resource overuse.

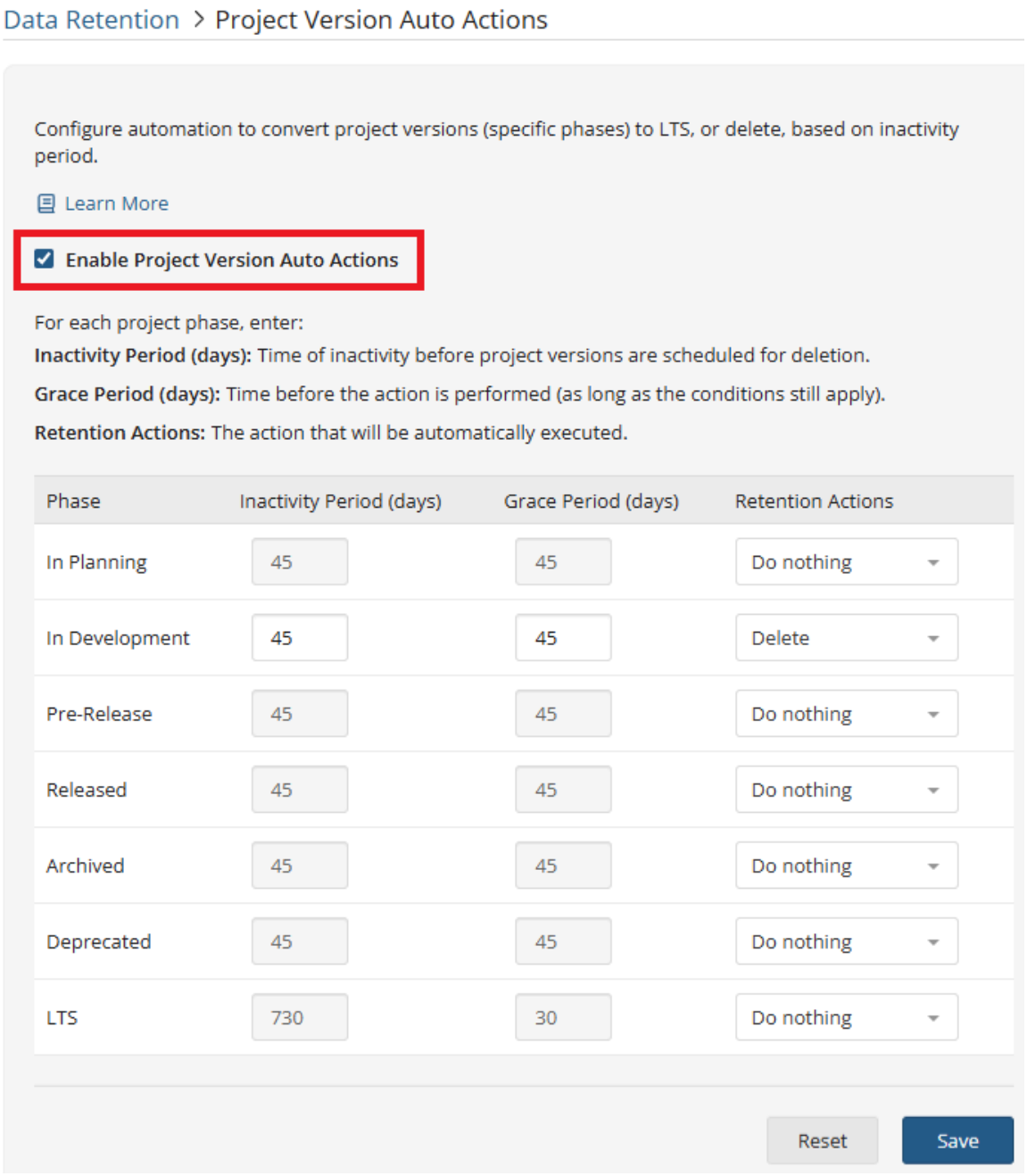
This feature is particularly beneficial for users generating multiple project versions over time, helping to enhance overall efficiency and project organization.

How to enable or disable project version auto actions

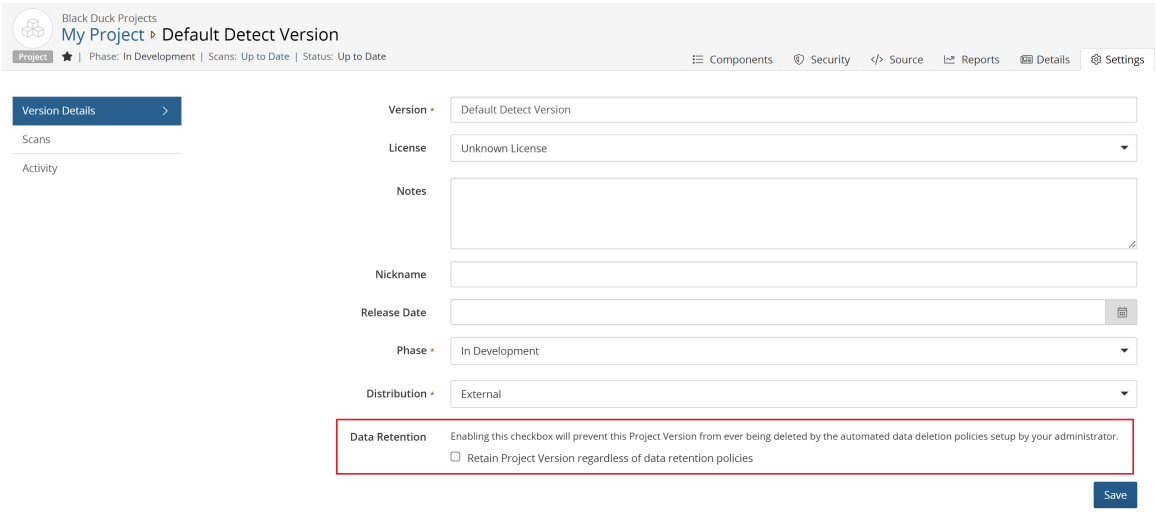
To enable or disable project version auto-actions:

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

→ **System Settings**.
3. Click **Data Retention**.
4. Click **Project Version Auto Actions**.
5. Check or uncheck the **Enable Project Version Auto Actions** checkbox.



Enabling this feature will cause new data retention fields on the project version's settings page to be displayed.



Project version auto action settings

Phase: Defines what project version phases are applicable to the auto-deletion process.

Inactivity Period (days): Defines the time period in which a version has to be inactive for it to get scheduled for the selected retention action in the future. Inactivity is defined as no scans and no edits for the project version including project version settings, component edits, vulnerability remediations and policy overrides. The default value is 45 days and maximum is 365 days.

Grace Period (days): Defines how far in the future project versions will be scheduled for the desired retention action (i.e. how long the warning icon shows up in the UI before the project version is actually deleted). The default value is 45 days and maximum is 365 days for active projects, 1825 days for LTS.

Retention Actions: The action that will be automatically executed. Selection options are:

- Delete: Delete this project version.
- Do nothing: Do not take any action for this project version.
- Convert to LTS: Convert this project version to Long-Term Support.

Criteria for project version auto action

The following criteria is used to determine if a project version is a candidate for auto action:

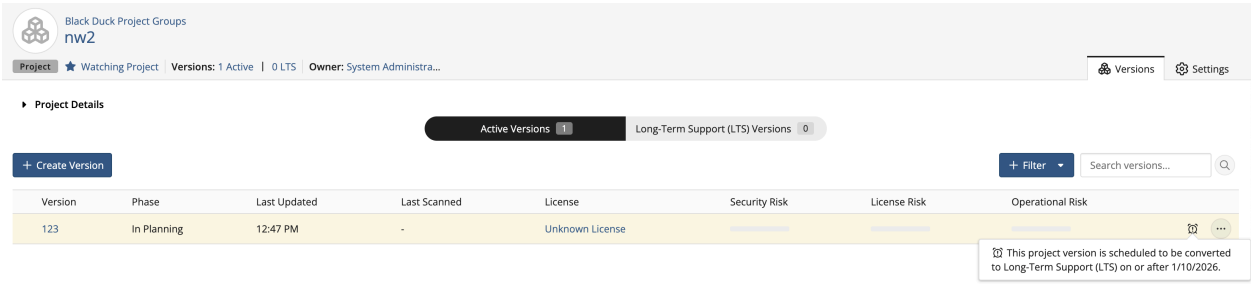
- The project version phase matches any of the release phase values set in **Project Version Phase Condition**.
- The project version is not a sub-project that is included as a component on other project version's BOM.
- The project version is not part of a project in which "Custom scan signatures" are enabled.
- The project version has been inactive during the timeframe specified in the environment variable.
- The project version is not protected from auto-deletion on the project version settings UI in the Data Retention section.

Any project versions which fall outside of these constraints will not be tagged for auto action.

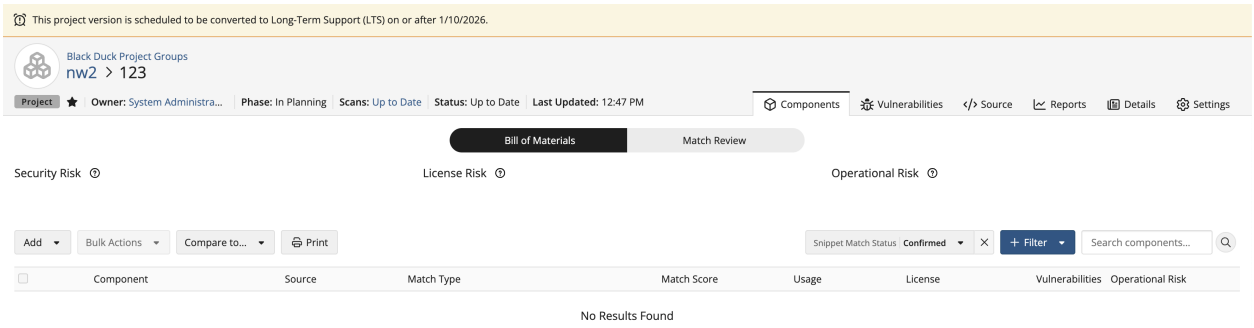
Project versions marked for auto action

Once a project version is tagged for auto action, a  icon is displayed on the project's overview screen:

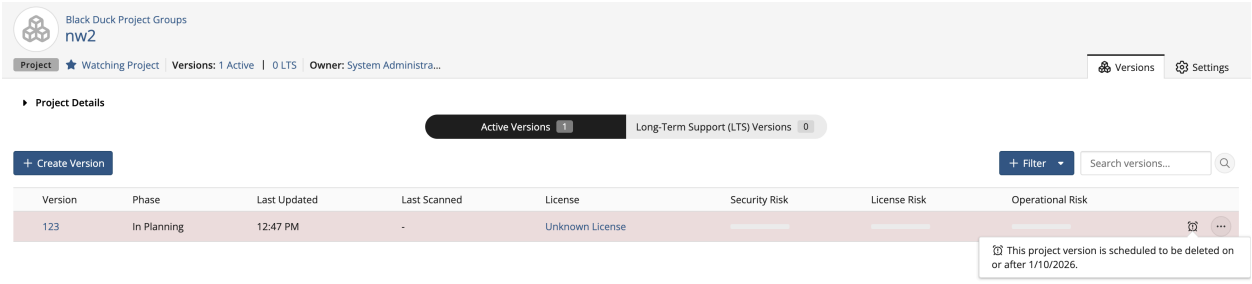
- Project versions marked for auto deletion are highlighted in yellow.



