revenera

Code Insight 2024 R1

User Guide

Legal Information

Book Name:	Code Insight 2024 R1 User Guide
Part Number:	RCI-2024R1-UG00
Product Release Date:	February 2024

Copyright Notice

Code Insight incorporates software developed by others and redistributed according to license agreements. Copyright notices and licenses for these external libraries are provided in a supplementary document that accompanies this one.

Intellectual Property

For a list of trademarks and patents that are owned by Flexera Software, see https://www.revenera.com/legal/intellectual-property.html. All other brand and product names mentioned in Flexera Software products, product documentation, and marketing materials are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The Software is commercial computer software. If the user or licensee of the Software is an agency, department, or other entity of the United States Government, the use, duplication, reproduction, release, modification, disclosure, or transfer of the Software, or any related documentation of any kind, including technical data and manuals, is restricted by a license agreement or by the terms of this Agreement in accordance with Federal Acquisition Regulation 12.212 for civilian purposes and Defense Federal Acquisition Regulation Supplement 227.7202 for military purposes. The Software was developed fully at private expense. All other use is prohibited.

Contents

1

Code Insight 2024 R1 User Guide

Code Insight empowers organizations to take control of and manage their use of open source software (OSS) and third-party components. It helps development, legal, and security teams use automation to create a formal OSS strategy that balances business benefits and risk management.

The *Code Insight User Guide* is intended for anyone who uses Code Insight for scanning, analyzing, and reviewing project codebases.

This guide describes how to use Code Insight to realize these benefits. It includes the following sections.

Table 1-1 ▪ Code Insight User Guide Navigation Table

Topic	Content
Using Code Insight	“How to” information for using Code Insight functionality.
Performing Advanced Searches	Overview and procedures for using advanced searches to find specific inventory.
Exporting and Importing Project Data	Explanation and procedures for exporting and importing project data.
Automated Analysis	Information about Code Insight automated analysis tools and features.
Performing Remote Scans	Information about the Code Insight agent scan, performed remotely.
Configuring Source Code Management	Procedures for using Source Code Management (SCM) systems with Code Insight.
Pages and Panels	Reference to field descriptions on the s, panes, tabs, and dialogs used in the Code Insight user interface.
Code Insight User Roles and Permissions	A reference to the various user roles and permissions available in Code Insight to control access to Code Insight functionality.

Product Support Resources

The following resources are available to assist you with using this product:

- Revenera Product Documentation
- Revenera Community
- Revenera Learning Center
- Revenera Support

Revenera Product Documentation

You can find documentation for all Revenera products on the Revenera Product Documentation site:

https://docs.revenera.com

Revenera Community

On the Revenera Community site, you can quickly find answers to your questions by searching content from other customers, product experts, and thought leaders. You can also post questions on discussion forums for experts to answer. For each of Revenera's product solutions, you can access forums, blog posts, and knowledge base articles.

https://community.revenera.com

Revenera Learning Center

The Revenera Learning Center offers free, self-guided, online videos to help you quickly get the most out of your Revenera products. You can find a complete list of these training videos in the Learning Center.

https://learning.revenera.com

Revenera Support

Customers who have purchased a maintenance contract for their product(s) can submit a support case or check the status of an existing case by making selections from the **Get Support** menu in the Revenera Community (https://community.revenera.com).

Contact Us

Revenera is headquartered in Itasca, Illinois, and has offices worldwide. To contact us or to learn more about our products, visit our website at:

http://www.revenera.com

You can also follow us on social media:

- Twitter
- Facebook
- LinkedIn
- YouTube
- Instagram

2

Using Code Insight

The following topics provide basic information about Code Insight that will enable you to start using the product effectively:

- Opening Code Insight
- About Roles and Permissions in Code Insight
- About Code Insight Projects
- About Code Insight Scans
- Creating a Project
- Applying a Scan Profile to the Project
- Uploading a Project Codebase (for Server Scans)
- Scanning the Codebase (Server Scans)
- Overview of Scan Results
- Auditing Scan Results in the Analysis Workbench
- Reviewing Published Inventory for a Project
- Using “Lookup Component” to Search for Components to Associate with Inventory
- Specifying a User-Preferred License Mapping
- Working with Security Vulnerabilities
- Managing Security Vulnerability Alerts
- Creating and Editing Custom Components
- Creating Custom Component Versions
- Creating and Editing Custom Licenses
- Managing Custom Detection Rules
- Finalizing the Notices Text for the Notices Report

- Generating Reports for a Project
- Managing Scan Queues Across All Scan Servers
- Monitoring the Code Insight Jobs Queue
- Viewing Inventory Across All Projects
- Exploring Components and Licenses in the Data Library
- Accessing Projects in Code Insight
- Managing Projects
- Managing Policies to Automatically Review Inventory
- Tracking the Progress of an Electronic Update
- Managing Authorization Tokens
- Downloading Code Insight Log Files

Opening Code Insight

Code Insight runs in your web browser. This section explains how to start up Code Insight, opening to the Code Insight dashboard.

Note ▪ *If this is the first time you have opened Code Insight or if you have recently upgraded Code Insight or shut down your Tomcat server, you must start up the Tomcat server with the startup command before opening Code Insight. For more information, see “Starting and Stopping Tomcat” in the “Installing Code Insight” chapter in the “Code Insight Installation and Configuration Guide”.*

Task ***To open Code Insight, do the following:***

1. Launch a web browser and navigate to: `http://<your_server_host_name>:8888/codeinsight`.

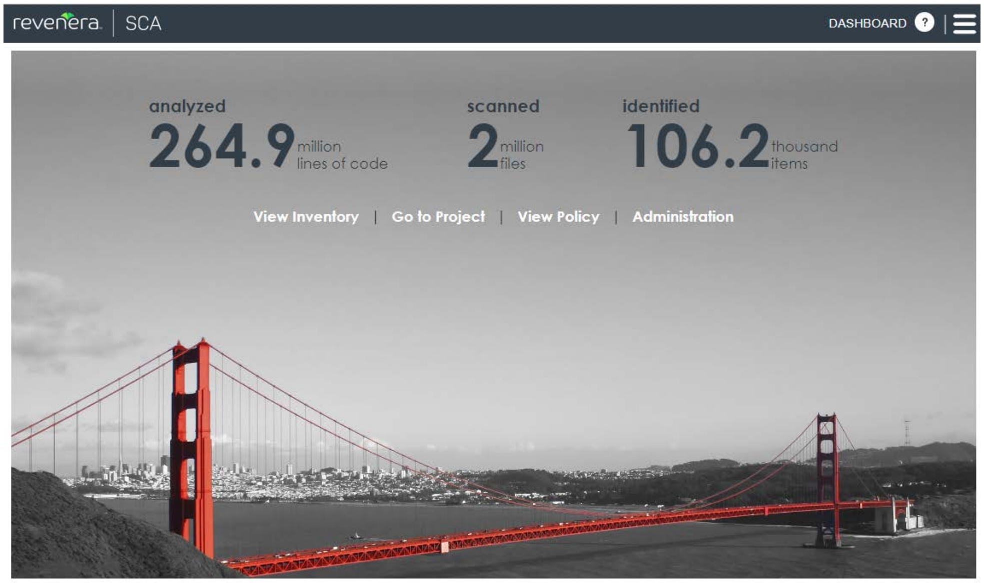
   If you are unsure about your server host name, contact your site’s system administrator or the Code Insight System Administrator for guidance.

2. Enter your Code Insight credentials in the **Username** and **Password** fields.

   Note ▪ *The default login name is* **`admin`**; *the default password is* **`Password123`**. *Your installation might require a different login name and password. If you are unsure about what credentials to enter, contact the Code Insight System Administrator for guidance.*

3. Click **Login**.

   The Code Insight dashboard is displayed. This dashboard shows statistics from the most recent codebase scans performed across all Code Insight projects and provides entry points to other parts of the Code Insight Web UI.


revenera | SCA
DASHBOARD
analyzed
264.9 million lines of code
scanned
2 million files
identified
106.2 thousand items
View Inventory | Go to Project | View Policy | Administration

4. From the Code Insight dashboard, navigate anywhere in Code Insight that you have permission to access, as described in the following table. (Options are not displayed for those areas to which you do not have access permission.) For more information about the dashboard, see Code Insight Dashboard.

Click this option...	...to go here
view inventory	The **Inventory** view, which provides a compilation of inventory across all current Code Insight projects. The list can be filtered at a basic level to show inventory for all projects, only your projects, or for a specific project. Filtering can be further refined by vulnerability severity, review status, and many other criteria. For more information, see Viewing Inventory Across All Projects.
go to project	The **Projects** view, which provides access to all current projects in Code Insight. For more information, see Accessing Projects in Code Insight.
view policy	The **Policy** page, where you have access to all policies that automate the review process of project inventory when it is published. For more information, see Managing Policies to Automatically Review Inventory. Access to this requires Manage Policy permissions.
administration	The **Administration** page, where you have access to Code Insight administrative functionality. See the *Code Insight Installation and Configuration Guide* for a description of administrative tasks. Access to this page requires Code Insight System Administrator permissions.

Viewing Online Help and Online Guides

Code Insight provides online help topics and online versions of its guides so you can find answers to your questions about the product while you are using it.

Task ***To access online help and guides, do the following:***

- To access the online help for the current Code Insight page, including the Code Insight dashboard, click the Help icon in the upper right corner of the Code Insight Web UI.

  ?

- To access the Code Insight online user guides, click the icon in the upper right corner of the Code Insight web page.

  ☰

  From the Code Insight main menu that opens, select **HELP**. The **Help** page is displayed, providing a list of links to the available online documentation.

Changing Your Password

If you want to change your current Code Insight password, use this procedure.

Task ***To change your Code Insight password, do the following:***

1. Click the icon in the upper right corner of the Code Insight web page.

   ☰

   The Code Insight main menu is displayed.

2. Select **Preferences** from the menu to open the **Preferences** page.
3. Enter your new password in **New Password**, and then reenter it in **New Password Confirm**.
4. Click **Update Password**.

About Roles and Permissions in Code Insight

Code Insight offers a set of user roles and permissions that enables your site to control access to Code Insight features and functionality.

The initial Code Insight System Administrator, identified during Code Insight installation, can assign users to system-level roles for managing Code Insight policies and creating Code Insight projects. The System Administrator can also create other System Administrators and define default Project Administrators, Analysts, and Reviewers that are automatically assigned to projects when they are created.

At the project level, a project creator automatically becomes the Project Contact as well as a Project Administrator (among other roles) for the project. A Project Administrator can assign users to project roles that enable these users to analyze and review project scan results. The administrator can also remove a user from any project role as needed, whether the user was manually assigned the role or had inherited it.

Note ▪ *When a project is migrated from a previous Code Insight version (2020 R3 or earlier), by default the Project Owner becomes the Project Contact and is assigned to the Project Administrator and Analyst roles.*

For more about the management of Code Insight roles and permissions, refer to the following:

- The Assigning and Removing Project Users describes the assignment of users to project roles.
- The Code Insight User Roles and Permissions section serves as a reference to the various Code Insight roles available and the permissions granted to each role. As you prepare use the Code Insight, refer to this section to determine the roles required to perform certain Code Insight functionality and the permissions the roles enable.

About Code Insight Projects

A project in Code Insight represents a version or release of an application or application module on whose codebase you run a Code Insight scan and analysis. The scan discovers of evidence of open-source software (OSS) and third-party code in the application files and generates an inventory of this software that you can then review and remediate within the project. The project also provides facilities that enable you to perform your own thorough audit of the scanned files to validate the generated inventory and create or modify inventory items if needed.

Typically, you would create a project for each one of your products or services, but you might also create projects to review vendor code for security or licensing issues, to screen an open-source component you are considering using, or to prepare for an open-source contribution.

For added flexibility, you can group projects in project folders that represent business units, teams, product lines, tools, or any other groupings that help you locate projects more easily. The folders can be nested to the desired level.

All projects have the same basic key elements but use various configurations, as described in these next sections:

- Key Project Elements
- Common Project Configurations
- Legacy Projects

Important ▪ *You can create projects only if the Code Insight System Administrator has granted you permission to do so, as described in the “Code Insight Installation and Configuration Guide”.*

Key Project Elements

The following key elements of a project are important to keep in mind when creating, configuring, and organizing projects:

- **Materials to scan or analyze**—Each project has an uploaded codebase or a configured remote scan location (such as on a build server, artifact repository, or version control system).

- **Scan profile**—Each project has an associated scan profile with a set of scan settings that are applied when the project is scanned. (The profile can be one of the default scan profiles or a custom scan profile).
- **Policy profile**—Each project has an associated policy profile with a set of intellectual property or security policies that are applied when project inventory is published to the project (such as during the first scan or when manually published by a project user).
- **Project visibility**—Each project has an associated visibility configuration that specifies which logged-in users have the ability to view or change the project.

Common Project Configurations

These are some examples of common project configurations:

- **Project that manages server scan results**—A project is configured for server scan (that is, a Scan Server scan) on one or more application source codebases that are uploaded to the Scan Server or synchronized to the server through a Source Control Management application. The scan profile for a given server scan can perform a full analysis of the source—including searches for exact matches of entire OSS or third-party files and for partial source-code matches (fingerprints)—and process files inside archives.
- **Project that manages remote-scan results**—The project manages the results of remote scans performed on one or more remote server codebases—each codebase typically consisting of built artifacts residing on a build server (for example, a Jenkins or GitLab server or another supported build server, artifact repository, or version control system). A given remote scan is performed by a Code Insight scan-agent plugin, which sends the scan results to the project. (Currently, only OSS or third-party license evidence that the scan discovers in the codebase files is viewable in the project.)
- **Project for manage results of server and remote scans**—The project is configured to perform server scans on the application's source code (either uploaded or synchronized to the Scan Server), *and* it manages the results of remote scans performed on build server codebases.
- **Security-focused project**—The policy profile selected for the project triggers an automatic review of project inventory based on the presence of security vulnerabilities, on the CVSS scores and severity of these vulnerabilities, and on other criteria.
- **Project focused on intellectual-property protection**—The policy profile selected for the project triggers an automatic review of project inventory based on the presence of allowed and not-allowed components, version ranges, and licenses.

Legacy Projects

The following section describes the changes that were introduced for the Code Insight project in the 2020 R3 release:

- Projects Prior to Code Insight 2020 R3
- Projects in Code Insight 2020 R3 and Later
- Resource for Additional Information

Projects Prior to Code Insight 2020 R3

Prior to Code Insight 2020 R3, you had the option of creating one of two types of projects:

- **Standard**—This project type was reserved for server scans only (that is, scans performed by the Scan Server and that require that codebase files be uploaded or synchronized to the Scan Server in order to be scanned.) The standard project enabled users to audit the OSS and third-party evidence discovered in the scanned source files and then review, remediate, and finalize an inventory of OSS and third-party components used by the application.
- **Inventory Only**—This project type was used for remote scans. Remote scans are performed by scan-agent plugins that scan remote server codebases—typically containing an application's built artifacts—and then send the scan results to a project on the Code Insight server. Previously, the plugin was designed to send the resulting inventory of OSS and third-party software to a project created in Code Insight as "Inventory Only". Users could then review, remediate, and finalize the inventory. However, because no file information was sent to the project, users could not audit the scanned codebase files.

Projects in Code Insight 2020 R3 and Later

Starting in Code Insight 2020 R3, only one type of project is supported—a *unified* project used by both server and remote scans. The unified project can manage the scan results of one or more server scans, one or more remote scans, or a combination of both scan types for a given application. It also manages remote-file information sent by the scan agents. Thus, in a single project, you can audit an application's codebase files loaded on the Scan Server and its remote codebase files—as well as review, remediate, and finalize the complete inventory of OSS and third-party software for the application, as captured from all server and remote scans.

If you are upgrading from a pre-2020 R3 Code Insight release, existing projects are handled in the upgrade as such:

- Standard projects are automatically migrated to the new release as unified projects, enabling you to add the results of remote scans to the previous server scan results in these projects if you want.
- Inventory-only projects are not automatically migrated to the new release as unified projects and will continue to be supported in future releases for a limited time. These legacy projects support only 2020 R2 or earlier scan-agent plugins and allow imports only from other legacy projects. You might consider manually migrating legacy projects to unified projects *now* even if you intend to continue to use these projects for remote scans only (see Resource for Additional Information).

In the user documentation for Code Insight 2020 R3 and later, a unified project is simply called a *project*. Previous inventory-only projects will be referred to as *legacy projects*.

Resource for Additional Information

For complete information about the concept of a unified project, its impact on projects existing before the 2020 R3 release, and the migration of the previous projects to the unified project type, see the Code Insight 2020 R3 Changes to Projects article in the Revenera Customer Community.

Creating a Project

You must create the project before running a scan on a codebase—whether the codebase resides locally on a Scan Server or is a remote codebase to be scanned by a scan agent.

The following procedure focuses on creating a public project, which is the default project type. However, you can use this same procedure to create a private project, which has limited user access. For more information about creating a private project, see Creating a Private Project.

Any user in the system has read-only access to a public project. To what degree a user can interact with the project depends on whether the user has a project role and what the role is.

Important ▪ *You can create projects only if the Code Insight System Administrator has granted you permission to do so, as described in the "Configuring Code Insight" chapter in the "Code Insight Installation and Configuration Guide". The* ***Add New*** *button and* ***Create New Project*** *right-click menu option referenced in the following procedure are available only if you have this permission.*

Task ***To create a project, do the following:***

1. Ensure that you are in the **Projects** view in the Code Insight Web UI. See Opening the Projects View if you need instructions.
2. In the **Projects** pane on the left, use one of the following methods to create the project. (The display of projects in the **Projects** pane is either in a tree or plain-list format. See Select the Projects Display Format.)
    - (Project tree only) To create a project *under a specific folder*, use either method:
        - Right-click the specific folder or any project directly under that folder, and select **Create New Project | At This Level**.
        - Select the specific folder or any project directly under that folder; then click the **Add New** button (located at the top of the **Projects** pane), and select **Project** from the associated dropdown menu.

      Once the project is defined, it will be stored under the selected folder or the folder to which the selected project belongs.
    - To add a project to the plain list or to store a project *at the root level* of the project tree, use one of these methods:
        - Click the **Add New** button, and select **Project** from the associated dropdown menu.
        - (Plain list only) Right-click anywhere in the plain list, and select **Create New Project**.
        - (Project tree only) Right-click anywhere in the project tree, and select **Create New Project | At Root Level**.

      Once the project is defined, it will be added to plain list or stored at the root level of project tree.

   The **Add Project** dialog is displayed.
3. Complete the following fields on the **Add Project** dialog:
    - **Name**—Type a name for the new project.

- **Project Visibility**—Select **Public** to allow general access to the project. All users in the system can view a public project. To what degree a user can interact with a public project depends on the project role of the user. (To create a project with the **Private** option to limit its visibility, see Creating a Private Project.)

  Note *▪ The **Project Visibility** setting can be later changed through the **Edit Project** option on the **Manage Project** menu on the **Summary** tab. For more information, see Editing the Project Definition and General Settings.*

- **Scan Server**—Select the Scan Server for this project. Even if the project will contain the results of only remote scans, you still need to specify a Scan Server. In this way, it is available should you need to perform a deep analysis evidence in the codebase files.

4. Click **Save** to save the new project.

   As project creator, you automatically become the Project Contact and are assigned to the Project Administrator, Analyst, and Reviewer roles. These roles enable you to initially manage the project and its users, analyze the project codebase, and review project inventory.

5. (Optional) Assign project roles to users who will interact with the project. You can also remove yourself and others from any roles as needed. For more information, see Assigning and Removing Project Users.

The new project appears in the list of projects under the appropriate folder or at the root level of the list. At this point, the new project’s dashboard in the right pane does not contain information about the project. The project dashboard will be populated once a scan is run on the project codebase files.

About Code Insight Scans

A Code Insight scan processes codebase files to identify evidence of OSS and third-party code. The scan results are then processed by Code Insight, which creates inventory of the detected OSS and third-party components, detects licenses and security vulnerabilities, applies policies for automated review, and creates review and remediation tasks per project configuration.

The following describes the types of Code Insight scans and the analysis techniques used by scans:

- Scan Types
- Scan Analysis Techniques
- Scan Profiles

Scan Types

Code Insight performs two type of codebase scans—server and remote. A single Code Insight project for a given application can contain of the results of either or both types of scans.

Server Scans

Server scans are performed by the Code Insight Scan Server. A server scan requires that the codebase files you want to scan reside on the Scan Server. These files are placed on the server either by uploading them (manually or through Code Insight) or by synchronizing one or more repositories in a Source Control Management (SCM) system, such as Git or Perforce, to the server. The complete codebase for a server scan typically represents the source code for a given application. Once the scan is complete, you can review and remediate the inventory of discovered open-source and third-party software, as well as audit the scanned files to verify the inventory findings and customize inventory as needed. For more information about configuring a project for server scans and performing these scans, see the following:

- Uploading a Project Codebase (for Server Scans)
- Configuring Source Code Management
- Scanning the Codebase (Server Scans)

Remote Scans

Remote scans are performed by a Code Insight scan-agent plugin, which is installed and configured on a remote instance to perform a scan within the context of an Engineering build server on that instance (for example, an IDE, an artifact repository, a CI product, or an product to build, test, or install your application).

The plugin allows a scan of built artifacts and source files and then sends the results to Code Insight as inventory for review and remediation. A representation of the scanned remote files is also available in the project, enabling you associate these files with inventory, mark them as reviewed, or perform other file-related actions.

For more information about configuring Code Insight scan-agent plugins and remote scanning, see the following:

- Performing Remote Scans
- *Code Insight Plugins Guide* (available for download in the Revenera Customer Community)

Scan Analysis Techniques

The Code Insight scan performs a static analysis of files of any type (source or binary) to find open source and third-party components, licenses, and security vulnerabilities and, depending on the scan profile, to identify file-level and snippet-level evidence to aid users in determining the origin of every file in the codebase. The end goal of the Code Insight scan is to build an accurate Bill of Materials and to eliminate any security and intellectual property (IP) risk associated with the materials.

During a codebase scan, Code Insight processes every file in the materials, regardless of programming language or file type. It processes source materials, scripts, object code, binaries, images, icons, and documents to identify both open source and closed source components, licenses, and security vulnerabilities.

Code Insight identifies these elements using a combination of Automated Analysis and Advanced Analysis techniques:

- **Automated Analysis**—The Scan Server uses automated detection rules to identify components, versions, licenses, and security vulnerabilities. In applying these rules, the Scan Server automatically generates inventory items that make up the Bill of Materials. The rules are found in the Code Insight Data Library, which is updated on your Code Insight server through both an internal process and as part of the weekly Electronic Update. For more information, see Revenera Support.

- **Advanced Analysis**—The Scan Server uses Advanced Analysis techniques to detect copyrights, emails, URLs, search terms, and source code of actual OSS and third-party software. This level of analysis requires the Code Insight Compliance Library (CL), downloaded from the Product and License Center. The CL is a database containing the source code and other elements found in OSS and third-party software. Advanced Analysis attempts to match the source code in the CL database with entire files and source-code fingerprints (snippets) in the scanned files to generate evidence of OSS and third-party software on which you can take action.

  Currently remote scans do not support the use of the CL.

Scan Profiles

A scan profile controls aspects of the scan behavior, such as the level of inventory dependencies to scan, whether to perform package discovery or license detection within archives, the type source-code matching to be performed against the Code Insight Compliance Library, and other behavior configuration. For more information about scan profiles, Applying a Scan Profile to the Project.

Applying a Scan Profile to the Project

Code Insight supports scan profiles for abstracting and reusing scan settings. Often, organizations are concerned about consistent scan or audit practices across their enterprise, and scan profiles support that need. The following describes scan profiles and how to create one:

- About Scan Profiles
- Applying a Scan Profile

About Scan Profiles

Code Insight includes the following default scan profiles:

- **Basic Scan profile (without a CL)**—Used to produce automated findings along with string-based third-party indicators at a file level. This profile disables both exact-file and source-code matching, and therefore does *not* require a Compliance Library (CL).
- **Standard Scan profile**—Expands the file-level third-party indicators with exact-file matches based on the Compliance Library.
- **Comprehensive Scan profile**—Further expands the file-level third-party indicators with exact file-level and source-code matches based on the Compliance Library.

Additional scan profiles can be defined by the Code Insight System Administrator for use across projects, as described in the *Code Insight Installation & Configuration Guide*.

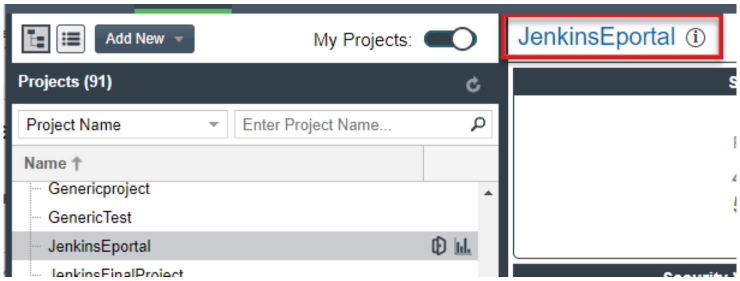
Applying a Scan Profile

The scan profile is used to abstract and reuse scan settings across projects. The scan profile currently selected for a project shows in the **Scan Settings** section on the **Summary** tab. The scan settings specified in the current scan profile are applied for each project scan. However, if you want to apply a different scan profile to the project, follow these steps.

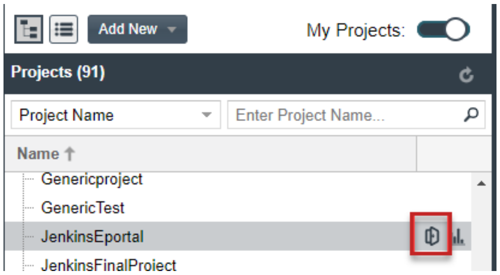
Task ***To select a new scan profile, do the following:***

1. Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.)
2. From the list of projects, click the project for which you want to apply a scan profile.

   (Optional) Click the **My Projects** toggle at the top of the list to show only those projects with which you are associated as Project Contact or through a project role. (For details, see Showing Only Your Projects.) You can also search projects by name, inventory, or security vulnerability as described in Searching Across All Projects in the System.
3. Do one of the following to open the project selected in the list:
   - Click the project name (in the example, *JenkinsEportal*) in the title bar of the right panel:

   - Click the **Open Project** icon to the right of the selected project in list:

   The project opens on its **Project Inventory** tab.
4. Click the project's **Summary** tab located left of the **Project Inventory** tab.
5. On the **Summary** tab, select **Edit Project** from the **Manage Project** menu.
6. From the **Edit Project** window, navigate to the **Scan Settings** tab, and select the desired scan profile for your project. (Click the information icon next to a selected scan profile to open a read-only view of its attributes.)

Uploading a Project Codebase (for Server Scans)

For a server scan, the codebase files that you want to scan for a project must reside on the Scan Server before you can perform the scan. One way to place these files on the Scan Server is upload them through Code Insight, as describe in this section. You can upload multiple codebases for a single project scan.

The following topics describe the codebase upload process:

- Performing the Codebase Upload
- Supported Archive Types for Uploading
- Expandable ArchivesExpandable Archives
- More About Archive Expansion Behavior During Codebase Uploads

Refer to the Code Insight User Roles and Permissions section for role requirements to upload the codebase.

As an alternative to (or in addition to) uploading codebases for a project, you can obtain codebase files for the project by synchronizing a Source Control Management codebase repository to the Scan Server. For more information, see Configuring Source Code Management. The complete codebase for a project can consist of files that were both uploaded and synchronized to the Scan Server.

Performing the Codebase Upload

The following information provides the basic requirements and the steps for uploading a codebase to the Scan Server:

- Basic Requirements for the Upload
- Uploading the Codebase

Basic Requirements for the Upload

The following are basic requirements for a codebase upload:

- The Scan Server to which you are uploading the codebase must be running (that is, the Tomcat server installed on the same instance as the Scan Server must be running).
- The codebase you are uploading must be archived in one of the supported upload formats. See Supported Archive Types for Uploading.
- Any archives that you want to expand within the codebase must be ones that the upload process supports for expansion. See Expandable Archives.
- The size limit for the codebase you are uploading is 10 GB.

  Although the 10 GB size limit for an upload cannot be changed, methods are available for providing a larger project codebase for scanning on the Scan Server. See the next section, Codebase Size Limitations for Uploads and Scans, for more information.

Codebase Size Limitations for Uploads and Scans

The size limit for a codebase upload is 10 GB and cannot be changed. However, the size of the project codebase that is actually scanned can be greater than 10 GB if you apply any of the following scenarios:

- Upload multiple 10 GB (or less) codebases in separate uploads for the same project (without overwriting previous files).
- Synchronize a codebase repository that is greater than 10 GB to the Scan Server (from Git, Subversion, or Perforce) using the Source Code Management functionality, as described in Configuring Source Code Management).
- Copy codebase files directly to the scan root (assuming you have access to the file system on the Scan Server).

Code Insight has been tested and is certified to successfully scan codebases that are within the following size limitations:

- When Code Insight uses a MySQL database, any codebase up to 35 GB in size and containing no more than 700,000 files.
- When a SQL Server database is used, any codebase up to 15 GB in size and containing no more than 300,000 files.

Uploading the Codebase

The following procedure describes how to upload the project codebase.

You can repeat these steps to upload each additional codebase for a project.

Note ▪ *You will be able to view the uploaded codebase in the* ***Codebase Files*** *tree in the* ***Analysis Workbench*** *once you scan the codebase. The resulting codebase display is based on how you configure the upload (for example, the level of archive expansion, whether archives are retained after expansion, and so forth), as described in this procedure.*

Task ***To upload a project codebase, do the following:***

1. Navigate to the **Summary** tab for the project for which you are uploading a codebase. (If necessary, see Opening the Project Summary Tab).
2. Click the **Upload Project Codebase** button to open the **File Upload** dialog.
3. Click **Select Archive File** to browse for the archive containing your codebase.
4. (Optional) Select **Delete existing project codebase files** to have Code Insight delete previously uploaded codebase files.
    - By keeping this option unselected, you can append new files or codebases to the existing codebase. (Default)
    - By selecting this option, you overwrite the existing codebase with the new one.

***Note** ▪ If you select to delete existing codebase files, a **Warning** dialog appears, asking you to confirm the deletion. Be aware that all existing codebase files for the project will be permanently removed from the Scan Server during the upload. If you rescan the project without replacing these files via a new upload, the scan results for the removed files will be permanently deleted.*

5. For **Archive File Expansion Options**, select the level of archive expansion you want to perform on the codebase during the upload.

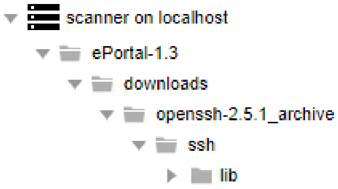
    - **Uploaded file only**—Extract the files from the uploaded archive only. Any extracted archives (called *first-level archives*) are not expanded. (Default)
    - **Uploaded file and first-level archives only**—Extract the files from the uploaded archive and expand all first-level archives in the codebase. See the next step for additional configuration available when you select this option.
    - **Uploaded file and all contained archives**—Extract the files from the uploaded archive and expand archives at all levels (that is, archives within archives) in the codebase. See the next step for additional configuration available when you select this option.

    For each expanded archive, the upload process extracts the archive contents to a folder automatically created with the archive name.

6. Configure settings that define the behavior of the upload process once archives are expanded. These settings are optional and are enabled only if **Uploaded file and first-level archives only** or **Uploaded file and all contained archives** has been selected.

    - **Delete archive files after expansion**—Remove those archives that have been expanded during an upload. (The archive is removed from the uploaded codebase after the upload is finished.) If you leave this option unselected, the archive is retained as an additional file directly under its parent folder. For examples of codebase trees that result based on how this option is configured, see More About Archive Expansion Behavior During Codebase Uploads. (Unselected by default)
    - **Append value to expanded archive directory name**—(Optional) Define a string to append to the name of any folder automatically created during the upload to store an archive's contents. After a scan, this appended string helps you to identify those folders in the codebase tree whose contents were extracted from archives, especially if the original archives were removed from the codebase during the upload (see the previous option).

        For example, suppose the appended value is `_archive`, and the upload process extracts an archive called `openssh-2.5.1`. After the upload process expands the archive, the name of the folder containing the archive contents becomes `openssh-2.5.1_archive`, as shown in this example of a scanned codebase in the **Codebase Files** tree. Note that the example also shows that the `openssh-2.5.1` archive has been removed due to the selection of **Delete archives after expansion**.



```
▼ scanner on localhost
   ▼ ePortal-1.3
      ▼ downloads
         ▼ openssh-2.5.1_archive
            ▼ ssh
               ▶ lib
```

The appended value has a maximum of 20 characters and does not support certain special characters.

(Hover over the ⓘ icon next to the **Append value to expanded archive directory name** field for a list of unsupported characters.)

7. Click **Upload**. Code Insight uploads your codebase file and attaches it to the selected project. You can now scan the uploaded codebase, enabling you to view the codebase and its associated third-party or open-source inventory in the **Analysis Workbench**.

Supported Archive Types for Uploading

The archive that you upload must be one of these types:

- `.zip`
- `.tar`
- `.tar.gz`
- `.7z`
- `.iso`
- `.ova`

Expandable Archives

The following archive types within the uploaded archive can be expanded either at the first-level or recursively, depending on the **Expand Archive** option you select on the **File Upload** dialog. An expanded archive enables you to view the evidence found in the specific files in the archive and determine the inventory associated with the evidence.

Table 2-1 ▪ Expandable Archives

- `.7z`
- `.apk`
- `.cpio`
- `.img`
- `.iso`
- `.jar`*****
- `.ova`
- `.tar`
- `.tar.bz2`
- `.tar.gz`
- `.tar.lzma`
- `.tar.xz`
- `.tgz`
- `.txz`
- `.zip`
- `.vmdk`

***** Only sources and uber/fat jars can be expanded.

More About Archive Expansion Behavior During Codebase Uploads

The following topics describe important information about how archives are expanded when a codebase is uploaded:

- About Archive Expansion
- Handling of the Archives After Their Expansion
- Expansion of an Uploaded Archive Containing an Intermediary .tar File
- Expansion of a Sources or Uber Jar
- Expansion of an Ova Archive
- Archive Expansion for Multiple Codebases Uploaded to the Same Project

About Archive Expansion

As described in Performing the Codebase Upload, when preparing to upload a project codebase to the Scan Server, you must specify the level of archive expansion (that is, the depth to which you want to see inside codebase archives) that will be visible once you scan the uploaded codebase. Depending on the level of expanded archives in the **Codebase Files** tree (in the **Analysis Workbench**), you can drill down to the files contained within the archive layers and view the specific evidence found within these files. From these files, you can then filter directly to those inventory items associated with the files.

For archives not expanded, you can still view the inventory directly associated with the archive, but you cannot access files within the archive nor view their evidence at the individual file level. Additionally, the file that is associated with a given inventory item is the unexpanded archive, not the actual file containing the evidence within the archive.

Handling of the Archives After Their Expansion

When an archive is expanded, its contents are extracted to a folder automatically created (with the archive's name) directly under the archive's parent folder. What happens to the archive once it is expanded depends on the **Delete Archive Files After Expansion** option selected.

For example, suppose the archive `AppsSport.zip` is located in the `coreApps` directory in your uploaded codebase. The `AppsSport.zip` archive contains the files `hockey1.exe` and `tennis.exe`.

Archive Retention Configured

If **Delete Archive Files After Expansion** is *not* selected, the archive `AppsSport.zip` is retained in its parent folder, `coreApps`, once it is expanded. The resulting codebase tree looks like this, where both `AppsSport.zip` and a folder `AppsSport`, containing the archive contents, are found directly under `coreApps`:

```
coreApps
---AppsSport
-----hockey1.exe
-----tennis1.exe
---AppsSport.zip
```

Note ▪ *When an archive is retained, both the archive and its extracted files are processed during a codebase scan. This can result in a duplication of inventory.*

Archive Removal Configured

If **Delete Archive Files After Expansion** is selected, the archive `AppsSport.zip` is removed once it is expanded, resulting in the following codebase tree:

```
coreApps
---AppsSport
-----hockey1.exe
-----tennis1.exe
```

Expansion of an Uploaded Archive Containing an Intermediary .tar File

The `.tar.gz`, `.tgz`, `.txz`, and `.tar.xz` archive types and similar archives contain an intermediary `.tar` archive. The codebase upload extracts the intermediary `.tar` file from the initial archive, but applies the **Archive Expansion Options** configuration starting with the expansion of the intermediary `.tar` file, not the initial archive. The following examples demonstrates this expansion behavior.

- Contents of the Archive with an Intermediary .tar File
- Expansion of Uploaded File Only
- Expansion of Uploaded File and First-Level Archives Only
- Expansion of Uploaded File and All Contained Archives

Note ▪ *The .jar files referenced in these examples represent normal jar files. Code Insight currently does not support the expansion of normal jar files. However, it does support the expansion of uber and sources jars. For examples of how these jars are expanded, see Expansion of a Sources or Uber Jar.*

Contents of the Archive with an Intermediary .tar File

Suppose the archive `jars.tar.gz` has these contents, where the intermediary file is `jar.tar`:

```
jars.tar.gz
--jars.tar
----file-1.txt
----file-2.txt
----jar.zip
------abc.jar (color)
------xyz.jar
------classes.zip
--------corporation.class
--------employee.class
```

Expansion of Uploaded File Only

The uploaded codebase looks like this if **Uploaded File Only** for **Archive Expansion Options** is applied:

```
file-1.txt
file-2.txt
jar.zip
```

In this case:

- The `jars.tar` is extracted from `jars.tar.gz`.
- The `jars.tar` archive is then expanded but not retained.
- Any first-level archives (in this case, `jar.zip` found in `jars.tar`) are retained but not expanded.

Expansion of Uploaded File and First-Level Archives Only

The uploaded codebase looks like this if the **Uploaded file and first-level archives only** option for **Archive Expansion Options** is selected but the **Delete archive files after expansion** option is *not* selected.

```
file-1.txt
file-2.txt
\jar
---abc.jar
---xyz.jar
---classes.zip
jar.zip
```

In this case:

- The `jars.tar` is extracted from `jars.tar.gz`.
- The `jars.tar` archive is then expanded but not retained.
- The first-level `jar.zip` archive (contained in `jars.tar`) is expanded (and retained).
- The second-level `classes.zip` archive (contained in `jar.zip`) is not expanded.

If **Delete archive files after expansion** *is* selected, the uploaded codebase looks like this:

```
file-1.txt
file-2.txt
\jar
---abc.jar
---xyz.jar
---classes.zip
```

Expansion of Uploaded File and All Contained Archives

The uploaded codebase looks like this if the **Uploaded file and all contained archives** option for **Archive Expansion Options** is selected but the **Delete archive files after expansion** option is *not* selected.

```
file-1.txt
file-2.txt
jars.tar
\jar
---abc.jar
---xyz.jar
---\classes
```

```
------Corporation.class
------Employee.class
---classes.zip
jar.zip
```

In this case:

- The `jars.tar` is extracted from `jars.tar.gz`.
- The `jars.tar` archive is then expanded but not retained.
- The first-level `jar.zip` archive (contained in `jars.tar`) is expanded (and retained).
- The second-level `classes.zip` archive (contained in `jar.zip`) is expanded (and retained).

If **Delete archive files after expansion** *is* selected, the uploaded codebase looks like this:

```
file-1.txt
file-2.txt
\jar
---abc.jar
---xyz.jar
---\classes
------Corporation.class
------Employee.class
```

Expansion of a Sources or Uber Jar

While Code Insight does not support the expansion of regular jar, it does support the expansion of the following special jars:

- **Sources jar**—Contains all the source code (that is, the `.java` and `.class` files) of a compiled Java program.
- **Uber (or fat) jar**—Contains all the source code of the compiled Java program but also embeds the program's dependencies, thus providing an "all-in-one" distribution of the software. An uber jar can contain other uber and sources jars in addition to regular jars.

Refer to the following topics for more information about the expansion of these jar types during an upload:

- Enabling the Expansion of Sources and Uber Jars
- Sources and Uber Jars Uploaded in Supported Archive
- Example Expansion of an Uber Jar

Enabling the Expansion of Sources and Uber Jars

The option to expand sources and uber jars during a codebase upload is set at the project level. If you want to expand uber and sources jars in the codebase, be sure that this option is enabled before performing a codebase upload. (By default, this option is disabled.)

Task ***To enable (or disable) the expansion of uber and sources jars during codebase uploads, do the following:***

1. Follow the steps in Editing the Project Definition and General Settings to access the settings for the project in whose codebase you want to be able to expand (or not expand) uber and sources jars.
2. On the **General** tab for the project, select (or unselect) the **Expand Source and Uber jar files** option.
3. Click **Save** to apply the change.

Sources and Uber Jars Uploaded in Supported Archive

The top-level sources or uber jar must be archived and uploaded as one of the supported archive types listed in Supported Archive Types for Uploading. You cannot directly upload one of these jar files.

Example Expansion of an Uber Jar

The following examples demonstrate how a codebase that is uploaded in an uber jar is displayed in the **Analysis Workbench**, depending on the archive level configured for the upload (as described in Performing the Codebase Upload).