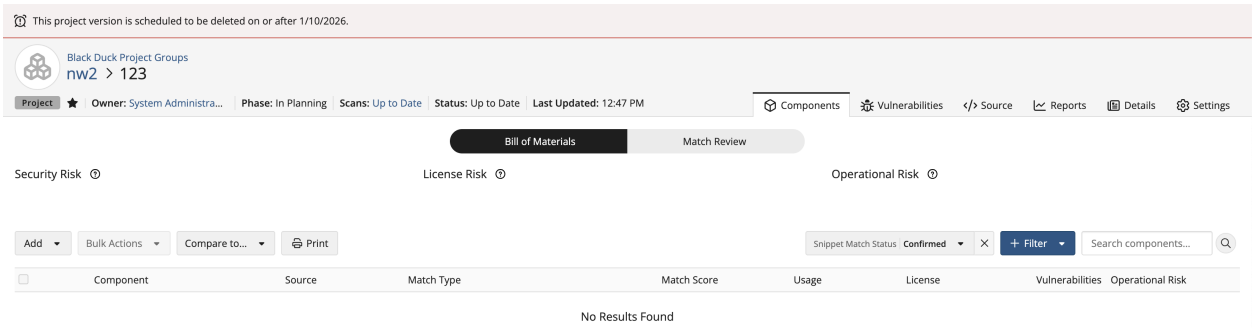
A banner will also be displayed on the top of the project version's BOM page.



- Project versions marked for auto deletion are highlighted in red.



A banner will also be displayed on the top of the project version's BOM page.



After a project has been flagged for auto action, if it subsequently has activity against it, it will be “unflagged” for the selected auto action and the inactivity counter will start again.

The jobs which sets the auto action status and deletes the project versions runs every 15 minutes.

After project version auto-deletion

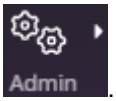
Once the elapsed grace period has passed, the project version will be deleted. This will orphan any scans for the project version as they will no longer be associated with a project version. Those scans will be deleted according to the “Cleaning up unmapped code locations” configuration as defined in the install guide (default = 30 days).

Please note, if all project versions for a project are deleted, then the project is also deleted.

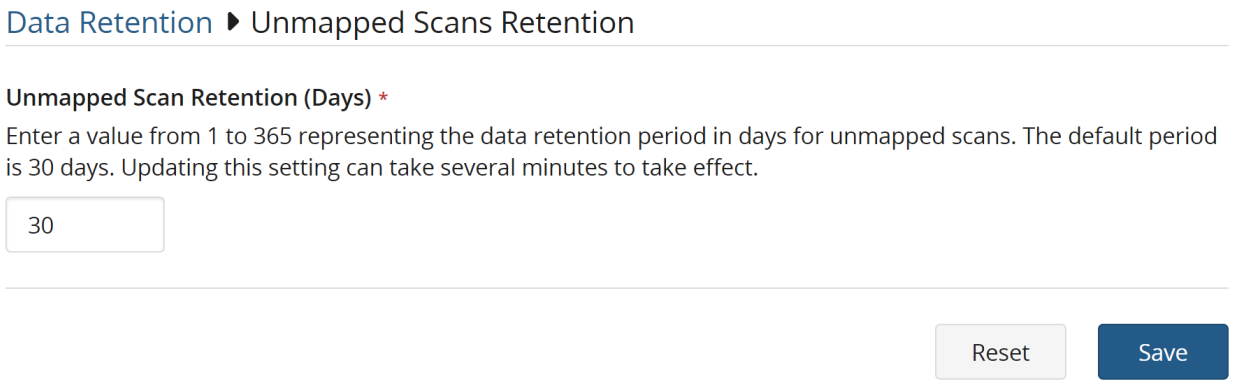
Changing the data retention period for unmapped scans

You can change the period of time unmapped scans are retained. By default, the time frame is set to 30 days and can be set to as low as 1 day and to as high as 730 days.

To change the data retention period for unmapped scans:

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

.
3. Select **System Settings**.
4. Click **Data Retention**.
5. Click **Unmatched Scans Retention**.



Note: Changing this setting will take effect the next time ScanPurgeJob runs, which occurs every 15 minutes.

Protecting scans by project version

You can prevent scans in designated project versions from being unmapped by **Scan Auto-Unmapping** by following these steps:

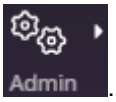
1. Navigate to the desired project version.
2. Click the **Settings** tab.
3. In the Version Details pane, find the **Scan Retention** section.
4. Check the **Prevent automatic unmapping of scans** checkbox.

SBOM Retention

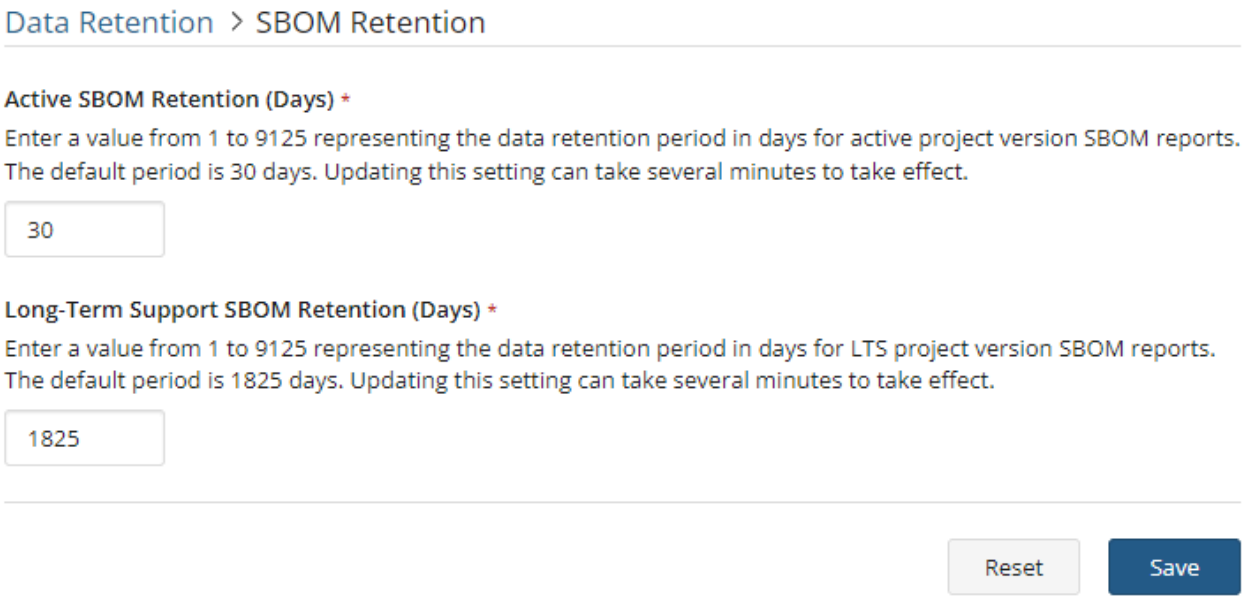
When you generate an SBOM that is meant to be distributed, it's important that an SBOM management solution retains the SBOM so it can be reproduced if needed. This is different than other type of Black Duck reports and while it typically happens as part of the release process at a point in time when no further

changes are expected to the BOM, that's not always the case. With SBM Retention, you have more control over how long SBOMs are retained for both active and long-term support projects.

To change the data retention period for SBOM reports:

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

.
3. Select **System Settings**.
4. Click **Data Retention**.
5. Click **SBOM Retention**.



6. Enter a valid value for the desired project version status:
   - **Active SBOM Retention (Days)**. Enter a value ranging from 1 to 9125 days. The default period is 30 days. Changing this value will affect all active project version SBOM reports.
   - **Long-Term Support SBOM Retention (Days)**. Enter a value ranging from 1 to 9125 days. The default period is 1825 days. Changing this value will affect all long-term support (LTS) project version SBOM reports.

Note: Updating these values can take several minutes to take effect.

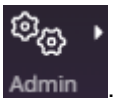
Retain Unmatched File Data

Changing the Retain Unmatched File Data setting globally

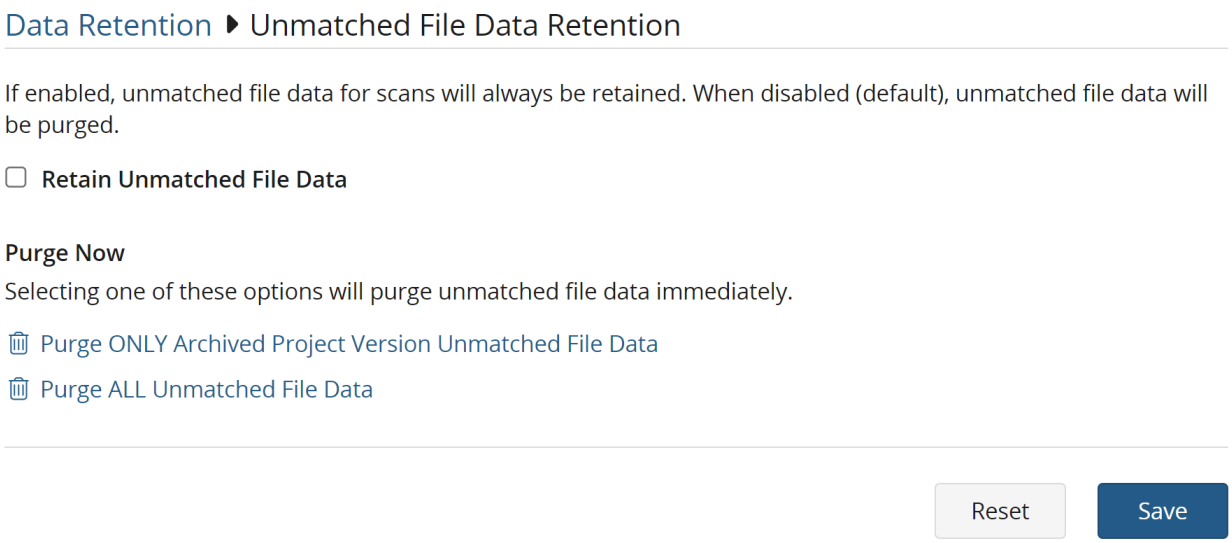
You can change whether or not your system retains unmatched files. By default, this setting is not enabled.

To enable the retain unmatched files data setting:

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

2. Click 

.
3. Select **System Settings**.
4. Click **Data Retention**.
5. Click **Unmatched File Data Retention**.



6. Check the **Retain Unmatched File Data** checkbox.

If enabled, unmatched file data for scans will always be retained. When disabled (default), unmatched file data will be purged. Note that the global setting only applies to projects and scans that do not explicitly specify their own setting; similarly, changing the global setting does not affect projects or scans that do specify their own setting.

⚠ **Warning:** Once unmatched files are purged, they cannot be recovered except by restoring from backup.

You can also manually purge unmatched data immediately by clicking either of the following buttons:

Purge ONLY Archived Project Version Unmatched File Data: Clicking this button will purge all unmatched data currently in the Archived project version phase only.

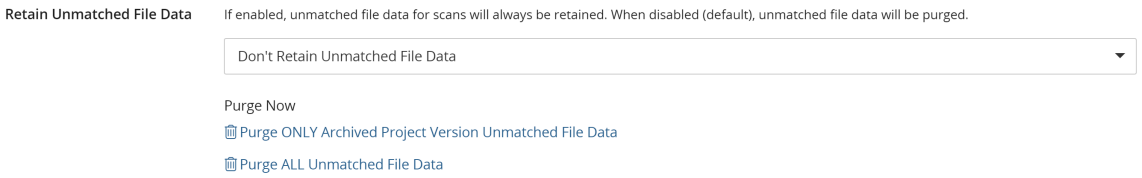
Purge ALL Unmatched File Data: Clicking this button will purge all purge all unmatched data regardless of its project version phase.

As indicated above, clicking either button only applies to projects and scans that do not explicitly specify their own setting.

Changing the Retain Unmatched File Data setting for a project

You can set a project's setting to have its own policy of whether or not the unmatched file data is purged by following these steps:

1. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
2. Select the **Settings** tab.
3. Scroll to the **Retain Unmatched File Data** section.



4. Select an option from the dropdown menu:

   Retain Unmatched File Data: Selecting this option will prevent this project's data from being purged regardless of the global system default setting. This also disables the buttons granting the ability to manually purge unmatched file data.

   Don't Retain Unmatched File Data: Selecting this option will allow this project's unmatched file data to be purged. This also enables the buttons granting the ability to manually purge unmatched file data.

   System Default: Selecting this option will use the global system default setting as set above. The buttons granting the ability to manually purge unmatched file data will either be enabled or disabled depending on how the system setting is configured.

5. Click **Save**.

You can also manually purge unmatched data immediately for this project like from the Retain Unmatched File Data administration page.

Scan auto-unmapping

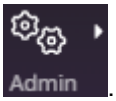
When enabled, inactive scans are scheduled to be unmapped from project versions where at least one scan of the same type, updated date is older (defined by the combined value of inactivity and grace periods) than the most recent scan of the same type. They need to meet the project version phase and inactivity time conditions, and will only be unmapped after the grace period.

Important: The last scan of a project version of a given type is always protected. For example, if a project version has 2 signature scans and 1 dependency scan, only one of the signature scans can end up being unmapped.

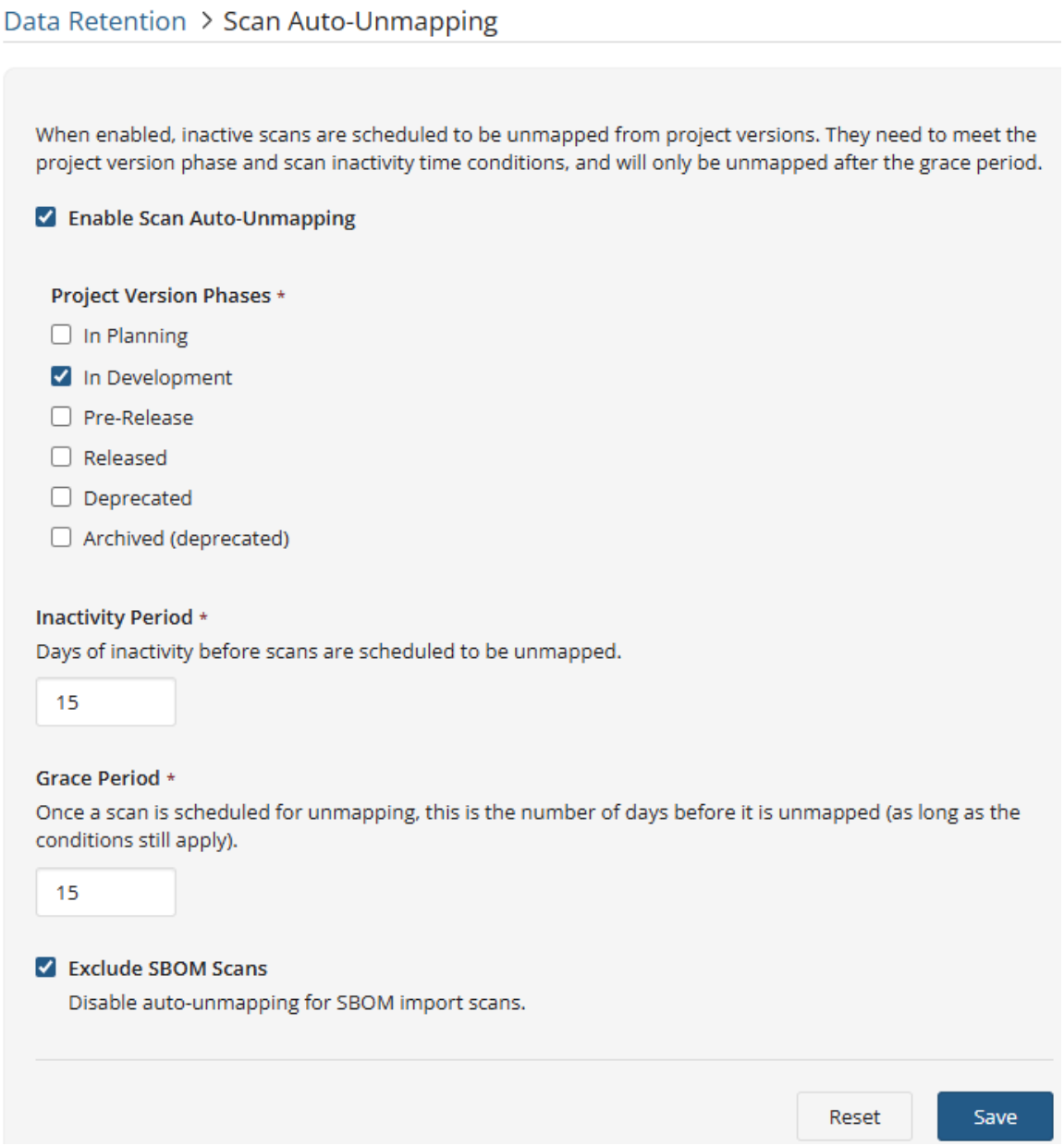
Warning: If this is the first time the Project Version Scan Auto-Unmapping is enabled, it may take a while to finish processing depending on the amount of scans in your environment.

Enabling project version scan auto-unmapping

To enable the project version scan auto-unmapping setting:

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

.
3. Select **System Settings**.
4. Click **Data Retention**.
5. Click **Scan Auto-Unmapping**.



6. Check the **Enable Project Version Scan Auto-Unmapping** checkbox.
7. Select all applicable project version phases.
8. Enter the amount of days of inactivity before scans in the project version phases selected above are scheduled to be unmapped. Default is 15 days.
9. Enter the amount of grace period days. Once a scan is scheduled for unmapping, this is the number of days before it is unmapped (as long as the conditions still apply). Default is 15 days.
10. Click **Save**.

Start-up grace period

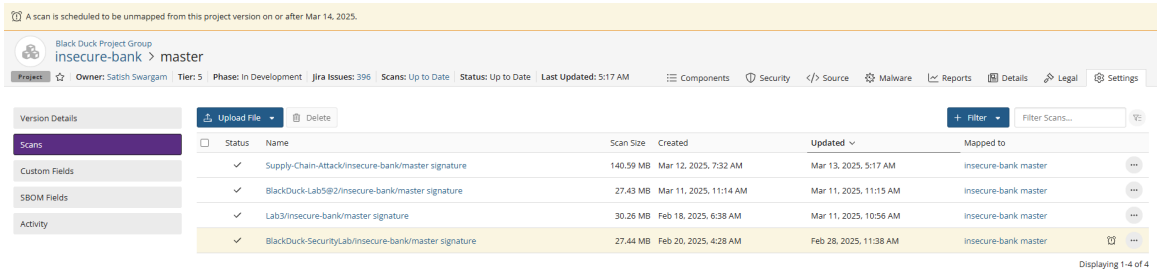
When the system starts for the first time with Scan Auto-Unmapping enabled or when Scan Auto-Unmapping transitions from disabled to enabled on the Settings page, the period of time entered in the **Grace Period** will act as a "start-up" grace period. The Start-up grace period freezes all activity until it ends.

For example, when you enable Scan Auto-Unmapping and set the **Grace Period** to 15 days, the system delays doing anything with scans until 15 days have passed.

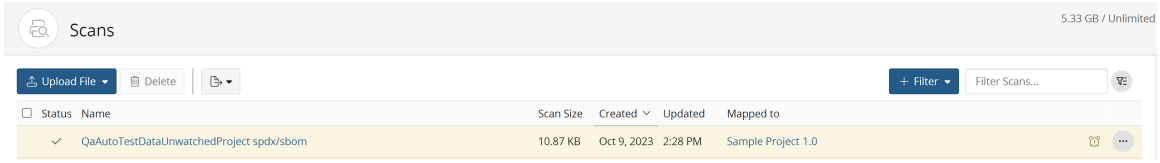
Scan lifecycle with Scan Auto-Unmapping enabled

Scans are considered to be in an active state for a period of time determined by the **Inactivity Period** starting from their Last Updated date. They remain active if they are re-scanned during this time frame. When a scan has not been re-scanned and exceeds the **Inactivity Period**, they enter the **Grace Period** for a period of time as configured above. During the **Grace Period**, affected scans will be indicated with a ⏰ icon and/or warning message.