- Contents of the Uber Jar
- Expansion of Uploaded File Only (Containing the Uber Jar)
- Expansion of Uploaded File (Containing the Uber Jar) and First-Level Archives Only
- Expansion of Uploaded File (Containing the Uber Jar) and All Archives

Note ▪ *The expansion of a sources jar is handled like that of an uber jar in these examples except that the sources jar will not contain other jars.*

In these examples, the uber jar `codebaseABC.jar` is uploaded in a zip file of the same name. Each example assumes that the **Expand Source and Uber jar files** option has been enabled for the project (as described in Enabling the Expansion of Sources and Uber Jars).

Contents of the Uber Jar

The contents of the zip file and the uber file it contains looks like this before uploading:

```
codebaseABC.zip (uploaded file)
---codebaseABC.jar (uber jar containing codebase)
------uber2.jar
---------regular1.jar
------------file1.txt
------------file2.txt
------sources1.jar
-----------file3.java
-----------file4.java
------folder
---------file5.bin
```

Expansion of Uploaded File Only (Containing the Uber Jar)

The uploaded codebase looks like this if **Uploaded File Only** for **Archive Expansion Options** is applied:

```
codebaseABC.jar
```

In this case:

- The codebaseABC.jar is extracted from codebaseABC.zip.
- The first-level archive, codebaseABC.jar (the main uber jar), is retained but not expanded.

Expansion of Uploaded File (Containing the Uber Jar) and First-Level Archives Only

The uploaded codebase looks like this if the **Uploaded file and first-level archives only** option for **Archive Expansion Options** is selected but the **Delete archive files after expansion** option is *not* selected.

```
codebaseABC
---uber2.jar
---sources1.jar
---folder1
------file5.bin
codebaseABC.jar
```

- The codebaseABC.jar is extracted from codebaseABC.zip.
- The first-level archive, codebaseABC.jar (the main uber jar) is expanded (and retained).
- The second-level archives inside codebaseABC.jar—uber2.jar and sources1.jar—are retained but not expanded.
- The folder, folder1 (inside codebaseABC.jar) and its contents, file5.bin, are visible in the codebase. Any evidence found in the file file5.bin can be explored in the **Analysis Workbench**.

If **Delete archive files after expansion** *is* selected, the uploaded codebase looks like this:

```
codebaseABC
---uber2.jar
---sources1.jar
---folder1
------file5.bin
```

Expansion of Uploaded File (Containing the Uber Jar) and All Archives

The uploaded codebase looks like this if the **Uploaded file and all contained archives** option for **Archive Expansion Options** is selected but the **Delete archive files after expansion** option is *not* selected.

```
codebaseABC
---uber2
------regular1.jar
---uber2.jar
---sources1
--------file3.java
--------file4.java
---sources1.jar
---folder1
------file5.bin
codebaseABC.jar
```

In this case:

- The codebaseABC.jar is extracted from codebaseABC.zip.
- This first-level archive, codebaseABC.jar (the main uber jar), is then expanded (and retained).
- The second-level archive inside codebaseABC.jar—uber2.jar and sources1.jar—are expanded (and retained).
- The regular1.jar inside the uber2.jar is retained but not expanded since it is a regular jar.
- The files, file3 and file4, contained in the sources1.jar are visible in the codebase. Any evidence available in these files can be explored in the **Analysis Workbench**.
- The folder, folder1 (inside codebaseABC.jar) and its contents, file5.bin, are visible in the codebase. Any evidence found in the file file5.bin can be explored in the **Analysis Workbench**.

If **Delete archive files after expansion** *is* selected, the uploaded codebase looks like this:

```
codebaseABC
---uber2
------regular1.jar
---sources1
--------file2.java
--------file3.java
---folder1
------file5.bin
```

Expansion of an Ova Archive

Code Insight supports the upload of a codebase in an .ova archive and the expansion of its .vmdk archives (usually found at the first level in the .ova file) and the .img archives (usually found in the .vmdk file). Other archives, such .iso files, might be part of codebase upload; and these too are expanded, as long as their expansion is supported by Code Insight (see Expandable Archives). As with all uploads, the expansion level of .ova archive is based on the **Archive Expansion Options** configuration for the upload.

If the .vmdk archive contains a large of amount of data, the upload might experience a disk space issue. In this case, the .vmdk archive is retained but not expanded, despite the level of expansion configured for the upload.

Archive Expansion for Multiple Codebases Uploaded to the Same Project

If multiple codebases are uploaded to the same codebase path for a given project (and existing codebase files are not deleted), the archives within all the codebases for the project are expanded based on the current **Archive Expansion Options** configuration. The following process demonstrates this behavior:

1. Create project1 and upload the codebase codefiles1.zip, using the **Uploaded file only** option. The contents of codefile1.zip are extracted. Archives in these contents are not expanded.
2. Upload codefile2.zip to the same project (at the same codebase path), this time using the **Uploaded file and all contained archives**. (Keep in mind that codebase1.zip was previously expanded with no further expansion of any archives in its contents.) Now all archives at all levels are expanded within the codefile1 and codefile2 codebases.
3. Upload codefile3.zip to the same project, this time using **Uploaded file and first-level archives only**. Now *only the first-level archives* in all three codebases are expanded.

If you upload multiple codebases to the same project, best practice is to keep track of the **Archive Expansion Options** configuration for each upload so that you can apply an appropriate configuration for the subsequent upload.

Scanning the Codebase (Server Scans)

After a project's codebase has been uploaded to (or synchronized to) the Scan Server and the appropriate scan profile is selected, the codebase is ready to be scanned. To perform the scan, you must have proper permissions (see Code Insight User Roles and Permissions), and the Scan Server must be running—that is, the Tomcat server (installed on the same instance as the Scan Server) must be running.

The following instructions describe how to start a server scan on your codebase. (For information about the differences between server and remotes scans, refer to About Code Insight Scans.)

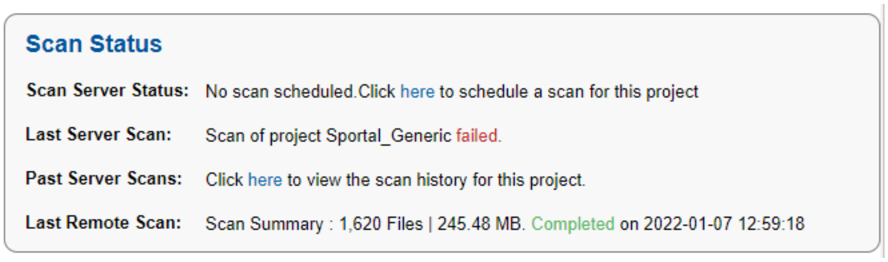
Note ▪ *When using a MySQL database, Code Insight is certified to scan a codebase up to 35 GB in size and containing no more than 700,000 files. When using a SQL Server database, Code Insight is certified to scan a codebase up to 15 GB and containing no more than 300,000 files. Also see Codebase Size Limitations for Uploads and Scans.*

Task ***To start the scan, do the following:***

1. Navigate to the **Summary** tab for the project that you want to scan. (If necessary, see Opening the Project Summary Tab).
2. Click the **Start Scan** button (or the link in **Scan Server Status**) to trigger the scan. The scan is queued and runs in the background. You can monitor the scan's progress by clicking the "here" link in **Past Server Scans** to obtain information about scheduled, active, and past scans on the project. You can also monitor the scan in the **Jobs** queue (see Monitoring the Code Insight Jobs Queue). Note the following:

    - If a scan is running on another project, your scan will automatically start based on queue order. Additionally, if the Scan Server is temporarily inactive, the scan will automatically start based on queue order once the server is running again.
    - The project currently open can have only one scan in queue or in progress at a time. If you attempt to schedule a scan when your project already has a scan queued or running, the **Start Scan** button will be disabled until the scan completes. For more reasons for **Start Scan** button disablement and the actions you can take, see Actions to Take When the Start Scan Button is Disabled.
    - If a report generation is currently in queue or in progress for the project, the scan is not triggered. Instead, a pop-up error message is displayed, explaining that you must wait until the report generation has completed before repeating this step to attempt to trigger the scan again.

    Information about the scan's progress appears in the **Scan Server Status** section on the **Summary** tab.



Scan Status

Scan Server Status: No scan scheduled.Click here to schedule a scan for this project

Last Server Scan: Scan of project Sportal_Generic failed.

Past Server Scans: Click here to view the scan history for this project.

Last Remote Scan: Scan Summary : 1,620 Files | 245.48 MB. Completed on 2022-01-07 12:59:18

When the scan completes, **Last Server Scan** will display one of the following messages:

- **Completed**—The scan succeeded with no warnings during scan or analysis. This message appears on the screen in green.
- **Completed with warnings**—The scan succeeded but the analysis produced warnings. For more information, check the **scanEngineDetail** log for the Scan Server.
- **Failed**—The scan failed. This message appears on the screen in red. For more information, see Scan Failure Reasons and Troubleshooting Measures.

For an overall understanding of the scan results, see Overview of Scan Results.

3. Do any of the following:

   - Manage the project. For example, you can assign users to project analyzer or reviewer roles, define the project's scan settings, configure an automated review and remediation workflow, configure a connection to a remote data source such as Perforce or Jira, and more. See Managing Projects for details.
   - Analyze the scan results, as described in Auditing Scan Results in the Analysis Workbench.
   - Generate the following standard reports and any applicable custom reports that have been added:
     - Project Report
     - Audit Report
     - Notices Report

Actions to Take When the Start Scan Button is Disabled

The **Start Scan** button on the **Summary** tab for a project is disabled under the following two conditions.

- **The project that you are attempting to scan is already in the scan queue or is currently being scanned**—Check the **Scan Server Status** field on the **Summary** tab to confirm the "Scan scheduled" or "Project being scanned" status. Then wait until the scan completes before attempting to schedule another on the same project.

  Note ▪ *Under certain circumstances, the **Scan Server Status** field might not update quickly enough to reflect the "Scan scheduled" or "Project being scanned" status. However, if a scan on the current project is indeed already in queue or running, an attempt to click the field's "here" link to schedule a scan will result in an error message, stating that you cannot start another scan on the project. For your reference, the message also provides the task ID for the currently queued or running scan. (This ID can be used with the **Get Scan Status** API to check the scan status outside of the UI when necessary.)*

- **The Scan Server associated with the project is disabled for scanning**—If no scan is scheduled or running for the current project, check with the Code Insight System Administrator to determine the status of the Scan Server. If it is disabled, you will need to create a new project for the codebase and associate it with an enabled Scan Server. (The **Start Scan** button and the "here" link in the **Scan Server Status** field should still be enabled if the Scan Server is only temporarily inactive.)

Scan Failure Reasons and Troubleshooting Measures

The following lists possible causes and troubleshooting help for the failures of a server scan.

Table 2-2 ▪ Scan Failure Causes and Troubleshooting Measures

Scan Failure Cause	Troubleshooting Measures
Scan server is not accessible	Verify that the correct hostname and port for the selected Scan Server have been identified in Code Insight.
Scan server is unable to access or read the CL files	Verify that the correct Compliance Library (CL) path has been identified for the Scan Server.
Scan server ran out of memory	Ensure that the JVM heap (memory) size is adequate for running the Scan Server. (Recommended JVM heap sizes are listed in the *Code Insight Installation and Configuration Guide*.)
Codebase file(s) are not accessible and cannot be read	Verify (and adjust if necessary) the codebase file permissions.
Codebase file(s) are encrypted and cannot be read	Attempt to open the codebase files in **7-zip** or **winzip**. This application might provide a clearer description of the error than the scan process can.
Codebase file(s) are corrupted and cannot be read	Attempt to open the files in an external text editor. The editor might provide a clearer description of the error than the scan process can.
Codebase file(s) contain unparseable characters	This type of error is rare. Should it occur, verify that your database character set and collation settings are correct and that they match the requirements listed in the *Code Insight Installation and Configuration Guide*.
Indexing of the scanned codebase files and results failed	To help you identify the problem and troubleshoot, review the **scanEngineDetail** log for the Scan Server.

Table 2-2 ▪ Scan Failure Causes and Troubleshooting Measures (cont.)

Scan Failure Cause	Troubleshooting Measures
Unable to communicate with CodeAware	This scan failure can occur when both of these conditions exist: ● Code Insight is running in a proxy-enabled environment. ● The Scan Server is running under its fully qualified domain name. The Scan Server must call Code Insight Automated Analysis to analyze the codebase files. If the time required by Automated Analysis to analyze files exceeds the proxy server “read timeout” limit, the scan fails (even though Automated Analysis might still finish). Try either of these methods to resolve this scan-failure issue: ● If the Core Server and Scan Server are running on the same instance, change the Scan Server hostname to `localhost`. (If you are running Code Insight in SSL mode, ensure that the SSL certificates accommodate the hostname change.) ● If the Core Server and Scan Server are not running on the same instance, try excluding the Scan Server from the proxy by adding its hostname to the `http.nonProxyHosts` property in the proxy details. The *Code Insight Installation and Configuration Guide* provides information about configuring Code Insight to run in SSL mode or in a proxy-enabled environment.
No alternative DNS name found that matches localhost	This scan failure occurs when all these conditions exist: ● Code Insight is running in a proxy-enabled environment. ● The Core Server and Scan Server are installed on separate instances. ● Both servers are configured for SSL. Try these methods to resolve the scan-failure issue: ● Ensure that the Secure Site SSL certificate on each instance has been properly configured. ● Try excluding the Scan Server from the proxy by adding its hostname to the `http.nonProxyHosts` property in the proxy details. The *Code Insight Installation and Configuration Guide* provides information about configuring Code Insight to run in SSL mode or in a proxy-enabled environment.

Table 2-2 ▪ Scan Failure Causes and Troubleshooting Measures (cont.)

Scan Failure Cause	Troubleshooting Measures
Unable to find valid certificate	This scan failure can occur when both of these conditions exist: ● The Core Server and Scan Server are installed on separate instances. ● Both servers are configured for SSL. The scan fails when the Core Server is unable to communicate with the Scan Server. Ensure that the Secure Site SSL certificate on each instance is valid and has been properly imported. (The *Code Insight Installation and Configuration Guide* provides information about procuring and importing these certificates as part of the SSL configuration for Code Insight.)

Overview of Scan Results

This section provides an overview of the following basic scan results, which can be examined in the **Analysis Workbench** and on the **Project Inventory** tab.

- Inventory
- Scan Evidence (**Analysis Workbench** only)
- License Information

Details about how these scan results pertain specifically to Analysts (who examine the results using the **Analysis Workbench**) and Reviewers (who examine the results using the **Project Inventory** tab) are found in later sections, Auditing Scan Results in the Analysis Workbench and Reviewing Published Inventory for a Project.

Inventory

System-generated inventory is created by Code Insight during a scan and is available for view in the **Analysis Workbench** and, if automatically published, on the **Project Inventory** tab. An inventory item represents an explicit finding in the scanned codebase and can represent any of the following: top-level component, bundled component, component found inside an archive, or direct or transitive dependency component.

***Note ▪** Alternatively, you can review the published inventory across all projects. For details, see Viewing Inventory Across All Projects.*

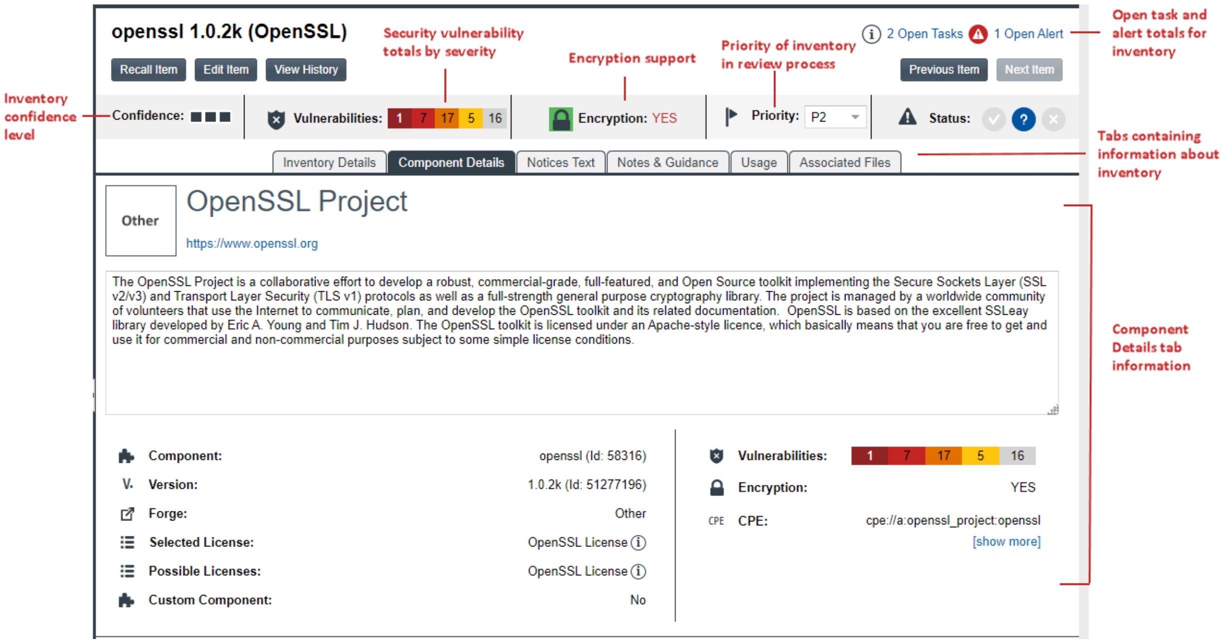
An inventory item typically has an associated component, version, license and list of security vulnerabilities, as well as other details about these elements. See Inventory Details Tab in the Analysis Workbench for a full description of the information collected by the scan.

These are some important elements about an inventory item that you can view at a glance:

- Review Status of Inventory
- Inventory Priority

- Inventory Confidence
- Security Vulnerabilities Associated with Inventory
- Inventory Usage Information

The following example highlights these elements for a given inventory item on the **Project Inventory** tab. Many of these same elements are available in the inventory view in the **Analysis Workbench**. (For more information about the **Analysis Workbench**, see Auditing Scan Results in the Analysis Workbench. For information about the **Project Inventory** tab, see Reviewing Published Inventory for a Project.)



openssl 1.0.2k (OpenSSL)

2 Open Tasks 1 Open Alert

Confidence: ■■■ | Vulnerabilities: 1 7 17 5 16 | Encryption: YES | Priority: P2 | Status:

OpenSSL Project

https://www.openssl.org

The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. The project is managed by a worldwide community of volunteers that use the Internet to communicate, plan, and develop the OpenSSL toolkit and its related documentation. OpenSSL is based on the excellent SSLeay library developed by Eric A. Young and Tim J. Hudson. The OpenSSL toolkit is licensed under an Apache-style licence, which basically means that you are free to get and use it for commercial and non-commercial purposes subject to some simple license conditions.

Component: openssl (Id: 58316)
Version: 1.0.2k (Id: 51277196)
Forge: Other
Selected License: OpenSSL License ⓘ
Possible Licenses: OpenSSL License ⓘ
Custom Component: No

Vulnerabilities: 1 7 17 5 16
Encryption: YES
CPE: cpe://a:openssl_project:openssl
[show more]

Inventory confidence level

Security vulnerability totals by severity

Encryption support

Priority of inventory in review process

Open task and alert totals for inventory

Tabs containing information about inventory

Component Details tab information

Review Status of Inventory

During a scan, all inventory is checked against existing policies as defined in the Policy Profile. As a result, inventory is either automatically approved or rejected by policy or unaffected by policy. If inventory is not affected by policy, it should be manually reviewed, and the Policy Profile should be updated to reflect the review decision for future scans. The manual review process is described in detail in Reviewing Published Inventory for a Project.

For information about setting up policies that automate the inventory review process, see Managing Policies to Automatically Review Inventory.

Note *▪ Unaffected inventory is labeled as **Draft** in the **Review Status** field in the **Analysis Workbench** and is represented as a circled X (for **Not Reviewed**) in the **Status** field on the **Project Inventory** tab.*

Inventory Priority

The priority of an inventory item is meant to highlight which items are in more need of a review than others during the inventory review process. During a scan, the priority for auto-published inventory is automatically assigned based on the associated license. Code Insight uses the following algorithms determine the default priority of an inventory item.

Note *▪ You can manually change the inventory priority by simply selecting a different priority from the* ***Priority*** *dropdown list either in the* ***Analysis Workbench*** *or on the* ***Project Inventory*** *tab.*

For a “Component” Inventory Type

Code Insight sets the inventory priority to P1 if any of these circumstances exist:

- The inventory item has at least one associated security vulnerability with a severity of High (for CVSS v2.0) or Critical (for CVSS v3.x).
- The **Selected License** priority is P1 (see License Priority).
- No licenses are found (that is, the **Selected License** value is **I don’t know** and no evidence of other licenses is found in the files associated with the inventory item).

Otherwise, when the user or system selects a component-version-license triad, the inventory priority is based on the license priority or highest associated security vulnerability severity, *unless* that would mean lowering an existing inventory priority.

Note *▪ If the* ***Selected License*** *value for an inventory item is* ***I don’t know*** *but evidence of other licenses is found in the files associated with inventory item, the inventory priority is based on the highest priority among the found licenses or the highest associated vulnerability severity.*

For a “License-Only” Inventory Type

When a user selects a license for a license-only inventory item, the inventory priority is set to the license priority (see License Priority) *unless* that would mean lowering an existing inventory priority.

Note *▪ Due to the algorithm used to calculate the priority, the system-generated inventory priority will never be lowered by the system. It can only be lowered explicitly by the user.*

Inventory Confidence

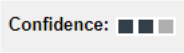
The Automated Analysis portion of the Code Insight Scan Server uses a variety of techniques to identify inventory items from the scanned code base. The Confidence level (High, Medium, or Low) of an inventory item is a measure of the strength of the discovery technique used to generate the inventory item and the certainty of the finding. It is derived by assigning a score to the following elements:

- The strength of the analysis technique that provided the metadata on the inventory item.

- The existence of this inventory item in the Code Insight Data Library: items that have matching components in the Data Library have higher levels of confidence.

The Confidence level is represented as a simple three-segment graph for each inventory item in the **Analysis Workbench** or on the **Project Inventory** tab. Three shaded segments indicate High confidence, two indicate Medium, and one indicates Low.

The following **Confidence** graph shows Medium confidence (with two of the three segments shaded):



Confidence: ■■□

The Confidence level is also available as a search criterion on the **Project Inventory** tab and can be used to quickly identify items that may require additional triage or review.

The following describes the Confidence levels:

- **High confidence**—An inventory item of High confidence means that either the item was identified with a specific and highly targeted rule or from the processing of a structured manifest file from a package manager (such as `pom.xml` for the maven package manager and `package.json` for the npm package manager). A High-confidence inventory item almost always matches with a component in the Code Insight Data Library and rarely requires further triage or review by the Analyst.
- **Medium confidence**—An inventory item of Medium confidence means that the item was identified using a more generic technique or by the processing of a secondary indicator to produce an inventory item. A Medium-confidence inventory item might or might not have a match to a component in the Code Insight Data Library and might require triage or review in order to be validate or further refine the finding.
- **Low confidence**—An inventory item of Low confidence means that the inventory item was identified using a very generic rule or an exploratory detection technique, and thus might represent a component of unknown origin. Inventory of Low confidence rarely have a match to a component in the Code Insight Data Library and should be further triaged and reviewed by an Analyst for accuracy and completeness.

The table below summarizes the various detection techniques and the corresponding confidence value:

Table 2-3 ▪ Confidences Levels Associated with Various Detection Techniques

Detection Technique	Rule or Configuration File Used	Confidence Level
Analyzers	Primary	Hight
Analyzers	Secondary	Medium
Search term analysis	Rules with versions	High
Search term analysis	Component-only rules	Medium
File name analysis	Specific rules	Low
File name analysis	Generic rules of certain type of components	Low
File name analysis	Generic rules	Low

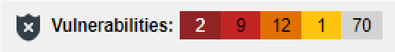
Table 2-3 ▪ Confidences Levels Associated with Various Detection Techniques (cont.)

Detection Technique	Rule or Configuration File Used	Confidence Level
Direct dependencies	Based on package manager files (`pom.xml`, `package.json`, and so forth)	Low by default, but can increase to Medium if matching component + version is found in the Code Insight Data Library
Transitive dependencies	Based on lookups against respective repositories (maven, npm, and so forth)	Low by default, but can increase to Medium if matching component + version is found in the Code Insight Data Library

Security Vulnerabilities Associated with Inventory

Code Insight uses data from the National Vulnerability Database (NVD) and other advisories such as RubySec to report security vulnerabilities associated with your inventory items. The information from these sources is used to create vulnerability rankings and alerts.

The **Vulnerabilities** bar graph shows the current security-vulnerability counts by severity level for a given inventory item listed in the **Analysis Workbench** and on the **Project Inventory** tab (and in other locations):

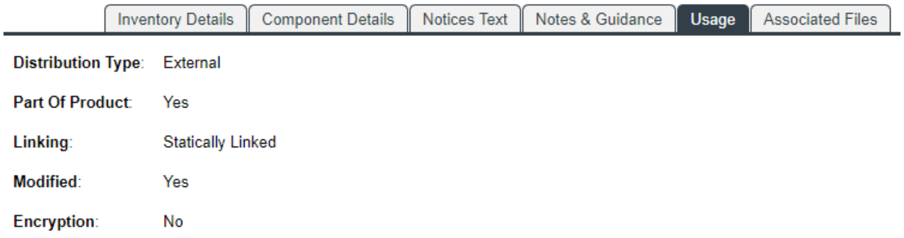


For more information about how to explore the security vulnerabilities associated with inventory, see Working with Security Vulnerabilities. This same section also describes how to suppress a vulnerability in your Code Insight instance if, for example, you have taken steps to protect your code against the vulnerability or if the vulnerability proves to be a “false positive”.

Inventory Usage Information

Code Insight provides the ability to see and edit usage information for a given OSS or third-party component associated with an inventory item. Usage information describes how a software package developed in your organization uses the OSS or third-party component. This information is important because it aids auditors and reviewers in determining how closely to monitor an inventory item for intellectual property (IP) and security risks and whether to approve or reject the item, create tasks for its remediation, and issue alerts and notifications pertaining to the item. Usage properties can also help users determine whether an inventory item should be included in Third-Party Notices and what steps need to be taken to satisfy license obligations and conditions of use. Finally, usage information can help to identify license conflicts and compatibility issues.

The inventory usage fields are available on the **Usage** tab for a given inventory item, as found both on the **Project Inventory** tab (shown below) and in the inventory view in the **Analysis Workbench**. You can update these fields when you manually create or edit inventory items.



Distribution Type: External

Part Of Product: Yes

Linking: Statically Linked

Modified: Yes

Encryption: No

- **Distribution Type**—Indicates how you are distributing the OSS or third-party component associated with the inventory item. The distribution type can affect license priority and obligations.
    - **Externally** with your product, shipped to customers (outside of your organization, including a private cloud deployment at the customer's site)
    - As an application **hosted** in your company's data center (such as a SAAS application)
    - **Internally** only (such as an internal test framework included in the codebase but not distributed with the product)
    - Distribution method **unknown**
- **Part of Product**—Indicates whether the OSS or third-party component is part of the core product or an infrastructure piece such as a build or test tool. This information can affect whether third-party notices are required for this item.
- **Linking**—Indicates how your software package links to libraries in the OSS or third-party component—statically (the component is included in the materials), dynamically (the component is brought in at runtime), or not linked at all. Linking can affect license priority and obligations.
- **Modified**—Indicates whether code from the OSS or third-party package has been modified for use by your organization.
- **Encryption**—Indicates whether the component provides encryption capabilities used in the product. Encryption can affect export controls.

For explicit directions on viewing or editing inventory usage either in the **Analysis Workbench** or on the **Project Inventory** tab, see the following:

- Viewing Security Vulnerabilities for Inventory in the Analysis Workbench
- Viewing Usage Information for Project Inventory
- Editing Inventory from the Project Inventory Tab

Scan Evidence

Scan evidence is generated by Code Insight during a scan and is available for view in the **Analysis Workbench** to any analyst assigned to the project. Scan evidence is typically an indicator of open-source or third-party content in the codebase. It can be useful for verifying system-generated inventory, identifying and creating additional inventory not discovered during scan, finding embedded licenses and copyrights in bundled code or archives, determining file origin, and locating stolen or borrowed code.

You can quickly view filter on the following evidence in codebase files in the **Analysis Workbench**. (For more details about examining evidence in the **Analysis Workbench**, see Examining and Managing Open-Source Evidence for a Given File and Viewing a Summary of Evidence Detected Across the Codebase.)

- **Exact Matches**—A whole-file match to a file in the Compliance Library
- **Source Matches**—Snippet-level matches to files in the Compliance Library
- **Copyrights**—Third-party copyright statements detected in the code
- **Emails/URLs**—Third-party emails and URLs detected in the code
- **Licenses**—Licenses detected in the code based on custom license patterns supplied by Electronic Update
- **Search Terms**—String matches based on pre-configured search terms provided by Code Insight and on custom search terms added by the user as part of the Scan Profile

Scan Evidence from Scan-Agent Plugins

For files scanned by a Code Insight scan-agent plugin on a remote system, only license evidence is currently reported in Code Insight.

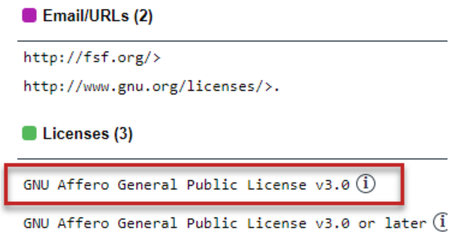
License Information

The Code Insight scan detects license text and references to licenses in your codebase and enables you to examine this information is various ways, such as viewing the license or license-reference text highlighted in the codebase file itself (see Examining Evidence of Open-Source Copyrights, Email Addresses, URLs, Licenses, and Search Terms in a Given Non-Binary File). The scan can also generate inventory to which it associates a license based on the open-source or third-party component. The following topics provide an overview of other license-related information you can examine or manage:

- License Details from the Code Insight Data Library
- License Priority
- Reporting of Detected License Text through the As-Found Text Inventory Field
- Notices Text Field

License Details from the Code Insight Data Library

The Code Insight scan can use information in the Code Insight Data Library to automatically select a license for a codebase file based on evidence found in the file. The scan also uses the Data Library to select a license for an automatically generated inventory item based on the inventory's open-source or third-party component. You can view details from the Data Library for this license by simply clicking ⓘ next to any license reference in the **Analysis Workbench** or the **Project Inventory** tab.



🟪 Email/URLs (2)

http://fsf.org/>
http://www.gnu.org/licenses/>.

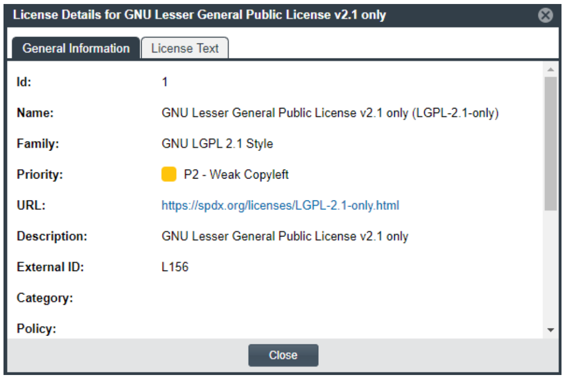
🟩 Licenses (3)

GNU Affero General Public License v3.0 ⓘ
GNU Affero General Public License v3.0 or later ⓘ

The **License Details** window is displayed containing the following information:

- A **General Information** tab that lists details such as the name of the license, its family, and the license priority assigned by Code Insight.
- A **License Text** tab that displays the complete license text (representing the external forge license text).

For descriptions of these fields in this window, see License Details Window.


License Details for GNU Lesser General Public License v2.1 only

Id:	1
Name:	GNU Lesser General Public License v2.1 only (LGPL-2.1-only)
Family:	GNU LGPL 2.1 Style
Priority:	🟨 P2 - Weak Copyleft
URL:	https://spdx.org/licenses/LGPL-2.1-only.html
Description:	GNU Lesser General Public License v2.1 only
External ID:	L156
Category:	
Policy:	

In addition to accessing the **License Details** window from the **Analysis Workbench** and the **Project Inventory** tab, you can access this window by clicking the ⓘ icon next to a license reference in these locations as well:

- Next to the **Name** column in the row for any license policy listed in the **Licenses** section on the Policy Details Window. For more information, see Managing Policies to Automatically Review Inventory.
- In the license column for any component listed on the **Components** tab in **Global Component & License Lookup**. For more information, see Exploring Components Globally.
- New to the license you select a component version your creating on the **Create Component Version** window. For more information, see Creating Custom Component Versions.

License Priority

You want to understand the priority of licenses in your codebase so you can handle them based on your corporate policies. Code Insight uses a default license priority to highlight which licenses associated with inventory need more attention than others in the inventory review process, helping to define day-one work items.

Each license referenced in the **Analysis Workbench** and on the **Project Inventory** tab has one of the following priority values:

Table 2-4 ▪ License Priorities

Priority	Characteristics	Icon	Description
P1	Viral/Strong Copyleft	🟥	Usually, P1 licenses require immediate attention due to the possibility of tainting proprietary application code, an issue that can have significant business impact.

Table 2-4 ▪ License Priorities (cont.)

Priority	Characteristics	Icon	Description
P2	Weak Copyleft/ Commercial/ Uncommon	🟨	The typical P2 license requires legal review and guidance based on corporate policies about the proper use of these types of licenses in your organization.
P3	Permissive/Public Domain	🟩	In general, P3 licenses are allowed and have minimal impact to an organization as long as license obligations are satisfied. The most common license obligation is properly attributing the use of an open source component to its author. This is the default priority.

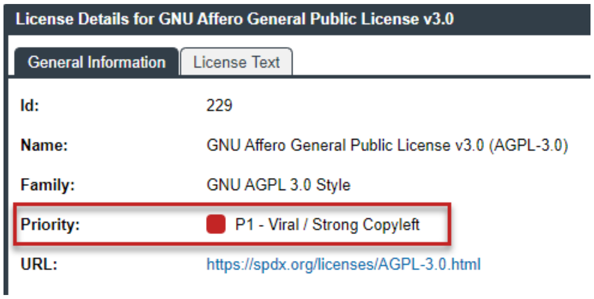
Inventory priority (see Inventory Priority) is a risk metric for the inventory item that takes license priority into account as one of the contributing factors. Inventory priority is set at scan time when the inventory item is created by the system or during inventory review. You can set or override the inventory priority at any time. License priority, on the other hand, is static and never changes. The license priority is supplied by the Electronic Update.

Inventory priority typically defaults to the license priority value unless a critical vulnerability exists or you manually override the inventory priority value (as described in Inventory Priority).

***Note** ▪ Code Insight REST APIs that reference the license entity, such as the Component Search API, include the license priority in the API response body.*

Viewing the License Priority

You can view the license priority from the **License Details** window associated with the license. See License Details from the Code Insight Data Library for details on accessing this window.



Id: 229

Name: GNU Affero General Public License v3.0 (AGPL-3.0)

Family: GNU AGPL 3.0 Style

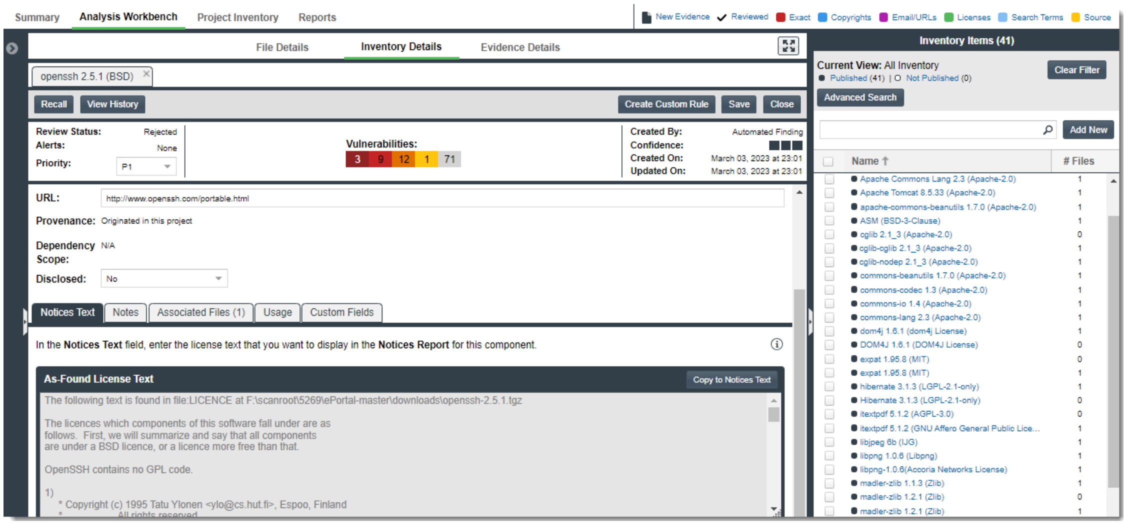
Priority: 🟥 P1 - Viral / Strong Copyleft

URL: https://spdx.org/licenses/AGPL-3.0.html

Reporting of Detected License Text through the As-Found Text Inventory Field

The **As-Found License** field (on the **Notices Text** tab for a selected inventory item in the **Analysis Workbench** and in **Project Inventory**) shows the license text or license references found in the scanned codebase.

The following shows the **Notices Text** tab in focus in the lower part of the **Analysis Workbench**.



openssh 2.5.1 (BSD)

Review Status: Rejected
Alerts: None
Priority: P1

Vulnerabilities: 3 9 12 1 71

Created By: Automated Finding
Confidence: ■■■
Created On: March 03, 2023 at 23:01
Updated On: March 03, 2023 at 23:01

URL: http://www.openssh.com/portable.html
Provenance: Originated in this project
Dependency Scope: N/A
Disclosed: No

In the **Notices Text** field, enter the license text that you want to display in the **Notices Report** for this component.

As-Found License Text

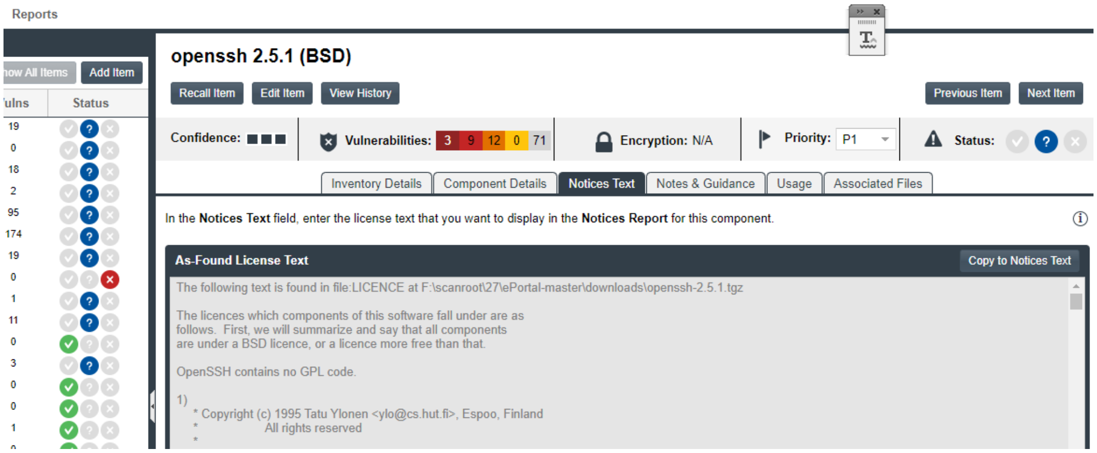
```
The following text is found in file:LICENCE at F:\scanroot\5269\ePortal-master\downloads\openssh-2.5.1.tgz

The licences which components of this software fall under are as
follows.  First, we will summarize and say that all components
are under a BSD licence, or a licence more free than that.

OpenSSH contains no GPL code.

1)
   * Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland
   *                    All rights reserved
```

Inventory Items (41)

Current View: All Inventory
● Published (41) | ○ Not Published (0)

Name	# Files
Apache Commons Lang 2.3 (Apache-2.0)	1
Apache Tomcat 8.5.33 (Apache-2.0)	1
apache-commons-beanutils 1.7.0 (Apache-2.0)	1
ASM (BSD-3-Clause)	1
cglib 2.1_3 (Apache-2.0)	0
cglib-cglib 2.1_3 (Apache-2.0)	1
cglib-nodep 2.1_3 (Apache-2.0)	1
commons-beanutils 1.7.0 (Apache-2.0)	1
commons-codec 1.3 (Apache-2.0)	1
commons-io 1.4 (Apache-2.0)	1
commons-lang 2.3 (Apache-2.0)	1
dom4j 1.6.1 (dom4j License)	1
DOM4J 1.6.1 (DOM4J License)	0
expat 1.95.8 (MIT)	0
expat 1.95.8 (MIT)	1
hibernate 3.1.3 (LGPL-2.1-only)	1
Hibernate 3.1.3 (LGPL-2.1-only)	0
itextpdf 5.1.2 (AGPL-3.0)	0
itextpdf 5.1.2 (GNU Affero General Public Lice...	1
libjpeg 6b (IJG)	1
libpng 1.0.6 (Libpng)	1
libpng-1.0.6(Accoria Networks License)	1
madler-zlib 1.1.3 (Zlib)	1
madler-zlib 1.2.1 (Zlib)	0
madler-zlib 1.2.1 (Zlib)	1

This shows the **Notices Text** tab in **Project Inventory**.



openssh 2.5.1 (BSD)

Confidence: ■■■ Vulnerabilities: 3 9 12 0 71 Encryption: N/A Priority: P1 Status:

In the **Notices Text** field, enter the license text that you want to display in the **Notices Report** for this component.

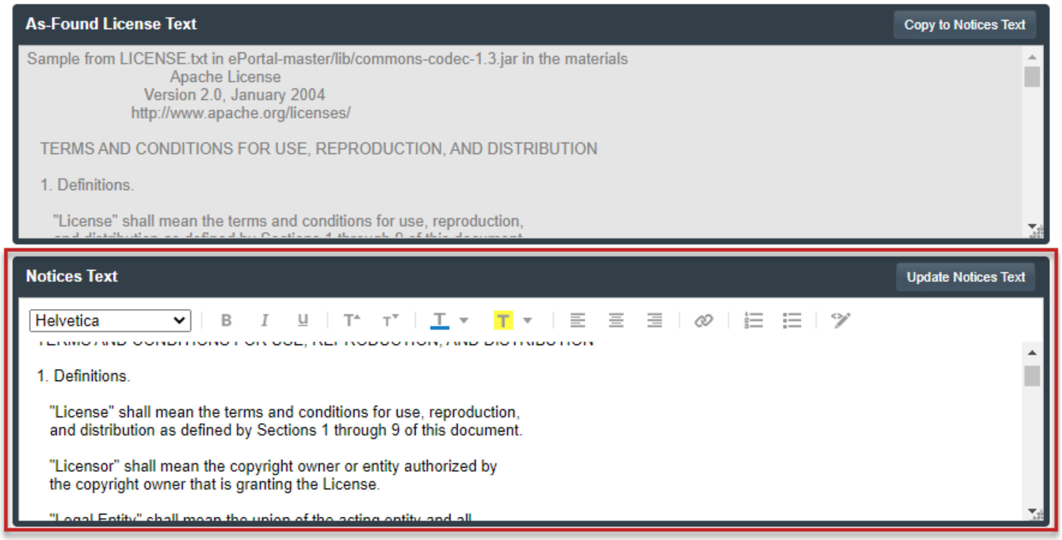
As-Found License Text

```
The following text is found in file:LICENCE at F:\scanroot\27\ePortal-master\downloads\openssh-2.5.1.tgz

The licences which components of this software fall under are as
follows.  First, we will summarize and say that all components
are under a BSD licence, or a licence more free than that.

OpenSSH contains no GPL code.

1)
   * Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland
   *                    All rights reserved
   *
```

The **As-Found License Text** content cannot be edited, but you can copy it to the **Notices Text** field (also on the **Notices Text** tab) if you need to modify it. Ultimately, the content (or lack of content) in the **Notices Text** field determines what license content is pulled into the Notices report for the specific inventory item. For more information, see Notices Text Field.

Notices Text Field

The **Notices Text** field is used to finalize the license text for use in the Notices report. (This field is located under the **As-Found License Text** field on the **Notices Text** tab for a selected inventory item in the **Analysis Workbench** and in **Project Inventory**.) For example, you can copy the contents of the **As-Found License Text** field to this field and modify the text as needed. Alternatively, you can pull in the current license text from the Code Insight Data Library (and modify it as needed); or you can simply provide your own Notices content in this field.



Sample from LICENSE.txt in ePortal-master/lib/commons-codec-1.3.jar in the materials
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.

1. Definitions.

"License" shall mean the terms and conditions for use, reproduction,
and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by
the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all

When the Notices report is run, the content of the **Notices Text** field item is pulled into the report if this field contains information. If the **Notices Text** field is empty, the content of the **As-Found License Text** field is used in the report. If both fields are empty, the report uses the license content from Code Insight Data Library.

For more information about finalizing license text for the Notices report, see Finalizing the Notices Text for the Notices Report.

Auditing Scan Results in the Analysis Workbench

After a scan has been performed on a project codebase uploaded to a Scan Server, an Analyst for the project analyzes, or *audits*, the results of the scan in the **Analysis Workbench**. The following sections describe the role of the Analyst and the tasks involved with a scan audit:

- Role of an Analyst
- Opening the Analysis Workbench
- Searching the Codebase Files
- Examining and Managing Open-Source Evidence for a Given File
- Viewing a Summary of Evidence Detected Across the Codebase
- Managing the Codebase Files
- Managing Inventory in the Analysis Workbench

Role of an Analyst

The role of a project Analyst in Code Insight is to transform the evidence uncovered by the Scan Server into an *inventory item*. Analysts create inventory items that associate files in your codebase to open-source and third-party projects, called *components* in Code Insight. For example, Analysts might evaluate files with a copyright of “Copyright (c) 2015 to 2021 Mark Smith” and a license match to the license used by the “zlib” component. The Analyst would then associate these files with an inventory item for the “zlib” open-source component and mark the files as *reviewed* to register progress.

The Analyst will evaluate all of the evidence within a codebase, create inventory items where appropriate, mark the analyzed files as reviewed, and finally *publish* them. The remaining sections in Auditing Scan Results in the Analysis Workbench describe these tasks.

Once published, the inventory will be available for reporting and review by Legal, Security, and Development teams, as described in Reviewing Published Inventory for a Project. The ultimate goal of both the audit and the review/remediation processes is to produce a complete and accurate inventory of open-source and third-party code within your products—sometimes referred to as a Bill of Materials (BOM).

Refer to the Code Insight User Roles and Permissions section for more information about Analyst role required to access the **Analysis Workbench** and to analyze and act on scan results.

Opening the Analysis Workbench

Use the following procedure to open the **Analysis Workbench**.

Task ***To open the Analysis Workbench, do the following:***

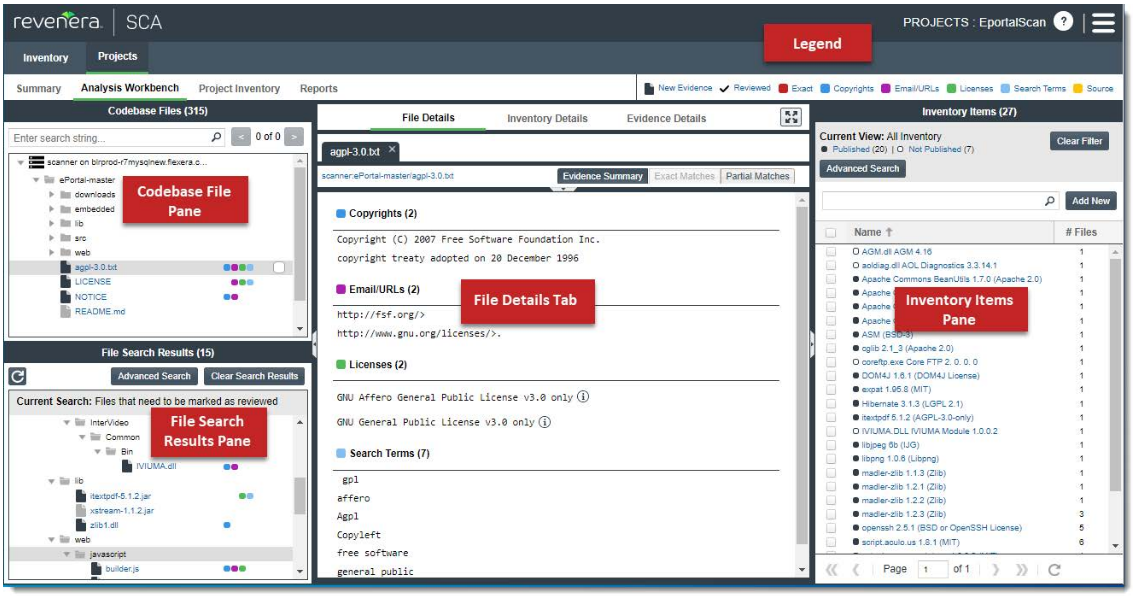
1. Open a project from the **Projects** view. (For instructions, see Opening a Project.)

   The project opens to either its **Project Inventory** or **Summary** tab. If you have Analyst permissions, the **Analysis Workbench** tab is available next to either or both of these tabs.

2. Open the **Analysis Workbench** tab to begin the analysis process. See the next section, The Analysis Workbench Layout.

The Analysis Workbench Layout

The following is a view of the Code Insight **Analysis Workbench**, showing the various areas of the page.



After you click **Analysis Workbench**, the following information appears in various panes and tabs in the workbench:

- Codebase Files Pane (Top Left Pane)
- File Search Results Pane (Bottom Left Pane)
- Files Details Tab (Middle Pane)
- Inventory Details Tab (Middle Pane)
- Evidence Details Tab (Middle Pane)
- Inventory Items Pane
- Legend

Codebase Files Pane (Top Left Pane)

This pane lets you browse a codebase tree listing the project's scanned files that you uploaded or synchronized to the Scan Server or that were scanned remotely by a Code Insight scan-agent plugin. The codebase tree provides the following:

- Scan Server's Base Node
- Scan Agent's Base Node
- Types of Evidence Found in a File
- Review Indicator
- Access to File Details

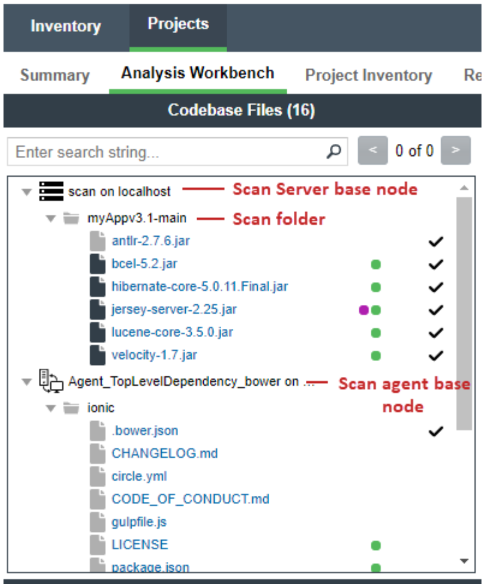
Scan Server's Base Node

The project's codebases scanned by a Scan Server are listed under the *Scan Server's base node*, which is identified both by the Scan Server's unique alias and by the name of instance on which the server is hosted. This base node has the format <scanServerAlias> on <scanServerHost > (such as **Scanner03 on localhost**).

Scan Agent’s Base Node

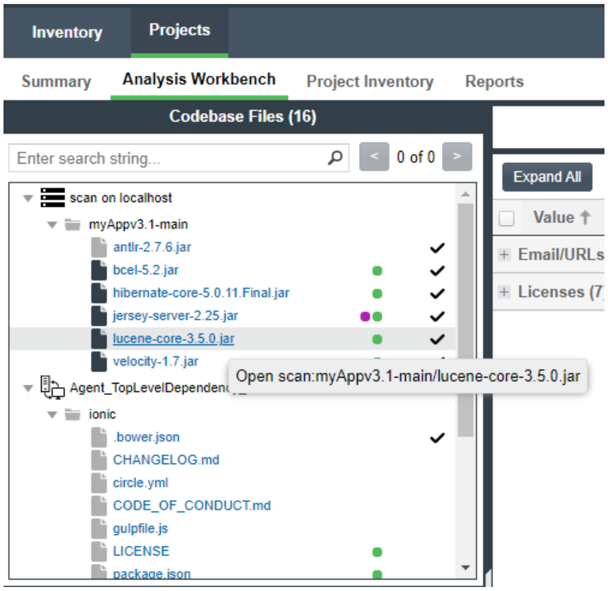
The remote codebases scanned by a scan-agent plugin are listed under the *scan agent’s base node*, identified both by a unique alias for the scan agent and by the instance on which the agent is hosted. This base node has the format <scanAgentAlias> on <scanAgentHost> (such as **EP_Remote on BLR-DT-100555.ECompany.com**).

The unique, user-defined alias provided during scanner setup (for either a Scan Server or a remote scan agent) is a descriptive name used to represent the scan-root container for the scanner. The base node then—as a combination of both the alias and the host instance name—provides a more meaningful representation for the absolute scan-root path for the scanner. (The actual absolute scan-root path for each scanner associated with the project is available on the project’s **Summary** tab.)



When the **Analysis Workbench** for a given project first opens, the codebase tree expands only the first base node. Under that node, only the first top-level (scan) folder is expanded, showing the first-level codebase folders and files directly under that scan folder. These first-level folders as well as all other base nodes and folders are collapsed and need to be expanded manually as needed.

When you hover over a file name in the codebase tree, the name is shown in an <alias>:<relativeFilePath> format, where <alias> is the alias of the Scan Server or scan agent and <relativeFilePath> is the file path relative to the absolute scan-root path on host instance. (See the following example where, when a user hovers over the codebase file **lucene-core-3.5.0.jar**, located directly under the scan folder **myApp3.1-main**, the file name is shown as **scan:myApp3.1-main/lucene-core-3.5.0.jar**.)



Types of Evidence Found in a File

The types of evidence found in a given file show as color-coded icons to the right of the file name. The color coding is identified in the legend located in the right side of the **Analysis Workbench** header. (See Legend.)

Note the following:

- For files scanned by a Code Insight scan-agent plugin on a remote system, only license evidence is currently reported in Code Insight (indicated by the green icon 🟩 for those files that contain license evidence). No other type of evidence is reported for such files at this time.
- Some source files contain indications that they are data files, generated code, or common code that is widely used in many open source projects. In those cases, Code Insight records the fact that source matches exist but does not store all of the source-matched data. These files are indicated in the Analysis Workbench with an icon (☒).

Review Indicator

A checkmark at the end of file row indicates that the file has been reviewed.

Access to File Details

When you click a file, its metadata, content, and evidence is shown in the middle pane (**File Details** pane).

File Search Results Pane (Bottom Left Pane)

This pane lists the results of file searches against the codebase. The results are shown in a codebase tree that has the same format, properties, and behavior as the codebase tree in the **Codebase Files** pane (see Codebase Files Pane (Top Left Pane)). For more information about file searches, see Searching the Codebase Files.

Files Details Tab (Middle Pane)

This tab lists a summary of evidence found in the file currently selected in the codebase tree. For non-binary files, the tab can show the actual file content, including evidence highlighted in color. For a binary file, the tab can show strings of possible third-party evidence. From this tab, you can research the source of the possible third-party code and, if necessary, ultimately create an inventory item explaining the scan findings.

Note ▪ *The middle pane toggles between the **File Details** tab, the **Inventory Details** tab, and the **Evidence Details** tab, depending on whether an codebase file or an inventory item is selected or if you explicitly click one of these tabs.*

Inventory Details Tab (Middle Pane)

This tab shows information about the inventory item selected in the **Inventory Items** pane (see Inventory Items Pane). This information includes component and license details, inventory priority, inventory confidence level, as well as auditing and guidance notes, associated files, and the third-party notices associated with the inventory item. From this tab, you can edit the inventory item, recall it from its published state, and create a custom rule based on your findings that future scans on other projects can use to automatically generate inventory. For a description of the fields in this pane, see Inventory Details Tab in the Analysis Workbench.

Evidence Details Tab (Middle Pane)

This tab displays all instances of copyright, license, email, URL, and search-string evidence uncovered by the scan across all files in the project. (You must explicitly click the **Evidence Details** tab to see this information.) The list of evidence instances is organized by evidence type and is sortable.

Note ▪ *For files scanned by a Code Insight scan-agent plugin on a remote system, only license evidence is currently reported in Code Insight. The **Evidence Details** pane visibly shows any license evidence found in remotely scanned files.*

To filter the files in the **File Search Results** pane to focus attention on those files containing particular evidence (such as specific copyright), select one or more evidence instances (rows) on the **Evidence Details** tab, and click **Search Files**. The **File Search Results** pane shows the files containing that evidence. For more information, see Filtering the Codebase by a One or More Specific Instances of Evidence.

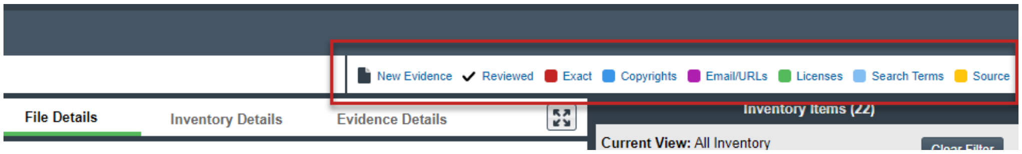
Inventory Items Pane

(Right pane) This pane lists all the inventory currently identified in the codebase. You can sort the inventory items in ascending or descending order by name (**Name** column) or the by the number of codebase files (**# Files** column) associated with each inventory item.

Click any inventory item in the list to display its details on the **Inventory Details** tab. For a description of the fields on this pane, see Analysis Workbench.

Legend

(Right side of the Analysis Workbench header) This legend provides a color key for the various types of evidence and for file-review status referenced in the **Analysis Workbench**. The **Legend** is interactive. You can click it to filter what appears in the **File Search Results** pane.For more details, see Using the Filter Legend Options to Filter the Codebase.



Searching the Codebase Files

Code Insight offers various methods for searching the list of scanned codebase file:

- Searching for Codebase Files Based on Name
- Searching for Codebase Files Based on Search Criteria
- Creating and Editing File Searches
- Using the Filter Legend Options to Filter the Codebase
- Filtering the Codebase by a One or More Specific Instances of Evidence

Searching for Codebase Files Based on Name

You can perform a file search to find files based on the file name to concentrate the **Analysis Workbench** display on files of interest for conducting your analysis of the scan results. The following limitations apply:

- There is no support for wildcard specifications. The comparison is a case-insensitive filename containing the complete search string.
- Only the first 1,000 matching files are returned by the file search.

This specific search highlights the search results in the **Codebase Files** pane, unlike the other file searches (described in the sections that follow) that display the search results in the **File Search Results** pane.

Task ***To perform a file search based on name, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the search text box in the **Codebase Files** pane, enter the partial or full name of the file or folder that you want to search and press **Enter**. You must type at least three characters to initiate the filename search. The text box is highlighted with a red border if you enter fewer than three characters, and an error message is shown in a tooltip.

   When a match is found, the tree in the **Codebase Files** pane is expanded as much as necessary to highlight the matching file. The file details are not open until you click on the file in the tree.

3. Select the **Next Match** (**>**) and **Previous Match** (**<**) buttons next to the search string box to navigate the results of the search.

    - **Files**—If the Previous or Next match button reaches a file, that file will be highlighted in the codebase tree, and the search term will be highlighted in yellow.
    - **Folders**— If the Previous or Next match button reaches a folder, that folder will be highlighted in the codebase tree and the search term will be highlighted in yellow. The folder will also be automatically expanded one level so that you can see its child items.

    The counter between the buttons indicates the total number of matches and the current match number.

4. (Optional) Click the name of a file to display its contents in the **File Details** tab.
5. (Optional) Click the **X** to clear the search string.

Searching for Codebase Files Based on Search Criteria

You can perform a file search to find files based on the file name to concentrate the **Analysis Workbench** display on files of interest for conducting your analysis of scan results.

Task ***To perform a file search by criteria, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. Click **Advanced Search** in the **File Search Results** pane. The **Advanced File Search** dialog appears.
3. Select an existing search filter or add a new one:

    - To select an existing search filter, click the name of the filter; and then click **Search** to begin the search with the selected filter. (For a list of built-in search filters that Code Insight provides, see Codebase Filters Provided with Code Insight.)
    - To create and run a new search filter, see Creating and Editing File Searches.

    Results are listed in the **File Search Results** pane.

Codebase Filters Provided with Code Insight

For your convenience, Code Insight provides the following built-in search filters by which to search codebases. You can copy these filters to create custom filters; or, if necessary, you can edit or delete them, as described in Creating and Editing File Searches.

Table 2-5 ▪ Codebase Filters Provided by Code Insight

Code Insight Predefined Filter	Filters to...
Files that require additional analysis	Files that contain evidence but are *not* reviewed or associated with inventory.
Files with new evidence requiring another look	Files that contain new evidence but are already marked as reviewed or are currently associated with inventory.

Table 2-5 ▪ Codebase Filters Provided by Code Insight

Code Insight Predefined Filter	Filters to...
Files that need to be marked as reviewed	Files that are associated with inventory but are not marked as reviewed.
Files with possible commercial content	Files containing copyrights that include commercial names.
Files not in inventory	File that are not associated with any inventory item.

Creating and Editing File Searches

You can supplement the built-in filters with custom filters to focus on scan data that are important to you. For example, you can create a new file search from scratch or from a copy of an existing search. You can also edit searches.

Refer to the following topics:

- Creating a New File Search
- Editing a File Search
- Copying a File Search
- Deleting a File Search
- Available Search Criteria for Building Codebase Filters

Any new searches you create or any copies or edits you make are available to all users in your Code Insight system. Likewise, any searches that you delete are no longer available to users in the system.

Creating a New File Search

Use either procedure to create a new file search:

- Create a File Search from Scratch
- Create a File Search from a Copy

Create a File Search from Scratch

Use this procedure to create a file search from scratch.

Task ***To create a new search from scratch, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **File Search Results** pane, click **Advanced Search**. The **Advanced File Search** dialog appears.
3. Click **Add New**. The **Create Filter** dialog appears.

4. In the **Name** field, type a name for the search.

5. (Optional) In the **Description** field, type a description of the search. For example, type text that explains what the filter will search for.

6. Use this procedure to enter values in the **Criteria** fields. (For details about the available criteria, see Available Search Criteria for Building Codebase Filters.)

    a. Select a criterion from the drop-down **Select Search Field** menu.

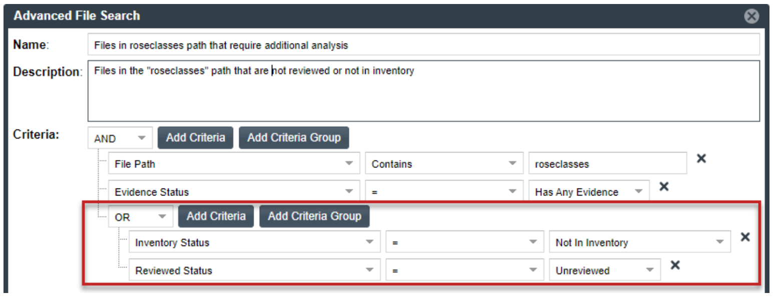
    Note *▪ When creating a new filter, consider that a Code Insight scan-agent plugin on a remote system currently reports only license evidence for its scanned files. The fields applicable for searching such files are limited to the following:* ***File Size****,* ***File Path****,* ***File Digest****,* ***Review Status****,* ***Inventory Status, Evidence status, Has license matches, Does not have license matches,*** *and* ***License****.*

    b. If applicable, select a search operation (for example, **Contains** or **=**) and provide a search string or value.

    c. To add another criterion, click **Add Criteria**, select a Boolean value to define how the criteria is applied, and repeat the previous steps to define the criterion. Repeat for each criterion added.

    d. To add a group of criteria that serves as a criterion, click **Add Criteria Group**, and repeat steps a through c to create the group.

    The following shows an example of a criteria group.



Advanced File Search

Name: Files in roseclasses path that require additional analysis

Description: Files in the "roseclasses" path that are not reviewed or not in inventory

Criteria: AND

File Path | Contains | roseclasses

Evidence Status | = | Has Any Evidence

OR

Inventory Status | = | Not In Inventory

Reviewed Status | = | Unreviewed

7. Determine how you want to proceed:

    - **Save**—Save your search but do not execute it.
    - **Save and Search**—Save your search filter and then execute it.
    - **Search without Saving**—Execute the search without saving it.
    - **Cancel**—Do not execute the search or save it.

Create a File Search from a Copy

Use this procedure to create a file search from a copy of an existing search. Using a copy keeps the existing search in tact and provides a template for creating the new one.

Task ***To create a search from a copy of an existing one, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **File Search Results** pane, click **Advanced Search**. The **Advanced File Search** dialog appears.
3. Locate the search you want to copy, and click 🗋 in its entry. A new file search is created with “Copy of...” in its title.
4. In the entry for the copy, click ✎ to open the filter properties.
5. In the **Name** field, type a new name for the search.
6. Modify the filter criteria as needed and save the changes. See Create a File Search from Scratch for any additional instructions.

Editing a File Search

Use these instructions to edit an existing file search.

Task ***To edit a file search, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **File Search Results** pane, click **Advanced Search**. The **Advanced File Search** dialog appears.
3. In the entry for the file search you want to edit, click ✎ to open the filter properties.
4. Modify the filter criteria as needed and save the changes. See Create a File Search from Scratch for any additional instructions.

Copying a File Search

Use these instructions to create a copy an existing file search (as a backup or a basis for creating a new search, for example).

Task ***To make a copy of an existing file search, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **File Search Results** pane, click **Advanced Search**. The **Advanced File Search** dialog appears.
3. Locate the search you want to copy, and click 🗋 in its entry. A new file search entry is created with “Copy of...” in its title.

Deleting a File Search

Use these instructions to delete a file search. When you delete the search, it is removed from the system and no longer available to users.

Task ***To delete an existing file search, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **File Search Results** pane, click **Advanced Search**. The **Advanced File Search** dialog appears.
3. Locate the search you want to delete, and click ✖ in its entry.

   A message is displayed to confirm that you want to delete the search.
4. Click **Yes** to remove the file search from the system.

Available Search Criteria for Building Codebase Filters

Code Insight provides the following search criteria on which to build codebase search filters.

Note ▪ *A Code Insight scan-agent plugin on a remote system currently reports only license evidence for its scanned files. The fields applicable for searching these scanned files are limited the following:* ***File Size****,* ***File Path****,* ***File Digest****,* ***Review Status****,* ***Inventory Status, Evidence status, Has license matches, Does not have license matches, License****.*

Table 2-6 ▪ Available Criteria for Building Codebase Search Filters

Criterion Type	Available Criterion	Operation	Criterion Value	Criterion will filter to those files with...
File Properties	File Size (in KB) *	Select **<** or **>**.	Enter or select the file-size.	A size is less than or greater than the specified size.
	File Path*	Select **Contains**, **Ends With**, or **Doesn’t Contain**.	Enter the file-path string or partial string.	A path containing a match to the specified string.
	Reviewed Status *	Only **=** is available.	Select **Reviewed** or **Unreviewed**.	The selected review status.
	File Digest *	Only **=** is available.	Enter the file’s MD5 value.	The exact MD5 specified.
	Evidence Status	Only **=** is available.	Select **Has Evidence**, **Has New Evidence** (since last scan), or **Has No Evidence**.	The evidence status specified.
	Inventory Status *	Only **=** is available.	Select one status: ● **In Inventory**—Files associated with inventory. ● **Not in inventory**—Files not associated with inventory. ● **Low Confidence File Inventory**—Files associated with low-confidence inventory. See Inventory Confidence for details.	The selected inventory status.

Table 2-6 ▪ Available Criteria for Building Codebase Search Filters (cont.)

Criterion Type	Available Criterion	Operation	Criterion Value	Criterion will filter to those files with...
	Scan Status	Only **=** is available.	Select one: ● **Successfully Scanned**—Files that were successfully scanned in the most recent scan. Conversely, this criterion is helpful in determining which files were *not* scanned in certain situations (for example, if you were forced to stop the scan before it finished or if the Scan Server crashed). ● **Skipped Source Matching**—Files that were ignored during source-code matching.	The selected scan status.
File Evidence—Source Matches	Has Source Matches	—	—	Source-code snippets that match snippets of open-source or third-party files stored in the Code Insight Compliance Library.
	Does Not Have Source Matches	—	—	No evidence of such source-code snippets.
File Evidence—Search Term Matches	Has Search Term Matches	—	—	The specified search-term string. (Search terms, defined in the scan profile, are used to identify open-source or third-party evidence in codebase files.)
	Does Not Have Search Term Matches	—	—	No evidence of the specified search term.
	Search Term	Select **=** or **Contains**.	Enter the full search-term or a partial search-term string.	—

Table 2-6 ▪ Available Criteria for Building Codebase Search Filters (cont.)

Criterion Type	Available Criterion	Operation	Criterion Value	Criterion will filter to those files with...
File Evidence—License Matches	Has License Matches	—	—	Evidence of the open-source or third-party license selected for **License**.
	Does Not Have License Matches	—	—	No evidence of the selected license.
	License	Only **=** is available.	Select the open-source or third-party license by which to filter codebase files.	—
File Evidence—Exact Matches	Has Exact Matches	—	—	Entire content that exactly matches the content of open-source or third-party files stored in the Code Insight Compliance Library.
	Does Not Have Exact Matches	—	—	No exact match to the entire content of any open-source or third-party file in the Compliance Library.
File Evidence—Email/URL Matches	Has Email/URL Matches	—	—	Evidence of the open-source or third-party email address or URL specified for **Email/URL**.
	Does Not Have Email/URL Matches	—	—	No evidence of the specified email or URL.
	Email/URL	Select **=** or **Contains**.	Enter the open-source or third-party email or URL (or partial value) by which to filter codebase files.	—

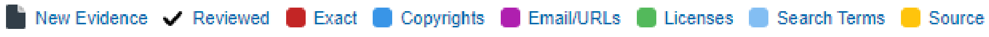
Table 2-6 ▪ Available Criteria for Building Codebase Search Filters (cont.)

Criterion Type	Available Criterion	Operation	Criterion Value	Criterion will filter to those files with...
File Evidence—Copyright Matches	Has Copyright Matches	—	—	Evidence of the copyright holder (specified for **Copyright Holder**) or copyright statement (specified for **Copyright Statement**).
	Does Not Have Copyright Matches	—	—	No evidence of the specified copyright or copyright holder.
	Copyright Holder	Select = or **Contains**.	Enter the open-source or third-party copyright holder (or partial value) by which to filter codebase files.	—
	Copyrights	Select = or **Contains**.	Enter the open-source or third-party copyright statement (or partial value) by which to filter codebase files.	—

***** Criterion currently supported for searches on scanned remote files (that is, files scanned by a Code Insight scan-agent plugin on a remote system).

Using the Filter Legend Options to Filter the Codebase

The codebase filter legend in the ribbon at the top right of the **Analysis Workbench** provides a means of filtering the codebase by evidence type or by files with a “Reviewed” status. For example, by simply clicking an icon (or its label), you can filter to all files containing copyright or email-address evidence or that are exact matches to third-party files.



The following describes the filter legend options:

Table 2-7 ▪ Filter Legend

Icon	Label	Filters to files...
🗎	**New Evidence**	...containing any evidence that the previous scan did *not* detect but that the most recent scan *did*.
✓	**Reviewed**	...marked as “reviewed”.

Table 2-7 ▪ Filter Legend (cont.)

Icon	Label	Filters to files...
🟥	**Exact**	...that are exact matches to known third-party files.
🟦	**Copyrights**	...containing copyright information.
🟪	**Email/URLS**	...containing email addresses or URLs.
🟩	**Licenses**	...containing license information.
🟦	**Search Terms**	...containing search terms defined in the scan profile.
🟨	**Source**	...containing code-snippet matches (fingerprints) of known third-party code.

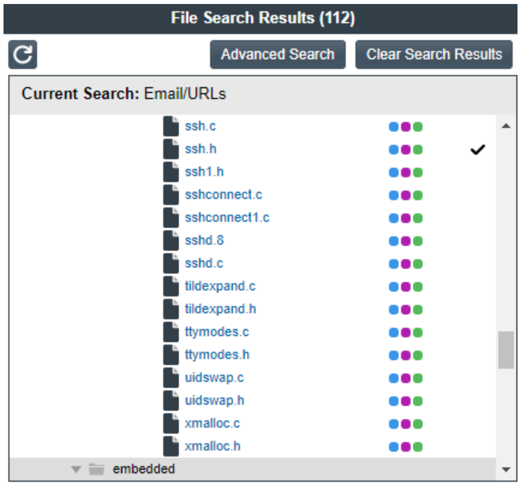
The color theme used for evidence types in this legend is also used to indicate the types of evidence found in a given file in the **Codebase Files** and **File Search Results** lists (see the following procedure) and on the **File Details** tab (see Examining and Managing Open-Source Evidence for a Given File).

***Note ▪** For files scanned by a Code Insight scan-agent plugin on a remote system, only license evidence is currently reported. Therefore, the **Licenses** filter is the only applicable filter for locating such files.*

Task ***To filter the codebase using the filter legend options, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. Click the option in the filter legend to identify how you want to filter the codebase files. Results are listed in the **File Search Results** pane.
3. Navigate to the **File Search Results** pane, which now shows a codebase tree containing the files that meet your criterion.
4. Drill down in the codebase tree to view the files.

   Note that each file entry is flagged not only with a icon that matches the filter-legend criterion you selected but also with icons representing all evidence or attributes associated with this file.



5. Select a file a from the filtered codebase list.

   Refer to these later sections for different ways to analyze and act on third-party evidence discovered in the files:

   - Examining and Managing Open-Source Evidence for a Given File
   - Examining and Managing Open-Source Evidence for a Given File
   - Managing the Codebase Files

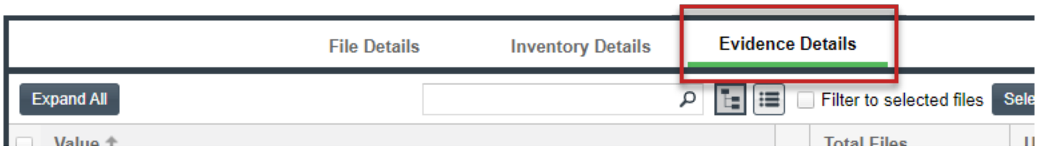
Filtering the Codebase by a One or More Specific Instances of Evidence

You can filter the codebase to show only those files that contain a one or more *specific* instances of copyright, email, URL, license, or search-term evidence. To do so, use the **Evidence Details** tab in the **Analysis Workbench** to set up a search of these instances in the codebase. This tab lists the actual instances of the various types of evidence found in the codebase and shows the total number of files that contain each instance.

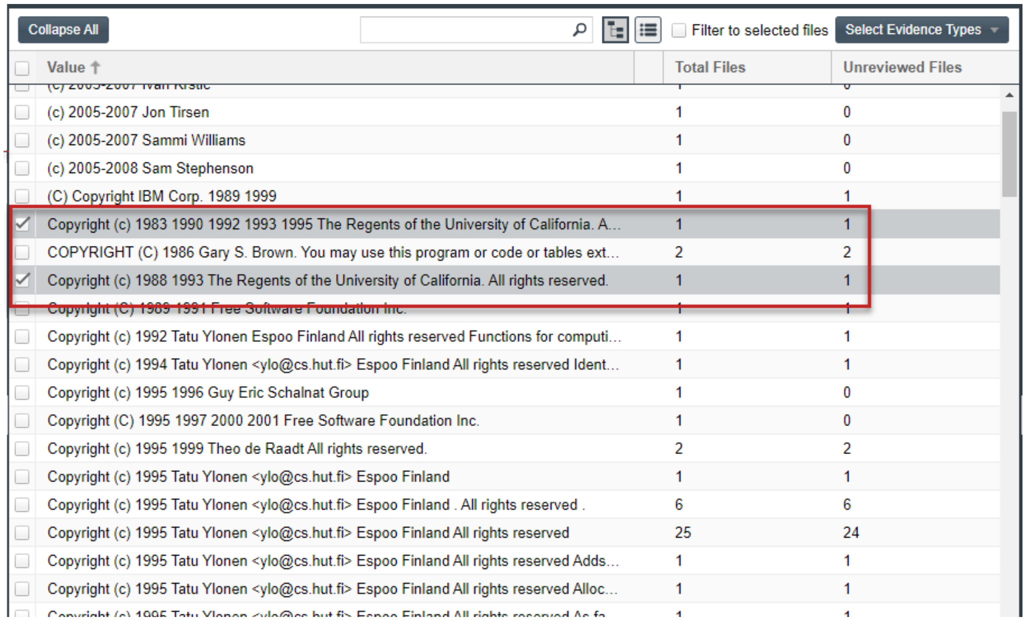
For example, suppose the **Evidence Details** tab indicates that a certain number of codebase files contain evidence of specific Twitter copyrights, and you want to know which codebase files contain this evidence. From the **Evidence Details** tab, you select the evidence instances—that is, the specific Twitter copyrights found in the codebase—by which to filter the codebase. When the search is complete, the files containing any of these copyrights are listed in the **File Search Results** pane.

Task ***To search the codebase for the files containing specific evidence instances, do the following.***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **Analysis Workbench**, click **Evidence Details** in the middle pane to open the tab.



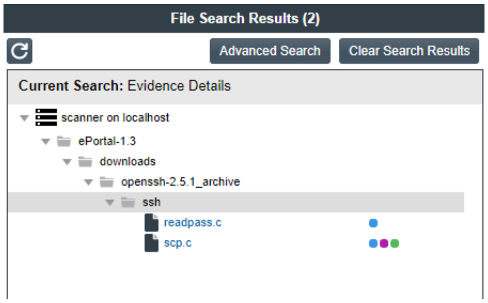
3. Select the checkbox to the left of one or more evidence instances in the list by which you want to search the codebase. (When you select multiple evidence instances, the search uses OR logic to obtain the results.) For example, you might want search for those files containing any of the two selected Twitter copyrights. Note that these copyrights are found in a total of three files.



□	Value ↑	Total Files	Unreviewed Files
□	(c) 2005-2007 Ivan Krstic	1	0
□	(c) 2005-2007 Jon Tirsen	1	0
□	(c) 2005-2007 Sammi Williams	1	0
□	(c) 2005-2008 Sam Stephenson	1	0
□	(C) Copyright IBM Corp. 1989 1999	1	1
☑	Copyright (c) 1983 1990 1992 1993 1995 The Regents of the University of California. A...	1	1
□	COPYRIGHT (C) 1986 Gary S. Brown. You may use this program or code or tables ext...	2	2
☑	Copyright (c) 1988 1993 The Regents of the University of California. All rights reserved.	1	1
□	Copyright (C) 1989 1991 Free Software Foundation Inc.	1	1
□	Copyright (c) 1992 Tatu Ylonen Espoo Finland All rights reserved Functions for computi...	1	1
□	Copyright (c) 1994 Tatu Ylonen <ylo@cs.hut.fi> Espoo Finland All rights reserved Ident...	1	1
□	Copyright (c) 1995 1996 Guy Eric Schalnat Group	1	0
□	Copyright (C) 1995 1997 2000 2001 Free Software Foundation Inc.	1	0
□	Copyright (c) 1995 1999 Theo de Raadt All rights reserved.	2	2
□	Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi> Espoo Finland	1	1
□	Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi> Espoo Finland . All rights reserved .	6	6
□	Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi> Espoo Finland All rights reserved	25	24
□	Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi> Espoo Finland All rights reserved Adds...	1	1
□	Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi> Espoo Finland All rights reserved Alloc...	1	1
□	Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi> Espoo Finland All rights reserved As fa...	1	1

4. Click **Search Files** in the lower right of the tab.

   The files containing one or more of the selected evidence instances are listed in the **Files Search Results** pane. In the following example, the three files containing the Twitter copyrights are listed.



Current Search: Evidence Details

- scanner on localhost
  - ePortal-1.3
    - downloads
      - openssh-2.5.1_archive
        - ssh
          - readpass.c ●
          - scp.c ●●●

Examining and Managing Open-Source Evidence for a Given File

The **File Details** tab provides metadata about a selected codebase file and detailed information about open-source and other third-party evidence detected in the file:

- File Metadata
- Examining a Codebase File Exactly Matching an Open-Source File
- Examining a Codebase File Containing Partial Matches to Open-Source File Content

File Metadata

The **File Details** tab includes a expandable header that lists metadata about the selected codebase file, as well as the three tabs—**Evidence**, **Exact Matches**, and **Partial Matches**—available to examine the file's open-source or third-party evidence. (These tabs are described in the procedural sections that follow.)

By default, the header is collapsed.

Table 2-8 ▪ Codebase File Metadata

File Property	Description
<alias>:<relativeFile Path> name	(Displays only when the header is collapsed) The file identified in an <alias>:<relativeFilePath> format, where <alias> and <relativeFilePath> are defined below for **Alias** and **Path**. As an example, if you selected the codebase file **agpl-3.0.txt**, located directly under the scan folder **ePortal-1.3**, which in turn is directly under the scan-root path for the scanner whose alias is "EP_remote", the file name shown here would be **EP_remote:ePortal-1.3/agpl-3.0.txt**.
Name	The name of the selected file.
Path	The file's path relative to the scan-root path on instance hosting the scanner (Scan Server or remote scan agent).
Alias	The unique name defined during scanner setup to represent the scanner's scan-root path containing the codebase where the file is located. The alias provides a more descriptive name for the scan-root path.
Digest	The MD5 value for the file. ***Note ▪*** *If SHA-1 support is enabled for your Code Insight instance, you can view the file's SHA-1 digest by using the Code Insight **Get details of a file by ID** or **Fetch all scanned files for a project** REST API. For more information about these APIs, refer to the Code Insight Swagger documentation, available from the **Help > REST API Guide** option on the* ☰ menu. *The SHA-1 digest does not display along with the MD5 digest in this metadata section. (To determine whether SHA-1 support is enabled, contact your Code Insight database or system administrator.)*
Modified	The value of the file's "Date Modified" property at the time of the most recent scan. This value shows the date when the file was last modified in the file system.
Type	The file format type of the scanned file, such FILE or ARCHIVE_BINARY.
File Size	The size of the file.
Lines of Code	The number of lines of code in the file.