- On the Project Version page, a scan targetted for unmapping will have a ⏰ at the end of its row. In the screenshot below, note that only the fourth scan has been marked for unmapping. The other scans will remain.

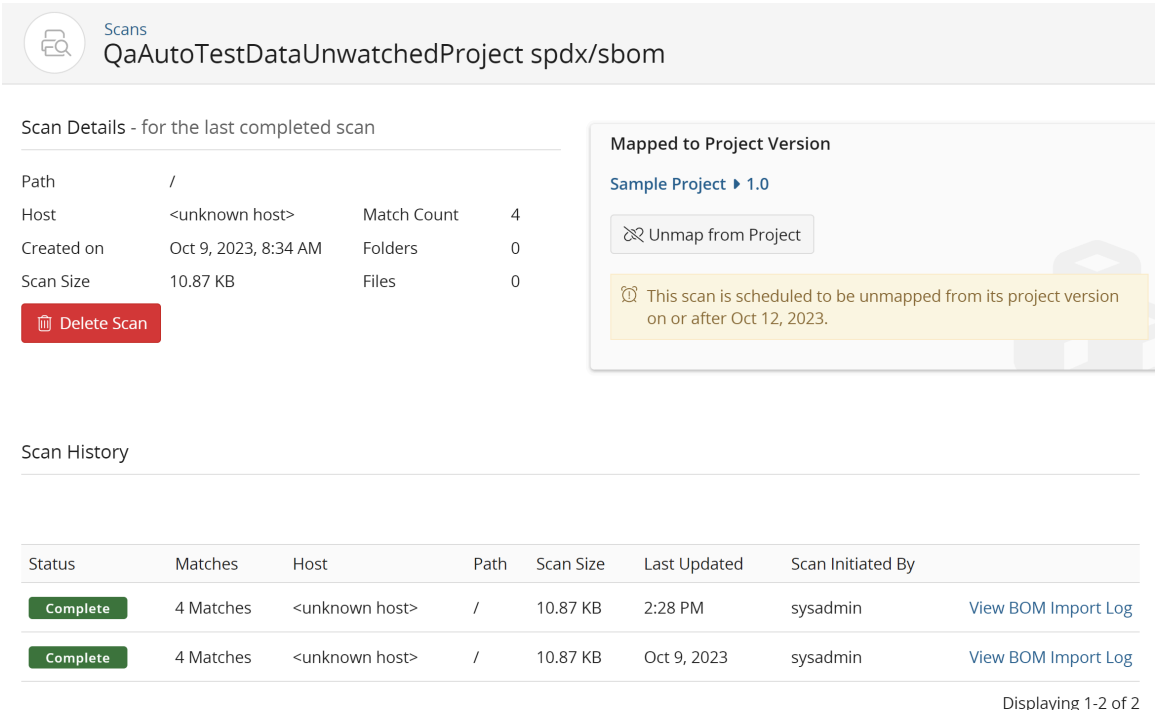


- On the Scans page, the scan will have a ⏰ at the end of its row.

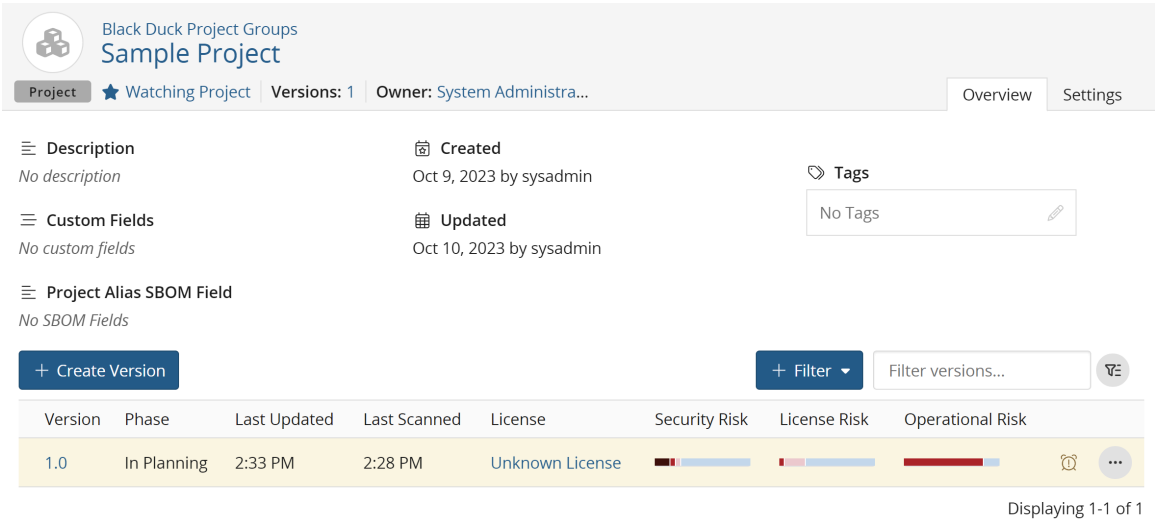


  Mousing over this icon will display a message stating when this scan will be unmapped from its project version.

- On the *Name of scan* scan page, in the **Mapped to Project Version** section, a warning message beneath the **Unmap from Project** button is displayed, stating when this scan will be unmapped from its project version.

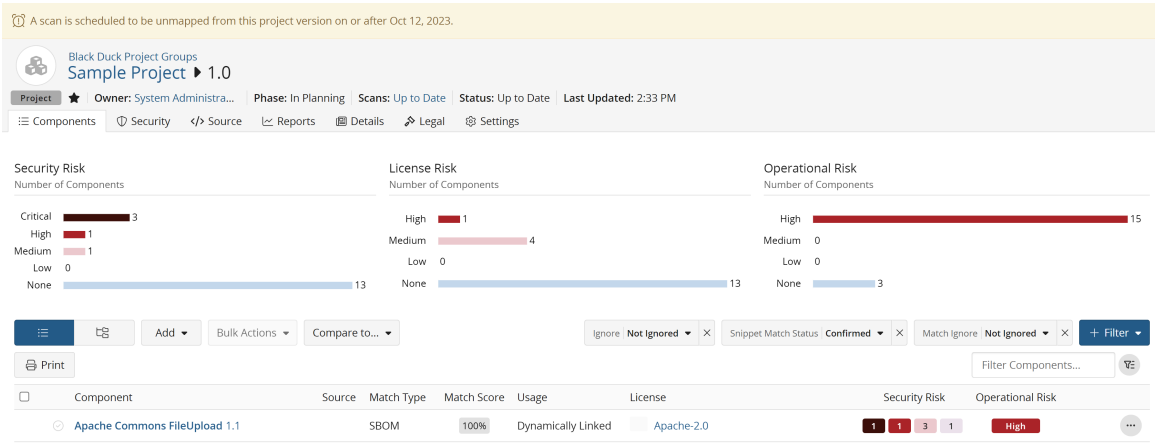


- On the *Project name* page, the project version will have a ⏰ at the end of its row.

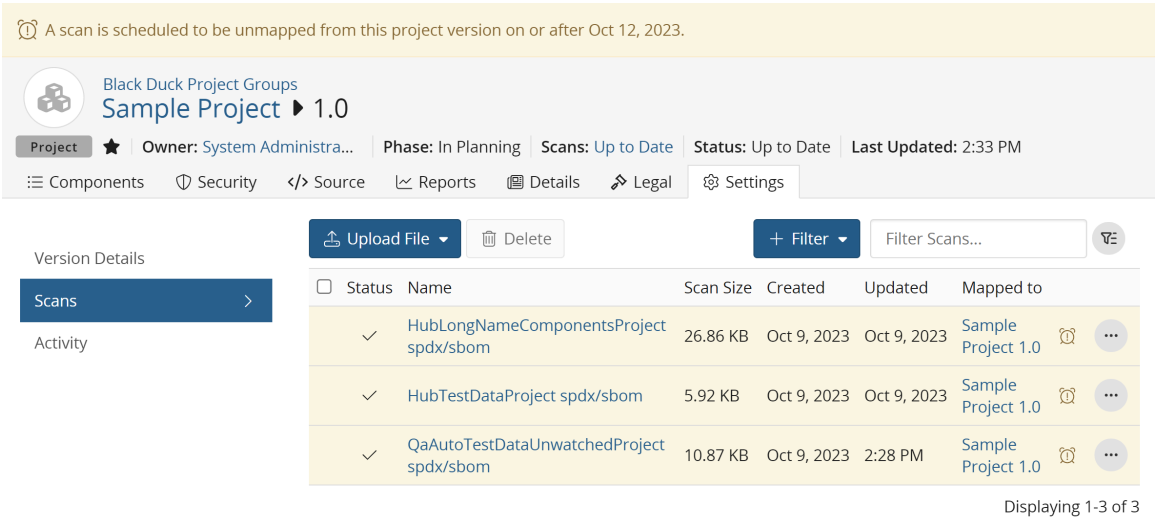


Mousing over this icon will display a message stating when this scan will be unmapped from its project version.

- On the **Components** tab of the *Project version* page, a banner will be displayed on the top of the page with a ⏰ and a message stating when this scan will be unmapped from this project version.



- On the **Settings** tab of the *Project version* page, the Scans section will display all scans for this project version.



  Affected scans will have a ⏰ at the end of its row. Mousing over this icon will display a message stating when this scan will be unmapped from this project version.

Once the **Grace Period** expires, the Purge Scan Data - Unmap stale scans job will unmap the scan.

Be aware that your code location could become unmapped even if you are conducting regular scans. This can occur if the following conditions are met:

- The unmapping period, plus grace periods, is set to a very short amount of time (less than 7 days)
- No code changes are made within the system for a period of 7 or more days

We strongly recommend setting the combined unmapping period plus grace period to at least 7 days to avoid this issue. The default setting is configured to a higher duration to ensure better mapping continuity.

Protecting scans by project version

You can prevent scans in designated project versions from being unmapped by **Scan Auto-Unmapping** by following these steps:

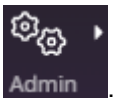
1. Navigate to the desired project version.

2. Click the **Settings** tab.
3. In the Version Details pane, find the **Scan Retention** section.
4. Check the **Prevent automatic unmapping of scans** checkbox.

Disabling auto-unmapping for SBOM import scans

You can globally disable the automatic unmapping of scan types for SBOM (Software Bill of Materials) imports. This prevents the unintended unmapping of SBOMs when multiple SBOMs are imported into the same project version. By excluding SBOM scan types from the automatic unmapping functionality, you can maintain their manual mappings, ensuring greater accuracy and control over the mapping process.

By default, Exclude SBOM Scans is enabled by default. To disable it:

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

.
3. Select **System Settings**.
4. Click **Data Retention**.
5. Uncheck the **Exclude SBOM Scans** checkbox.
6. Click **Save**.