Table 2-8 ▪ Codebase File Metadata (cont.)

File Property	Description
Reviewed	The Yes or No indicator showing whether the file has been reviewed.

Examining a Codebase File Exactly Matching an Open-Source File

If your project is configured for exact-match scanning, the scan will identify files in the codebase whose content exactly matches files in the Compliance Library (CL). Follow these steps to examine a scanned codebase file whose content exactly matches one or more open-source or other third-party files (called *remote files*) in the CL. The **Exact Matches** tab for a given codebase file shows the matching remote files, along with the open-source or third-party component versions and licenses associated with each.

Note *▪ By default, Code Insight does not perform source-code matching on files that are exact matches to CL files. However, you can enable your project scan to force source-code matching on files that are also exact matches. See Updating Scan Settings for a Project. For information about the results of source-code matching, see Examining Evidence of Open-Source Code in a Given Non-Binary File. Currently, exact matching is not available for files that are scanned by a scan agent plugin.*

Task ***To examine a codebase file that exactly matches one or more remote files, do the following:***

1. Ensure that you have run a scan with the **Comprehensive Scan Profile** selected for the desired project (or a custom scan profile with the Exact Matches feature enabled). For more information, see Updating Scan Settings for a Project.
2. Open the **Analysis Workbench** for the project. (For instructions, see Opening the Analysis Workbench.)
3. Click the **Exact** link in the legend at the top right of the page to find all files with exact matches (see Using the Filter Legend Options to Filter the Codebase). Results are listed in the **File Search Results** pane.
4. Select a codebase file from the **File Search Results** list, and select the **Exact Matches** tab.

   Three Remote Files panels are displayed:

   - The information in the **Remote Files** panel on the left consists of a set of files from the open-source community that are an exact match to the scanned file. This means that the scanned file in the codebase likely originated from outside the organization, and thus its origin needs to be identified.
   - The **Components** panel lists the open-source or third-party components associated with each remote file.
   - The **Licenses** panel lists the licenses normally associated with each component.

   See the More About the “Remote Files” Panels on the Exact or Partial Matches Tabs for more information about the functionality available from the three panels.
5. Select a remote file in the **Remote Files** panel to see the associated component and license information (on the **Components** and **Licenses** panels, respectively).

6. (Optional) Associate the codebase file to an inventory item based on the open-source or third-party component associated with a matching remote file. See Adding a Codebase File to Inventory Associated with a Remote File's Open-Source Component for details.

Examining a Codebase File Containing Partial Matches to Open-Source File Content

The following sections describe how to examine a given codebase file that contains code snippets or textual-string evidence that partially matches open-source or other third-party file content that is stored in the Code Insight Data Library or in the Compliance Library (CL):

- Viewing a Summary of Open-Source Evidence in a Given File
- Viewing Details for Licenses Associated with Codebase Files
- Examining Open-Source Evidence in a Given Binary File
- Examining Evidence of Open-Source Copyrights, Email Addresses, URLs, Licenses, and Search Terms in a Given Non-Binary File
- Examining Evidence of Open-Source Code in a Given Non-Binary File
- More About the "Remote Files" Panels on the Exact or Partial Matches Tabs
- Adding a Codebase File to Inventory Associated with a Remote File's Open-Source Component

Note *▪ Currently, for files scanned by a Code Insight scan-agent plugin on a remote system, only license evidence is reported in Code Insight. The* ***Evidence Summary*** *tab (described in this section) for such a file will list any license evidence discovered in the file as part of the remote scan. However, both the* ***Exact Matches*** *and* ***Partial Matches*** *tabs (also described in this section) are disabled for the file, as exact and partial match information is available for only scans performed by a Scan Server.*

Viewing a Summary of Open-Source Evidence in a Given File

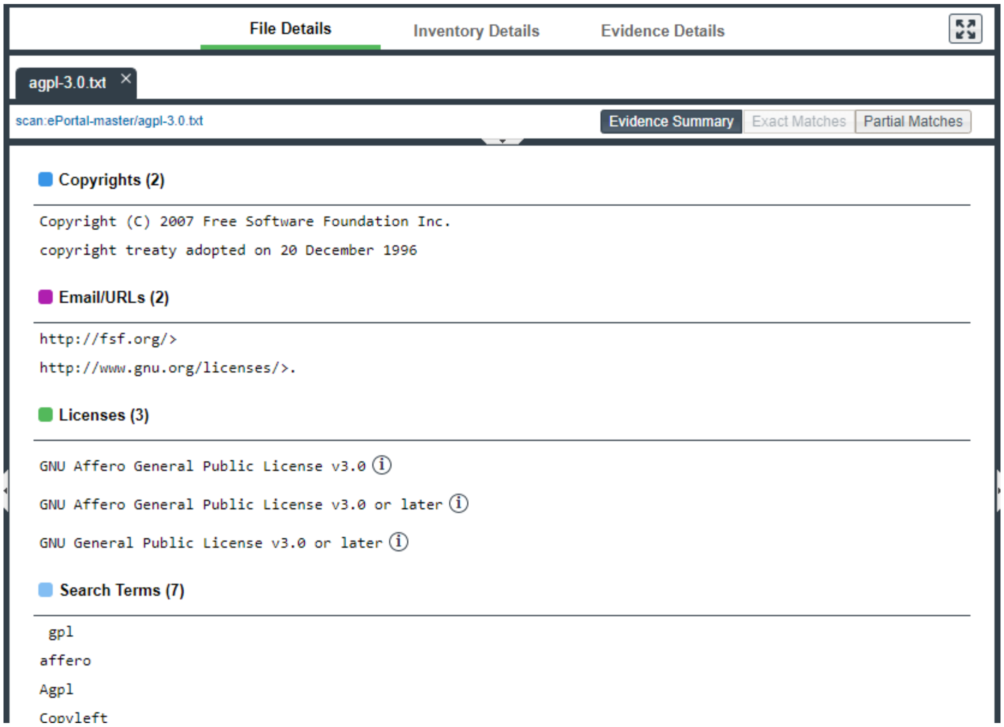
The **Evidence Summary** tab on the **File Details** tab provides a *summary* of the open-source and third-party evidence identified for a given scanned file (binary or non-binary). You can use this information to write review comments in new or existing inventory items associated with this file. The **Evidence Summary** tab lists the string-based scan results (and result totals) for each of the following evidence types:

- Copyrights
- Emails/URLs
- Licenses
- Search Terms

This listing is especially useful for examining a concise view of the open-source and third-party evidence in a binary file (such as an object file, image, executable, and so forth).

Task ***To view the evidence summary, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. Select a file in the **Codebase Files** panel.
3. Select the **File Details** tab.
4. Select the **Evidence Summary** tab. Summary information about the selected file appears in the center pane:
5. (Optional) To view additional information for the selected file, click the expand arrow (▾). The top portion of the tab expands to show details about the file.



Copyrights (2)

```
Copyright (C) 2007 Free Software Foundation Inc.
copyright treaty adopted on 20 December 1996
```

Email/URLs (2)

```
http://fsf.org/>
http://www.gnu.org/licenses/>.
```

Licenses (3)

```
GNU Affero General Public License v3.0 ⓘ
GNU Affero General Public License v3.0 or later ⓘ
GNU General Public License v3.0 or later ⓘ
```

Search Terms (7)

```
gpl
affero
Agpl
Copyleft
```

Viewing Details for Licenses Associated with Codebase Files

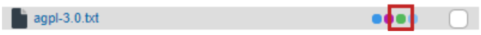
In the **Analysis Workbench**, you can view details about the licenses discovered in codebase files. When you click the information ⓘ icon next to a license reference on the **Evidence Summary** tab on the **File Details** tab, detailed information for a given license is displayed on the following tabs in the **License Details** window:

- A **General Information** tab that lists details such as the name of the license, its family, and the license priority assigned by Code Insight.
- A **License Text** tab that displays the complete license text (representing the external forge license text).

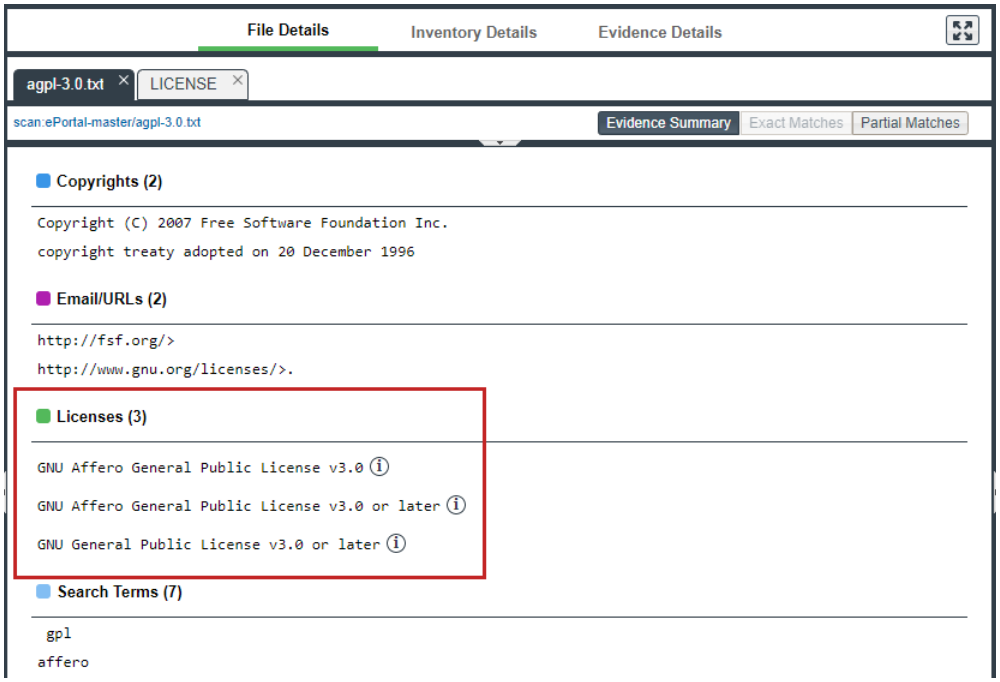
The license information shown in the **License Details** window is pulled from the Code Insight Data Library.

Task ***To view details for a license, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. (Optional) To make file selection easier, you can filter the codebase files to only those containing license evidence. See Using the Filter Legend Options to Filter the Codebase.
3. In the **Codebase Files** pane or **File Search Results** pane, select the codebase file containing the license evidence you want to review. A file with license evidence will show a green icon in its entry:



4. Locate a license reference on the **File Details** tab, as in this example of the **Evidence Summary** subtab on **File Details** tab.



🟦 Copyrights (2)

```
Copyright (C) 2007 Free Software Foundation Inc.
copyright treaty adopted on 20 December 1996
```

🟪 Email/URLs (2)

```
http://fsf.org/>
http://www.gnu.org/licenses/>.
```

🟩 Licenses (3)

```
GNU Affero General Public License v3.0 ⓘ
GNU Affero General Public License v3.0 or later ⓘ
GNU General Public License v3.0 or later ⓘ
```

🟦 Search Terms (7)

```
gpl
affero
```

Note *▪ License references are also displayed when create or edit an inventory item or perform a Lookup Component from the* ***Inventory Details*** *tab.*

5. Click the information ⓘ icon next to the license name. The **License Details** window appears with the **General Information** tab in focus.

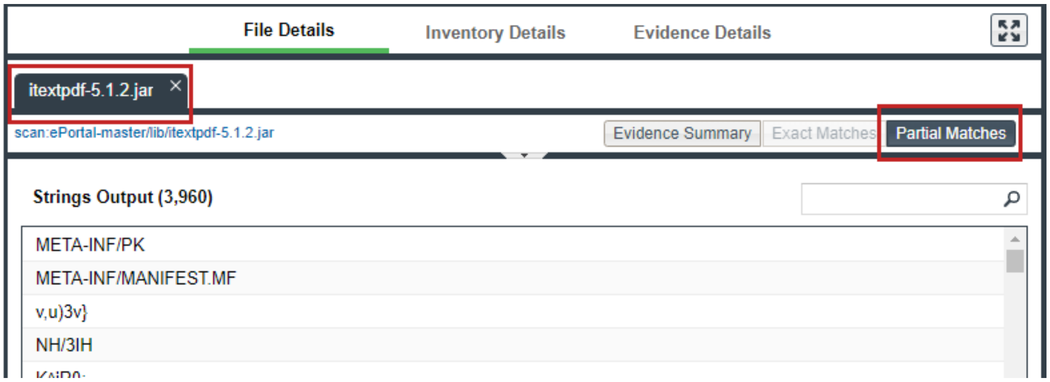
   For descriptions of these fields on this tab, see License Details Window. Also see License Priority for background on how the license priority is used.
6. Select the **License Text** tab to view the license text.
7. When you have finished examining the license details, click **Close**.

Examining Open-Source Evidence in a Given Binary File

In the content of a given binary file (such as an object file, image, executable, and so forth) in the codebase, Code Insight can locate textual strings that might be evidence of open-source or other third-party code. Each string, consisting of at least three consecutive printable characters, might part of a comment, copyright, URL, email address, and another evidence type. Code Insight simply lists these strings on the **Partial Matches** tab; it does not show them highlighted within the context of the actual binary code (unlike non-binary files, in which Code Insight highlights evidence within the actual file content).

Task ***To examine open-source evidence present in a binary file, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. Select a binary file in the **Codebase Files** panel, and click **File Details**.
3. Click **Partial Matches**. The **File Details** panel displays the strings that are output.

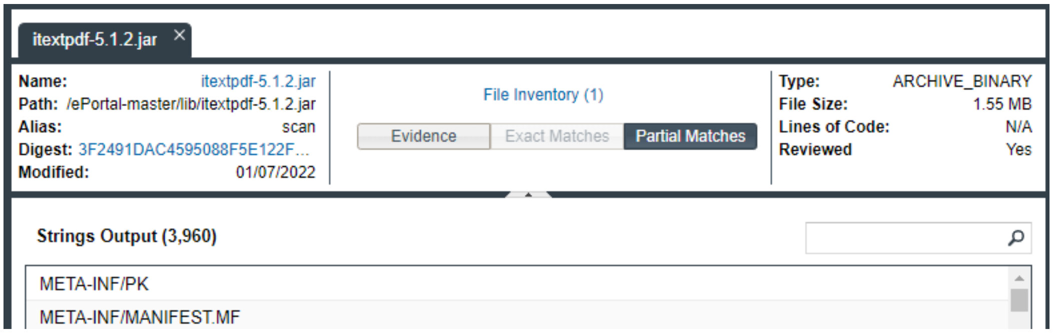


scan:ePortal-master/lib/itextpdf-5.1.2.jar

Strings Output (3,960)

META-INF/PK
META-INF/MANIFEST.MF
v,u)3v}
NH/3IH

4. (Optional) Click the expand arrow (▾), to view additional options. The top portion of the tab expands to show details about the binary file.



Name: itextpdf-5.1.2.jar
Path: /ePortal-master/lib/itextpdf-5.1.2.jar
Alias: scan
Digest: 3F2491DAC4595088F5E122F...
Modified: 01/07/2022

Type: ARCHIVE_BINARY
File Size: 1.55 MB
Lines of Code: N/A
Reviewed Yes

Strings Output (3,960)

META-INF/PK
META-INF/MANIFEST.MF

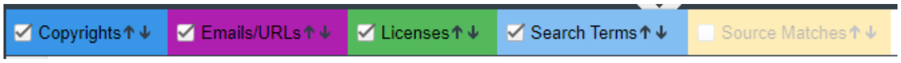
Examining Evidence of Open-Source Copyrights, Email Addresses, URLs, Licenses, and Search Terms in a Given Non-Binary File

You can view open-source or third-party evidence of copyrights, URLs, licenses, email addresses, or search terms (or a combination of these) as highlighted within the content of a given non-binary file.

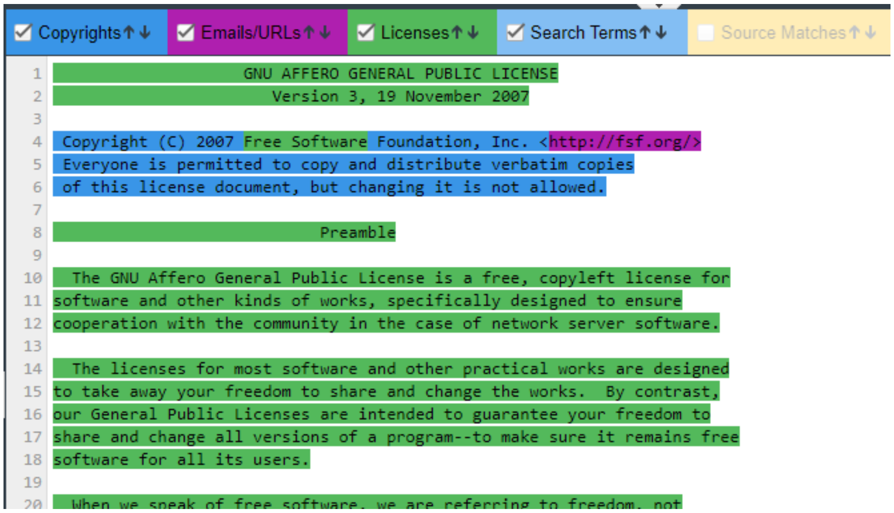
Task ***To view copyright, email, URL, and license content in a file, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **Codebase Files** pane or **File Search Results** pane, select the codebase file containing the evidence you want to review. (Optionally, to make file selection easier, you can filter the codebase files to only those containing a specific type of evidence. See Using the Filter Legend Options to Filter the Codebase.)
3. Click **File Details**.
4. Select the **Partial Matches** tab to show the contents of the file.

   Color-coded selection boxes at the top of the **Partial Matches** tab are used to indicate the type of evidence you want to highlight in the file. (Based on your screen size, labels on these selection boxes might not be visible. In this case, hover over a box to see its label.) Depending on the types of evidence existing in the file, certain selection boxes might already be selected; others might be disabled.



5. If necessary, select (or unselect) one or more selection boxes to highlight the evidence you want to view in the file. For example, the following selections will highlight instances of copyright, email, URL, license, and search-term evidence in the file:



```
                    GNU AFFERO GENERAL PUBLIC LICENSE
                       Version 3, 19 November 2007

 Copyright (C) 2007 Free Software Foundation, Inc. <http://fsf.org/>
 Everyone is permitted to copy and distribute verbatim copies
 of this license document, but changing it is not allowed.

                            Preamble

  The GNU Affero General Public License is a free, copyleft license for
software and other kinds of works, specifically designed to ensure
cooperation with the community in the case of network server software.

  The licenses for most software and other practical works are designed
to take away your freedom to share and change the works.  By contrast,
our General Public Licenses are intended to guarantee your freedom to
share and change all versions of a program--to make sure it remains free
software for all its users.

  When we speak of free software, we are referring to freedom, not
```

Considerations for Viewing License Evidence

When a given source or text file in the **Codebase Files** list contains license evidence (as indicated by a green icon 🟩 in the file entry and by the one or more licenses listed on **Evidence Summary** tab), the **Partial Matches** tab usually shows the specific evidence for each license highlighted in green within the file content. However, the following exceptions can occur:

- **Licenses are detected but not highlighted in the file**—Cases can occur during a scan when a license is discovered in the scan results and listed on the **Evidence Summary** tab, but no associated license text is highlighted on the **Partial Matches** tab. The lack of highlighting occurs because the scanner is unable to calculate the offsets for license text it cannot explicitly identify in the file.
- **All content is highlighted, including large sections of non-license-related text**—If a file containing license evidence uses a non-supported file extension, all content in the file is highlighted in green, including large sections of non-license-related information. (This is different from the scenario in which all or nearly all the content in a file *is* the license text, and thus the entire file is highlighted as such.)

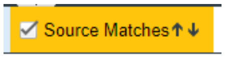
Examining Evidence of Open-Source Code in a Given Non-Binary File

If your project scan is configured to perform source-code matches, the scan will identify source-code snippets (also called *fingerprints*) in your non-binary code that match open-source and other third-party code stored in the Compliance Library (CL). The **Partial Matches** tab for a given codebase file shows the snippet matches as highlighted within the actual file content. This tab also includes a list of the CL files (called *remote files*) associated with the discovered snippets. When you select one of these remote files, the source-code highlights are refreshed to highlight only those snippets associated with the remote file.

Note ▪ *The size limit for a file that you open in the* ***Partial Matches*** *tab is 2 MB. If the file you want to inspect is too large, you can download and open it outside of Code Insight to inspect it manually for evidence.*

Task ***To view source matches, do the following:***

1. Ensure that you have run a scan with **Comprehensive Scan Profile** selected in the desired project (or a custom scan profile with the Source Code Matches feature enabled). For more information, see Updating Scan Settings for a Project.
2. Open the **Analysis Workbench** for the project. (For instructions, see Opening the Analysis Workbench.)
3. Click the **Source** link in the legend at the top right of the page to filter to all files with source-code matches (see Using the Filter Legend Options to Filter the Codebase). Results are listed in the **File Search Results** pane.
4. Click a codebase file in the list in **File Search Results**, and select the **Partial Matches** tab.
5. On the **Partial Matches** tab, click the **Source Matches** selection box at the top of the tab to enable *source code fingerprint match* results.



Three Remote Files panels are displayed:

- The information in the **Remote Files** panel on the left consists of a set of files stored in the Code Insight Data Library (and thus identified in the open-source community) that contain code snippets identical to code snippets detected in the scanned file. This matching code can indicate that the scanned file in the codebase contains content that originated from outside the organization, and its origin needs to be identified.

  ***Note** ▪ On these panels, the files in the Data Library are called "remote" to identify them separately from the from the actual codebase files to which they correspond.*

- The **Components** panel lists the open source or third-party components associated with the remote file.
- The **Licenses** panel lists the licenses normally associated with the component.

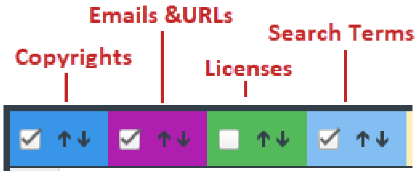
See the More About the "Remote Files" Panels on the Exact or Partial Matches Tabs for details about the functionality available from the three panels.

6. Select a remote file in the **Remote Files** panel on the left to highlight the source-code snippets in the scanned file that match those in the remote file and to view the lists of associated component and license information (on the **Components** and **Licenses** panels, respectively).

   Note that the **Remote Files** panel will additionally contain the following CodeRank™ values:

   - **CodeRank (CR%)** —A composite heuristic comprised of Coverage, Clustering, and Uniqueness values. The higher the number, the stronger the match confidence.
   - **Coverage (CV%)** —The percentage of remote-file content contained in your scanned file.
   - **Clustering (CL%)** —The density or proximity of remote-file matches within your scanned file.
   - **Uniqueness (U%)** —An indication of how often the remote-file matches detected in the scanned file occur in the Compliance Library (CL).
   - **Matches** —The number of unique matches in the scanned file.

7. To view the instances of other types of evidence (for example, copyrights, licenses, URLs, email addresses, and search terms) in the codebase file, click the appropriate color-coded selection boxes at the top of the **Partial Matches** tab. (The following shows selections boxes that have been contracted because of a reduced screen size.)



   Each instance of evidence is highlighted in the same color as its corresponding selection box.

More About the “Remote Files” Panels on the Exact or Partial Matches Tabs

When you open the **Exact Matches** tab or the **Partial Matches** tab (and select the **Partial Matches** checkbox) for a codebase file selected in the **Analysis Workbench**, a **File Details** view is shown in the center of the screen with the following panels:

- Remote Files Panel
- Components Panel
- Licenses Panel

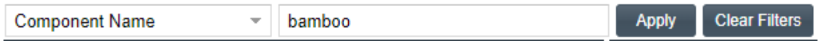
Note About Filtering in the Panels

The items in each panel can be filtered in these ways:

- When you a select a specific item in one panel, the items in the other panels area filtered to show only those items associated with the selected item.

  For example, when you select a specific *remote file* (that is, a file found in the Compliance Library) that matches a codebase file either exactly or partially in the **Remote Files** panel, the **Components** list is filtered to show only items associated with the remote file, and the **Licenses** list is filtered to show only items associated with the items now listed in the **Components** panel. Similarly, if you select a specific component in the **Components** list, the **Remote Files** and **Licenses** lists are filtered to show only those items associated with the selected component.

- Use the filter above the panel section to filter items in a given panel. Select the panel name from the filter dropdown list and enter a search string to show only items in that panel containing the string. When the filter is applied, the other panels are automatically filtered to show only items associated with the items now listed in the panel filtered by the search string.


Component Name
bamboo

Remote Files Panel

This panel initially lists all the files from the Compliance Library (CL), called *remote files*, that are either a perfect match (exact match) or contain partial-match content (source-code fingerprint match) to the scanned file. The partial-match content also ranks the remote files by CodeRank™ values, described in the previous section, Examining Evidence of Open-Source Code in a Given Non-Binary File.

The remote files list can be filtered as discussed in Note About Filtering in the Panels.

Components Panel

This panel initially lists all the component versions that contain the remote files listed in the **Remote Files** panel. The list can be filtered as discussed in Note About Filtering in the Panels.

You can perform the following operations for a given component in the **Components** panel:

- To review the path of a remote file within a component, select the file in the **Remote Files** panel, and then click the **Remote File Paths** icon in the component row. A remote file is a file found within an open source component release that is either identical to the scanned file, or contains similar partial content as the scanned file. The remote file path is important because similar file structures between the scanned codebase and the remote file content is a potential strong indicator of code reuse from an open source project.
- To view information about the component, click the **Information** icon ⓘ.
- To add the selected codebase file to an inventory item associated with the component, click the **Add File to Inventory** icon . For more information, see Adding a Codebase File to Inventory Associated with a Remote File's Open-Source Component.

Licenses Panel

This panel lists all the licenses associated with the component versions listed in the **Components** panel but can be filtered as discussed in Note About Filtering in the Panels.

You can view information about the license by clicking the **Information** icon ⓘ in the license entry.

Adding a Codebase File to Inventory Associated with a Remote File's Open-Source Component

When a given codebase file exactly or partially matches a remote file (that is, a file in the Compliance Library), you can use the following procedure to easily add the codebase file to an inventory item based on an open-source or third-party component associated with the remote file.

Task ***To add codebase file to an inventory item based on the remote file's open-source or third-party component, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. Click the **Exact** or **Source** matches link in the legend at the top right of the workbench to search for codebase files that are exact or partial matches to files in the Compliance Library. Results are listed in the **File Search Results** pane.
3. From the list in **File Search Results**, locate and click the codebase file you want to add to an inventory item based on a specific component version associated with the file.
4. Open the **File Details** tab, and, at the top of the tab, select the **Exact Matches** or **Partial Matches** tab.

   Additionally, if you are on the **Partial Matches** tab, select the **Source Matches** checkbox.
5. From the **Remote Files** panel, select the remote file associated with the component on which the inventory item to which you want to add the file is based (or will be based if you need to create an inventory item).

6. In the **Components** panel, locate the component version that you believe is the origin of the matching code in the scanned codebase file, and click the **Add File to Inventory** icon ✚ in that component row.

   Code Insight searches for existing inventory items associated with the given component version. If one or more inventory items exist, the **Add to Inventory** dialog is displayed, showing the list of available inventory items. Continue with step 7.

   Otherwise, if no inventory items are currently associated with the given component version, the **Lookup Component** window is displayed, showing the given component version. From this window, you can register an instance for the component version (by selecting a license), register a new component version, search for a new component altogether, or create a custom component. Once you click **Use This Instance** for a component version in the **Lookup Component** window, Code Insight creates the inventory item based on the selected component instance and associates the selected file with the inventory item. The **Inventory Details** tab is opened for the inventory item. (Ignore the remaining steps in this procedure.)

7. Click the checkbox next to the inventory item to which you want to add the file.

8. (Optional) To mark the selected codebase file as reviewed, click **Mark file as reviewed**.

9. Click **Submit**. Code Insight adds the codebase file to the inventory item.

Viewing a Summary of Evidence Detected Across the Codebase

The **Evidence Details** pane in the **Analysis Workbench** enables you to view the list of open-source and third-party textual evidence detected across the codebase during the scan. The list shows instances of evidence for the following entities and, for each instance, includes the total number of files in which the instance was found and the number of those files that are not marked as reviewed:

- **Copyrights**—The copyright text of potential third-party software code found in your codebase.
- **Email/URLs**—Email addresses and website URLs of potential owners of third-party software found in your codebase.
- **Licenses**—Third-party licenses in your codebase that should be reviewed for IP compliance.
- **Search Terms**—Terms related to open-source or third-party software in your codebase (based on the terms defined in the Scan Profile).

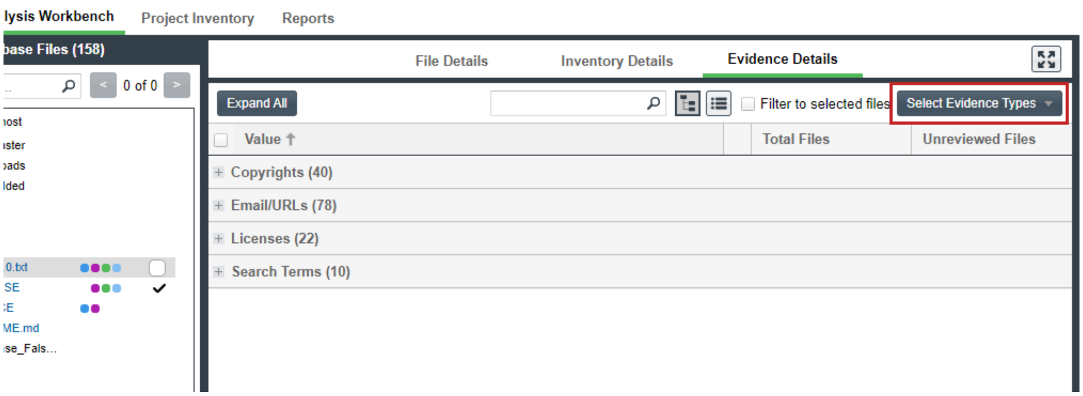
Note ▪ *Currently, for files scanned by a Code Insight scan-agent plugin on a remote system, only license evidence is reported in Code Insight. The* ***Evidence Details*** *pane will list any license evidence found in such files as part of the remote scan.*

Task ***To view all open-source or third-party copyrights, email addresses, URLs, licenses, and search terms in the codebase, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)

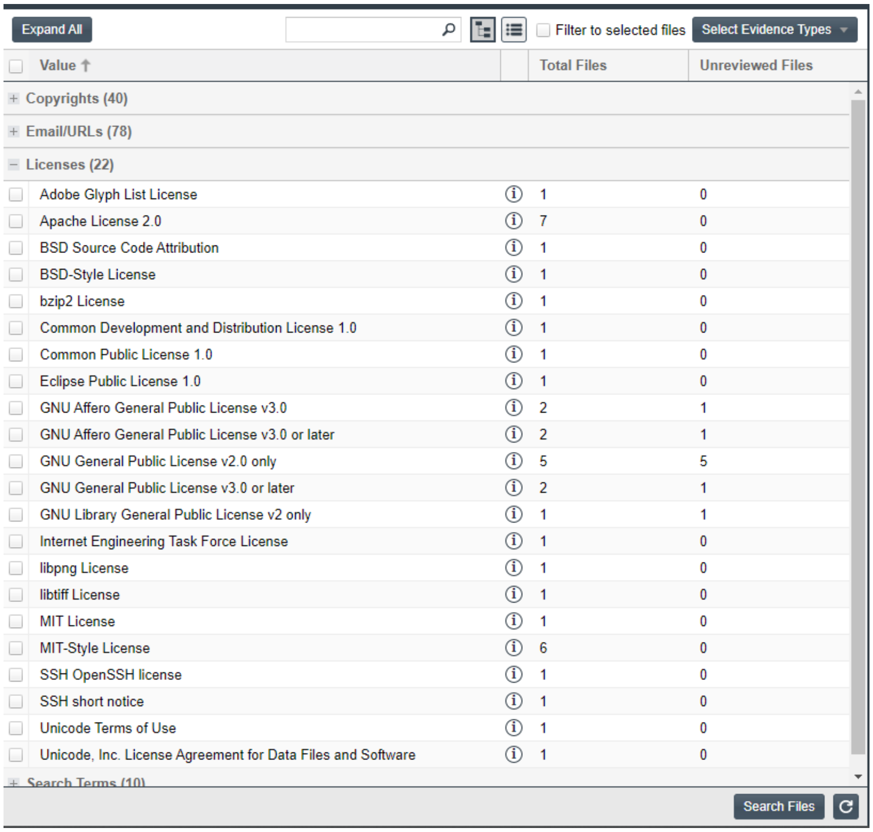
2. Open the **Evidence Details** tab in the center pane.

By default, the evidence is displayed in a tree-view, where you can expand or collapse the evidence instances (entries) under each evidence category (**Copyrights**, **Email/URLs,** and so forth).



Value	Total Files	Unreviewed Files
Copyrights (40)		
Email/URLs (78)		
Licenses (22)		
Search Terms (10)		

3. View the list of evidence instances under specific categories as needed. (Alternatively, to see all evidence instances, click **Expand All**; click **Collapse All** to return the display to only collapsed category headings.)



Value		Total Files	Unreviewed Files
Copyrights (40)			
Email/URLs (78)			
Licenses (22)			
Adobe Glyph List License	ⓘ	1	0
Apache License 2.0	ⓘ	7	0
BSD Source Code Attribution	ⓘ	1	0
BSD-Style License	ⓘ	1	0
bzip2 License	ⓘ	1	0
Common Development and Distribution License 1.0	ⓘ	1	0
Common Public License 1.0	ⓘ	1	0
Eclipse Public License 1.0	ⓘ	1	0
GNU Affero General Public License v3.0	ⓘ	2	1
GNU Affero General Public License v3.0 or later	ⓘ	2	1
GNU General Public License v2.0 only	ⓘ	5	5
GNU General Public License v3.0 or later	ⓘ	2	1
GNU Library General Public License v2 only	ⓘ	1	1
Internet Engineering Task Force License	ⓘ	1	0
libpng License	ⓘ	1	0
libtiff License	ⓘ	1	0
MIT License	ⓘ	1	0
MIT-Style License	ⓘ	6	0
SSH OpenSSH license	ⓘ	1	0
SSH short notice	ⓘ	1	0
Unicode Terms of Use	ⓘ	1	0
Unicode, Inc. License Agreement for Data Files and Software	ⓘ	1	0
Search Terms (10)			

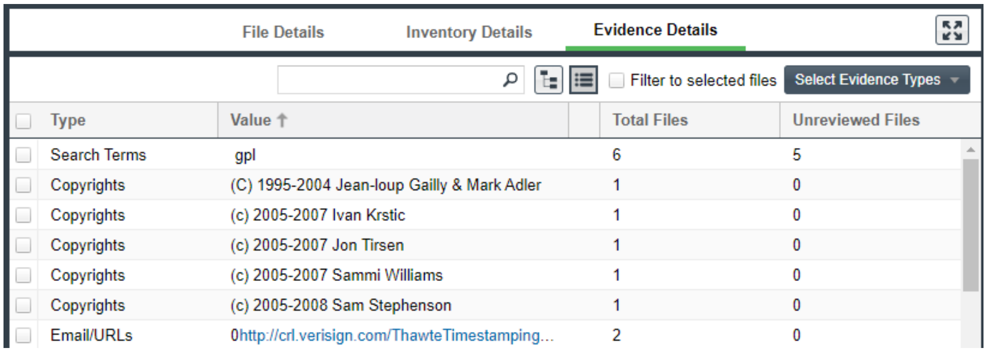
In each category list, you can view the total number of files containing evidence at the category level, the total number of files containing evidence for each instance, and the number of those files that are not marked as reviewed. For each instance of license evidence, you can click the ⓘ icon to view information about the license.

4. (Optional) Configure the **Evidence Details** tab to show alternative views of the available evidence. See the next section, Configuring Various Views of Evidence Details.

Configuring Various Views of Evidence Details

The **Evidence Details** tab provides options to configure various views of the available evidence. You can use a combination of these configurations to obtain the view you want.

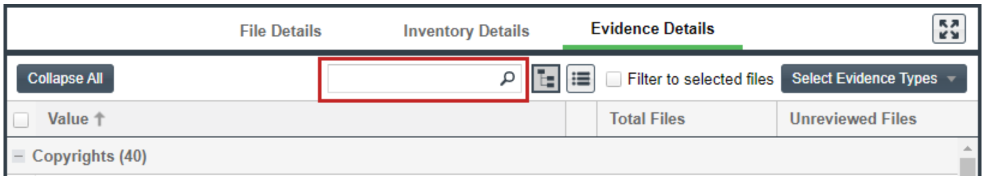
- **Display the evidence as a simple list instead of in a tree-view**—Click ☰ at the top of the **Evidence Details** tab. The list is reformatted without expandable and collapsible categories. A **Type** column is added to show the category for each evidence instance.



□	Type	Value ↑	Total Files	Unreviewed Files
□	Search Terms	gpl	6	5
□	Copyrights	(C) 1995-2004 Jean-loup Gailly & Mark Adler	1	0
□	Copyrights	(c) 2005-2007 Ivan Krstic	1	0
□	Copyrights	(c) 2005-2007 Jon Tirsen	1	0
□	Copyrights	(c) 2005-2007 Sammi Williams	1	0
□	Copyrights	(c) 2005-2008 Sam Stephenson	1	0
□	Email/URLs	0http://crl.verisign.com/ThawteTimestamping...	2	0

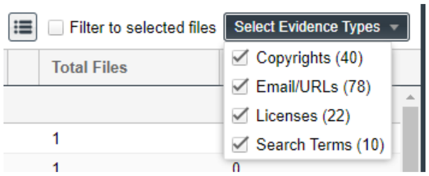
To return to the tree-view list, click ⊞ at the top of the **Evidence Details** tab.

- **Search for evidence that contains a specific string**—Enter the string in the search box at the top of the **Evidence Details** pane, and then click the **Refresh** ⟳ button in the lower right of the pane.

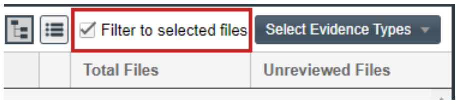


The list is refreshed to show only evidence containing the string.

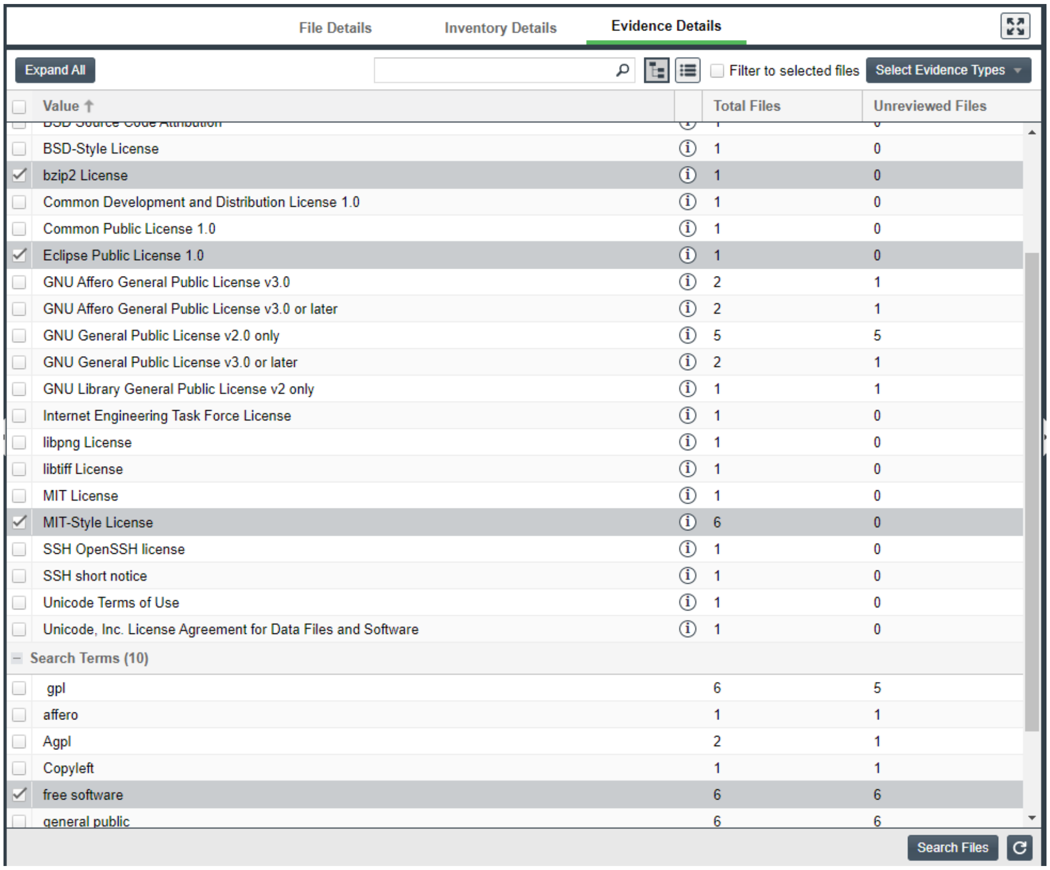
- **Show only specific categories of evidence**—From the **Select Evidence Types** dropdown list at the top of the **Evidence Details** tab, select the categories you want to display.



- **List the evidence contained in selected codebase files only**—Select one or more files in the **Codebase Files** pane, and click **Filter to Selected Files** on the **Evidence Details** tab. The evidence instances on the **Evidence Details** tab filters to only those instances found in the selected files.

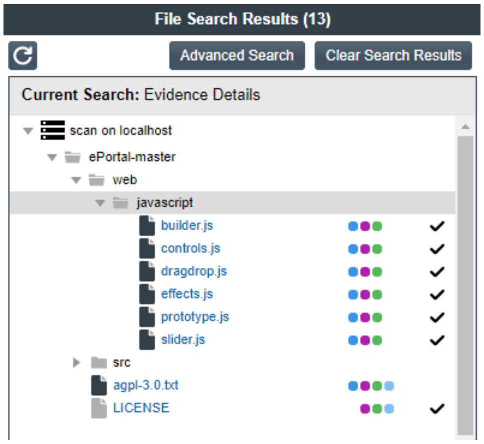


- **Filter to a list of codebase files in the File Search Results pane that contain only selected evidence**—Select the checkbox to the left of one or more evidence instances in the list on the **Evidence Details** tab, and click **Search Files** in the lower right of the tab. (When you select multiple evidence instances, the search uses OR logic to obtain the results.)



☐	Value ↑		Total Files	Unreviewed Files
☐	BSD Source Code Attribution	ⓘ	1	0
☐	BSD-Style License	ⓘ	1	0
☑	bzip2 License	ⓘ	1	0
☐	Common Development and Distribution License 1.0	ⓘ	1	0
☐	Common Public License 1.0	ⓘ	1	0
☑	Eclipse Public License 1.0	ⓘ	1	0
☐	GNU Affero General Public License v3.0	ⓘ	2	1
☐	GNU Affero General Public License v3.0 or later	ⓘ	2	1
☐	GNU General Public License v2.0 only	ⓘ	5	5
☐	GNU General Public License v3.0 or later	ⓘ	2	1
☐	GNU Library General Public License v2 only	ⓘ	1	1
☐	Internet Engineering Task Force License	ⓘ	1	0
☐	libpng License	ⓘ	1	0
☐	libtiff License	ⓘ	1	0
☐	MIT License	ⓘ	1	0
☑	MIT-Style License	ⓘ	6	0
☐	SSH OpenSSH license	ⓘ	1	0
☐	SSH short notice	ⓘ	1	0
☐	Unicode Terms of Use	ⓘ	1	0
☐	Unicode, Inc. License Agreement for Data Files and Software	ⓘ	1	0
−	Search Terms (10)			
☐	gpl		6	5
☐	affero		1	1
☐	Agpl		2	1
☐	Copyleft		1	1
☑	free software		6	6
☐	general public		6	6

A list of only those files that contain the selected evidence appears in a tree view in the **File Search Results** pane.



Current Search: Evidence Details

- scan on localhost
  - ePortal-master
    - web
      - javascript
        - builder.js ✓
        - controls.js ✓
        - dragdrop.js ✓
        - effects.js ✓
        - prototype.js ✓
        - slider.js ✓
    - src
    - agpl-3.0.txt
    - LICENSE ✓

Managing the Codebase Files

The following topics describe basic operations you can perform on one or more selected codebase files in the **Codebase Files** and **File Search Results** panes:

- Showing Inventory Associated with Files Selected in the Codebase List
- Adding Files to Inventory From the Codebase List
- Listing Copyright, Email, URL, License, and Search-Term Evidence for Files Selected in the Codebase List
- Marking Codebase Files as Reviewed
- Reverting Codebase Files to Unreviewed Status
- Downloading a Codebase File
- Copying Codebase File and Folder Paths

As an alternative to performing these operations on files listed in the **Codebase Files** or **File Search Results** pane, you can also perform them on the files listed on the **Associated Files** tab for a selected inventory item in the **Inventory Items** pane in the **Analysis Workbench**. See Viewing Associated Files for Inventory in the Analysis Workbench.

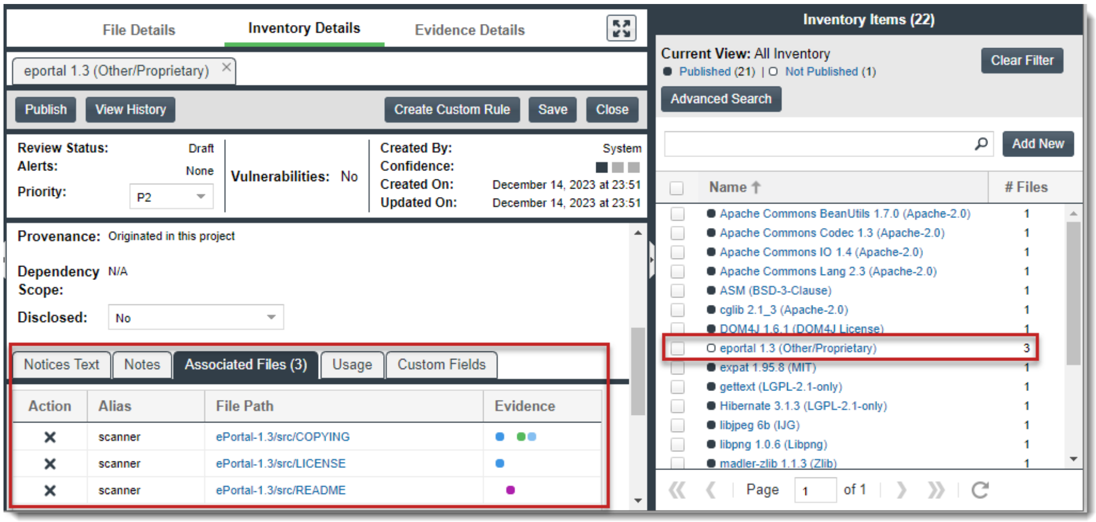
Showing Inventory Associated with Files Selected in the Codebase List

You can filter inventory items in the **Inventory Items** pane in the **Analysis Workbench** to only those items associated with the codebase files you have selected in the **Codebase Files** or **File Search Results** pane. This procedure is helpful in quickly locating the inventory items to which a codebase file is associated.

***Note** ▪ The results from this filter overwrite any filtered or non-filtered inventory results currently in the **Inventory Items** pane.*

***Task** To show only inventory items associated with files selected in the codebase, do the following:*

1. Open the **Analysis Workbench** for a project. (For instructions, see Opening the Analysis Workbench.)
2. In the **Codebase Files** or **File Search Results** pane of the **Analysis Workbench**, select and right-click an individual file or a set of files to view the inventory items with which the files are associated. (You can also right-click a directory to select all files in that directory and its subdirectories.)
3. From the pop-up menu, select **Show file inventory**. The **Inventory Items** pane on the right side of the **Analysis Workbench** filters to all inventory items to which the selected files are associated.
4. (Optional) To view all files associated with a displayed inventory item, do the following:
    a. Select the inventory item in the **Inventory Items** pane. The **Inventory Details** tab for that item opens in the middle pane of the **Analysis Workbench**.
    b. Within the **Inventory Details** tab for the inventory item, click the **Associated Files** tab to see all the files associated with the inventory item. (For more information about the file list and the options available on the **Associated Files** tab, see Viewing Associated Files for Inventory in the Analysis Workbench.)



Review Status: Draft
Alerts: None
Priority: P2
Vulnerabilities: No
Created By: System
Confidence:
Created On: December 14, 2023 at 23:51
Updated On: December 14, 2023 at 23:51

Provenance: Originated in this project

Dependency Scope: N/A

Disclosed: No

Action	Alias	File Path	Evidence
✕	scanner	ePortal-1.3/src/COPYING	● ●●
✕	scanner	ePortal-1.3/src/LICENSE	●
✕	scanner	ePortal-1.3/src/README	●

Current View: All Inventory
● Published (21) | ○ Not Published (1)

	Name	# Files
□	● Apache Commons BeanUtils 1.7.0 (Apache-2.0)	1
□	● Apache Commons Codec 1.3 (Apache-2.0)	1
□	● Apache Commons IO 1.4 (Apache-2.0)	1
□	● Apache Commons Lang 2.3 (Apache-2.0)	1
□	● ASM (BSD-3-Clause)	1
□	● cglib 2.1_3 (Apache-2.0)	1
□	● DOM4J 1.6.1 (DOM4J License)	1
□	○ eportal 1.3 (Other/Proprietary)	3
□	● expat 1.95.8 (MIT)	1
□	● gettext (LGPL-2.1-only)	1
□	● Hibernate 3.1.3 (LGPL-2.1-only)	1
□	● libjpeg 6b (IJG)	1
□	● libpng 1.0.6 (Libpng)	1
□	● madler-zlib 1.1.3 (Zlib)	1

Page 1 of 1

Adding Files to Inventory From the Codebase List

This section describes how to “inventory” selected codebase files from the **Codebase Files** or **File Search Results** list in the **Analysis Workbench**, either by associating these files with existing inventory or by creating a new inventory item with which to associate them.

Task ***To create or update an inventory item with files from the Codebase Files list, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. (Perform this step only when adding files to a *single* existing inventory item. Otherwise, skip to step 3.) Navigate to the **Inventory Items** pane, and select the inventory item to open its **Inventory Details** tab.
3. On the left side of the **Analysis Workbench**, navigate to the **Codebase Files** pane or the **File Search Results** pane.
4. Select one or more codebase files that you want to add to inventory (or select one or more folders whose files you to add).

   Note ▪ *You cannot select the base node for a Scan Server or a remote scan agent for this function.*

5. Proceed with either procedure:
   - If adding files to the inventory item whose **Inventory Details** tab you opened in step 2, drag and drop the selected files to the **Associated Files** tab on that tab.

- If adding the selected files to multiple existing inventory items or if creating an inventory item using the selected files as associated files:

  a. Right-click the selected files to open a pop-up menu, and select **Add to Inventory** to open the **Add to Inventory** dialog.

  b. Continue with either Add Selected Codebase Files to Existing Inventory or Create a New Inventory Item from Codebase Files Selected in Analysis Workbench.

Add Selected Codebase Files to Existing Inventory

This procedure describes how to use the **Add to inventory** dialog to add the selected codebase files to one or more existing inventory items in the **Analysis Workbench**.

Task ***To add the selected codebase files to existing inventory, do the following:***

1. Access the **Add to inventory** dialog by following the steps in the preceding main procedure, Adding Files to Inventory From the Codebase List.
2. In the **Add to inventory** dialog, select one or more inventory items to which to add the selected codebase file or files. (You can use the search field to search for the inventory.)
3. (Optional) Click **Mark files as reviewed**.
4. Click **Submit** to add codebase files to the **Associated Files** tab for each selected inventory item in the **Analysis Workbench**.

Create a New Inventory Item from Codebase Files Selected in Analysis Workbench

This procedure describes how to use the **Add to inventory** dialog to create a new inventory item with which to associate the selected codebase files in the **Analysis Workbench**.

Task ***To create a new inventory item with which to associated selected codebase, do the following:***

1. Access the **Add to inventory** dialog by following the steps in the preceding main procedure, Adding Files to Inventory From the Codebase List.
2. In the **Add to inventory** dialog, click **Add New**. A new inventory item “candidate”, showing default values, opens in its own tab on the **Inventory Details** tab.

   Note that the selected codebase files for which you are creating the inventory item are automatically added to the **Associated Files** tab for the new inventory item in the **Analysis Workbench**.
3. Complete the fields to define the new inventory item, as described in the later section, Creating Inventory From the Inventory Items List.
4. (Optional) Drag and drop one or more additional files from the **Codebase Files** list (or **File Search Results** list) to the **Associated Files** tab.
5. When you have completed the details for the new inventory item, click **Save**. The name of the inventory item is added (with a **Review Status** of **Draft**) in the **Inventory Items** pane.

Listing Copyright, Email, URL, License, and Search-Term Evidence for Files Selected in the Codebase List

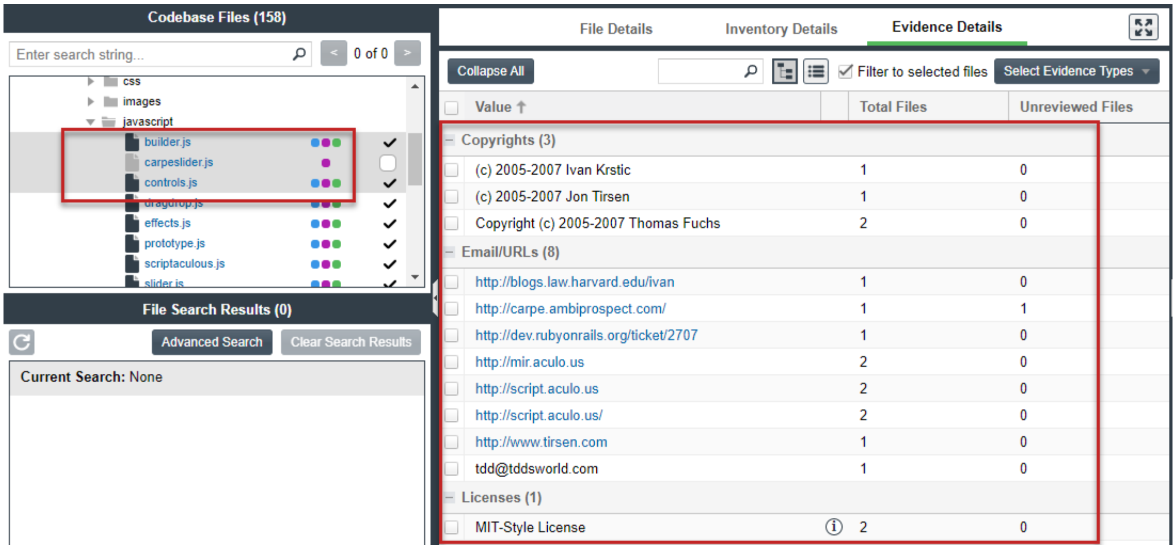
Use the following procedure to view a list of all copyright, email, URL, license, and search-term evidence found in one or more files selected in the **Codebase Files** or **File Search Results** pane. The **Evidence Details** tab, which by default lists such evidence for the entire codebase, is filtered to list evidence for the selected files only.

Task ***To list the evidence for one or more codebase files, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)

2. In the **Codebase Files** or **File Search Results** pane of the **Analysis Workbench**, select and right-click an individual file or a set of files whose evidence you want to view. (You can also right-click a directory to select all files in that directory and its subdirectories.)

3. From the pop-up menu, select **Show file evidence**. The **Evidence Details** tab opens in the center of the **Analysis Workbench**, listing the evidence found in the selected files.

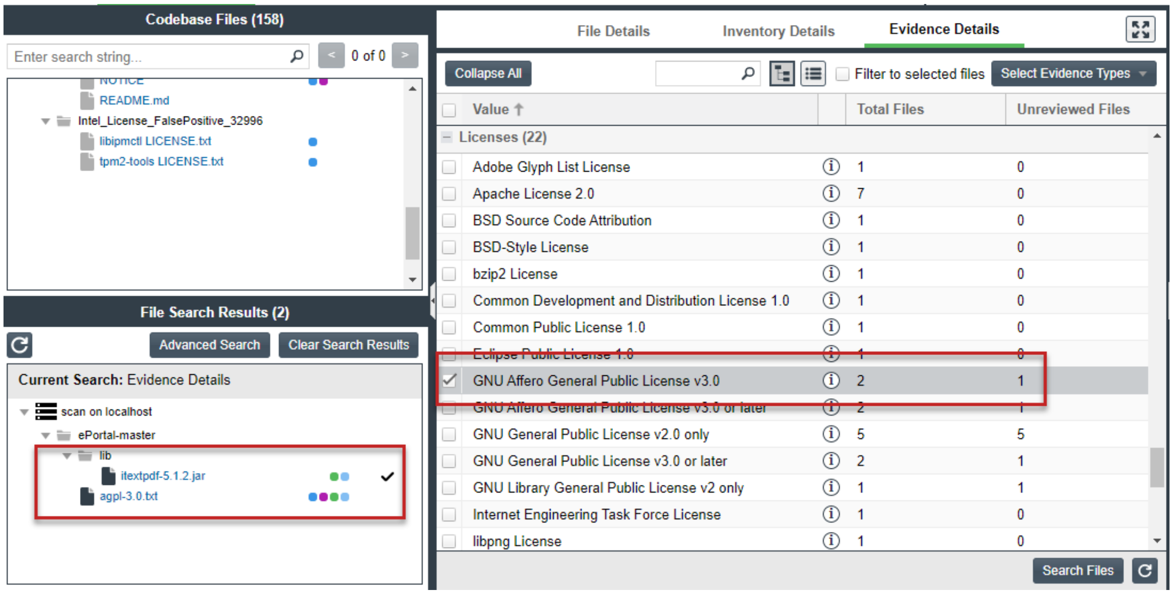


Current Search: None

Value	Total Files	Unreviewed Files
Copyrights (3)		
(c) 2005-2007 Ivan Krstic	1	0
(c) 2005-2007 Jon Tirsen	1	0
Copyright (c) 2005-2007 Thomas Fuchs	2	0
Email/URLs (8)		
http://blogs.law.harvard.edu/ivan	1	0
http://carpe.ambiprospect.com/	1	1
http://dev.rubyonrails.org/ticket/2707	1	0
http://mir.aculo.us	2	0
http://script.aculo.us	2	0
http://script.aculo.us/	2	0
http://www.tirsen.com	1	0
tdd@tddsworld.com	1	0
Licenses (1)		
MIT-Style License	2	0

4. (Optional) To determine which of the selected files contains a specific instance of evidence (for example, a specific copyright or search term), select the checkbox next to the instance on the **Evidence Details** tab, and click **Search Files** (bottom right of the tab).

   The associated files are listed in the **Files Search Results** pane. (You can also select multiple evidence instances in this step.) For more details about using the **Evidence Details** tab, see Examining Evidence of Open-Source Copyrights, Email Addresses, URLs, Licenses, and Search Terms in a Given Non-Binary File.



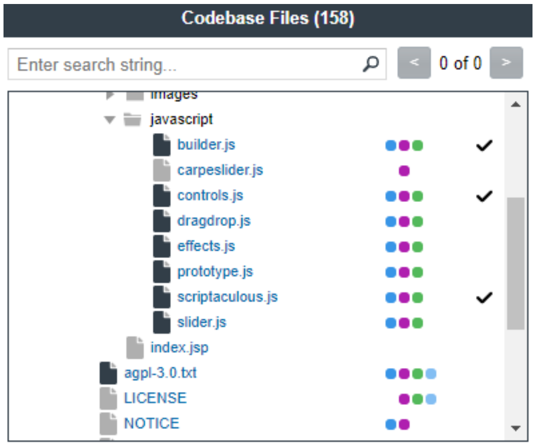
Current Search: Evidence Details

- scan on localhost
  - ePortal-master
    - lib
      - itextpdf-5.1.2.jar ✓
    - agpl-3.0.txt

	Value ↑		Total Files	Unreviewed Files
	Licenses (22)			
□	Adobe Glyph List License	ⓘ	1	0
□	Apache License 2.0	ⓘ	7	0
□	BSD Source Code Attribution	ⓘ	1	0
□	BSD-Style License	ⓘ	1	0
□	bzip2 License	ⓘ	1	0
□	Common Development and Distribution License 1.0	ⓘ	1	0
□	Common Public License 1.0	ⓘ	1	0
□	Eclipse Public License 1.0	ⓘ	1	0
☑	GNU Affero General Public License v3.0	ⓘ	2	1
□	GNU Affero General Public License v3.0 or later	ⓘ	2	1
□	GNU General Public License v2.0 only	ⓘ	5	5
□	GNU General Public License v3.0 or later	ⓘ	2	1
□	GNU Library General Public License v2 only	ⓘ	1	1
□	Internet Engineering Task Force License	ⓘ	1	0
□	libpng License	ⓘ	1	0

Marking Codebase Files as Reviewed

It is important to keep track of which files have been audited by marking files as reviewed when you are finished auditing them. If necessary, you can use the **Advanced Search** button on the **File Search Results** pane to filter to only un-reviewed files to see what is left to evaluate. You can also see the progress of the audit on the **Summary** tab. Files that have been marked as reviewed show a checkmark to the right of the file name in the **Codebase Files** and **File Search Results** panes.



- images
- javascript
  - builder.js ✓
  - carpeslider.js
  - controls.js ✓
  - dragdrop.js
  - effects.js
  - prototype.js
  - scriptaculous.js ✓
  - slider.js
- index.jsp
- agpl-3.0.txt
- LICENSE
- NOTICE

When all files have been marked as reviewed, an overview-style audit can be considered completed.

Task ***To mark files as reviewed, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **Codebase Files** or **File Search Results** pane of the **Analysis Workbench**, select and right-click an individual file or a set of files that you want to mark as reviewed. (You can also right-click a directory to select all files in that directory and its subdirectories.)

   Note • *You cannot select the base node for a Scan Server or a remote scan agent for this function.*
3. From the pop-up menu, select **Mark as reviewed**. A checkmark is added to the right of each selected file to indicate that it now has a reviewed status.

   Note • *If you enabled the project scan setting that automatically publishes inventory, you can also enable the setting that automatically marks files associated with this inventory as reviewed. For more information about these settings, see* Edit Project: Scan Settings Tab*.*

Reverting Codebase Files to Unreviewed Status

In some cases, a files marked as reviewed might need to be reverted to an unreviewed status. This can happen, for example, if new evidence or security vulnerabilities require more investigation of the file contents.

Task ***To revert reviewed files to unreviewed status, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **Codebase Files** or **File Search Results** pane of the **Analysis Workbench**, select and right-click an individual file or a set of files that you want to revert to unreviewed. (You can also right-click a directory to select all files in that directory and its subdirectories.)

   Note • *You cannot select the base node for a Scan Server or a remote scan agent for this function.*
3. From the pop-up menu, select **Mark as unreviewed**. The checkmark to the right of each selected file is removed.

Downloading a Codebase File

You can download an individual codebase file to your browser's default download location.

Note *▪ Currently, this option is available only for files scanned by the Scan Server, not for files scanned by a Code Insight scan-agent plugin on a remote system.*

Task ***To download a codebase file, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **Codebase Files** or **File Search Results** pane of the **Analysis Workbench**, select and right-click the file that you want to download.

   Note *▪ You can select only a single file for this function, not a folder or the base node for a Scan Server or a remote scan agent.*

3. From the pop-up menu, select **Download File**. The file is downloaded to your browser's default location.

Copying Codebase File and Folder Paths

You can copy the complete paths of files and folders listed in **Codebase Files** or **File Search Results** pane, enabling you to paste accurate path information to other parts of the project (for example, to the **Audit Notes** field), between projects, or in personal locations. The following describes the format in which the paths are copied:

- The output for a file-path copy uses the format <alias>:<relativeFilePath>, where <alias> is the meaningful name given to a scanner (Scan Server or remote scan agent) to represent its scan-root container, and <relativeFilePath> is the file path relative to the absolute scan-root path on host instance. For example, if you copy the codebase file **agpl-3.0.txt**, located directly under the scan folder **ePortal-1.3**, which in turn is directly under the scan-root path for the Scan Server whose alias is "EP_remote", the output for the copy is **EP_remote:ePortal-1.3/agpl-3.0.txt**.
- Likewise, the output for a folder-path copy uses a similar format, <alias>:<relativeFolderPath>. For example, if you copy the **src** folder, located directly under the scan folder **ePortal-1.3**, which in turn is directly under the scan-root path for the Scan Server whose alias is "EP_remote", the folder for the copy is **EP_remote:ePortal-1.3/src**.

Note *▪ When you copy a folder path, only the path for the folder is copied, not the paths for the files within the folder.*

Task ***To copy codebase file and folder paths, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)

2. In the **Codebase Files** or **File Search Results** pane of the **Analysis Workbench**, perform any of these tasks to copy codebase file or folder paths to the operating system Clipboard:

    - **To copy the path for a single file**—Right-click the file, and select **Copy File Path** from the pop-up menu.
    - **To copy paths for multiple files**—Select the files (using the Ctrl or Shift key), right-click anywhere in the group, and select **Copy Paths** from the pop-up menu.
    - **To copy the path for a single folder**—Right-click the folder, and select **Copy Folder Path** from the pop-up menu.
    - **To copy paths for multiple folders**—Select the folders (using the Ctrl or Shift key), right-click anywhere in the group, and select **Copy Paths** from the pop-up menu.

    When selecting multiple codebase items to copy, select only all files or all folders. Do not select a combination of files and folders. If you do select a combination, the **Copy Paths** option is disabled. Additionally, you cannot select the base node for a Scan Server or a remote scan agent for this function.

3. Use Ctrl + v to paste the copied path or paths to the desired location.

Managing Inventory in the Analysis Workbench

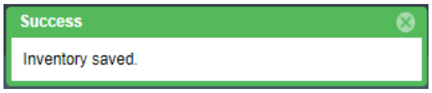
The **Inventory Details** tab allows you to view and manage details for a selected inventory item. Refer to the following topics for more information. You can find descriptions of the fields available on this tab in Inventory Details Tab in the Analysis Workbench.

- Success Messages When Working with Inventory
- Using the Inventory Items Context Menu
- Performing Inventory Searches in the Analysis Workbench
- Viewing Security Vulnerabilities for Inventory in the Analysis Workbench
- Viewing Details About the Component Associated with Inventory in the Analysis Workbench
- Viewing Details About Licenses Associated with Inventory in the Analysis Workbench
- Viewing Associated Files for Inventory in the Analysis Workbench
- Viewing or Editing Inventory Usage Information from the Analysis Workbench
- Viewing and Updating Detection and Auditing Notes in the Analysis Workbench
- Creating an Inventory Item from the Analysis Workbench
- Editing Inventory from the Analysis Workbench
- Creating a Custom Rule Based on the Files Associated with Inventory in the Analysis Workbench
- Publishing or Recalling Inventory Manually from the Analysis Workbench
- Viewing the Update History for an Inventory Item in the Analysis Workbench

Success Messages When Working with Inventory

When you perform certain inventory operations such as creating or editing inventory (and others), a message box is displayed in the upper right corner of the Code Insight user interface to inform you of one of the following:

- The operation has successfully completed.
- The operation has been successfully initiated as a background job in the **Jobs** queue. The message includes the job ID so that you can track the job (as described in Monitoring the Code Insight Jobs Queue).



Success

Inventory saved.

The message persists for a couple of seconds, but you can click the ⊗ button in the box to close the message sooner.

Using the Inventory Items Context Menu

The **Inventory Items** pane has a context menu containing shortcuts to common inventory tasks. The following tasks are available on the context menu:

- **Publish Inventory**—Select inventory items that you would like to publish, right-click, and choose **Publish Inventory** to quickly publish your selected items. Publishing an inventory item makes it visible in the **Project Inventory** view.
- **Recall Inventory**—Select published inventory items that you would like to recall back to an unpublished state, right-click, and choose **Recall Inventory**. The selected items are removed from the **Project Inventory** view and are only visible in the **Analysis Workbench**.

  ***Note** ▪ Editing an inventory item does not require a recall of the inventory item. The item's field values may be edited from the **Analysis Workbench** or the **Project Inventory** view at any time, even if the item has already been published.*

- **Show Inventory Files**—To see files associated with the selected inventory items, select the list of inventory items, and right-click and choose **Show Inventory Files**. The associated files will be shown in the **File Search Results** pane.
- **Delete Inventory**—Select inventory items that you want to delete, right-click, and select **Delete Inventory**. The selected items will be deleted from the project.

***Note** ▪ When you republish an inventory item by selecting the Recall and Publish tasks, the published date on the item is reset. This action in turn affects the age of the inventory item. Republished items are treated as newly published items.*

Performing Inventory Searches in the Analysis Workbench

You can filter the **Inventory Items** list to focus on the inventory you want to examine. The following sections describe the filtering methods:

- Filtering Inventory by Name
- Filtering by Publication Status
- Performing an Advanced Inventory Search

Note ▪ *Results from an inventory search (using one or more of these filtering methods) and the results of an inventory search based on associated codebase files are mutually exclusive and will overwrite each other in the* ***Inventory Items*** *pane. (For more information about inventory searches based on an inventory's associated codebase files, see Showing Inventory Associated with Files Selected in the Codebase List.)*

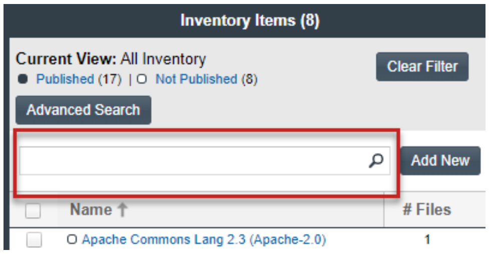
Filtering Inventory by Name

You can filter the inventory by an inventory name or a string within the name.

Note ▪ *The name filter you define is automatically copied to the Advanced Inventory Search feature should you use this feature (see Performing an Advanced Inventory Search). Likewise, if you enter a name filter when using the Advanced Inventory Search feature, it is copied to the name-filter field on the* ***Inventory Items*** *pane. This behavior enables you to keep the name filter persistent. In either location, the filter can be removed or replaced as needed.*

Task ***To filter the inventory by name, perform this step:***

In the name filter field above the **Inventory Items** list, provide the inventory name or a partial-name string. As you type each character in the string, the list is automatically filtered according to the entered characters.

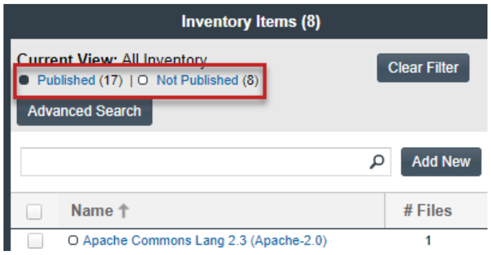

Inventory Items (8)
Current View: All Inventory
● Published (17) | ○ Not Published (8)
Clear Filter
Advanced Search
Add New
Name
# Files
○ Apache Commons Lang 2.3 (Apache-2.0)
1

Filtering by Publication Status

You can filter inventory by its published or not published status.

Task ***To filter the inventory by its publication status, perform this step:***

At the top of the **Inventory Items** pane, click the **Published** or **Not Published** link to filter to the list of published or not-published items, respectively.


Inventory Items (8)
Current View: All Inventory
● Published (17) | ○ Not Published (8)
Clear Filter
Advanced Search
Add New
Name
# Files
○ Apache Commons Lang 2.3 (Apache-2.0)
1

Performing an Advanced Inventory Search

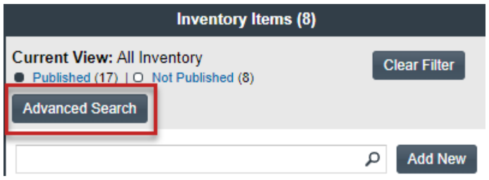
Code Insight provides the **Advanced Inventory Search** dialog that enables you to easily filter the list of inventory items to those of interest based on many available criteria—inventory attributes, selected license attributes, and associated security vulnerabilities, tasks, and security alerts. In this way, you can focus on only those inventory items in which you are interested. The following procedure shows you how to access and use this dialog. Refer also to Performing Advanced Searches for practical applications of this search feature.

Note the following when using the Advanced Inventory Search feature in the **Analysis Workbench**.

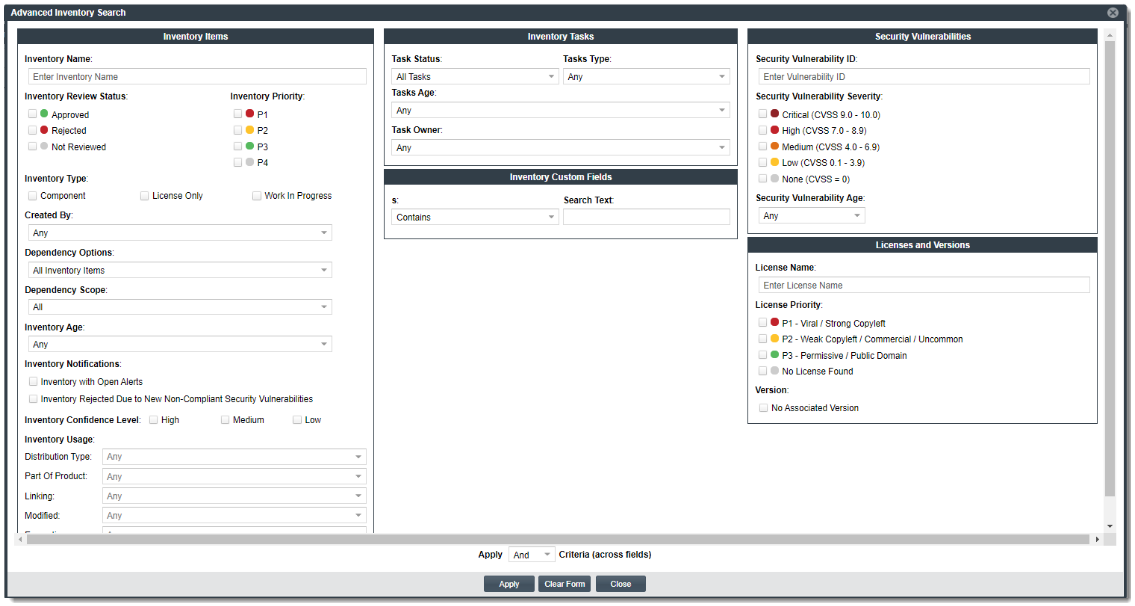
- If the **Inventory Items** list is filtered by published or not-published items (before or after using an Advanced Inventory search), the resulting inventory list is based on the published/not-published filter AND the **Advanced Inventory Search** criteria.
- If you entered a name filter on the **Inventory Items** pane, it is automatically displayed for the **Inventory Name** filter on the **Advanced Inventory Search** dialog. (Likewise, if you enter a name filter on the **Advanced Inventory Search** dialog, it is automatically copied to the **Inventory Items** pane.) This behavior enables you to keep the name filter persistent. However, you can remove or replace this filter as needed in either location.

Task ***To filter inventory in the Analysis Workbench, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **Inventory Items** pane, click the **Advanced Search** button above the list of inventory items.


Inventory Items (8)
Current View: All Inventory
● Published (17) | ○ Not Published (8)
Clear Filter
Advanced Search
Add New

The **Advanced Inventory Search** dialog is opened.



3. From this dialog, select search criteria as needed from the following categories. For a detailed description of the search criteria, see Advanced Inventory Search Dialog.

   - **Inventory Items**—Search for inventory items of a certain name (or string), review status, priority, type, creator, dependency scope, age, usage, or confidence level or that have open vulnerability alerts and work items. (For details on alerts and work items, see Managing Security Vulnerability Alerts and Creating and Viewing External Work Items for a Project Inventory Task.)

   - **Inventory Tasks**—Search for inventory items that have been assigned tasks. You can refine the search to locate inventory with open or closed tasks, tasks of a certain age or type (such as manual reviews or source-code remediation), or tasks assigned to a specific user.

   - **Inventory Custom Fields**—Search for inventory whose custom inventory fields contain the value you specify as criteria (or contain no value). Custom inventory fields are defined specifically for your site. If no such fields have been defined, this section is not visible.

   - **Security Vulnerabilities**—Search for inventory items that have vulnerabilities of a certain vulnerability ID, CVSS severity, or age. (Note that list of available severities for **Security Vulnerability Severity** varies depending on the CVSS version being used by Code Insight. The picture above shows the severities for CVSS v3.x. See Working with Security Vulnerabilities for details.)

   - **Licenses**—Search for inventory items that have licenses of a certain of a certain name or license priority.

4. In **Apply Criteria** field, select the boolean operator to apply to the criteria:

   - **Or**—To be included in the search results, an inventory item must contain at least one of the criteria you selected on this dialog.

   - **And**—To be included in the search results, an inventory item must meet all the criteria across the advanced search, as selected in this dialog. (This is the default operator.)

5. Click **Apply** to filter the inventory on the **Inventory Items** pane to display only those inventory items that meet the selected criteria.

6. To refresh the **Inventory Items** pane to show all inventory items, click **Show All Items**.

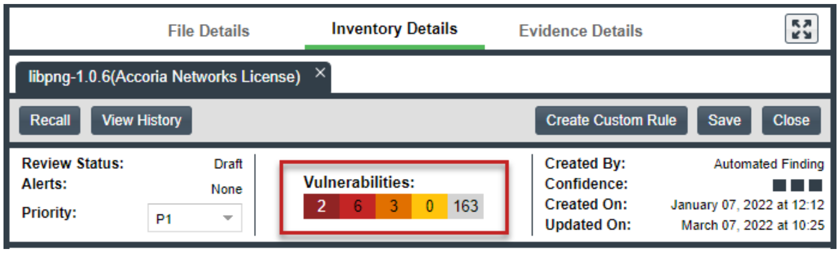
Viewing Security Vulnerabilities for Inventory in the Analysis Workbench

Code Insight uses data from the National Vulnerability Database (NVD), Secunia advisories (as published by the Secunia Research team from Revenera), and other advisories such as RubySec to report security vulnerabilities associated with your inventory items. The vulnerabilities information from these sources is used to create vulnerability rankings and alerts.

Use this procedure to access details about the security vulnerabilities associated with an inventory item in the **Analysis Workbench**.

Task ***To view security vulnerabilities for an inventory item, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. From the **Inventory Details** tab for a selected inventory item in the **Analysis Workbench**, locate the **Vulnerabilities** bar graph. (No graph is displayed if the inventory item has no known associated security vulnerabilities.)

   Review Status: Draft
   Alerts: None
   Priority: P1
   Vulnerabilities: 2 6 3 0 163
   Created By: Automated Finding
   Confidence: ■■■
   Created On: January 07, 2022 at 12:12
   Updated On: March 07, 2022 at 10:25

   The severities depicted on the graph differ depending on the CVSS version Code Insight is using (see Working with Security Vulnerabilities). This example shows vulnerability severity counts using CVSS v3.x.
3. Click any of the counts in the graph to open the **Security Vulnerabilities** window, which lists the current security vulnerabilities for the inventory item.

   Note ▪ *Suppressed vulnerabilities are neither reflected in the counts on* ***Vulnerabilities*** *bar graph nor are they visible on* ***Securities Vulnerabilities*** *window.*

   For more information about vulnerabilities, see Working with Security Vulnerabilities.
4. When you have finished viewing the reported vulnerabilities, click **OK** to return to the **Inventory Items** list.

Viewing Details About the Component Associated with Inventory in the Analysis Workbench

You can view details about the OSS or third-party component with which an inventory item is associated.

Task ***To view details for the component with which an inventory item is associated, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. From the **Inventory Details** tab for a selected inventory item in the **Analysis Workbench**, locate the **Component** field. This field lists the OSS or third-party component with which the selected inventory item is associated.
3. Click ⓘ next to the **Component** value. The **Component Details** window is displayed, showing information about the component.

   For descriptions of these fields on this tab, see Component Details Window.
4. Click the **View all versions** link to examine the list of versions for a component and, for each version, its associated licenses and security vulnerabilities by severity.
5. When you have finished examining the details, click **Close**.

Viewing Details About Licenses Associated with Inventory in the Analysis Workbench

You can view more details about the licenses associated with the OSS or third-party component on which an inventory item is based. This information is pulled from the Code Data Library and is displayed on the **License Details** window, which includes the following:

- A **General Information** tab that lists details such as the name of the license, its family, and the license priority assigned by Code Insight.
- A **License Text** tab that displays the complete license text (representing the external forge license text).

While the forge license text is what is likely to be used by the component, the scan might have found actual license text associated with this component in your codebase that can be different from the forge version. To view the exact license text, if any, that the scan discovered and to prepare the final license text, when necessary, to include in the Notices report for distribution to customers, navigate to the **Notices Text** tab. Refer to Finalizing the Notices Text for the Notices Report for more information.

The following procedure describes how to access the **License Details** window from the **Inventory Details** tab in the **Analysis Workbench**.

Note *▪ This window is also accessible when create or edit an inventory item or perform a Lookup Component from the* ***Analysis Workbench****.*

Task ***To view details for the inventory license, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. From the **Inventory Details** tab for a selected inventory item in the **Analysis Workbench**, locate the **License** field. This field lists the license currently that is associated with the component identified for the inventory item.
3. Click ⓘ next to the **License** value. The **License Details** window appears with the **General Information** tab in focus.

   For descriptions of these fields on this tab, see License Details Window. Also see License Priority for background on how the license priority is used.
4. Select the **License Text** tab to view the license text.
5. When you have finished examining the license details, click **Close**.