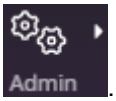
Activity Audit Trail

The Activity Audit Trail feature allows for the retention of activity audit records of user actions and key events, such as project version component and vulnerability records, in the application affecting a project and/or project version.

This feature is disabled by default for fresh installations.

Tip: Disabling this feature can increase performance and decrease storage costs.

To enable Activity Audit Trail:

1. Log into Black Duck as a System Administrator user.
2. Click 

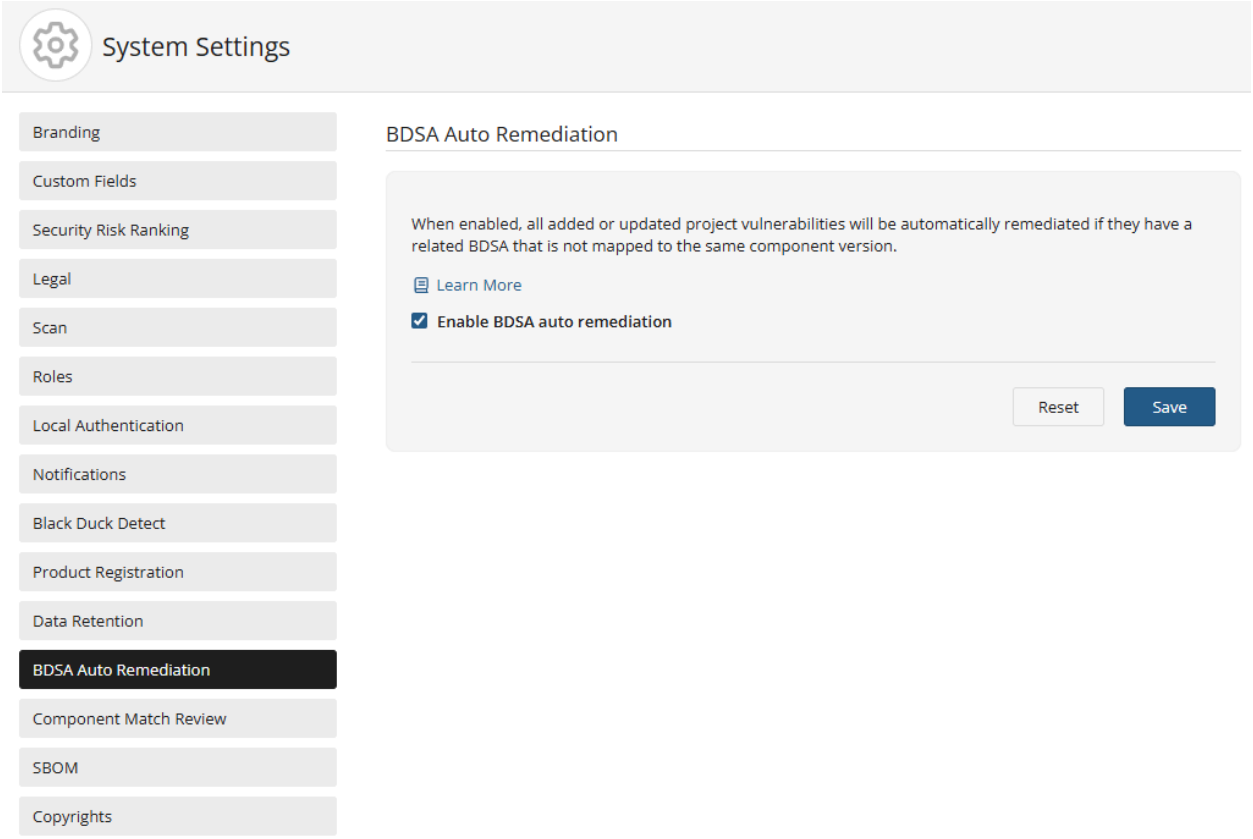
.
3. Click **Data Retention** on the left-hand menu.
4. Click the **Activity Audit Trail** tile.
5. Click the checkbox slider to display **Enabled**.

For more information on how the Knowledge Base update job interacts with the audit trail and affects vulnerability management, please refer to Understanding the Knowledge Base Update Job

BDSA Auto Remediation

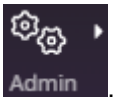
When the Black Duck Security Advisory (BDSA) team analyzes a CVE vulnerability, they check to see what component versions are affected by the vulnerability. Sometimes they find that the vulnerability applies to a different set of versions. This feature will give you the ability to automatically ignore CVE vulnerabilities if the BDSA team has found that the vulnerability does not apply to that component version.

This setting only applies to CVE vulnerabilities with a related BDSA vulnerability. If the CVE is mapped to a component version, but its related BDSA is not also mapped to that component version then the system may automatically remediate the CVE vulnerability based on the feature setting.



Changing the BDSA Auto Remediation setting

To change the BDSA Auto Remediation setting:

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

.
3. Select **System Settings**.
4. Click **BDSA Auto Remediation**.
5. Check the **Enable BDSA auto remediation** checkbox to activate the feature or remove the check to disable it.
6. Click the **Save** button to save the current setting (on or off) for the BDSA Auto Remediation feature and apply the changes going forward for all new scans. It will not update any existing CVEs.

For more information regarding Black Duck Security Advisories and how to remediate vulnerabilities, please refer to:

- Black Duck Security Advisories
- Remediating security vulnerabilities

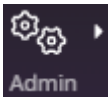
Configuring the Component Match Review Threshold

The Component Match Review process allows for the management of components in the Bill of Materials (BOM) based on match scores during scans. This helps ensure that components requiring further review are identified and addressed appropriately.

Match Score Thresholds:

- Components with a match score below 4% will not be generated in the BOM.
- Components with a match score greater than or equal to 4% but less than 90% will be directed to the Match Review page for human review. This allows users to evaluate ambiguous matches before making decisions on their inclusion in the BOM.

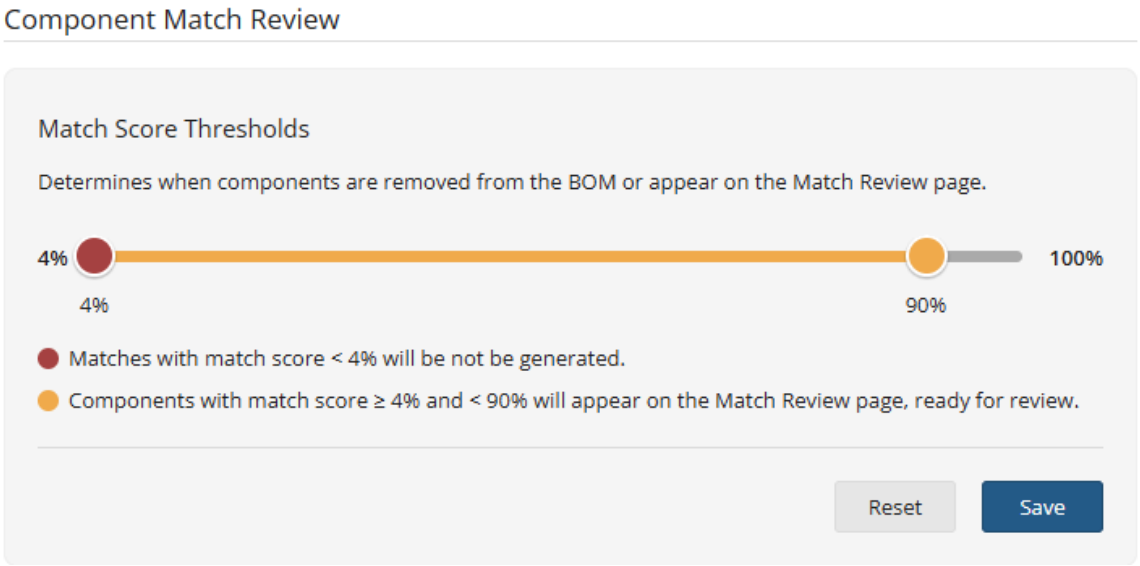
To configure the Component Match Score thresholds:

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

3. Select **System Settings**
4. Select **Component Match Review**
5. Use the slider control to adjust the lower and upper thresholds according to your organization's requirements or policies. The slider can also be adjusted using the keyboard, where the Tab key switches between the two sliders and the arrow keys are used for fine-tuning.

   The red slider sets the "remove" threshold, which determines components that will be directed to the Match Review page after the next scan. The yellow slider sets the "warning" threshold, marking components scoring below this threshold as low score, which will be highlighted in gray in the BOM.

   Note: It is crucial to configure the thresholds appropriately to avoid losing true positives in your match results.



6. Click **Save**.

Slider control limitations

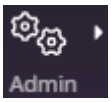
The slider has specific limitations: the "warning" threshold cannot be lower than the "remove" threshold, and the "remove" threshold cannot be adjusted above the "warning" threshold or set lower than 4%. As a result, components with a match score of less than 4% will not appear in the BOM.

Configuring the default license for unmatched components

The licence for auto-created unmatched components found when uploading a report file on the Scans page can be configured from the SBOM page in the System Settings.

Important: This license will exclusively apply to components where the SBOM license value is `NOASSERTION`. It will not add the default license to components where license has no value.

To set the default license:

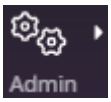
1. Log in to Black Duck as a System Administrator.
2. Click 

and select **System Settings**.
3. Select **SBOM** from the lefthand menu.
4. Select the desired license from the **License Name** dropdown box. By default, the selected license is Unknown License.

Configuring copyright options

You can configure options to improve copyright lists, which are used in SBOM and Notices File reports.

To configure copyright options:

1. Log in to Black Duck as a System Administrator.
2. Click 

and select **System Settings**.
3. Select **Copyrights** from the lefthand menu.
4. Check the checkbox for any of the following options:
   - **Normalize Copyright Entries from the KnowledgeBase**. Standardize the format of copyright entries, applying the transformations described below. This option must be enabled to apply any of these transformations.

     Warning: By enabling any of these options, you are modifying the fundamental characteristics of copyrights obtained from the KnowledgeBase copyright inventory, and these may now deviate from how they appeared in the identifying source code.

   - **Merge Copyrights**. Enabling this option will merge identical copyrights with different date ranges. If a component has multiple copyright entries for different years, they will be combined into one entry displaying the range of years.

     For example, the following copyrights would be combined into one merged copyright:

     ```
     Copyright 2021 Component Corporation
     Copyright 2022 Component Corporation
     Copyright 2023 Component Corporation
     Copyright 2024 Component Corporation
     ```

     The resulting merger displays `Copyright 2021-2024 Component Corporation`.

- **Remove Copyrights Without Dates**. Enabling this option will remove any copyrights that do not have a date.