Viewing Associated Files for Inventory in the Analysis Workbench

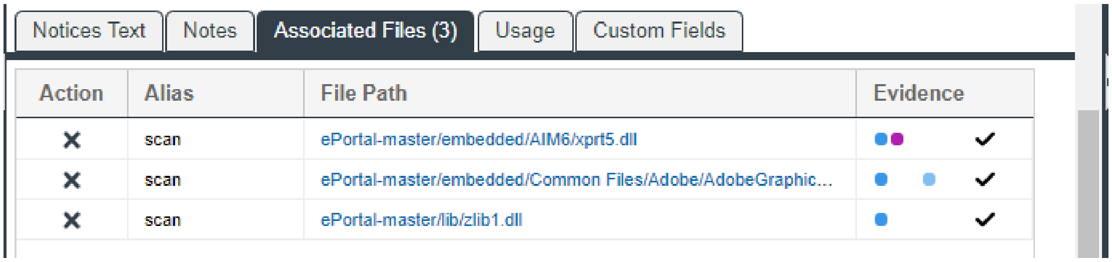
Use this procedure to view the codebase files that have been automatically or manually associated with the inventory item currently selected in the **Inventory Items** pane in the **Analysis Workbench**.

Task ***To view the codebase files currently associated with a given inventory item, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. Select an inventory item from the **Inventory Items** pane in the **Analysis Workbench**.
3. In the **Inventory Details** tab that is opened in the middle pane for the inventory item, click the **Associated Files** tab. The tab lists the codebase files currently associated with the selected inventory item. Each file entry shows the following details. (Note that you cannot sort the file list.)

   - **Action**—Icons that you can click to perform certain actions on the file. Currently, only the ✖ icon shows, enabling you to disassociate the file from the inventory item.
   - **Alias**—The unique, user-defined name provided during scanner setup (for a Scan Server or a remote scan agent) to represent its scan-root path containing the codebase in which the file is located. The alias provides a name that is more meaningful than the scan-root path. (The actual absolute scan-root path for each scanner associated with the project is available on the project’s **Summary** tab.)
   - **File Path**—The file’s path relative to the scan-root path on instance hosting the scanner. You can click the file-path link to open the **File Details** tab for that file in the Codebase Files view.
   - **Evidence**—The color-coded icons representing the types of open-source or third-party evidence found in the file (see Using the Filter Legend Options to Filter the Codebase for a description of the icons). A check mark indicates that the file has been reviewed.

Note *▪ Currently, license evidence is the only type of open-source and third-party evidence reported for files scanned by a Code Insight scan-agent plugin on a remote system.*



Action	Alias	File Path	Evidence
✕	scan	ePortal-master/embedded/AIM6/xprt5.dll	●● ✓
✕	scan	ePortal-master/embedded/Common Files/Adobe/AdobeGraphic...	● ● ✓
✕	scan	ePortal-master/lib/zlib1.dll	● ✓

4. (Optional) Right-click a file entry for a list of options that enable you to perform certain operations on the file, such as marking it as reviewed, reverting its reviewed status to unreviewed, and other operations. See Managing the Codebase Files for details about these same options that are also available from the **Codebase Files** and **File Search Results** panes in the **Analysis Workbench**.

Viewing or Editing Inventory Usage Information from the Analysis Workbench

Inventory usage information is important because it aids users in determining how closely to monitor inventory items for intellectual property (IP) and security risk and in taking appropriate action to approve or reject inventory, create tasks for remediation, and issue alerts and notifications. Usage fields can determine whether the item should be included in Third-Party Notices and what steps need to be taken to satisfy license obligations and conditions of use. They can also help to identify license conflicts and compatibility issues.

Task ***To view or edit inventory usage information, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)

2. From the **Inventory Details** tab for a selected inventory item in the **Analysis Workbench**, select the **Usage** tab.

3. View and, if necessary, edit the usage fields.

   For details about the these fields and how they are used, see Inventory Usage Information.

   Note *▪ When you save edits made to usage fields for a published inventory item, an automatic review of the inventory can be triggered. See Automatic Review Triggered When Saving Existing Inventory in the Analysis Workbench for more information.*

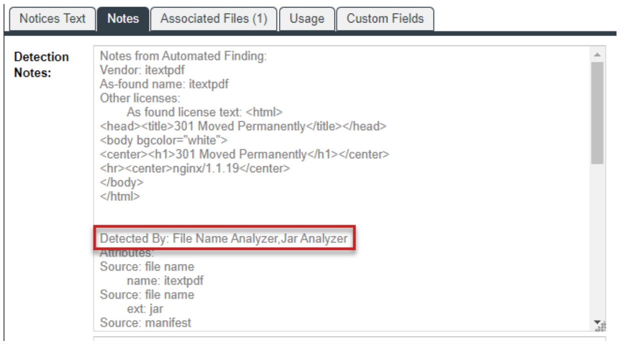
Viewing and Updating Detection and Auditing Notes in the Analysis Workbench

The **Notes** tab can provide information about automated and manual analysis of codebase as it relates to an inventory item. This can help you in your analysis of your product’s use of the OSS or third-party software identified by the inventory item.

Task ***To view notes, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. From the **Inventory Details** tab for a selected inventory item in the **Analysis Workbench**, select the **Notes** tab.
3. Review or update content in the following fields as needed:
    - **Detection Notes**—Information generated during the scan to explain the means by which the scan detected OSS or third-party software in the codebase. The **Detected By** attribute in the **Detection Notes** content indicates the automated detection technique(s) responsible for the inventory finding. The following example shows that the detection techniques used to find the inventory item included File Name Analyzer and Jar Analyzer.

      The **Detection Notes** content is not editable.

      Detection Notes:

      ```
      Notes from Automated Finding:
      Vendor: itextpdf
      As-found name: itextpdf
      Other licenses:
              As found license text: <html>
      <head><title>301 Moved Permanently</title></head>
      <body bgcolor="white">
      <center><h1>301 Moved Permanently</h1></center>
      <hr><center>nginx/1.1.19</center>
      </body>
      </html>


      Detected By: File Name Analyzer,Jar Analyzer
      Attributes:
      Source: file name
              name: itextpdf
      Source: file name
              ext: jar
      Source: manifest
      ```

    - **Audit Notes**—Information recorded about the manual analysis of the codebase associated with this software. You can add your own notes. For example, you might indicate that you needed to create this inventory item manually.
4. Click **Save** in the upper right corner of the **Inventory Details** tab to save the updates. If this inventory item is currently published, the save can trigger an automatic review of the inventory item. See Automatic Review Triggered When Saving Existing Inventory in the Analysis Workbench for details.

Creating an Inventory Item from the Analysis Workbench

When you identify third-party code in your codebase, you should create an inventory item to record it. Inventory items contain information critical for review and approval. The following describes the overall process for creating inventory in the **Analysis Workbench**.

Table 2-9 ▪ Inventory Creation Process in the Analysis Workbench

Phase	Description
1	Filter files that contain evidence of third-party code, such as copyright text or content from an open source license. See Searching for Codebase Files Based on Search Criteria and Viewing a Summary of Evidence Detected Across the Codebase.
2	Research the findings and identify the origin of the files.
3	Create an inventory item with details about the origin of the code. This is typically an open source project, such as zlib, OpenSSL, or ReactJS. If you do not know code's origin, you have options to create either a **License Only** inventory item (if the codebase files are governed by a common license) or a **Work In Progress** inventory item to serve as placeholder until you obtain more information. Inventory types are described in more detail in the procedure below.
4	When all of the evidence is explained in the files you are looking at (bearing in mind that some files might have code from several origins), mark the files as "reviewed".
5	When you are finished analyzing evidence for the inventory items, publish the ones you would like to report on. (For example, you might not want to publish internal or test tools.)

The following sections provide more details about creating inventory items from two locations in the **Analysis Workbench**:

- Creating Inventory From the Inventory Items List
- Creating Inventory From Files Currently Selected in the Codebase List

Creating Inventory From the Inventory Items List

This section describes how to create inventory from the **Inventory Items** list in the **Analysis Workbench**. (For instructions on creating inventory items based on *codebase files* in **Codebase Files** list or **File Search Results** list in the **Analysis Workbench**, see Creating Inventory From Files Currently Selected in the Codebase List.)

Task ***To create inventory from the Inventory Items list, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. Navigate to the **Inventory Items** list.
3. Click **Add New** at the top of the **Inventory Items** list. A new item, showing default values, opens in its own tab in the **Inventory Details** pane.
4. For the **Name** field, perform the appropriate step, based on the inventory type you intend to select for the **Type** field (see the next step):
   - For inventory of the type **Work in Progress**, specify a name for the inventory item. Best practice is to provide a name in the following conventional syntax used by Code Insight, even if the elements represented in the name are not available in the Data Library:

     ```
     <Component_name> <version> <License_name>
     ```

   - For inventory of the type **Component** or **License only**, leave the **Name** field blank. The field will be automatically populated based on the registered component or license instance.
5. From the **Type** dropdown list, select the type of inventory item you want to create and then perform the related step or steps:
   - **Work in Progress**—Create this type of inventory item if you want to quickly represent third-party code or an artifact without having to select an associated component, version, or license from the Code Insight Data Library. (You can later edit this inventory item to convert it to one of the other inventory types.) This option is typically used if you need a placeholder or cannot find the associated element in the Data Library. Items of type **Work in Progress** are not affected by policies and do not receive vulnerability updates or alerts.
   - **Component**—Create this type of inventory item if third-party code or artifacts point to a definite component version and possibly its license. You need to associate this type of inventory with a registered component instance—that is, a unique component-version-license combination found in the Code Insight Data Library. Use the Lookup Component feature, made available when you select the **Component** type, to locate this component instance and associate it with the inventory item. If are unable to locate the appropriate instance, the Lookup Component feature enables you to create a custom component. See Using “Lookup Component” to Search for Components to Associate with Inventory for further instructions.

     Once the instance is associated with inventory item, the **Name**, **Description**, **Component**, and **License** fields on the **Inventory Details** tab are automatically populated with information based on the selected instance. Additionally, Information ⓘ icons are available next to the **Component** and **License** fields so that you can view publicly available information about the selected component or its license.

     This inventory type is affected by policies and receives vulnerability updates and alerts.
   - **License Only**—Create this type of inventory item if evidence shows groups of codebase files of unknown origin are governed by a specific license. (You can later edit this inventory item to convert it to one of the other inventory types.) This inventory type is affected by policies.

     When creating **License Only** inventory, select the appropriate license from **License** dropdown list, which is enabled when you select this type.

The **Name** field for the inventory item is automatically populated with the name **Files under**

<**License_name**> **License**, where <License_name> is license you selected.The ⓘ icon is added so that you can view details about the selected license.(You can also click **New** to create a custom license. See When Creating or Editing a “License Only” Inventory Item for further instructions.)

6. Update the remaining fields if appropriate. For a description of each field, see Inventory Details Tab in the Analysis Workbench.

7. When you completed the details for the new inventory item, click **Save**. A new inventory item is added to the **Inventory Items** pane. The item is unpublished and its **Review Status** property is **Draft**.

Creating Inventory From Files Currently Selected in the Codebase List

You can create a new inventory item based on files you have selected in the **Codebase Files** or **File Search Results** list in the **Analysis Workbench**. For more information, see Create a New Inventory Item from Codebase Files Selected in Analysis Workbench.

Editing Inventory from the Analysis Workbench

Use the following steps to edit an inventory item from the **Analysis Workbench** as needed.

Task ***To edit an inventory item in the Analysis Workbench, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)

2. Navigate to the **Inventory Items** list in the right pane.

3. Select the inventory item that you want to edit.

   A new tab, labeled with the inventory name and showing information about the inventory item, is opened within the **Inventory Details** tab.

4. Make changes to the fields as needed. Refer to Inventory Details Tab in the Analysis Workbench for a description of each field. Also note the following information:

   - **Using the Lookup Component feature**—For a **Component** inventory item, you can use the Lookup Component feature to select a different registered instance (version and license) for the current component, select a different component altogether, or select a component instance for the first time (if you are changing the inventory item from a **Work in Progress** or **License Only** type). You can also use the Lookup Component feature to create a custom component instance to associate with the inventory item. Once the appropriate instance is selected, the **Name**, **Component**, and **License** fields on the **Inventory Details** tab are updated accordingly. See Using “Lookup Component” to Search for Components to Associate with Inventory for complete information.

   - **Converting to a different inventory type**—To convert a **Work In Progress** or **License Only** inventory item to a different inventory type, select the appropriate option in the **Type** field and perform any additional steps (if any) required for the type. See Creating Inventory From the Inventory Items List information about the additional steps. The **Name** and other fields on the **Inventory Details** tab are updated accordingly.

- **Quickly updating the license or version**—You can quickly update only the component version or its associated license by clicking the Edit ✎ icon next to either the **Component** or the **License** field. The **Edit Version/License** dialog is displayed, enabling you to select a different license, version, or both. (This procedure avoids performing the longer Lookup Component process to edit these elements.) You can also create a custom license from the dialog.

  If you select a new version or license (or create a new license), the **Update License Mapping** window might be displayed. This window provides the option to save the license mapping for the component version at the system level. If you select **Yes**, all future inventory system-generated for the component version will be mapped to this license. If you select **No**, the license mapping for the component version is updated in the database for the current inventory item only. (For more information about the **Update License Mapping** window and the option to save your license mapping at the system level, see Specifying a User-Preferred License Mapping.)

- **Access to details about component versions**—To help you make an informed selection of a component version, you can click the **View all versions** link to open the **Versions for** <**component**> window. From here, view the list of all versions for the component and, for each version, its associated licenses and security vulnerability totals (by severity). You can also delve into more detail for each associated vulnerability. If the appropriate version-license instance is not available, the window provides the option to create the missing instance. For further instructions, see Versions for <component> Window.

5. Click **Save** to save the changes to the inventory item. See the next section Automatic Review Triggered When Saving Existing Inventory in the Analysis Workbench for details about what happens during the save.

Automatic Review Triggered When Saving Existing Inventory in the Analysis Workbench

When you click **Save** after editing an existing inventory item, not only are your changes are saved, but, under certain conditions, an automatic review of the inventory item can also occur to approve or reject the item. This review is based on the policy profile associated with the project. See the following sections for more information:

- Conditions Triggering an Automatic Review
- Option for Not Overwriting a Current Review Status Set by a User
- Automatic Review Process
- Events After a Status Is Overwritten

For more information about the review policy profile, see Managing Policies to Automatically Review Inventory.

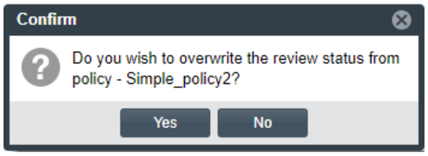
Conditions Triggering an Automatic Review

An automatic policy review is triggered upon saving an updated inventory item if the item you updated was already published *and* you edited the inventory item’s component, version, or license or any of its usage properties.

If these conditions are *not* met, no automatic review takes place. The edits to the inventory item are saved, but its status remains as is.

Option for Not Overwriting a Current Review Status Set by a User

If the current review status for the inventory item was manually set by a user, a pop-up window is displayed when you click **Save**, asking whether to allow the status to be overwritten by criteria in the current review policy (identified in the message).


Confirm

Do you wish to overwrite the review status from policy - Simple_policy2?

If you select **Yes**, your edits to the inventory item are saved and the automatic review proceeds. If you select **No**, the edits are saved, but no automatic review takes place (and the current status is maintained).

Note *▪ If the current review status was set automatically, no pop-up window is displayed. The automatic review simply proceeds.*

Automatic Review Process

When an automatic review is performed, it applies the current criteria in the review policy profile against the inventory item:

- If the inventory item meets at least one of the criteria, the item is assigned an **Approved** or **Rejected** status, overwriting the current status. (This same status is shown for the inventory item on the **Project Inventory** tab.)
- If the inventory item meets no criteria, its status remains the same.

Events After a Status Is Overwritten

When the review status is overwritten for an inventory item during an automatic review, the following events occur:

- The status change is recorded in **Inventory History** for the inventory item.
- Based on your project's configuration, a remediation task can be automatically created for the inventory item if it is rejected. For more information, see Updating Inventory Review and Remediation Settings for a Project.

Associating Codebase Files with Inventory in the Analysis Workbench

For instructions on associating codebase files with existing or newly created inventory, see Adding Files to Inventory From the Codebase List.

Creating a Custom Rule Based on the Files Associated with Inventory in the Analysis Workbench

During the auditing process for a project, you might find that one or more codebase files that are evidence of a specific third-party or OSS component are not being associated with inventory in your project. You must manually fix the situation—either by updating the existing inventory item to include the associated files or by creating the missing inventory item associated with the files.

Code Insight enables you to create a custom detection rule based on the file criteria of the inventory item that you had to create or update. Because you are creating this rule within the context of an existing inventory item, most of the fields that define the rule are pre-populated with details from the item, including the MD5 value for each file currently associated with the inventory item.

Refer to Managing Custom Detection Rules for further instructions.

*Note ▪ To create the rule based on an existing inventory item, the inventory item **Type** must be **Component**.*

Publishing or Recalling Inventory Manually from the Analysis Workbench

If you have performed manual work on your inventory items, you must publish the items to **Project Inventory** before anyone can review your work. Likewise, you can recall a published inventory item (that is, remove it from Project Inventory) for further auditing.

In the **Analysis Workbench**, you publish or recall inventory from either the **Inventory Details** tab or the **Inventory Items** pane:

- Publishing or Recalling Inventory from the Inventory Details Tab
- Publishing or Recalling Inventory from the Inventory Items Pane
- Automatic Review of Inventory at Publication

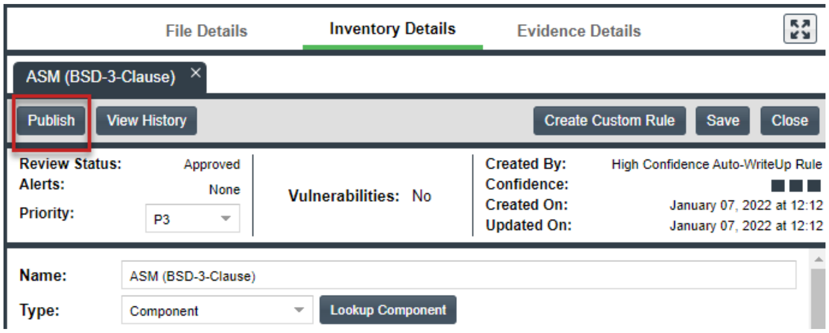
Note ▪ If you enabled the auto-publish feature in the project scan settings, you do not need to perform the steps below because system-created inventory items are automatically published.

Publishing or Recalling Inventory from the Inventory Details Tab

You these steps to publish or recall an inventory item from the Inventory Details tab.

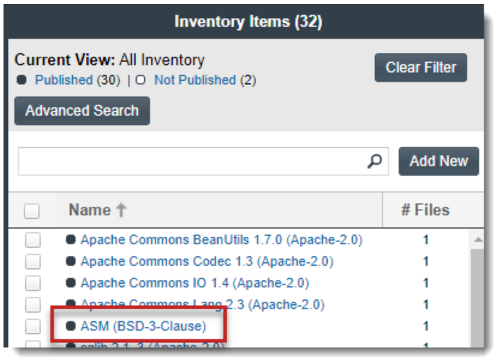
Task ***To publish or recall an inventory item from its Inventory Details tab, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. Select the inventory item from the **Inventory Items** pane so that the item’s details are displayed in the **Inventory Details** pane.
3. Do one of the following:
    - **Publish the inventory item**—For the unpublished inventory item currently in focus on the **Inventory Details** tab in the **Analysis Workbench**, click the **Publish** button.



Review Status: Approved
Alerts: None
Priority: P3
Vulnerabilities: No
Created By: High Confidence Auto-WriteUp Rule
Confidence: ■■■
Created On: January 07, 2022 at 12:12
Updated On: January 07, 2022 at 12:12
Name: ASM (BSD-3-Clause)
Type: Component

The newly published item now appears in the **Inventory Items** list with a filled box icon before its name (and is now also visible on the **Project Inventory** tab).

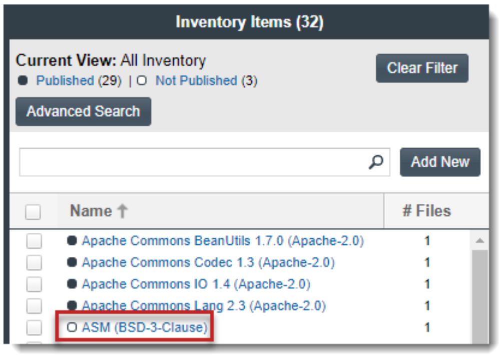


Current View: All Inventory
● Published (30) | ○ Not Published (2)

	Name	# Files
□	● Apache Commons BeanUtils 1.7.0 (Apache-2.0)	1
□	● Apache Commons Codec 1.3 (Apache-2.0)	1
□	● Apache Commons IO 1.4 (Apache-2.0)	1
□	● Apache Commons Lang 2.3 (Apache-2.0)	1
□	● ASM (BSD-3-Clause)	1

Once published, the inventory item is automatically reviewed by the policy profile associated with the project. For more information, see Automatic Review of Inventory at Publication.

- **Recall the inventory item**—For the published inventory item currently in focus, click the **Recall** button. The item now appears in the **Inventory Items** list with a clear-box icon before its name (and is no longer visible in **Project Inventory**).



Current View: All Inventory
● Published (29) | ○ Not Published (3)

	Name	# Files
□	● Apache Commons BeanUtils 1.7.0 (Apache-2.0)	1
□	● Apache Commons Codec 1.3 (Apache-2.0)	1
□	● Apache Commons IO 1.4 (Apache-2.0)	1
□	● Apache Commons Lang 2.3 (Apache-2.0)	1
□	○ ASM (BSD-3-Clause)	1

The review status of the inventory item before the recall is retained until item is re-published (and the latest review policy is applied).

Publishing or Recalling Inventory from the Inventory Items Pane

Use the following procedure to publish or recall one or more inventory items from the **Inventory Items** pane.

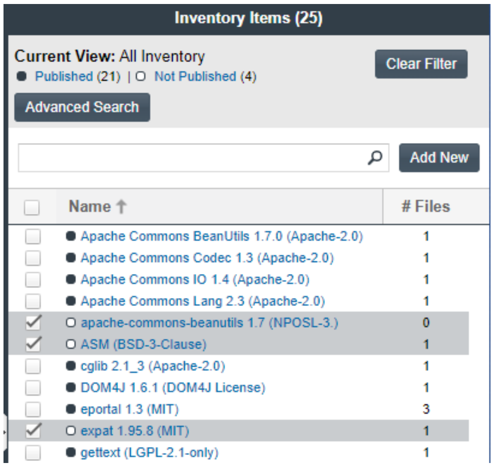
Task ***To publish or recall inventory from the Inventory Items pane, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. From the **Inventory Items** pane of the **Analysis Workbench**, select the items to publish so that a checkmark appears next to each item.

   or

   Select the published items you want to recall so that a checkmark appears to the left of each item.

   The following example shows unpublished items (with an clear-box icon before each name) selected for publication.



Inventory Items (25)

Current View: All Inventory
● Published (21) | ○ Not Published (4)

□	Name ↑	# Files
□	● Apache Commons BeanUtils 1.7.0 (Apache-2.0)	1
□	● Apache Commons Codec 1.3 (Apache-2.0)	1
□	● Apache Commons IO 1.4 (Apache-2.0)	1
□	● Apache Commons Lang 2.3 (Apache-2.0)	1
☑	○ apache-commons-beanutils 1.7 (NPOSL-3.)	0
☑	○ ASM (BSD-3-Clause)	1
□	● cglib 2.1_3 (Apache-2.0)	1
□	● DOM4J 1.6.1 (DOM4J License)	1
□	● eportal 1.3 (MIT)	3
☑	○ expat 1.95.8 (MIT)	1
□	● gettext (LGPL-2.1-only)	1

Note *▪ To help you locate the inventory item(s) by name, enter a name string in the search box above the list and click the search button.*

3. Right-click to open the context menu, and choose either **Publish Inventory** or **Recall Inventory**.
   - If you selected **Publish Inventory**, the newly published items appear in the **Inventory Items** list with a filled-box icon before their names (and are now visible in **Project Inventory**).

     Once published, the inventory item is automatically reviewed by the policy profile associated with the project. For more information, see Automatic Review of Inventory at Publication.
   - If you selected **Recall Inventory**, the recalled items appear in the **Inventory Items** list with a clear-box icon before their name (and are no longer visible in **Project Inventory**). The review status of the inventory item before the recall is retained until item is re-published (and the latest review policy is applied).

Automatic Review of Inventory at Publication

Upon publication, the inventory item is automatically reviewed by the review policy profile associated with the project. The item is either approved or rejected based on the policy criteria; or, if no criteria applies, the item is placed in a **Draft** state (or **Not Reviewed** on the **Project Inventory** tab).

Note *▪ Based on your project's configuration, additional events can occur once an inventory item is rejected or assigned a **Not Reviewed** status. (For example, a **Rejected** status can automatically create a remediation task for the inventory item.) See Updating Inventory Review and Remediation Settings for a Project for more information.*

With the availability of the inventory item on the **Project Inventory** tab, users can then further review the item's security or legal issues and, if appropriate, take steps to remediate and prepare the item for inclusion in the final Third-Party Notices report for the project.

Viewing the Update History for an Inventory Item in the Analysis Workbench

From the **Analysis Workbench**, you can view a history of the updates made to a specific inventory item within the context of a project.

Note *▪ You have access to this same history from the **Project Inventory** tab. See Viewing the Update History for an Inventory Item in Project Inventory.*

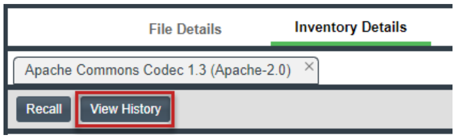
Task ***To view the update history of an inventory item, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)

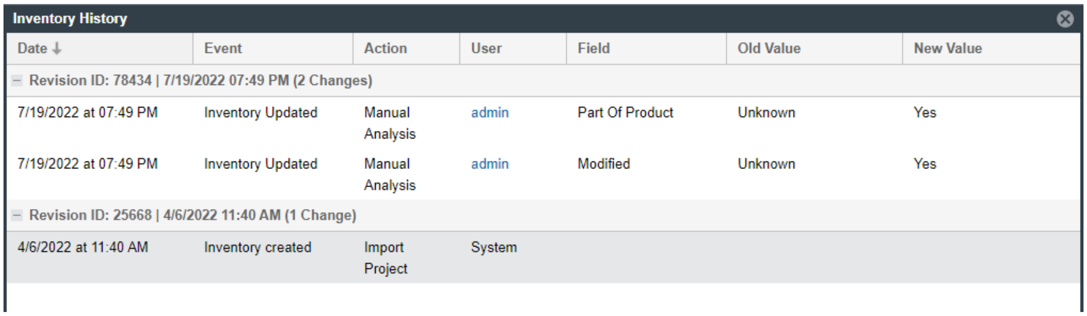
2. From the **Inventory Items** pane of the **Analysis Workbench**, select the inventory item whose update history you want to view.

   The **Inventory Details** pane is opened (or refreshed) with information about the selected inventory item.

3. In the **Inventory Details** pane, click the **View History** button.

   The **Inventory History** window is opened, showing the list of updates made to the inventory item within the project. By default, the updates are listed in descending order by date so that you see the most recent updates first. Each update record identifies—among other details—the update type, the user who made the update, and the before-and-after values in the update. For a description of all features on this window, see Inventory History Window.



Inventory History

Date ↓	Event	Action	User	Field	Old Value	New Value
Revision ID: 78434 | 7/19/2022 07:49 PM (2 Changes)						
7/19/2022 at 07:49 PM	Inventory Updated	Manual Analysis	admin	Part Of Product	Unknown	Yes
7/19/2022 at 07:49 PM	Inventory Updated	Manual Analysis	admin	Modified	Unknown	Yes
Revision ID: 25668 | 4/6/2022 11:40 AM (1 Change)						
4/6/2022 at 11:40 AM	Inventory created	Import Project	System			

Reviewing Published Inventory for a Project

The **Project Inventory** tab shows a list of all the inventory items that have been published for the current project, either automatically by the system or manually by a Reviewer or Analyst. From the **Project Inventory** tab, users can view details for the inventory item, and designated reviewers for the project can manage existing inventory (that is, set the status, change inventory priority, edit details, create and inventory, and manage review and remedial tasks).

Refer to the Code Insight User Roles and Permissions section for the project roles (in addition to the Reviewer and Analyst roles) required to review and act on published project inventory.

The following topics describe the various actions you can perform review and manage project inventory:

- Goal of the Reviewer
- Success Messages When Working with Inventory
- Displaying Project Inventory
- Searching Published Inventory
- Viewing Security Vulnerabilities for Project Inventory
- Viewing Details About the Licenses Associated with Project Inventory
- Viewing and Updating Notes and Guidance
- Viewing Usage Information for Project Inventory
- Viewing Associated Files
- Creating Inventory from the Project Inventory Tab
- Editing Inventory from the Project Inventory Tab
- Approving or Rejecting Inventory Items
- Creating and Managing Tasks for Project Inventory
- Creating and Viewing External Work Items for a Project Inventory Task
- Recalling a Published Inventory Item
- Viewing the Update History for an Inventory Item in Project Inventory

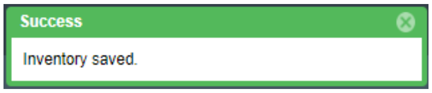
Goal of the Reviewer

The goal of the inventory review is to assess every inventory item and categorize it as *approved* or *rejected* for use in the current project based on your company policy. To review inventory, the user first must be assigned the role of Reviewer (or a role with Reviewer permissions). See Assigning and Removing Project Users.

Success Messages When Working with Inventory

When you perform certain inventory operations such as creating or editing inventory (and others), a message box is displayed in the upper right corner of the Code Insight user interface to inform you of one of the following:

- The operation has successfully completed.
- The operation has been successfully initiated as a background job in the **Jobs** queue. The message includes the job ID so that you can track the job (as described in Monitoring the Code Insight Jobs Queue).



Success

Inventory saved.

The message persists for a couple of seconds, but you can click the ⊗ button in the box to close the message sooner.

Displaying Project Inventory

When an inventory item has been published, it can be reviewed, updated, and reported on from the **Project Inventory** tab. Use this procedure to display the **Project Inventory** tab.

Task ***To view project inventory, do the following:***

1. Open the project whose published inventory you want to review. (For instructions, see Opening a Project.)
2. Click the **Project Inventory** tab. For details about this tab, see Project Inventory Tab.
3. From the **Inventory Items** list on the left, select the inventory item you want to review.

   Details for the selected inventory item populate the **Project Inventory Details** pane on the right. From this pane, you can edit inventory properties, recall an inventory item, finalize the third-party Notices content, set up review and remediation tasks, and provide audit, usage-guidance, and remediation notes—with the ultimate goal of approving or rejecting the item for the Bill of Materials. For complete information about this pane, see Project Inventory Details Pane.

Searching Published Inventory

You can filter the **Inventory Items** list to focus on the inventory you want to examine. The following sections describe the filtering methods:

- Filtering Inventory by Name
- Performing an Advanced Inventory Search

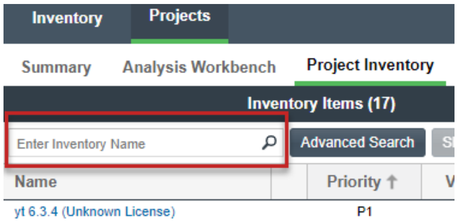
Filtering Inventory by Name

You can filter the inventory by an inventory name or a string within the name.

***Note** ▪ The name filter you define is also copied to the Advanced Inventory Search feature should you use this feature (see Performing an Advanced Inventory Search). Likewise, if you enter a name filter in the Advanced Inventory Search feature, it is copied to the **Inventory Items** pane. This behavior enables you to keep the name filter persistent. In either location, the filter can be removed or replaced as needed.*

***Task** To filter the inventory by name, perform this step:*

In the **Enter Inventory Name** field above the **Inventory Items** list, provide the inventory name or a partial-name string. As you type each character, the list is automatically filtered according to the entered characters.

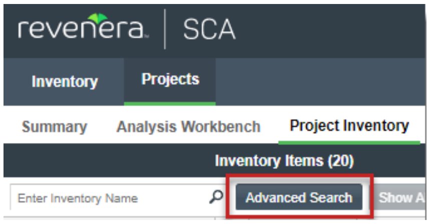


Performing an Advanced Inventory Search

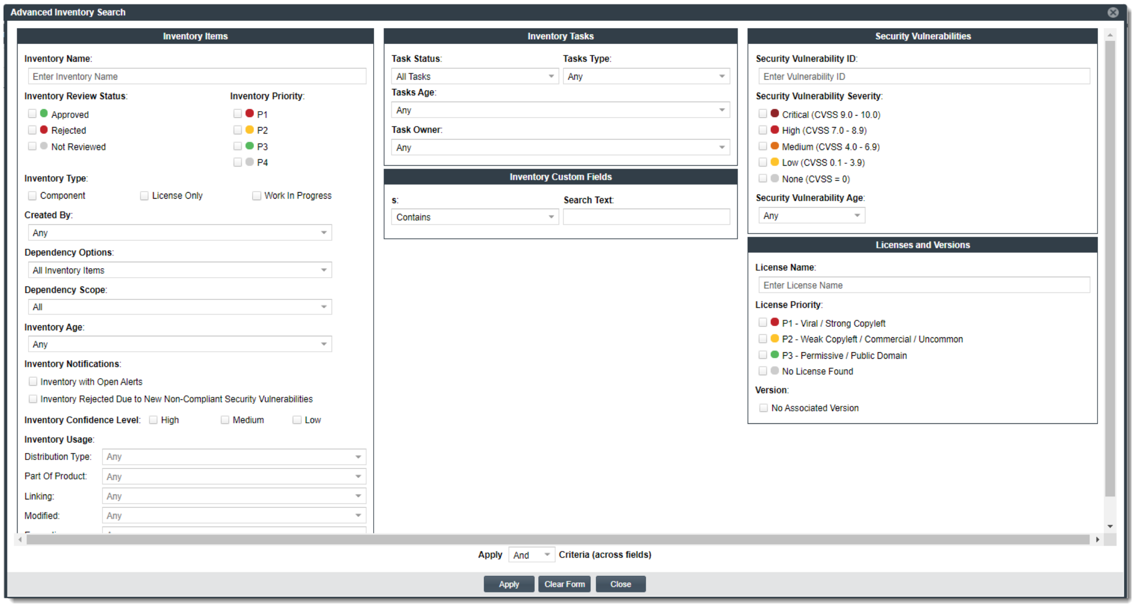
Code Insight provides the **Advanced Inventory Search** dialog that enables you to filter the list of published inventory items to those of interest based on many available criteria—inventory attributes, selected license attributes, and associated security vulnerabilities, tasks, and security alerts. In this way, you can easily focus on only those inventory items in which you are interested within the list of published items. The following procedure shows you how to access and use this dialog. Refer also to Performing Advanced Searches for practical applications of this search feature.

***Task** To filter published inventory, do the following:*

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory). The **Inventory Items** pane appears, showing the list of inventory items.
2. Click the **Advanced Search** button above the list of inventory items.



The **Advanced Inventory Search** dialog is opened.



Advanced Inventory Search

Inventory Items

Inventory Name: Enter Inventory Name

Inventory Review Status: □ ● Approved □ ● Rejected □ ● Not Reviewed

Inventory Priority: □ ● P1 □ ● P2 □ ● P3 □ ● P4

Inventory Type: □ Component □ License Only □ Work In Progress

Created By: Any

Dependency Options: All Inventory Items

Dependency Scope: All

Inventory Age: Any

Inventory Notifications:
□ Inventory with Open Alerts
□ Inventory Rejected Due to New Non-Compliant Security Vulnerabilities

Inventory Confidence Level: □ High □ Medium □ Low

Inventory Usage:
Distribution Type: Any
Part Of Product: Any
Linking: Any
Modified: Any

Inventory Tasks

Task Status: All Tasks
Tasks Type: Any
Tasks Age: Any
Task Owner: Any

Inventory Custom Fields

s: Contains
Search Text:

Security Vulnerabilities

Security Vulnerability ID: Enter Vulnerability ID

Security Vulnerability Severity:
□ ● Critical (CVSS 9.0 - 10.0)
□ ● High (CVSS 7.0 - 8.9)
□ ● Medium (CVSS 4.0 - 6.9)
□ ● Low (CVSS 0.1 - 3.9)
□ ● None (CVSS = 0)

Security Vulnerability Age: Any

Licenses and Versions

License Name: Enter License Name

License Priority:
□ ● P1 - Viral / Strong Copyleft
□ ● P2 - Weak Copyleft / Commercial / Uncommon
□ ● P3 - Permissive / Public Domain
□ ● No License Found

Version: □ No Associated Version

Apply And Criteria (across fields)

3. From this dialog, select search criteria as needed from the following categories. For a detailed description of the search criteria, see Advanced Inventory Search Dialog.

    - **Inventory Items**—Search for inventory items of a certain name (or string), review status, priority, type, creator, dependency scope, age, usage, or confidence level or that have open vulnerability alerts and work items. (For details on alerts and work items, see Managing Security Vulnerability Alerts and Creating and Viewing External Work Items for a Project Inventory Task.)

      If you entered a name filter in the **Inventory Name** field on the **Inventory Items** pane, it is automatically displayed for the **Inventory Name** filter on the **Advanced Inventory Search** dialog. (Likewise, if you enter a name filter on the **Advanced Inventory Search** dialog, it is copied to the **Inventory Items** pane.) This behavior enables you to keep the name filter persistent. However, you can remove or replace this filter as needed in either location.

    - **Inventory Tasks**—Search for inventory items that have been assigned tasks. You can refine the search to locate inventory with open or closed tasks, tasks of a certain age or type (such as manual reviews or source-code remediation), or tasks assigned to a specific user.

    - **Inventory Custom Fields**—Search for inventory whose custom inventory fields contain the value you specify as criteria (or contain no value). Custom inventory fields are defined specifically for your site. If no such fields have been defined this section is not visible.

    - **Security Vulnerabilities**—Search for inventory items that have vulnerabilities of a certain vulnerability ID, CVSS severity, or age. (Note that list of available severities for **Security Vulnerability Severity** varies depending on the CVSS version being used by Code Insight. The picture above shows the severities for CVSS v3.x. See Working with Security Vulnerabilities for details.)

    - **Licenses**—Search for inventory items that have licenses of a certain of a certain name or license priority.

4. In **Apply Criteria** field, select the boolean operator to apply to the criteria:

    - **Or**—To be included in the search results, an inventory item must contain at least one of the criteria you selected on this dialog.

- **And**—To be included in the search results, an inventory item must meet all the criteria across the advanced search, as selected in this dialog. (This is the default operator.)

5. Click **Apply** to filter the inventory on the **Inventory Items** pane to display only those inventory items that meet the selected criteria.
6. To refresh the **Inventory Items** pane to show all inventory items, click **Show All Items**.

Viewing Security Vulnerabilities for Project Inventory

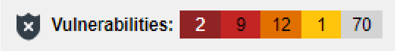
Code Insight uses data from the National Vulnerability Database (NVD), Secunia advisories (as published by the Secunia Research team from Revenera), and other advisories such as RubySec to report security vulnerabilities associated with your inventory items. The vulnerabilities information from these sources is used to create vulnerability rankings and alerts.

Use this procedure to access details for the vulnerabilities associated with an inventory item on the **Project Inventory** tab.

Task ***To view security vulnerabilities for an inventory item, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. Click a published inventory item from the **Inventory Items** list. The Project Inventory Details Pane on the right opens to the **Inventory Details** tab.

   If known security vulnerabilities exist for the inventory item, the **Vulnerabilities** bar graph is displayed:

   The severity levels depicted in the graph differ depending on the version of CVSS Code Insight is using (see Security Vulnerabilities Associated with Inventory). This example shows vulnerability severity counts using CVSS v3.x.
3. Click any of the counts in the graph to open the **Security Vulnerabilities** window, which lists current security vulnerabilities for the inventory item.

   Note ▪ *Suppressed vulnerabilities are neither reflected in the counts on* ***Vulnerabilities*** *bar graph nor are they visible on* ***Securities Vulnerabilities*** *window.*

   For more information about how to use this dialog to obtain details about the vulnerabilities, see Working with Security Vulnerabilities.
4. When you have finished viewing the reported vulnerabilities, click **OK** to return to the **Inventory Items** list.

Viewing Details About the Licenses Associated with Project Inventory

You can view more details about the licenses associated with the OSS or third-party component on which the current inventory item is based. This information is pulled from the Code Data Library and is displayed on the **License Details** window, which includes the following:

- A **General Information** tab that lists details such as the name of the license, its family, and the license priority assigned by Code Insight.
- A **License Text** tab that displays the complete license text (representing the external forge license text).

While the forge license text is what is likely to be used by the component, the scan might have found actual license text associated with this component in your codebase that can be different from the forge version. To view the exact license text, if any, that the scan discovered and to prepare the final license text, when necessary, to include in the Notices report for distribution to customers, navigate to the **Notices Text** tab. Refer to Finalizing the Notices Text for the Notices Report for more information.

The following procedure describes how to access the **License Details** window from the **Component Details** tab on the **Project Inventory** tab.

***Note** ▪ This window is also accessible when create or edit an inventory item or perform a Lookup Component search from the **Project Inventory** tab.*

Task ***To view details for the inventory license, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. Select a published inventory item from the **Inventory Items** list.
3. From the Project Inventory Details Pane on the right, do either:

    - For a component-based inventory item, click the **Component Details** tab. Then click information ⓘ icon next to the **Selected License** value (the license currently associated with the component) or the **Possible Licenses** (other valid license candidates with which you could associate the inventory item).
    - For a License Only inventory item, click the **License Details** tab, and then click the information ⓘ icon next to the license name.

    The **License Details** window appears with the **General Information** tab in focus. For descriptions of these fields on this tab, see License Details Window. Also see License Priority for background on how the license priority is used.
4. Select the **License Text** tab to view the license text.
5. When you have finished examining the license details, click **Close**.

Viewing and Updating Notes and Guidance

The **Notes & Guidance** tab can provide notes about the automated and manual analysis performed on the codebase as it relates to the current inventory item. The tab can also include guidance on how to remediate issues associated with your product's use of the OSS or third-party software identified by the inventory item.

Task ***To view notes and guidance, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. Select an inventory item from list.
3. From the Project Inventory Details Pane on the right, select the **Notes & Guidance** tab.
4. Review or update content in the following fields as needed. All information is editable except for the information in the **Detection Notes** field:
    - **Detection Notes**—Information generated during the scan to explain the means by which OSS or third-party component was detected in the codebase. This information is not editable. For more details, see Viewing and Updating Detection and Auditing Notes in the Analysis Workbench.
    - **Audit Notes**—Information recorded about the analysis of the code associated with the component in your codebase. For example, these notes might indicate that the inventory item for the component needed to be manually created based codebase evidence that was not detected in scan.
    - **Usage Guidance**—Two kinds of Information: 1) Information propagated from policies that rejected or approved the inventory during the automatic review process that occurred when the inventory was published. This content can explain why the item was rejected or provide requirements and recommendations for using those items that were approved. You cannot edit this content. 2) Reviewers' own notes and concerns about the use of the component in your product software. This information is editable.
    - **Remediation Notes**—A description of items to be addressed or actions to be taken before the use of this software in your product is acceptable from a legal or security standpoint.
5. Click **Save** in any field in which you have made changes.
6. When you have finished with this tab, navigate to another tab for the inventory item, or select another inventory item.

Viewing Usage Information for Project Inventory

Inventory usage information is important because it aids users in determining how closely to monitor inventory items for intellectual property (IP) and security risk and in taking appropriate action to approve or reject inventory, create tasks for remediation, and issue alerts and notifications. Usage fields can determine whether the item should be included in Third-Party Notices and what steps need to be taken to satisfy license obligations and conditions of use. They can also help to identify license conflicts and compatibility issues.

Task ***To view inventory usage information, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. Click an inventory item from the **Inventory Items** list.
3. From the Project Inventory Details Pane on the right, select the **Usage** tab in the inventory details. For details about the inventory usage fields and how they are used, see Inventory Usage Information.

Viewing Associated Files

Associated files are files in your codebase that have been automatically or manually associated with the inventory item selected in the **Inventory Items** pane due to evidence found in the files.

Task ***To view associated files, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. Click an inventory item from the **Inventory Items** list.
3. From the Project Inventory Details Pane on the right, select the **Associated Files** tab in the inventory details to view the list of files associated with the selected inventory item. Each file entry shows the following:
    - **Alias**—The unique, user-defined name provided during scanner setup (for a Scan Server or a remote scan agent) to represent its scan-root path containing the codebase in which the file is located. The alias provides a name that is more meaningful than the scan-root path.
    - **File Path**—The file’s path relative to the scan-root path on instance hosting the scanner.

      If you have Analyst permissions, the path is hyperlinked to open to the file’s **File Details** tab in the **Analysis Workbench**, where you can view file evidence. If necessary, while in the **Analysis Workbench**, you can also add or remove files associated with the inventory. If you do not have Analyst permissions, the path remains in plain text.
4. When you have finished viewing associated files, select another tab or click another item listed in the **Inventory Items** pane.

Creating Inventory from the Project Inventory Tab

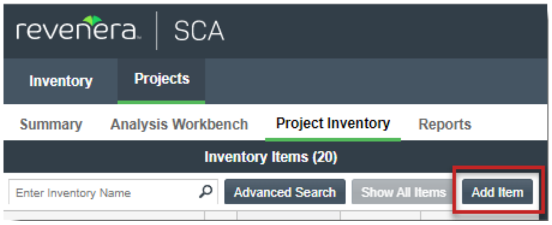
Reviewers can create an inventory item to represent any third-party code or artifact that is not automatically detected by the system.

Use the following steps to create an inventory item from the **Project Inventory** tab as needed. Note the following:

- When you save the inventory item, it is automatically published.
- No files can be associated with an inventory item when it is created from the **Project Inventory** tab.
- If you register a new component instance (a unique component-version-license combination) when creating inventory, the registered instance becomes available for selection across the system.
- Inventory of type **Work in Progress**, **Component**, or **License Only** can be created.

Task ***To create an inventory item from the Project Inventory tab, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. Click **Add Item** at the top of the **Inventory Items** list.



The **New Inventory** dialog opens.

3. For the **Name** field, perform the appropriate step, based on the inventory type you intend to select for the **Type** field (see the next step):

   - For inventory of the type **Work in Progress**, specify a name for the inventory item. Best practice is to provide a name in the following conventional syntax used by Code Insight, even if the elements represented in the name are not available in the Data Library:

     ```
     <Component_name> <version> <License_name>
     ```

   - For inventory of the type **Component** or **License only**, leave the **Name** field blank. The field will be automatically populated based on the registered component or license instance.

4. From the **Type** dropdown list, select the type of inventory item you want to create and then perform the related step or steps:

   - **Work in Progress**—Create this type of inventory item if you want to quickly represent third-party code or an artifact without having to select an associated component, version, or license from the Code Insight Data Library. (You can later edit this inventory item to convert it to one of the other inventory types.) This option is typically used if you need a placeholder or cannot find the associated element in the Data Library. Items of type **Work in Progress** are not affected by policies and do not receive vulnerability updates or alerts.

   - **Component**—Create this type of inventory item if third-party code or artifacts point to a definite component version and possibly its license. You need to associate this type of inventory with a registered component instance—that is, a unique component-version-license combination found in the Code Insight Data Library. Use the Lookup Component feature, made available when you select the **Component** type, to locate this component instance and associate it with the inventory item. If are unable to locate the appropriate instance, the Lookup Component feature enables you to create a custom component. See Using “Lookup Component” to Search for Components to Associate with Inventory for further instructions.

     Once the instance is associated with inventory item, the **Name**, **Description**, **Component**, and **License** fields on the **Inventory Details** tab are automatically populated with information based on the selected instance. Additionally, Information ⓘ icons are available next to the **Component** and **License** fields so that you can view publicly available information about the selected component or its license.

     This inventory type is affected by policies and receives vulnerability updates and alerts.

- **License Only**—Create this type of inventory item if evidence shows groups of codebase files of unknown origin are governed by a specific license. (You can later edit this inventory item to convert it to one of the other inventory types.) This inventory type *is* affected by policies

  When creating **License Only** inventory, also select the appropriate license from **License** dropdown list, which is enabled when you select this type.

  The **Name** field for the inventory item is automatically populated with the name **Files under**

  <**License_name**>, where <License_name> is license you selected. The ⓘ icon is added so that you can view details about the selected license.(You can also click **New** to create a custom license. See When Creating or Editing a “License Only” Inventory Item for further instructions.)

5. Update the remaining fields if appropriate. For a description of each field, see Project Inventory Details Pane.

   Click **Save**. The inventory item is added as a published item to the **Inventory Items** list on the **Project Inventory** tab and in the **Analysis Workbench**.

   Additionally, when the new inventory item is saved, it is it automatically reviewed by the review policy profile associated with project. (See Managing Policies to Automatically Review Inventory for more information.)

   - If the inventory item meets at least one of the criteria in the review policy, the item is assigned an **Approved** or **Rejected** status, overwriting the current status.
   - If the inventory item does not meet any of the policy criteria, the item is assigned the status **Not Reviewed**, indicating that a manual review is required.

   Note ▪ *Based on your project’s configuration, additional events can occur once an inventory item is rejected or assigned a **Not Reviewed** status. (For example, a **Rejected** status can automatically create a remediation task for the inventory item.) See Updating Inventory Review and Remediation Settings for a Project for more information.*

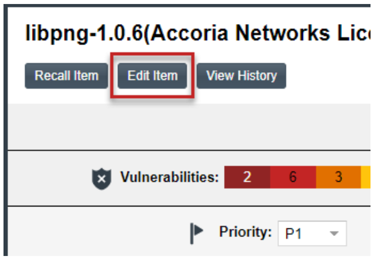
6. (Optional) If you created a **License Only** inventory item, view details about the license selected for the new inventory item on the **Licenses Details** tab in the right pane.

Editing Inventory from the Project Inventory Tab

Use the following steps to edit an inventory item from the **Project Inventory** tab for a project as needed.

Task ***To edit an inventory item from the Project Inventory tab, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. In the **Inventory Items** list, select the inventory item that you want to edit. Information about the inventory item is displayed in the right pane.
3. In the header on the right pane, click the **Edit Item** button next to the component name.



The **Edit Inventory** dialog opens.

4. Make changes to the fields as needed. Refer to Project Inventory Details Pane for all field descriptions, and see Creating Inventory from the Project Inventory Tab for additional steps required when updating the inventory type. Also note the following information:

    - **Using the Lookup Component feature**—For a **Component** inventory item, you can use the Lookup Component feature to select a different registered instance (version and license) for the current component, select a different component altogether, or select a component instance for the first time (if you are changing the inventory item from a **Work in Progress** or **License Only** type). You can also use the Lookup Component feature to create a custom component instance to associate with the inventory item. Once the appropriate instance is selected, the **Name**, **Component**, and **License** fields on the **Inventory Details** tab are updated accordingly. See Using “Lookup Component” to Search for Components to Associate with Inventory for complete information.

    - **Converting to a different inventory type**—To convert a **Work In Progress** or **License Only** inventory item to a different inventory type, select the appropriate option in the **Type** field and perform any additional steps (if any) required for the type. See Creating Inventory from the Project Inventory Tab information about the additional steps. The **Name** and other fields on the **Inventory Details** tab are updated accordingly.

    - **Quickly updating the license or version**—You can quickly update only the component version or its associated license by clicking the Edit ✎ icon next to either the **Component** or the **License** field. The **Edit Version/License** dialog is displayed, enabling you to select a different license, version, or both. (This procedure avoids performing the longer Lookup Component process to edit these elements.) You can also create a custom license from the dialog.

      If you select a new version or license (or create a new license), the **Update License Mapping** window can be displayed. This window provides the option to save the license mapping for the component version at the system level. If you select **Yes**, all future inventory system-generated for the component version will be mapped to this license. If you select **No**, the license mapping for the component version is updated for the current inventory item only. (For more information about the **Update License Mapping** window and the option to save your license mapping at the system level, see Specifying a User-Preferred License Mapping.)

    - **Access to details about component versions**—To help you make an informed selection of a component version, you can click the **View all versions** link to open the **Versions for** <**component**> window. From here, view the list of all versions for the component and, for each version, its associated licenses and security vulnerability totals (by severity). You can also delve into more detail for each associated vulnerability. If the appropriate version-license instance is not available, the window provides the option to create the missing instance. For further instructions, see Versions for <component> Window.

    - No additional files can be associated with an inventory item from **Project Inventory** tab.

5. Click **Save** to change the changes to the inventory item. See the next section Automatic Review When Saving Existing Inventory on the Project Inventory Tab for details about what happens during the save.

Automatic Review When Saving Existing Inventory on the Project Inventory Tab

When you click **Save** after editing an existing inventory item, not only are your changes are saved; but, under certain conditions, an automatic review of the inventory item can also occur to approve or reject the item. This review is based on the policy profile associated with the project. See the following sections for more information:

- Condition Triggering an Automatic Review
- Option for Not Overwriting a Current Review Status Set by a User
- Automatic Review Process
- Events After a Status Is Overwritten

For more information about the review policy profile, see Managing Policies to Automatically Review Inventory.

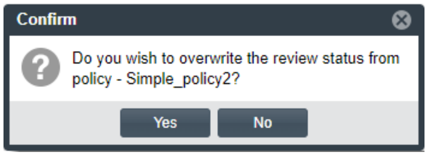
Condition Triggering an Automatic Review

An automatic policy review is triggered upon saving an updated inventory item if you edited the inventory item's component, version, or license or any of its usage properties.

If you did not edit any of these properties, no automatic review takes place and the inventory item's status remains as is.

Option for Not Overwriting a Current Review Status Set by a User

If the current review status for the inventory item was manually set by a user, a pop-up window is displayed when you click **Save**, asking whether to allow the status to be overwritten by criteria in the current review policy (identified in the message).



Do you wish to overwrite the review status from policy - Simple_policy2?

If you select **Yes**, your edits to the inventory item are saved and the automatic review proceeds. If you select **No**, the edits are saved, but no automatic review takes place (and the current status is maintained).

***Note** ▪ If the current review status was set automatically, no pop-up window is displayed. The automatic review simply proceeds.*

Automatic Review Process

When an automatic review is performed, it applies the current criteria in the review policy profile against the inventory item:

- If the inventory item meets at least one of the criteria, the item is assigned an **Approved** or **Rejected** status, overwriting the current status. (This same status is shown for the inventory item on the **Project Inventory** tab.)
- If the inventory item meets no criteria, its status remains the same.

Events After a Status Is Overwritten

When the review status is overwritten for an inventory item during an automatic review, the following events occur:

- The status change is recorded in **Inventory History** for the inventory item.
- Based on your project's configuration, a remediation task can be automatically created for the inventory item if it was rejected. For more information, see Updating Inventory Review and Remediation Settings for a Project.

Approving or Rejecting Inventory Items

The next step in the Code Insight workflow is to have security and legal experts review all published inventory and categorize them as approved or rejected for use in the software project. To approve or reject an inventory item manually, perform the following steps.

Task ***To approve or reject inventory items, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. In the row for the inventory item you want to approve or reject, click the green checkmark to approve the item or the red X to reject the item.

   A circle appears around the status icon to indicate it has been selected. A circle around the question mark indicates that no status selection has been made (that is, the inventory item requires further review to determine its status).

   Note that, depending on the inventory review and remediation options defined for the project, selecting the **Reject** status can automatically create a Remediate Inventory task. For more information, see Updating Inventory Review and Remediation Settings for a Project.

Creating and Managing Tasks for Project Inventory

The following sections describe the types of tasks that can be associated with project inventory and how to manage them:

- Task Types
- About Work Items
- Manually Creating a Task
- Opening the Tasks List
- Editing a Task
- Closing or Reopening a Task Directly from the Tasks List
- Effects of Closing Manual Review Tasks on Inventory Status

Task Types

Users with access to the project inventory (and edit privileges) can create and manage one or more tasks for a given inventory item. Tasks can be one of three types:

- **Manual Review Inventory**—A task to track the manual review of an inventory item, typically an inventory item that has not already been auto-reviewed by policy. A **Manual Review Inventory** task alerts the assignee to the need to review the inventory item within its current context. Closing this type of task with an **Approve** or **Reject** resolution can automatically approve or reject the inventory item. See Effects of Closing Manual Review Tasks on Inventory Status for details.

- **Remediate Inventory**—A task to track a remediation effort on the inventory item (typically a rejected item). A remediation task signals to the assignee to perform some action to make the inventory item acceptable for use (for example, to upgrade to a new version due to discovered vulnerabilities or to use a specific license and to comply with license obligations). Closing a remediation task does not automatically change the inventory review status.

- **Miscellaneous**—A task to track any other effort for an inventory item. Closing a **Miscellaneous** task does not automatically change the inventory review status.

Note that a task can also be created automatically in an automated workflow process (along with work items) based on review and remediation options up for the project, as described in Updating Inventory Review and Remediation Settings for a Project. Users with proper permissions can then manage these and manually created tasks, using the procedures described in this section. (For user roles that can manage tasks, see Roles and Permissions to Manage the Review Task Flow.)

About Work Items

If the Code Insight project is configured to connect to an external ALM (application lifecycle management) system such as Jira, a given task for an inventory item can have one or more associated work items. Work items keep track of the work that needs to be performed outside of Code Insight to address the task. When successfully created, a work item automatically sets up a corresponding issue in the external ALM system. A periodic synchronization process keeps the work item in Code Insight up to date with the state of its corresponding issue in the ALM system, enabling you to keep track of the remedial work performed for the inventory item.

Currently, Code Insight supports the creation of issues in a Jira ALM system only.

Creating Work Items

A work item can be created manually in the UI or automatically as part of a project’s automated workflow settings. A work item can be created only if the project is associated with *an ALM instance*, which defines a set of attributes used to connect to the ALM system and to set up work item. Once the project is associated with one of these instances, these attributes can be customized as needed to create work items.

For Further Instructions

For further instructions about setting up work items, refer to the following documentation:

- For details about associating a project with an ALM instance, refer to Associating the Project with an Application Life Cycle System to Create Work Items.

- For instructions about manually creating work items through the UI, see Creating and Viewing External Work Items for a Project Inventory Task.

- For instructions about creating work items as part of a project's automated workflow for inventory review and remediation, see Updating Inventory Review and Remediation Settings for a Project.

Manually Creating a Task

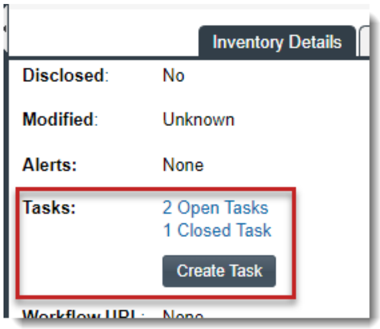
The following procedure describes the manual process for creating a task from the **Inventory Details** tab for a selected inventory item on the **Project Inventory** tab. You can also create a task from the **Tasks** list (see Opening the Tasks List).

Note ▪ *A task can also be created automatically in an automated workflow process (along with external work items) based on review and remediation options up for the project, as described in Updating Inventory Review and Remediation Settings for a Project. Users with proper permissions can then manage these tasks and manually created tasks, using the procedures described in this section.*

Task ***To create a task manually, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. Select the inventory item to which you want to add a task. Alternatively, to help you locate the inventory item, click the **Advanced Search** button to open the **Advanced Inventory Search** dialog. From here you can filter inventory items accordingly.

   When you select the specific inventory item, the Project Inventory Details Pane on the right is populated with information about the inventory item. The **Inventory Details** tab in focus.
3. In the **Tasks** section on the tab, click the **Create Task** button to open the **Create Task** dialog.

   Disclosed: No

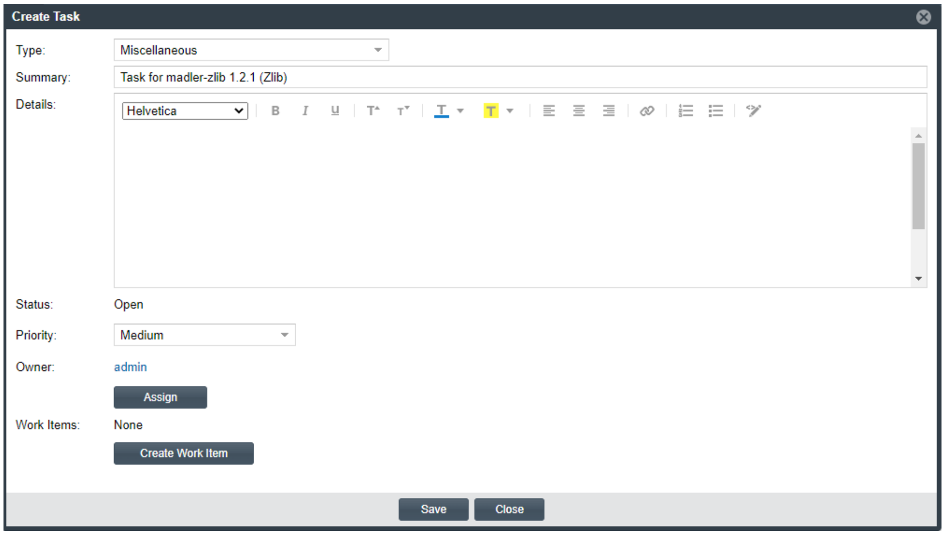
   Modified: Unknown

   Alerts: None

   Tasks: 2 Open Tasks
   1 Closed Task

   Workflow URL: None

   The **Create Task** window is opened.



Create Task

Type: Miscellaneous

Summary: Task for madler-zlib 1.2.1 (Zlib)

Details:

Status: Open

Priority: Medium

Owner: admin

Work Items: None

4. Select the type of task you want to create—**Manual Inventory Review**, **Remediate Inventory**, or **Miscellaneous**. (See Task Types.)

5. Complete the following fields as needed:

   - In the **Summary** field, provide a summary or title for the task.
   - In the **Details** field, provide instructions or requirements for completing this task (or provide any information that will be useful to the reviewer).
   - In the **Priority** field, assign a **High**, **Medium**, or **Low** priority to the task.
   - Keep the **Status** as **Open** for a new task.

6. To change the task owner, click the **Assign** button under the **Owner** field, and select a new owner.

   The initial task owner defaults to one of the following contacts, depending on the task type:

   - The Project Contact for **Miscellaneous** tasks
   - The project’s Legal Contact for **Manual Inventory Review** tasks
   - The project’s Developer Contact for **Remediate Inventory** tasks

   For more information about the these contacts, see Summary Tab.

7. To create an external work item associated with the task, click the **Create Work Item** button. (See Creating and Viewing External Work Items for a Project Inventory Task for details.)

   A “Success” message is displayed if the work item is created successfully on the corresponding ALM system (currently, a Jira server) associated to this project.

   You can repeat this step to create another work task.

8. Click **Save** to create the task. The **Tasks** list opens, showing the task you created.

   A notification email is automatically sent to the task owner, notifying that person about the new task assigned to them.

Opening the Tasks List

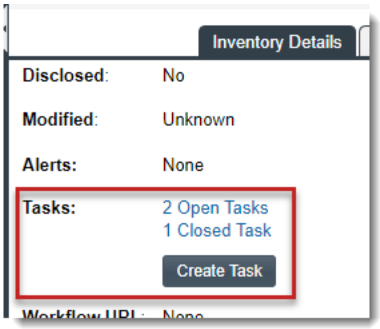
Use the following procedure to open the **Tasks** list, which shows the open and closed tasks associated with the current inventory item. From this list, you create a task or open the **Task Details** dialog for an existing task to edit its status and other task attributes. Alternatively, you can change the status for a task directly from the list.

From the **Tasks** list, you can also click a link to send an email to the owner or creator of a given task or to the user who closed a task.

Task ***To open the Tasks list, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. Select the inventory item to which the task you want to edit is associated. Alternatively, click the **Advanced Search** button to open the **Advanced Inventory Search** dialog, where you can select filters that help you pinpoint the inventory item. For example, you can select to filter on those inventory items associated with tasks that are assigned to a specific user or created within a certain date range.

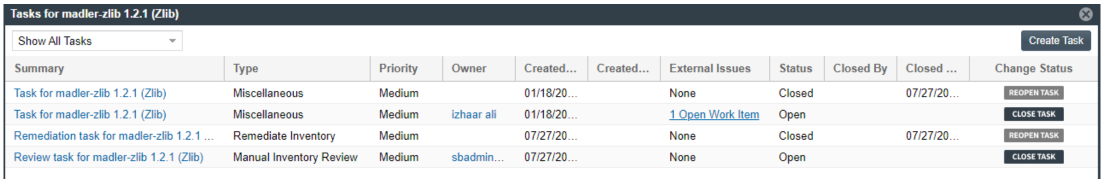
   When you select the specific inventory item, the Project Inventory Details Pane on the right is populated with information about the inventory item. The **Inventory Details** tab in focus.
3. In the **Tasks** section on the tab, click the ***x*** **Open Tasks** or ***x*** **Closed Tasks** link.



Disclosed: No
Modified: Unknown
Alerts: None
Tasks: 2 Open Tasks
1 Closed Task

Workflow URL: None

The **Tasks** list for the inventory item is displayed.


Tasks for madler-zlib 1.2.1 (Zlib)

Summary	Type	Priority	Owner	Created...	Created...	External Issues	Status	Closed By	Closed ...	Change Status
Task for madler-zlib 1.2.1 (Zlib)	Miscellaneous	Medium		01/18/20...		None	Closed		07/27/20...	REOPEN TASK
Task for madler-zlib 1.2.1 (Zlib)	Miscellaneous	Medium	izhaar ali	01/18/20...		1 Open Work Item	Open			CLOSE TASK
Remediation task for madler-zlib 1.2.1 ...	Remediate Inventory	Medium		07/27/20...		None	Closed		07/27/20...	REOPEN TASK
Review task for madler-zlib 1.2.1 (Zlib)	Manual Inventory Review	Medium	sbadmin...	07/27/20...		None	Open			CLOSE TASK

4. If needed, use the search filter at the top of the list to show open, closed, or all tasks.
5. Perform any of the following:
   - Change the status of a task directly from the **Tasks** list without having to open a task. See Closing or Reopening a Task Directly from the Tasks List.
   - Open a task to edit its details, including its status. See Editing a Task.
   - Click **Create Task** to create another task. See Manually Creating a Task for details about completing the fields on the **Create Task** dialog that opens.

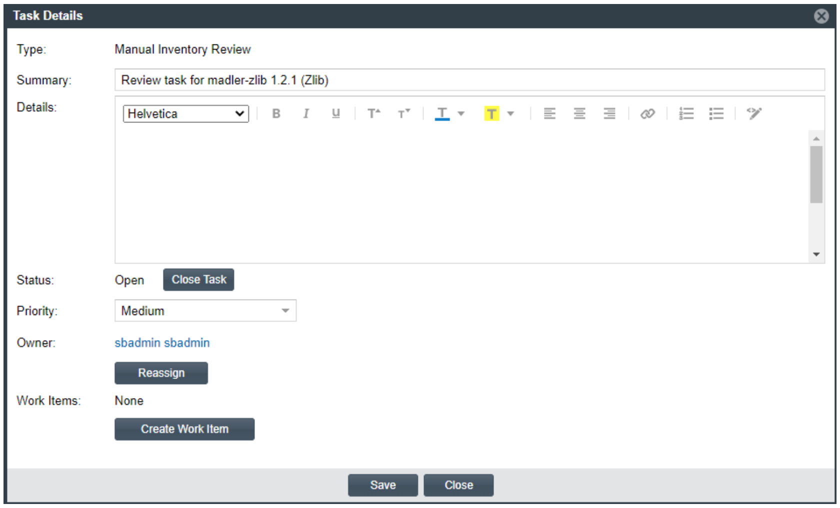
- Send an email to the task **Owner** or **Created By** user by clicking linked name in either column.
- Click the **Closed By** link to send an email to the user who closed the task. (If the task is open, the **Closed By** and **Closed On** values are blank.)

Editing a Task

The following describes how to edit and change the open or closed status of a task associated with a given inventory item on the **Project Inventory** tab.

Task ***To edit a task, do the following:***

1. For the selected inventory item on the **Project Inventory** tab, open the **Tasks** list. (See Opening the Tasks List.) The list shows all review tasks associated with the inventory item.
2. Locate the appropriate task from the **Tasks** list, and click the link in the **Summary** column for the task to open the **Task Details** dialog.



Task Details

Type: Manual Inventory Review

Summary: Review task for madler-zlib 1.2.1 (Zlib)

Details: Helvetica

Status: Open

Priority: Medium

Owner: sbadmin sbadmin

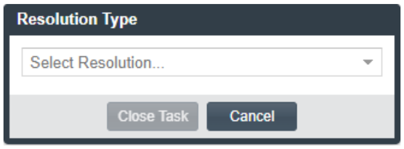
Work Items: None

3. To update the task fields or add an external work item, refer to the descriptions in Manually Creating a Task.
4. To change the status of the task, go to the **Status** section of the window, and do either of the following, depending on the current task status:

    - To *close* an open task, click the **Close Task** button.

      If the task type is **Remediate Inventory** or **Miscellaneous**, the task is immediately closed. Closing either of these tasks types has no effect on the current status of the inventory item. If you need to change the inventory status upon closing the task, do so manually.

      If the task type is **Manual Inventory Review**, select **Approved** or **Rejected** from the **Resolution Type** pop-up to indicate the task resolution. Then click **Close Task** from the pop-up. (To under stand how the resolution affects the status of the inventory item, see Effects of Closing Manual Review Tasks on Inventory Status.)



- To *reopen* a closed task, click the **Reopen** button. This does not affect the current status of the inventory item.

5. Click **Save** to save the updates and return to the **Tasks** list.

If you closed the task, its entry in the **Tasks** list shows your user ID and the date of task closure in the **Closed By** and **Closed On** fields, respectively. If you reopened a task, these values are blank.

Additionally, when task is closed, a notification email is automatically sent to the task owner and task creator. When a task is reassigned or reopened, a notification email is automatically sent to the task owner.

Closing or Reopening a Task Directly from the Tasks List

You can close or reopen a task directly from the **Tasks** list without having to open the task to edit it, as described in the following procedure.

Task ***To close or reopen a task directly from the Tasks list for a given project inventory item, do the following:***

1. For a given inventory item on the **Project Inventory** tab, open the **Tasks** list from the **Inventory Details** tab on the Project Inventory Details Pane on the right. This list shows all review tasks associated with the inventory item. To complete this step, follow the procedure in Opening the Tasks List.

2. In the **Tasks** list, locate the task whose status you want to change. (If necessary, use the search filter at the top of the list to show open, closed, or all tasks.)

3. In the **Change Status** column for the task, click the available status button:

   - CLOSE TASK—If the task type is **Remediate Inventory** or **Miscellaneous**, the task is immediately closed. Closing either of these tasks types has no effect on the current status of the inventory item. If you need to change the inventory status based on closing the task, do so manually.

     If the task type is **Manual Inventory Review**, select **Approved** or **Rejected** from the **Resolution Type** pop-up to indicate the task resolution. Then click **Close Task** from the pop-up. (To understand how the selected resolution can automatically change the status of the inventory item, see Effects of Closing Manual Review Tasks on Inventory Status.)

     Once the **Tasks** list is refreshed, it shows your user ID and the date of task closure in the **Closed By** and **Closed On** fields for the task.

     When a task is closed, a notification email is automatically sent to both the task owner and the task creator.

   - REOPEN TASK—The task is immediately reopened. This does not affect the current status of the inventory item.

     Once the **Tasks** list is refreshed, the **Closed By** and **Closed On** fields for the task show blanks.

     When a task is reopened, a notification email is automatically sent to the task owner.

4. Repeat these steps to change the status of other tasks from the **Tasks** list.

Effects of Closing Manual Review Tasks on Inventory Status

When you can close a **Manual Inventory Review** task, you must select an **Approve** or **Reject** resolution which, in turn, has an effect on the status of the inventory item, as follows:

- If the inventory item has only one review task associated with it, the **Approve** or **Reject** status of the task sets the inventory item status to **Approve** or **Reject** accordingly.
- If the inventory item has two or more review tasks associated with it, the **Reject** status of a single review task automatically sets the inventory item status to **Reject**. All review tasks are closed but can be reopened for further investigation.
- If an inventory item has two or more review tasks associated with it and these tasks are a combination of open tasks and tasks with an **Approve** status, the inventory item retains its **Not Reviewed** status.

Note that, depending on the inventory review and remediation options defined for the project, the **Reject** status that is automatically set when you close a **Manual Inventory Review** task can automatically create a **Remediate Inventory** task. For more information about these options, see Updating Inventory Review and Remediation Settings for a Project.

Creating and Viewing External Work Items for a Project Inventory Task

The Code Insight includes an ALM (application lifecycle management) facility that enables Code Insight users with access to the project inventory (and edit privileges) to create one or more external work items for a task associated with a given inventory item. An external work item helps users keep track of the remediation work that needs to be performed outside of Code Insight to address the inventory task.

The following topics provide more information about the external work items created from Code Insight:

- More About the Creation of External Work Items
- Prerequisites for Creating a Jira Issue from Code Insight
- Manually Creating a Jira Issue from Code Insight
- Viewing Jira Issues Assigned to an Inventory Task

More About the Creation of External Work Items

To create an external work item for a given task associated with a project inventory item, a user provides details in Code Insight that describe the work item. Code Insight then uses specific ALM connection information configured for the project to access the external ALM system and automatically create the work item there.

Periodic synchronization between Code Insight and the ALM system keeps Code Insight up to date with the status of the work item in the ALM system. Additionally, the Code Insight UI provides a direct link to the external work item in the ALM system.

Currently, Code Insight supports integration only with the ALM system Jira. Therefore, this chapter focuses the creation of the Jira issues (that is, Jira “work items”) on a Jira server from Code Insight.

Prerequisites for Creating a Jira Issue from Code Insight

You can create Jira issues from Code Insight only if the Code Insight project for which you are creating the issues has been associated with a specific Jira ALM instance, as described in Associating the Project with an Application Life Cycle System to Create Work Items.

Manually Creating a Jira Issue from Code Insight

The following procedure describes the manual process for creating a Jira issue from Code Insight for a open task associated with a specific project inventory item.

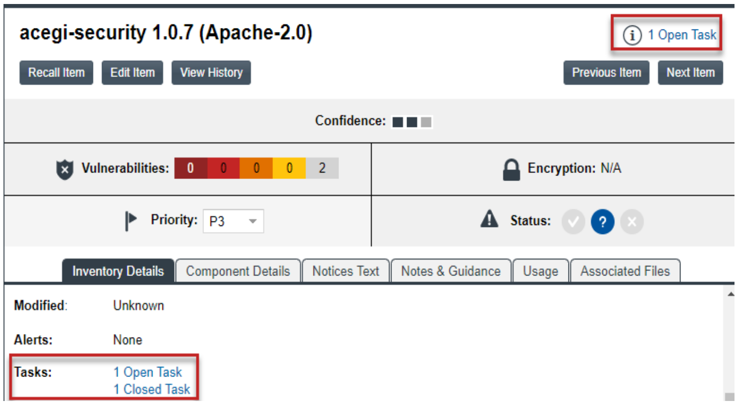
Note ▪ *A Jira issue can also be created automatically in an automated workflow process that you can set up for the project. See Updating Inventory Review and Remediation Settings for a Project for details on editing the automated workflow options and Edit Project: Review and Remediation Settings Tab for field descriptions.*

Task ***To create a Jira issue manually from Code Insight, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. Select the inventory item associated with the open task to which you want to add a work item.

   Alternatively, click the **Advanced Search** button to open the **Advanced Inventory Search** dialog, where you can select filters that help you locate the inventory item. For example, you can select to filter to inventory with open tasks (or you can further filter to inventory with open tasks assigned to a specific user or created within a certain date range).

   When you select the specific inventory item, the Project Inventory Details Pane on the right side of the **Project Inventory** tab is populated with information about the inventory item. (Note that, when you initially open the pane, the **Inventory Details** tab is in focus.)
3. Click the ***x*** **Open Tasks** link either in the upper right corner of the **Inventory Details** pane or in the **Tasks** section on the **Inventory Details** tab.



acegi-security 1.0.7 (Apache-2.0)

1 Open Task

Confidence: ■■□

Vulnerabilities: 0 0 0 0 2

Encryption: N/A

Priority: P3

Status: ✓ ? ✕

Modified: Unknown

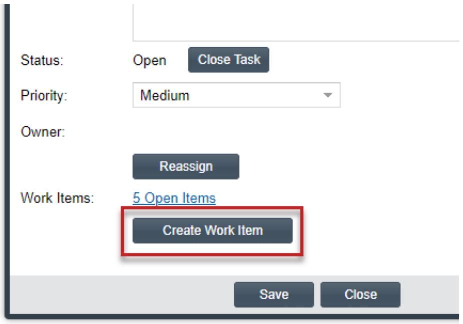
Alerts: None

Tasks: 1 Open Task
1 Closed Task

   The **Tasks for...** window listing all tasks for the inventory item is displayed.
4. In the list, locate the task for which you want to create a work item.

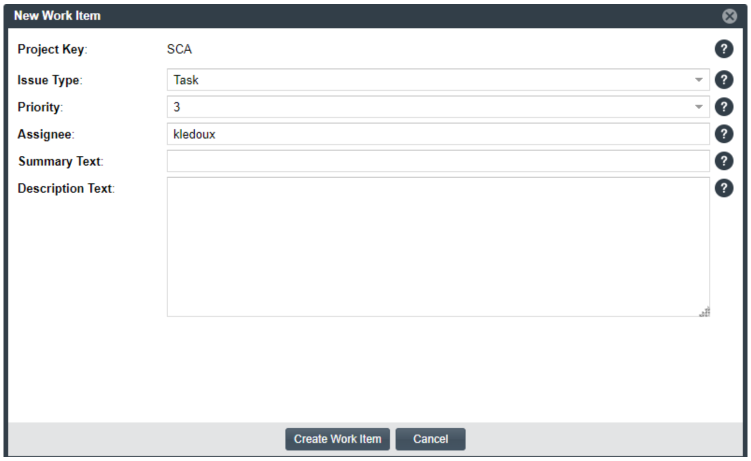
5. In the **Summary** column for task, click the link to open the task in the **Task Details** dialog.
6. At the bottom of the **Task Details** dialog, click the **Create Work Item** button.

   Note ▪ *The **Create Work Item** button on the **Task Details** dialog is enabled only if the project has been configured to an instance on the Jira server, as described in Associating the Project with an Application Life Cycle System to Create Work Items.*

   Status: Open
   Priority: Medium
   Owner:
   Work Items: 5 Open Items

   The **New Work Item** dialog is opened.

   New Work Item

   Project Key: SCA
   Issue Type: Task
   Priority: 3
   Assignee: kledoux
   Summary Text:
   Description Text:

7. Complete the fields to define the work item, as described in Fields Defining an External Jira Issue, editing any default values as needed. **Issue Type**, **Priority**, and **Summary Text** values are required.
8. When you have completed the fields, click the **Create Work Item** button.

   You are returned to the **Task Details** dialog.

Viewing Jira Issues Assigned to an Inventory Task

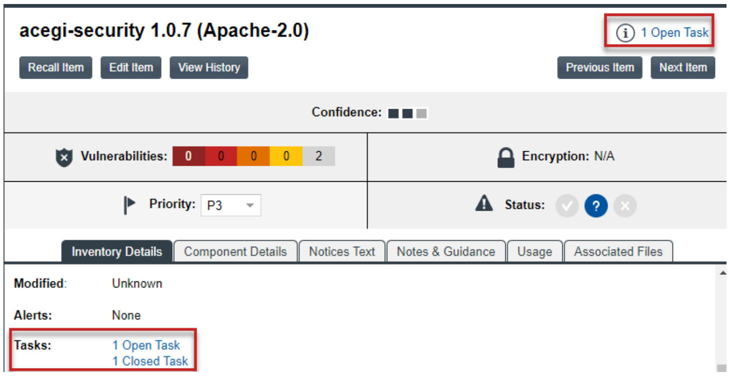
The following procedure describes how to access the list of Jira issues assigned to a given inventory task. Included in the details for each Jira issue is its current status obtained from Code Insight’s latest synchronization with the Jira server. The list also includes a link for each Jira issue that opens the issue directly on the Jira server.

Task ***To view the Jira issues assigned to a given inventory task, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory).
2. Select the inventory item associated with the task whose Jira issues you want to view.

   Alternatively, click the **Advanced Search** button to open the **Advanced Inventory Search** dialog, where you can select filters that help you locate the inventory item. For example, you can select to filter to inventory with open tasks only (or you can further filter to inventory with open tasks assigned to a specific user or created within a certain date range).

   When you select the specific inventory item, the Project Inventory Details Pane on the right side of the **Project Inventory** tab is populated with information about the inventory item. (Note that, when you initially open the pane, the **Inventory Details** tab is in focus.)
3. Click the ***x*** **Open Tasks** link either in the upper right corner of the **Inventory Details** pane or the ***x*** **Open Tasks** or ***x*** **Closed Tasks** link in the **Tasks** section on the **Inventory Details** tab.

   acegi-security 1.0.7 (Apache-2.0)

   Confidence:

   Encryption: N/A

   Priority: P3

   Modified: Unknown

   Alerts: None

   Tasks: 1 Open Task
1 Closed Task

   The **Tasks for...** window listing the open tasks for the inventory item is displayed.
4. In the list of tasks, locate the task that has the assigned Jira issues you want to view.
5. In the **External Issues** column for the task, click the **# Open Work Items** or **# Closed Work Items** link. The **Work Items for...** window is displayed listing the Jira issues for the selected task. (See Jira Issue Statuses.)
6. (Optional) Use the search filter at the top of the window to show **All**, **Open**, or **Closed** Jira issues for the task. (See Jira Issue Statuses.)
7. In the **Status** column in the list, view the status of a given Jira issue. (See Jira Issue Statuses.).
8. (Optional) Click the **External ID** link for a Jira issue to open it directly on the Jira server.

Jira Issue Statuses

If the status of a Jira issue on the Jira server changes, the change is reflected in **Status** column on for the issue in the **Work Items for...** window once a synchronization with the Jira server is run. The change can also result in an update to the **# Open Work Items** and **# Closed Work Items** for tasks assigned to a given inventory item.