  For example the following copyright would be removed from validated copyright lists:

  ```
  Copyright Component Corporation
  ```

- **Standard Copyright Tag**. Selecting one of options below will modify all copyrights to use the desired tag.
  - None, use existing string
  - Copyright ©
  - Copyright (C)
  - Copyright (c)
  - ©
  - (C)
  - (c)

- **Truncate Long Copyright Entries**. Truncates long copyrights to the first 200 characters. This transformation does not require **Normalize Copyright Entries from the KnowledgeBase** to be enabled to function. However, if that setting is enabled, **Truncate Long Copyright Entries** will automatically be enabled and cannot be disabled.

Transformations when normalizing copyright entries

When enabling **Normalize Copyright Entries from the KnowledgeBase**, the following changes will be applied to copyright entries from the KnowledgeBase. Text normalization is case insensitive.

- Escaped character markers like \n and \u003c are replaced with real characters.
- HTML copyright markers, © and © are replaced with ©.
- Comment markers /*, *, */, //, and # are removed.
- All sequences of white space, like \n, \r, \t and spaces are replaced with a single space.
- Copyright entries are truncated to 200 characters.
- All text after "all rights reserved" is removed.
- Copyright entries which do not include the word copyright, start with (C) or have (c) or © without a valid date (19dd or 20dd) are rejected.
- Copyright entries with no text (numbers are not text) after the copyright marker are rejected.
- Copyright entries shorter than 15 characters with no valid date are rejected.
- When copyrights are normalized duplicate entries are removed in reports.

Note: Rejected entries are truncated to 200 characters, but otherwise remain unedited.

Viewing project and project version audit information

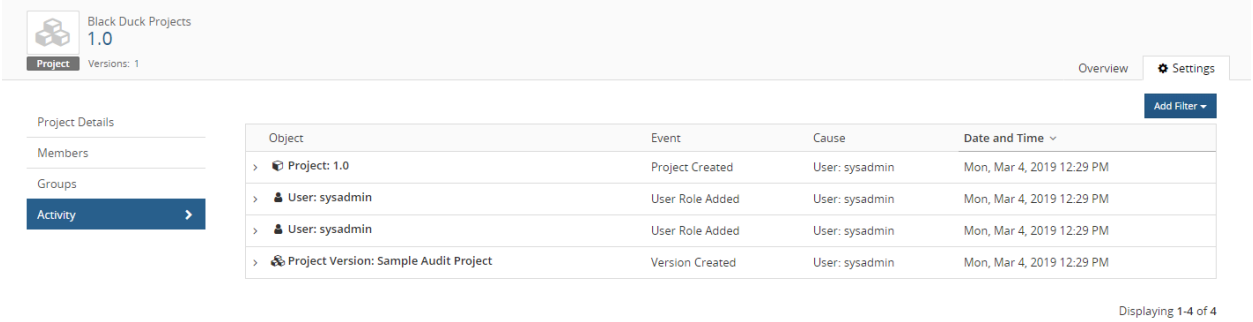
Black Duck tracks and displays all updates and changes that affect a project and/or project version. Use this information to understand who made changes or the events that caused changes to a project or project version. With this audit trail, you can determine, for example:

- 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 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:



Note the following:

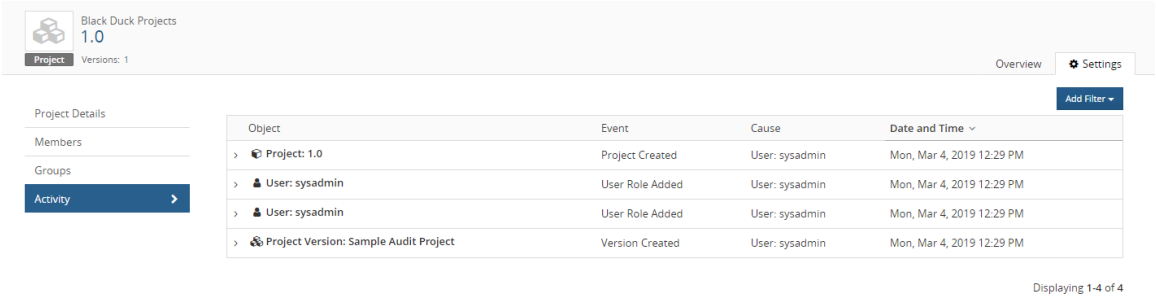
- Information is shown for the past 24 hours with the most recent changes appearing at the top of the table. Use the date filter to view information for different periods of time.
- While the deletion of a project version appears at the project level, deletion of a project will not appear here.

To view audit information:

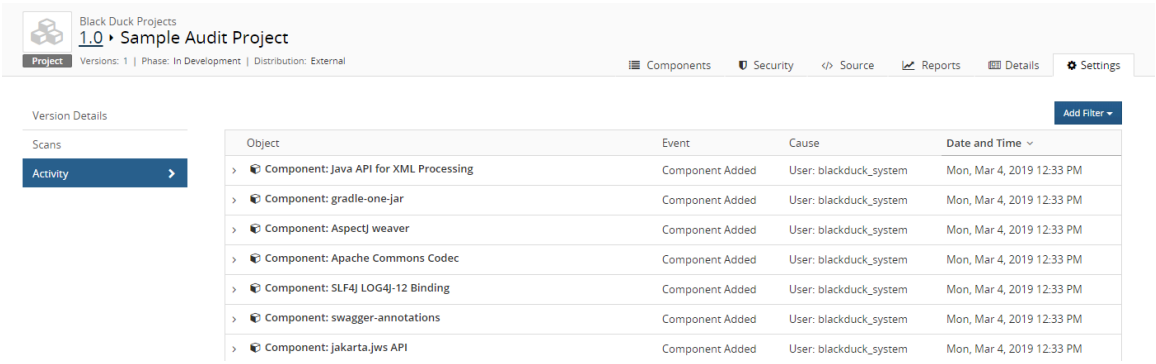
Audit information appears on the **Settings** tab of the project or project version.

1. Log in to Black Duck SCA.
2. Select the project name using the **Watching** or **My Projects** dashboard. The *Project Name* page appears.
3. Do one of the following:

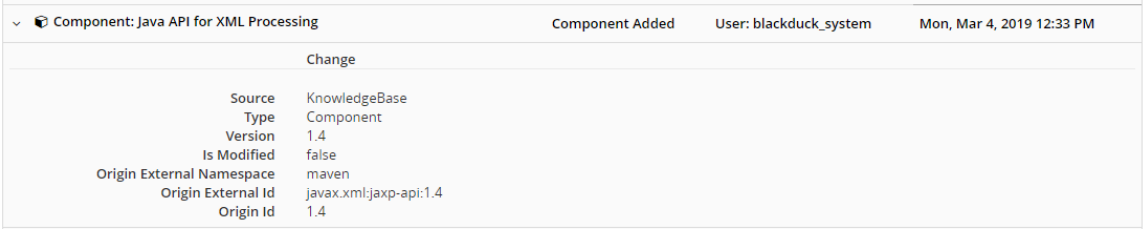
- To view *project* level audit information, select the **Settings** tab and then select **Activity**.



- To view *project version* level audit information, select the version, select the **Settings** tab, and then select **Activity**.



4. From this page:
   - Click > located to the left of the object name to view details of this event.



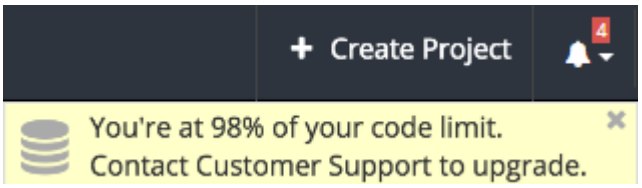
   - Filter the table to view specific information, such as activity during a specific date range or a specific type of event.

SBOM fields

If SBOM fields are enabled, adding or modifying these fields on the project version or component level will be logged in the Activity tab as well. Click here for more information on how to edit SBOM fields.

Managing your code size limits

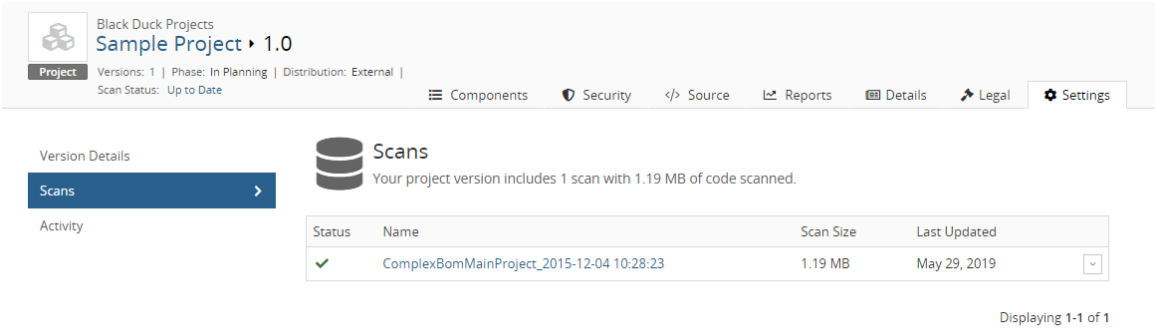
Black Duck will notify you when you are approaching your code size limit (as declared in your license). A notification, such as the following, appears in the UI when you are at 80% or higher of your code size limit:



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 Black Duck 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.



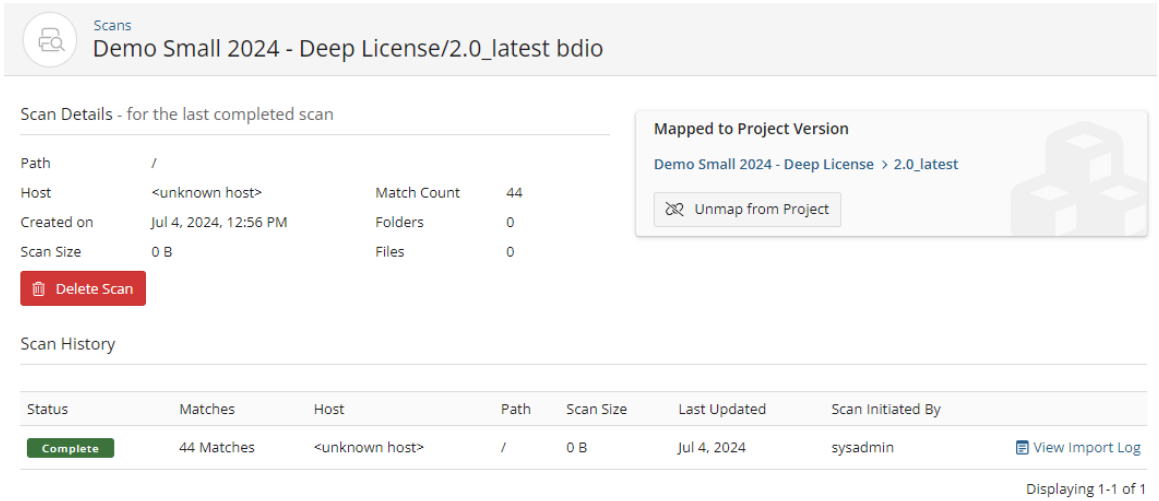
  The scan size appears above the list of scans.

- 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 **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.

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.
- Estimated Security Risks that have been added or removed from components without a version.
- 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 .
2. To manage the notifications, select **See All Notifications** located at the bottom of the list.

Filtering notifications

By default, the Notifications page is filtered. To further refine your search, select from the following options:

- **Created**: See all notifications created in a specified timeframe, such as Today, Past three days, Past week, and Past month.
- **Notification State**: Filter notifications based on their current state, such as New, Seen, Visited, or Hidden. Please note that the Seen and Visited filters currently share the same behavior.
- **Notification Type**: Display notifications based on their type, such as Component with Unknown Version, Rule Violation, Policy Override, Vulnerability, Project, Project Version, License Limit, or Rule Violation Cleared.

Note: If no options are selected for the Notification State or Notification Type filters, they default to including all options.

Viewing more information

To view more information on security vulnerabilities and BOM component adjustments:

1. Open the Notifications page 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.

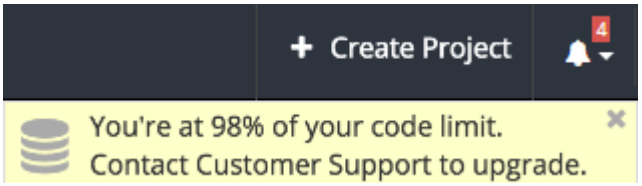
To view more information on policy violations and overrides:

1. Open the Notifications page 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.
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 Notifications page.

1. Open the notifications list by selecting .
2. Select **See All Notifications** located at the bottom of the list to display the Notification page.
3. Click  located at the end of the notification's row. Conversely, click  to unhide a hidden notification.

Internal SSL certificate expiration alert

The internal SSL certificate expiration alert notifies users when their SSL certificates are approaching their expiration date. This proactive alert system notifies users 30 days in advance, ensuring timely renewals and uninterrupted secure connections.

The internal SSL certificate expiration alert is automatically enabled within the application. Users will receive notifications through the application interface or via configured alert channels as the expiration date approaches.

About the Tools page

The Tools page provides download links and links to Black Duck integrations on GitHub, the Black Duck Community and Black Duck Academy 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 Black Duck Detect Desktop, Black Duck Detect CLI, and Legacy Downloads (the Signature Scanner).

- **Black Duck Detect (Desktop)**. Select the link to download the Mac OS X, LInux, or Windows version of Black Duck Detect Desktop from Google Cloud Storage. This tool scans your file system and generates a Bill of Materials (BOM).

  Black Duck Detect Desktop client systems must meet the following requirements:

  - Mac OS X. Version 12 or later. A minimum of 8 GB of RAM.
  - Windows. Windows 11. A minimum of 8 GB of RAM.
- **Black Duck Detect(CLI)**. Select the link to go to the Integrations Documentation page. From here, select **Black Duck Detect** to view download instructions and documentation for Black Duck 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:

  - macOS x64. Version 12 or later. A minimum of 8 GB of RAM.
  - macOS arm64. A minimum of 8 GB of RAM.
  - Windows. Windows 11. A minimum of 8 GB of RAM.
  - Linux. A minimum of 8 GB of RAM on the supported operating systems.

  Note: The Signature Scanner is included with Black Duck Detect. We recommend you use Black Duck 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:

- **Black Duck Community**. Clicking this link on the Tools page displays an online resource for customer support, solutions, and information.
- **Black Duck Academy**. Clicking this link on the Tools page opens a web page that provides more information on education courses for Black Duck products.

Hosted vs Air-gapped KnowledgeBase

Notice: This is a non-exhaustive list and any feature not explicitly listed may not be available for air-gapped KnowledgeBase installations.

Core features & scanning techniques

Feature	Included in Hosted	Included in Air-gapped	Notes
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.
Signature scanning	✓	✓	Signature scanning scans an arbitrary file system directory or archive and matches to known components in the Black Duck KnowledgeBase. 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.
Package manager scanning	✓	✓	Uses the project's package manager to derive the hierarchy of dependencies known to that package manager.
Snippet scanning	✓	✓	Snippet matching is beneficial to managing legal risk and detecting possible license infringement. A snippet match occurs when a portion of code in your

Feature	Included in Hosted	Included in Air-gapped	Notes
			file matches code in one or more KnowledgeBase files.
Binary scanning	✓	✓	Binary scanning identifies the open-source security, compliance, and quality risks in the software libraries, executables, and vendor-supplied binaries in use within your codebase.
Black Duck Secure Container scans	✓	✓	When scanning a container using this feature, Black Duck creates a new type of project that manages a new container scan. The container project displays the aggregated BOM and risk, but it also provides a way to view risk layer by layer, specifically adding support for components that are added or removed on a layer.
Vulnerabilities	✓	✓	Vulnerability information, triaging and management features are available in Black Duck.
Real time updates	✓	✗	Update of the hosted KB is done in real-time so new vulnerabilities or components appears continuously. With an air-gapped KB, new vulnerabilities and components only appear when the KB update is performed.

License text and copyright text

Feature	Included in Hosted	Included in Air-gapped	Notes
Template license text	✓	✓	Each license comes with a template version of the license.
Local license text scan	✓	✓	The license search feature to search for license text in your projects. This helps with the reconstruction of the specific custom license text which might be contained in the project source files. See License Detection for more information.
Local copyright text scan	✓	✓	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. See Copyright Detection for more information.
Actual license text	✓	✗	Black Duck KnowledgeBase provides the actual license text contained in each version of every component. No big data is shipped with air-gapped KnowledgeBase.

Feature	Included in Hosted	Included in Air-gapped	Notes
Full copyrights	✓	✗	Black Duck manages copyright statements by the origin name/ID for a component version. 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. No big data is shipped with air-gapped KnowledgeBase.
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. Deep license data is not enabled by default. No big data is shipped with air-gapped KnowledgeBase.

Cloud scanning services & specific match features

Feature	Included in Hosted	Included in Air-gapped	Notes
SCA Scan Service	✓	Coming in 2026	SCA Scan Service (SCASS) is a scalable solution for performing software composition analysis scans outside of the traditional Black Duck SCA environment. SCASS supports Package Manager and Signature Scans, making it a versatile choice for various scanning needs.
Match as a Service (MaaS)	✓	✓	Match as a Service (MaaS) is a cloud service that identifies OSS components, using data gathered by Component Scanning. MaaS moves the matching logic from the Black Duck server to the KB server for efficiency.
Scan correlation	✓	✗	Correlated Scanning is a scanning method which allows different matching technologies to scan the same application target and correlate results together to perform more accurate component and component version matching.
AI Model Scanning	✓	✗	AI Model Scanning enables organizations to maintain visibility into the AI models they use, monitor associated risks, and ensure compliance with relevant policies.

External integrations

Feature	Included in Hosted	Included in Air-gapped	Notes
SCM Onboarding	✓	✓	SCM repository auto-scanning allows Black Duck to check daily for any changes such as commits, pushes, or merges in the repository branch mapped

Feature	Included in Hosted	Included in Air-gapped	Notes
			to your SCM projects and perform scans if changes were made.
Artifactory Integration	✓	✗	BD SCA can be integrated with Artifactory to scan the assets for OSS. Artifactory Integration automatically blocks downloads from scanned Artifactory Repositories that have a Black Duck Policy Violation.

Other features

Feature	Included in Hosted	Included in Air-gapped	Notes
Stateless signature scanning	✓	✓	Stateless Scan is a scan mode that does not create or use any permanent storage within Black Duck, thus there is no bill of material (BOM) stored. It is used to quickly find policy violations within the designated scan target.
Auto BOM re-computation (components)	✓	✗	When the KnowledgeBase (KB) is updated, Black Duck automatically refreshes vulnerability and component information. For air-gapped KB deployments, this refresh is not automatic—projects must be re-scanned or scans manually unmapped and remapped after KB updates to trigger a data refresh.

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 a 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 SCA.
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. To obtain the zip file, please contact Black Duck Support at https://community.blackduck.com/s/contactsupport.

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 **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*>	Lists all Protex project identifiers for all projects to which you have access, one per line, on the console. <*SearchQuery*> is optional. To export multiple Protex projects, use the output from this parameter to write a script which iterates over multiple project identifiers.
--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.

Parameter	Description
-u. **--user** <*user*>	Specifies the username to log in to the Protex server.
-w, **--password**	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. 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.
-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
   ```

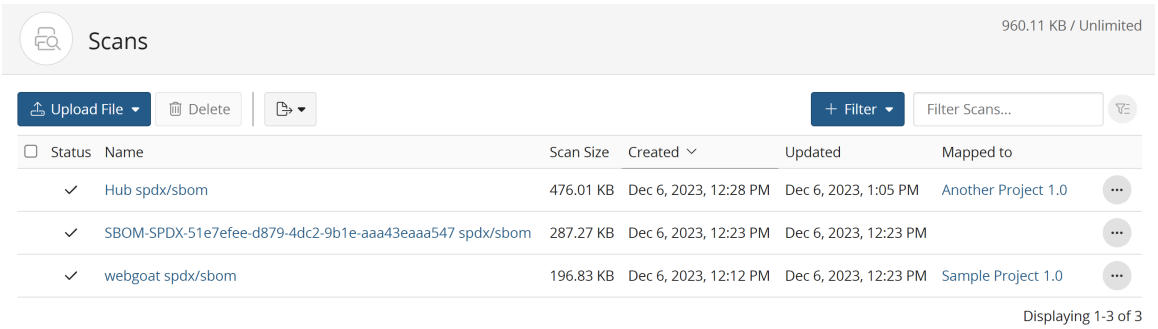
Importing the Protex BOM file

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 SCA.
2. Click 

.



3. In the Scans page, click **Upload Files**.
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.

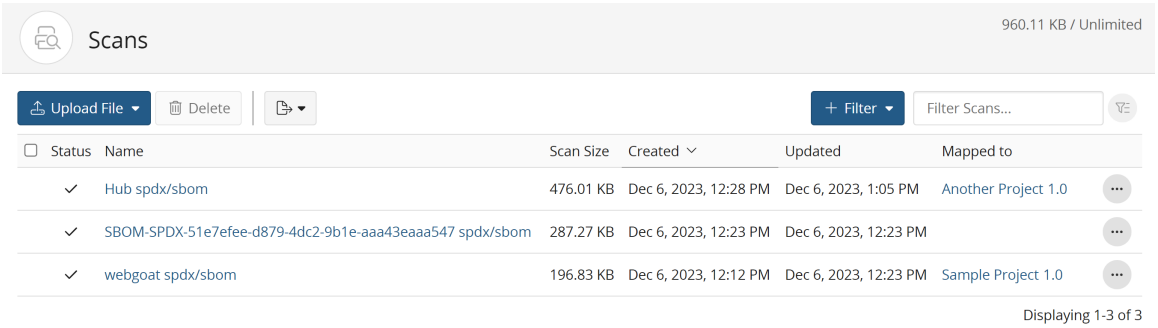
Mapping or unmapping a Protex BOM

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 SCA.
2. Click Scans.

   The Scans page appears.



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 SCA.
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.

Black Duck C/CPP Tool

C and C++ projects don't have a standard package manager or method for managing dependencies. It is therefore more difficult to create an accurate BOM for these projects. This leaves Software Composition Analysis tools fewer options than with other languages. The primary options which are available in this context are: file system signatures. Black Duck has a variety of old and new signatures which can be used to build a BOM. In order to effectively use signatures, the tool first needs to know which files to take signatures from. In the past SCA tools have pointed a scanner at a build directory, getting signatures from a subset of files within the directory sub-tree. The problem with this approach is that there are many environmental variables, parameters and switches provided to the build tools, which make reference to files outside of the build directory to include as part of the build. Further, there are, commonly, files within the build directory, which are not part of the build and can lead to false positives within the BOM.

The new Black Duck C/CPP tool avoids the pitfalls described above by using a feature of Coverity called Build Capture. Coverity Build Capture, wraps your build, observing all invocations of compilers and linkers and storing the paths of all compiled source code, included header files and linked object files. These files are then matched using a variety of methods described in the section of this document called "The BOM".

This overview is an excerpt from the documentation that can be found in the Black Duck C/CPP Tool Documentation Portal page along with other details on how to use the Blackduck-C-CPP tool.

Black Duck integrations & APIs

We created multiple options so you can choose the type of integration that best supports the task and your way of doing CI/CD development.

First, choose your approach:

- **Command line (CLI)**: Script the command line client to run in your environment, either directly or integrated into a build script of CI workflow.
- **Continuous integration (CI)**: Add a plug-in to existing pipelines that run in your SCM repo or Jenkins. Plug-ins are available in the marketplace and in many cases can be added to your config file by following step-by-step instructions in the documentation.
- **API**: Use REST APIs to call services built into the product (e.g. retrieve issue data and import it into your internal dashboards).

Secondly, select the option tailored to the security products and CI/CD platforms you use.

Command line clients

For scripting in the command line, users have a choice between the original **Detect client** (built by the Black Duck team for Black Duck SCA) and the **Bridge CLI** client.

Detect client

Black Duck Detect is a scan client that analyzes code in your projects and associated folders to perform compositional analysis and find vulnerabilities.

It can be configured to send scan results to Black Duck, which generates risk analysis when identifying open-source components, licenses, and security vulnerabilities.

How Detect works:

1. Uses the project's package manager to derive the hierarchy of dependencies.

2. Runs the Black Duck signature scanner on the project. This might identify additional dependencies not known to the package manager.
3. Uploads both sets of results (dependency details) to Black Duck, which creates the Bill Of Materials (BOM) for the project/version.
4. You can view the output and analysis results in Black Duck SCA.

Detect consolidates the functionality of Black Duck, package managers, and continuous integration plugin tools to perform the following tasks:

- Discover open-source components in your code.
- Map components to known security vulnerabilities.
- Identify license compliance and component quality risks.
- Set and enforce open-source use and security policies.
- Integrate open-source management into your DevOps environment.
- Monitor and alert users when new security threats are reported.
- Calculate security vulnerability risk in your code.
- Produce reports of the open-source analysis findings.
- Provide malware information if identified.

Note: Some scan types require specific feature licenses to execute. Contact your Black Duck representative for further information.

Bridge CLI client

Bridge is useful when you want a unified CLI for all the security tools offered by Black Duck Software: **Polaris**, **Coverity Connect**, **Black Duck SCA**, **Software Risk Manager**.

Bridge does all the following:

- SAST and SCA scanning
- Scan in synchronous or asynchronous (non-blocking) mode
- Scan whenever new code is merged to a branch
- Scan whenever a pull request is created/updated
- Decorate PRs with comments
- Create Fix PRs (Black Duck SCA only)
- Generates a SARIF file
- Post results to SCM (GitHub advanced security)
- Post results to any supported server (see the list of products above).

For more information **see, Bridge documentation.**

Note: Bridge can do any of the above in an air gapped environment. See the individual tool pages for more information.

Plug-in integrations

Plug-ins are the easiest way to integrate testing into your CI/CD pipeline. Choose from CI/CD plug-in integrations derived from either of the CLI clients above: **Detect** or the **Bridge CLI**.

Detect-based CLI plug-ins

Jenkins

The Detect Extension for Jenkins enables you to install and run Black Duck Detect in your Jenkins instance.

Capabilities include:

- Performing compositional analysis and functioning as a Black Duck intelligent scan client.
- Sending scan results to your Black Duck SCA server, which generates risk analysis when identifying open source components, licenses, and security vulnerabilities.
- Running Detect as either of the following:
  - A post-build action in a Jenkins Freestyle job.
  - Pipeline step using a Pipeline script in a Pipeline job.

Azure

The Detect Extension for Azure DevOps is designed to integrate Black Duck Detect seamlessly into Azure DevOps build and release pipelines.

It includes the ability to:

- Run a component scan in an Azure DevOps job.
- Create projects and releases in Black Duck SCA through the Azure DevOps job.
- Make results available on the Black Duck SCA server.

GitHub

The Detect GitHub Action plug-in integrates Black Duck Detect into GitHub action workflows.

Capabilities include all of the following:

- Run a component scan in a GitHub workflow.
- Upload results to a project in Black Duck SCA.
- Configure Detect in either of two modes:
  - Rapid scan mode to get detailed Black Duck policy reports (default behavior)
  - Intelligent scan mode to upload your data into Black Duck for more detailed analysis.

Note: As of October 2024, we recommend using the newer, Bridge-based GitHub Action for creating new pipelines, rather than Detect GitHub Action.

Bridge-based CI Plug-ins

Our latest plug-ins are built with the Bridge CLI Client under the hood, so you get the same benefits without writing the code.

Capabilities include:

- SAST and SCA scanning
- Scan in synchronous or asynchronous (non-blocking) mode
- Scan whenever new code is merged to a branch
- Scan whenever a pull request is created/updated
- Decorate PRs with comments
- Create Fix PRs (Black Duck SCA only)

- Generates a SARIF file
- Post results to SCM (GitHub advanced security)
- Post results to any supported server (see the list of products above).
- Make issues available in your instance of Black Duck SCA, Coverity, Polaris, or SRM.
- Fail the build in your CI system when a high-severity issue is found.

Bridge plug-ins for Black Duck are available on the following platforms:

- Azure
- GitHub
- GitLab
- Jenkins

API reference

Black Duck APIs offer a convenient way to retrieve ad-hoc information from Black Duck or to perform automated review or workflows. However, customers looking to build out customized views or dashboards using bulk data from Black Duck would be best served by using the Black Duck Reporting DB schema.

Tip: Although APIs offer a way to do most functions that Black Duck UI can do, you might choose the Black Duck Bridge command line interface or an SCM integration for building tests or automation into your pipeline.

After tests run, APIs can help with the following:

- Generate and download a notices file or SBOM
- Collate component copyright information
- Automatically assign issues to developers
- Add comments to Pull Requests with Black Duck data

To explore what Black Duck SCA APIs have to offer:

- Access the REST API documentation found in Black Duck SCA by opening the Help menu from the top navigation bar and selecting **REST API Developers Guide**.
- Visit the REST API documentation directly at https://<Black Duck Server URL>/api-doc/public.html.

Black Duck REST API Python bindings

This Python library provides a streamlined interface to interact with Black Duck APIs, enabling customers to efficiently automate and customize their workflows. This library is distributed as an open-source extension of Black Duck and is licensed under the Apache 2.0 license.

For more details or to contribute, visit the Black Duck API Python bindings GitHub repository.

Downloading the Black Duck API Specification

Black Duck provides the capability to download the full API specification using either a Postman collection or an OpenAPI Specification (OAS). These options allow customers to directly import the documentation into tools like Postman, simplifying the process of working with the APIs.

- **Postman Collection**: You can generate a Postman collection from the API documentation by downloading the `postman-collection-public.json` file from `/api-doc/postman-`

`collection-public.json`. This file can be imported into Postman to interact with the Black Duck APIs.

- **OpenAPI Specification (OAS)**: Similarly, the OpenAPI Specification can be generated via the `/api-doc/openapi3-public.json` endpoint. This allows you to explore and document the Black Duck API with tools that support OAS.

Integrating Black Duck with Business Intelligence tools

Black Duck offers a Reporting Database (Reporting DB) interface that enables seamless integration with Business Intelligence (BI) tools, such as PowerBI. This interface provides customers with the flexibility to generate custom reports, visualize data, and build comprehensive dashboards that reflect key performance indicators (KPI) and risk management views.

To accelerate the integration process, Black Duck provides sample reports specifically designed for use with PowerBI. These templates serve as an efficient starting point for customers, allowing them to quickly build customizable dashboards using data from Black Duck's Reporting DB.

For more details, visit Black Duck Dashboards using Microsoft Power BI.

Legal Information

License Agreement

Use of this product is governed by the terms of the Black Duck License and Subscription Agreement ("License Agreement").

Privacy Policy

Black Duck respects your right to privacy. Any use of your personal information, beyond that specified in the License Agreement, will be governed by Black Duck's Privacy Policy.

Included Third Party Software

Black Duck Software, Inc. uses third-party software in the development and operation of Black Duck™. Use of this software is governed by the applicable license agreements from each of the software vendors.

Customer support

If you have any problems with the software or the documentation, please contact Black Duck Customer Support:

- Online: https://community.blackduck.com/s/contactsupport
- To open a support case, please log in to the Black Duck Community site at https://community.blackduck.com/s/contactsupport.
- Another convenient resource available at all times is the online Community portal.

Black Duck Community

The Black Duck 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 Black Duck 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 Black Duck 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 Black Duck 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 Black Duck experts and our Knowledgebase.
- Share – Collaborate and connect with Black Duck 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@blackduck.com.