The following lists the default status values. (Custom statuses are not currently supported.)

- The default Open status values include **Open**, **Reopen**, **New**, **To Do**, **In Progress**, and **Backlog**.
- The default Closed status values include **Done**, **Resolved**, **Verified**, and **Closed**.

Fields Defining an External Jira Issue

The following fields are used to define an external Jira issue in Code Insight. These fields correspond to the fields that define a Jira issue on the Jira server.

Table 2-10 ▪ Fields Defining a Work Item

Field	Description
Project Key	The key that identifies the Jira project on the Jira server with which the Jira issue you are creating is associated. This field is set by the Project Manager and is not editable.
Issue Type	Select the type of issue you are creating on the Jira server—**Bug** or **Task**. This field is required.
Priority	Select the priority level for the Jira issue: **1**—Highest **2**—High **3**—Medium **4**—Low **5**—Lowest This field is required.
Assignee	Enter the email for the user on the Jira server to whom you are assigning the Jira issue.
Summary Text	Enter the text that will display as the summary for the issue on the Jira server. The field supports the use of Code Insight variables, as described in Using Code Insight Variables in Text. This field is required.
Description Text	Enter the text that will display as the description for the issue on the Jira server. This field supports the use of Code Insight variables, as described in Using Code Insight Variables in Text

Using Code Insight Variables in Text

The **Summary Text** and **Description Text** fields support Code Insight variables that automatically pass information about the Code Insight project and inventory item to the content in these fields.

Supported Variables

The following table lists the variables available for use in the text entered in the fields:

Table 2-11 ▪ Supported Code Insight Variables For Use in Work-Item Summary and Description Text

$PROJECT_NAME	Name of the Code Insight project containing the issue
$INVENTORY_ITEM_NAME	Name of the inventory item containing the issue
$COMPONENT_NAME	Name of the component associated with the inventory item
$VERSION_NAME	Version of the component associated with the inventory item
$LICENSE_NAME	Name of the selected license for the inventory item
$NUMBER_VULNERABILITIES	Total number of security vulnerabilities associated with the inventory item
$NUMBER_FILES	Total number of files associated with the inventory item
$INVENTORY_URL	Link to the inventory item

When the Jira issue is created, the included variables are replaced by their respective values.

Example Use of Variables

The following is example text you might enter in the **Default Summary Text** field. The text includes some of the available variables:

```
The $INVENTORY_ITEM_NAME inventory item in the project $PROJECT_NAME contains
$NUMBER_VULNERABILITIES vulnerabilities that require review. Go to $INVENTORY_URL to see the
vulnerable inventory item.
```

If your Code Insight project name is *MySampleProject* and the name of the inventory item name for which you create a work item is *Apache Commons BeanUtils*, the work item and Jira issue will display the following summary:

```
The Apache Commons BeanUtils 1.7.0 (Apache 2.0) inventory item in the project MySampleProject
contains 18 vulnerabilities that require review. Go to https://my.sample.server:8888/codeinsight to
see the vulnerable inventory item
```

Recalling a Published Inventory Item

You can recall (remove) a published inventory item from **Inventory Items** list if it does not fit the criteria for inclusion. Once you recall an inventory item, it is removed from the **Project Inventory** tab but is still available as an unpublished item in the **Analysis Workbench**. From this location, you can update the item and re-publish it if necessary.

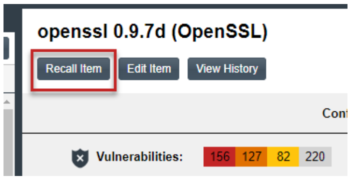
Recalling the item and publishing it again will affect the publish date on the item as well as the age of the inventory item.

Task ***To recall a published inventory item, do the following:***

1. Open the **Project Inventory** tab for the desired project, and select the inventory item that you want to recall (see Displaying Project Inventory).

   Information about the inventory item is displayed in the right pane.

2. In the pane header, click **Recall Inventory Item**.

   The item is removed from the **Inventory Items** list. In the **Analysis Workbench**, the review status of the inventory item before the recall is retained until item is re-published (and the latest review policy is applied).

Viewing the Update History for an Inventory Item in Project Inventory

From the **Project Inventory** tab, you can view a history of the updates made to a specific inventory item within the context of a project.

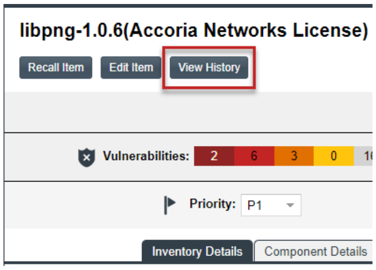
Note *▪ You have access to this same history from the* ***Analysis Workbench****. See Viewing the Update History for an Inventory Item in the Analysis Workbench.*

Task ***To view the update history of an inventory item, do the following:***

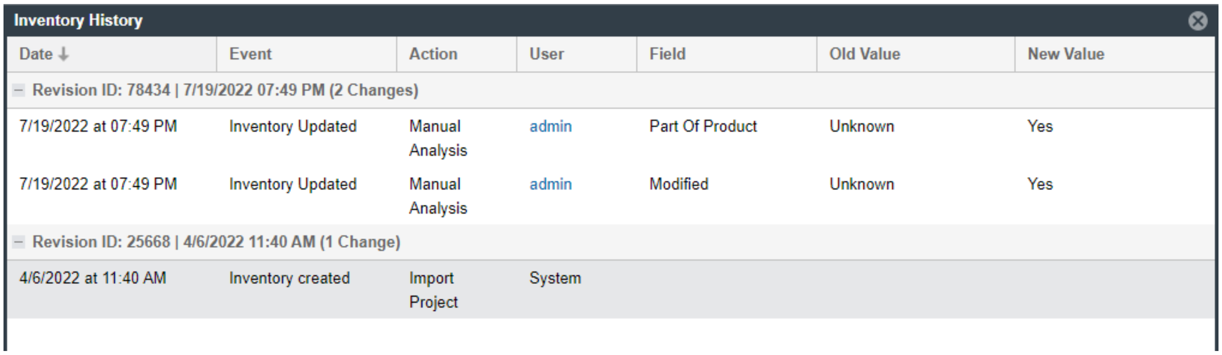
1. Open the **Project Inventory** tab for the desired project, and select the inventory item whose history you want to view (see Displaying Project Inventory).

   Information about the inventory item is displayed in the right pane.

2. In the pane header, click the **View History** button.

The **Inventory History** window is opened, showing the list of updates made to the inventory item within the project. By default, the updates are listed in descending order by date so that you see the most recent updates first. Each update record identifies—among other details—the update type, the user who made the update, and the before-and-after values in the update. For a description of all features in this window, see Inventory History Window.



Inventory History

Date ↓	Event	Action	User	Field	Old Value	New Value
Revision ID: 78434 | 7/19/2022 07:49 PM (2 Changes)						
7/19/2022 at 07:49 PM	Inventory Updated	Manual Analysis	admin	Part Of Product	Unknown	Yes
7/19/2022 at 07:49 PM	Inventory Updated	Manual Analysis	admin	Modified	Unknown	Yes
Revision ID: 25668 | 4/6/2022 11:40 AM (1 Change)						
4/6/2022 at 11:40 AM	Inventory created	Import Project	System			

Using "Lookup Component" to Search for Components to Associate with Inventory

The Lookup Component feature in Code Insight searches the Code Insight Data Library to find information about the open-source or third-party components available for association with inventory. This information can include security vulnerabilities and potential license issues associated with component versions and their licenses. The Lookup Component feature is available when creating or updating inventory, enabling you to select or update the component version and associated license for an inventory item.

The following sections describe how to use Lookup Component:

- Performing a Lookup Component Search
- Guidelines for Lookup Component Searches
- Priority of Results from a Lookup Component Search

Performing a Lookup Component Search

The following instructions describe on how you can look up an existing component with which to associate with an inventory item when you create or edit inventory within a project.

Task ***To perform a Lookup Component search, do the following:***

1. Follow the instructions to start the process of creating or updating an existing inventory item in the **Analysis Workbench** or in **Project Inventory**. The Lookup Component feature is available only if you are creating or editing an inventory item with a **Type** of **Component**.

   In the Analysis Workbench:

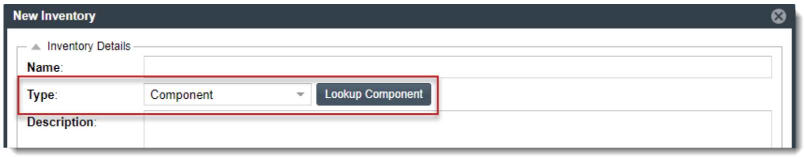
   - Creating Inventory From the Inventory Items List

- Editing Inventory from the Analysis Workbench

In Project Inventory

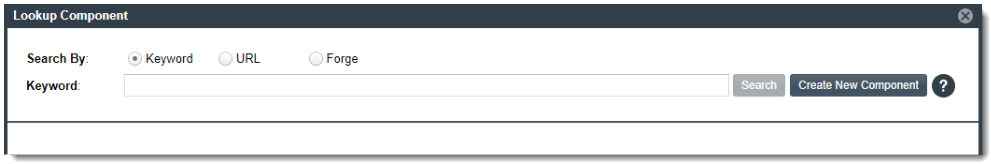
- Creating Inventory from the Project Inventory Tab
- Editing Inventory from the Project Inventory Tab

2. Once you access the **Edit** (or **Create**) **Inventory** window in **Project Inventory** or open the tab for the inventory item you are editing or creating in the **Analysis Workbench**, continue with the next steps.
3. Ensure that you select **Component** for the **Type** field.
4. Click **Lookup Component** next to the **Type** field.



Type: Component

The **Lookup Component** window is displayed.

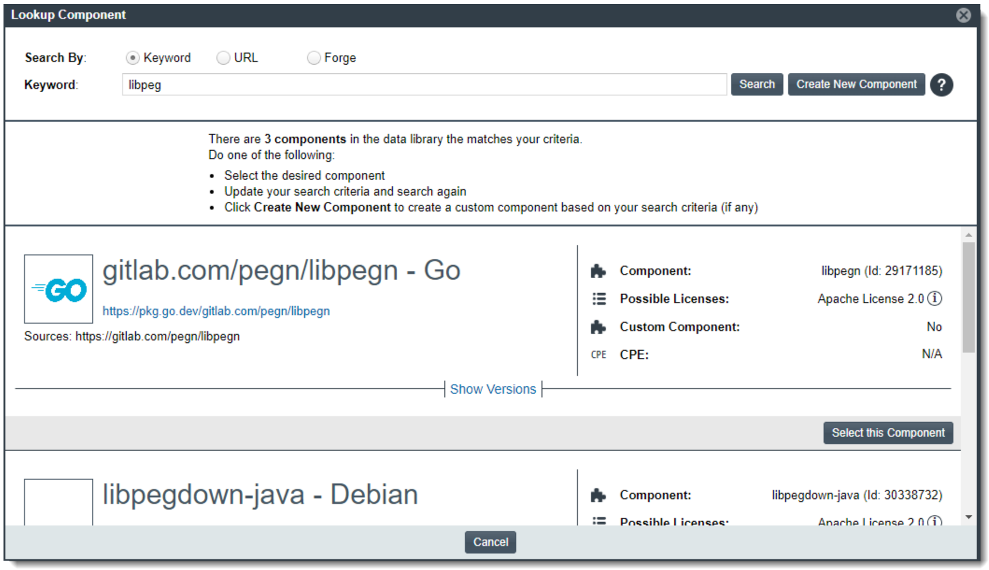


Search By: ● Keyword

5. Proceed with the appropriate step:
   - If you are creating a new inventory item, continue with the next step to search for the correct component-version-license instance to associate with the item.
   - If you are editing an inventory item, the component currently associated with that item is automatically displayed in the **Lookup Component** window. If you want to select a different version-license instance for the component currently associated with the inventory item, proceed to step 7. If you want to select a different component altogether for the item, continue with the next step. Otherwise, to leave the inventory item as, click **Cancel** to close the **Lookup Component** window and continue editing the inventory item.
6. To search for a component, select the type of search you want to perform (**Keyword**, **URL**, or **Forge**), and enter the required search criteria. For information on these searches and the criteria you must enter for each, see Guidelines for Lookup Component Searches.
7. Click **Search** to filter to the components matching your search criteria.

   ***Note** ▪ The search results are listed by priority. See Priority of Results from a Lookup Component Search.*

8. In the list of search results in the **Lookup Component** window, scroll to the appropriate component section, and click **Show Versions** (or **Show Instances**) to display the list of instances (version-and-license combinations) for that component.



Search By: ● Keyword ○ URL ○ Forge

Keyword: libpeg

There are 3 **components** in the data library the matches your criteria.
Do one of the following:

- Select the desired component
- Update your search criteria and search again
- Click **Create New Component** to create a custom component based on your search criteria (if any)

gitlab.com/pegn/libpegn - Go

https://pkg.go.dev/gitlab.com/pegn/libpegn

Sources: https://gitlab.com/pegn/libpegn

Component:	libpegn (Id: 29171185)
Possible Licenses:	Apache License 2.0 ⓘ
Custom Component:	No
CPE:	N/A

libpegdown-java - Debian

Component:	libpegdown-java (Id: 30338732)
Possible Licenses:	Apache License 2.0 ⓘ

Those instances that use a user-preferred license show the icon next to the license. A user-preferred license is one that a user has specified to be automatically associated with the given component version for all future system-generated inventory in your Code Insight instance. For more information, see Specifying a User-Preferred License Mapping.

9. Associate an instance with the current inventory item. Use one of the following procedures:

   - Associating an Existing Instance As Is
   - Changing the License for an Instance Before Associating with the Inventory Item
   - Registering a New Instance to Associate with the Inventory Item

Once you click **Use This Instance** from the **Show Versions** (or **Show Instances**) list to associate an instance with the inventory item, you are returned to the window or tab in which you were editing or creating the inventory item. The **Name**, **Description**, **Component**, and **License** fields for the inventory item are automatically populated with information based on the selected instance. Additionally, Information ⓘ icons are available next to the **Component** and **License** fields so that you can view publicly available information about the selected component or its license.

Associating an Existing Instance As Is

Use this step to select an existing instance in the **Show Versions** (or **Show Instances**) list for the displayed component in the **Lookup Component** window and associate the instance with the current inventory item.

Task ***To associate the current inventory item with an existing registered instance as is, do the following:***

In the **Show Versions** (or **Show Instances**) list, click **Use This Instance** next to the instance that you want to associate with the current inventory item.

You are returned to the window or tab in which you were editing or creating the inventory item. Details about the component instance that you selected automatically populate this window or tab, as described at the end of the previous Performing a Lookup Component Search section.

Changing the License for an Instance Before Associating with the Inventory Item

Use this procedure if, in the **Show Versions** (or **Show Instances**) list for the current component in the **Lookup Component** window, you have located the instance for the component version that you want to associate with the inventory item but want to switch the license. (Alternatively, use this same procedure, minus the last step, to simply change the license in an instance but not associate it with the current inventory item.)

Note *▪ You cannot create a custom license from the **Lookup Component** window to associate with a component version. However, you can create a custom license for the inventory item from the tab or window from which you care creating or editing the item and from other locations as well. For more information, see Creating and Editing Custom Licenses.*

Task ***To change the license for the instance you want to associate with the inventory item, do the following:***

1. In the **Show Versions** (or **Show Instances**) list, locate the instance for the component version whose license you want to change.
2. Click within the **Selected License** column for the instance to display a dropdown listing available licenses by categories.
3. Select the license you want to associate with the component version.

   If you selected the license from a group of licenses under the **System Suggested License** category or selected the license from the **Other Licenses** category, the **Update License Mapping** window is displayed. This window gives you the option to save the license mapping at the system level for all future system-generated inventory using this component version. Continue with step 4 for further information.

   Otherwise, no other window is displayed and the new license is mapped to the instance. Continue with step 5.
4. From the **Update License Mapping** window, select the appropriate option:
   - **Yes**—All future inventory system-generated for the component version across projects are automatically mapped to this license. Once you select **Yes**, the instance is updated with the new license and a user-preferred-license icon is displayed next to the license. Any other instances for the same component version in the list are replaced with this single instance that uses the user-preferred license.
   - **No**—This component-version-license combination will be saved in the database for the inventory items with which it is manually associated. It is also made available in **Lookup Component** window (as a registered instance) and in the **Versions for** <**component**> window. However, any future system-generated inventory for this component version will be mapped to the license associated with this version in Code Insight Data Library.

   For complete information about user-preferred licenses, see Specifying a User-Preferred License Mapping.

5. Click **Use This Instance** next to the instance whose license you updated to associate the instance to the current inventory item.

You are returned to the window or tab in which you were editing or creating the inventory item. Details about the component instance that you selected automatically populate this window or tab, as described at the end of the previous Performing a Lookup Component Search section.

Registering a New Instance to Associate with the Inventory Item

Follow this procedure to *register* (that is, create) a new instance in the **Show Versions** (or **Show Instances**) list for the current component in the **Lookup Component** window and associate the instance with the inventory item you are editing or creating. (Alternatively, use this same procedure, minus the last step, to simply create a new instance but not associate it with the current inventory item.)

Task ***To create a new instance to associate with the inventory item you editing or creating, do the following:***

1. In the **Show Versions** (or **Show Instances**) list for the current component, click **Register New Instance** to create a new version-and-license instance of the component to associate with the inventory item.
2. In the new instance row, click within the **Version** column to display a dropdown from which to select a component version (or to select **Create Custom Version** to create a new version).
3. Click within the **Selected License** column for the new instance to display a dropdown that lists the licenses in categories.
4. Select the license you want to associate with the component version.

   If you selected the license from a group of licenses under the **System Suggested License** category or selected the license from the **Other Licenses** category, the **Update License Mapping** window is displayed. This window gives you the option to save the license mapping at the system level. Continue with step 5.

   Otherwise, no other window is displayed and the new license is mapped to the instance. Continue with step 6.
5. From the **Update License Mapping** window, select the appropriate option:

   - **Yes**—All future inventory system-generated for the component version across projects are automatically mapped to this license. Once you select **Yes**, the instance is updated with the new license and a user-preferred-license icon is displayed next to the license. Any other instances for the same component version in the list are replaced with this single instance that uses the user-preferred license.
   - **No**—Whenever inventory is manually associated with this component-version-license combination, the association will be saved in the database. Additionally, the combination is also made available in **Lookup Component** window (as a registered instance) and in the **Versions for <component>** window. However, any future system-generated inventory for this component version will be automatically mapped to the license associated with this version in Code Insight Data Library.

   For complete information about user-preferred licenses, see Specifying a User-Preferred License Mapping.)
6. Click **Use This Instance** next to the new instance to associate it with the current inventory item.

You are returned to the window or tab in which you were editing or creating the inventory item. Details about the component instance that you selected automatically populate this window or tab, as described at the end of the previous Performing a Lookup Component Search section.

Guidelines for Lookup Component Searches

The following sections provide information about the types of searches you can perform using Lookup Component.

- Keyword
- URL
- Forge
- Unable to Locate the Component

Basically, use a **URL** or **Forge** search to obtain the most targeted results. The **Keyword** search can provide a broader set of results to explore.

If the search finds no results that meet your criterion, a pop-up message is displayed, stating as such.

Keyword

In the **Lookup Component** window, select the **Keyword** option to search components by their name. For the search criterion, enter a single string for the component name. This can be the full component name, such as **jquery-jquery-ui**, or a string within the name, such **jquery**. (Multiple strings are not allowed.)

Note ▪ *The search is case-insensitive, so it filters to all components whose names contain the string, no matter the case used in the entered string or in the actual component name.*

For common name conventions used for components in various forges, see the next section.

Component Name Conventions Used in Various Forges

In general, the name of a component is a unique identifier that can be based on the project, package, or gem name of the component or on another convention such as the component’s author or repository. The following shows the common conventions used for component names in certain forges. You can use this as a reference for helping you enter an appropriate string for the component name.

- **Apache**—<PROJECT_NAME>, for example “apache-batik”
- **Debian**—<PACKAGE_NAME>, for example “0ad”
- **Github**—<AUTHOR>-<REPOSITORY_NAME>, for example “jquery-jquery-ui”
- **GitLab**—<AUTHOR/ORGANIZATION>-<REPOSITORY_NAME>, for example:
    - “cryptsetup-cryptsetup” (as found in the component URL: `https://gitlab.com/`**`cryptsetup`**`/`**`cryptsetup`**)
    - “redhat-bison” (as found in the component URL: `https://gitlab.com/`**`redhat`**`/centos-stream/rpms/`**`bison`**)

- **NuGet Gallery**—<PACKAGE_NAME>, for example “newtonsoft.json”
- **Pypi**—<PACKAGE_NAME>, for example “hash_ring”
- **RubyGems**—<GEM_NAME>, for example “x-editable-rails”
- **Other**—<PROJECT_NAME>, for example “openssl”

URL

In the **Lookup Component** window, use the **URL** search option if you know the URL of the forge containing the component you want to locate. In the **URL** value, you can enter the complete path for the forge, such as **https://github.com/jquery/jquery**, or a string in the path, such as **jquery**.

Note ▪ *The search is case-insensitive, so the results will include all components with a matching forge path or path string (whichever criterion you entered in the* ***URL*** *field), no matter the case used in the filter or the actual URL.*

Forge

In the **Lookup Component** window, select the **Forge** option to search for a specific component when you know the exact name of its forge and project or repository within the forge. This search first prompts you to select the forge and then to enter other criteria that exactly identifies the component within the forge, such as the name of its repository, project, package, module, or other such information.

Note ▪ *The search is case-insensitive, so the results will include all components with a matching forge and repository, no matter the case used in the filter or in the actual repository name.*

Unable to Locate the Component

If a component you want to associate with inventory is not available by any of the searches, the component might not exist in the Code Insight Data Library nor be saved as a custom component. In this case, consider either of these options:

- Create your inventory item as **Work in Progress** and name it using the convention *<COMPONENT> <VERSION> (<LICENSE>)*. For example, `myComponent 1.2 (MIT)`. You can later edit this inventory item to convert it to one of the other inventory types—**Component** or **License**. See Creating Inventory From the Inventory Items List or Creating Inventory from the Project Inventory Tab for details.
- Create a custom component. See Creating and Editing Custom Components for details.

Priority of Results from a Lookup Component Search

The results for a Lookup Component search are prioritized in the following order:

1. **Registered Components**—Components with a history of use (one or more instances of the component are registered for use in the system).
2. **Important Components**—Components that are marked by Code Insight as important due to popularity or presence of security vulnerabilities.
3. **All other Components**—Components that are neither registered nor important.

   If no results are returned, the component might not exist in the Code Insight Data Library. See the previous section for options in dealing with components not available in the library.

Specifying a User-Preferred License Mapping

Project scans in Code Insight automatically create inventory for detected third-party or open-source components. The scans use resources such as the Code Insight Data Library to obtain inventory details, including licenses commonly associated with component versions. During a review of the inventory resulting from a scan, users might find that the license associated with given a component version is incorrect or unknown or that the version is associated with multiple licenses with no single license selected for the inventory item. In such cases, the reviewer must manually edit the inventory item to select the correct license. Because this same license-mapping issue can reoccur for new inventory generated across projects after each scan, users might be required to repeatedly edit inventory.

Code Insight provides a feature that enables users to specify a single license that is to be mapped to all future inventory automatically generated for a given component version across all projects. In this way, users do not have to repeat the manual mapping process for new inventory items system-wide every time scans are run. The license that the user specifies is called the “user-preferred license”.

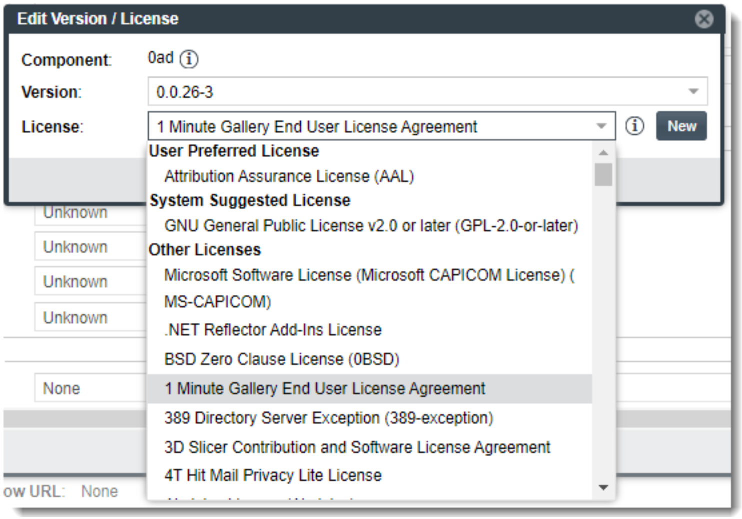
The following topics provide the information you need to know about identifying a user-preferred license.

- License Categories in the License Dropdown
- Accessing the Option to Identify a User-Preferred License
- About the Update License Mapping Window
- After You Save the License Mapping

License Categories in the License Dropdown

Code Insight classifies licenses into the categories that are described in the table later in this section. When you select a license to map to a specific component version, the license dropdown for that version can contain one or more of these categories from which to choose the license. The categories available for a component version are based on whether the version has a user-preferred license, one or more license associations (as specified in the Code Insight Data Library), or no known license associations.

The following screenshot shows an example license dropdown listing available licenses in categories.



Edit Version / License

Component: 0ad

Version: 0.0.26-3

License: 1 Minute Gallery End User License Agreement

New
User Preferred License
Attribution Assurance License (AAL)
System Suggested License
GNU General Public License v2.0 or later (GPL-2.0-or-later)
Other Licenses
Microsoft Software License (Microsoft CAPICOM License) (MS-CAPICOM)
.NET Reflector Add-Ins License
BSD Zero Clause License (0BSD)
1 Minute Gallery End User License Agreement
389 Directory Server Exception (389-exception)
3D Slicer Contribution and Software License Agreement
4T Hit Mail Privacy Lite License
Unknown
Unknown
Unknown
Unknown
None
ow URL: None

The following table identifies the available license categories and the type of license(s) each contains.

Table 2-12 ▪ License Categories Available on the License Dropdown

License Category	Type of License(s) Listed for the Category
User Preferred License	A license (previously classified as **Other Licenses**) that a user has specified to be mapped to all future inventory that the system generates across projects for the component version.
System Suggested License	The one or more licenses associated with the component version, as specified in the Code Insight Data Library.
User Preferred and System Suggested License	A license (previously classified as **System Suggested License**) that a user has specified to be mapped to all future inventory that the system generates across projects for the component version.
Other Licenses	Licenses that are reported in the Code Insight Data Library but are neither user-preferred nor system-suggested for the component version.

Accessing the Option to Identify a User-Preferred License

The option to identify a user-preferred license is available on the **Update License Mapping** window, which is accessible once you select the license you want to map to a component version. You can select the license by editing the inventory item associated with the component version, registering or editing an instance for the version in the **Lookup Component** feature, or creating a custom component version.

Refer to the following sections:

- Selecting the License to Map
- License Selections That Allow Access to the “Update License Mapping” Window

Selecting the License to Map

Use one of the following methods to select the license you want to identify as the user-preferred license.

- By Editing the Inventory Item
- By Registering or Editing Instances in the Lookup Component Feature
- By Creating a Custom Component Version

By Editing the Inventory Item

You can access the **Update License Mapping** window within the context of the inventory item to which you are mapping the user-preferred license.

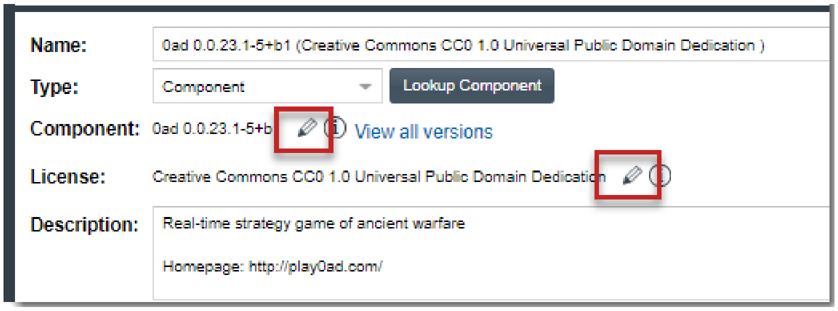
Task ***To access the Update License Mapping window by editing the inventory item, do the following:***

1. On the **Inventory Details** pane in **Project Inventory**, select the inventory item for whose component version you want to identify the user-preferred license, and then click the **Edit Item** button to open the **Edit Inventory** window.

   Or

   On the **Inventory Details** tab in the **Analysis Workbench**, select the inventory item for whose component version you want to identify the user-preferred license.

2. Click the **Edit** icon next to the **Component** or **License** value.



Name: 0ad 0.0.23.1-5+b1 (Creative Commons CC0 1.0 Universal Public Domain Dedication)

Type: Component

Component: 0ad 0.0.23.1-5+b

License: Creative Commons CC0 1.0 Universal Public Domain Dedication

Description: Real-time strategy game of ancient warfare

Homepage: http://play0ad.com/

   The **Edit Version/License** window is displayed.

3. From the **License** dropdown, select the license that you want to identify as the user-preferred license. Be sure that you select the license from an appropriate license category on the dropdown, as described in License Selections That Allow Access to the “Update License Mapping” Window to ensure that **Update License Mapping** window will be opened.

   Alternatively, click **New** to create a custom license to map to the component version. Once you have defined the license on the **Create Custom License** window, click **OK** to return to the **Edit Version/License** window.

4. Click **OK** on the **Edit Version/License** window.

   The **Update License Mapping** window is displayed. Its contents explain what happens if you accept this license mapping for all future inventory automatically created for the component version. For more information about the window contents, see About the Update License Mapping Window.

5. Choose the appropriate option:

    - **Yes**—All future inventory items that the system generates for the component version across projects are automatically mapped to the license you selected. For more information about what happens when you set up a user-preferred license, see After You Save the License Mapping.

      You are returned to the **Edit Version/License** window.

    - **No**—This component-version-license combination is saved for the specific inventory in the database and made available in the **Versions for** <**component**> window. However, any future system-generated inventory for this component version will be mapped to the license associated with this version in Code Insight Data Library.

      You are returned to the **Edit Version/License** window.

    - **Cancel**—Return to the **Edit Version/License** window.

6. Click **Save** to save the edits.

By Registering or Editing Instances in the Lookup Component Feature

You can access the **Update License Mapping** window when registering or editing an instance of the component version and license in Lookup Component. This process described next is performed within the context of editing an inventory item.

Task ***To access the Update License Mapping window by editing the inventory item, do the following:***

1. In the **Inventory Items** list in **Project Inventory**, select the inventory item associated with the component version for which you want to identify the user-preferred license, and then click the **Edit Item** button to open the **Edit Inventory** window.

   Or

   In the **Inventory Items** list in the **Analysis Workbench**, select the inventory item associated with the component version for which you want to identify the user-preferred license. The details tab for the inventory item is displayed.

   Or

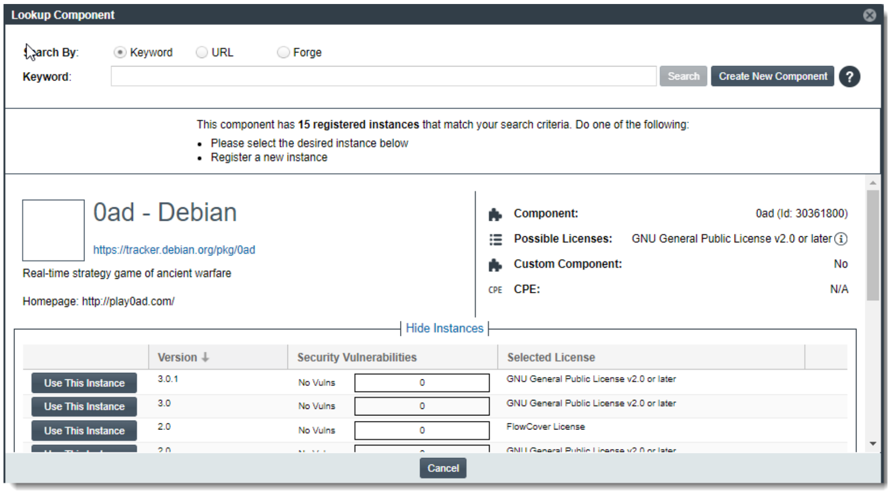
   At the top of the **Inventory Items** list in either location, click **Add Item** (in **Project Inventory**) or **Add New** (in **Analysis Workbench**) to create an inventory item. In the **New Inventory** window or tab that is displayed, you must select **Component** in the **Type** field for the new inventory item.

2. Click the **Lookup Component** button to open the **Lookup Component** window.

    - For an inventory item you are editing, the page for the component associated with the inventory item is automatically displayed in the **Lookup Component** window.

    - For an item you are creating, use the **Lookup Component** window to search for and locate the component to associate with the item (or create a new component).

3. On the **Lookup Component** window, click **Show Versions** to display a grid showing registered instances of component versions and their mapped licenses for the component.



Lookup Component

Search By: ● Keyword ○ URL ○ Forge

Keyword:

This component has **15 registered instances** that match your search criteria. Do one of the following:

- Please select the desired instance below
- Register a new instance

0ad - Debian

https://tracker.debian.org/pkg/0ad

Real-time strategy game of ancient warfare

Homepage: http://play0ad.com/

Component:	0ad (Id: 30361800)
Possible Licenses:	GNU General Public License v2.0 or later
Custom Component:	No
CPE:	N/A

	Version	Security Vulnerabilities		Selected License
Use This Instance	3.0.1	No Vulns	0	GNU General Public License v2.0 or later
Use This Instance	3.0	No Vulns	0	GNU General Public License v2.0 or later
Use This Instance	2.0	No Vulns	0	FlowCover License

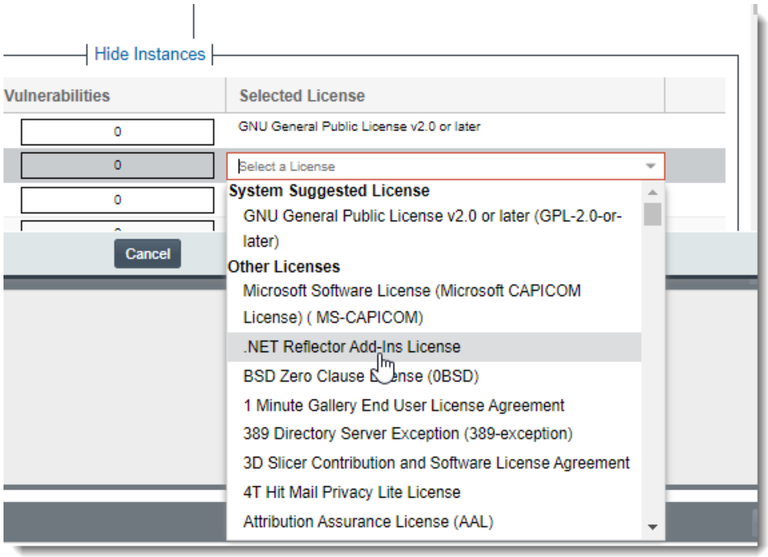
4. Locate the component version instance for which you are setting up the license.

   Or

   Click **Register a New Instance** to select or create a new component version.

5. Click within the license field for the instance.

6. From the **License** dropdown, select the license that you want to identify as the user-preferred license. You must select the license from an appropriate license category on the dropdown, as described in License Selections That Allow Access to the “Update License Mapping” Window, to ensure that **Update License Mapping** window is opened.



Vulnerabilities	Selected License
0	GNU General Public License v2.0 or later
0	Select a License
0	

System Suggested License
- GNU General Public License v2.0 or later (GPL-2.0-or-later)

Other Licenses
- Microsoft Software License (Microsoft CAPICOM License) (MS-CAPICOM)
- .NET Reflector Add-Ins License
- BSD Zero Clause License (0BSD)
- 1 Minute Gallery End User License Agreement
- 389 Directory Server Exception (389-exception)
- 3D Slicer Contribution and Software License Agreement
- 4T Hit Mail Privacy Lite License
- Attribution Assurance License (AAL)

The **Update License Mapping** window is displayed. Its contents explain what happens if you accept this license mapping for all future inventory automatically created for the component version. For more information about the window contents, see About the Update License Mapping Window.

7. Choose the appropriate option:

    - **Yes**—All future inventory items that the system generates for the component version across projects are automatically mapped to the license you selected. For more information about what happens when you set up a user-preferred license, see After You Save the License Mapping.

      You are returned to the **Lookup Component** window.

    - **No**—This component-version-license combination will be saved for the specific inventory item in the database and made available in **Lookup Component** window (as a registered instance) and in the **Versions for** <**component**> window. However, any future system-generated inventory for this component version will be mapped to the license associated with this version in Code Insight Data Library.

      You are returned to the **Lookup Component** window.

      Cancel—You are returned to the **Lookup Component** window.

8. Select **Use This Instance** next to the instance to apply the updated instance to inventory.

   You are returned to the **Edit Inventory** or **New Inventory** tab or window.

9. Click **Save**.

By Creating a Custom Component Version

You can access the **Update License Mapping** window when creating a custom component version. For instructions, see Creating Custom Component Versions.

License Selections That Allow Access to the “Update License Mapping” Window

The **Update License Mapping** window once you make a license selection only if you select one of these licenses:

- A license from a list of multiple licenses in the **System Suggested License** category
- A license from the **Other Licenses** category

If you select from any one of the categories, the **Update License Mapping** window is not displayed because you are selecting a valid license that Code Insight would normally map to the component version:

- **User Preferred License**
- **User Preferred and System Suggested License**
- **System Suggested License** (as long as it contains only one license and no **User Preferred and System Suggested License** also exists)

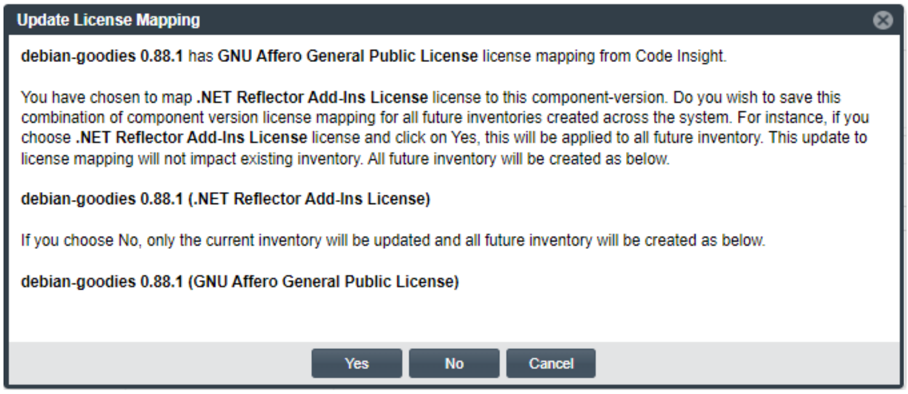
About the Update License Mapping Window

The **Update License Mapping** window provides the option to save your new license mapping at the system level so that all future inventory system-generated for the component version across projects will be associated with the license. The window also explains what happens when the mapping is save at the system level to help you make an informed decision.

The following shows the content variations in the window and the conditions under which the given content is displayed.

Content Variation in the Window

This content is displayed when a license is already mapped to the component version either automatically through data retrieved from the Code Insight Data Library or manually by a user.



debian-goodies 0.88.1 has **GNU Affero General Public License** license mapping from Code Insight.

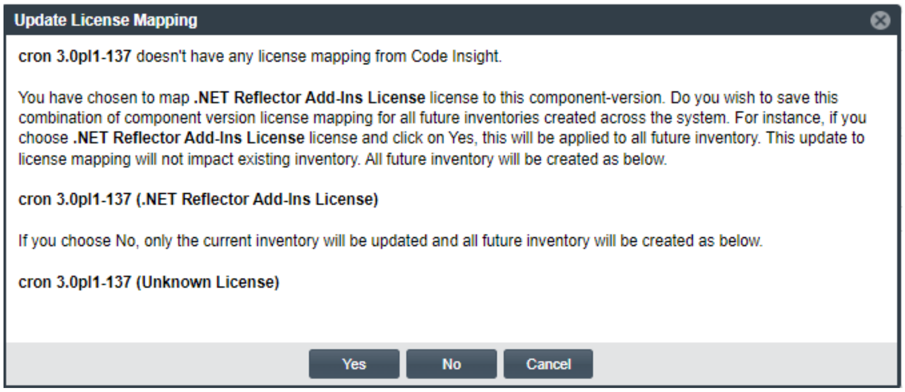
You have chosen to map **.NET Reflector Add-Ins License** license to this component-version. Do you wish to save this combination of component version license mapping for all future inventories created across the system. For instance, if you choose **.NET Reflector Add-Ins License** license and click on Yes, this will be applied to all future inventory. This update to license mapping will not impact existing inventory. All future inventory will be created as below.

debian-goodies 0.88.1 (.NET Reflector Add-Ins License)

If you choose No, only the current inventory will be updated and all future inventory will be created as below.

debian-goodies 0.88.1 (GNU Affero General Public License)

The following content is displayed when no license has been mapped automatically either through data retrieved from the Code Insight Data Library or through user-preferred mapping.



cron 3.0pl1-137 doesn't have any license mapping from Code Insight.

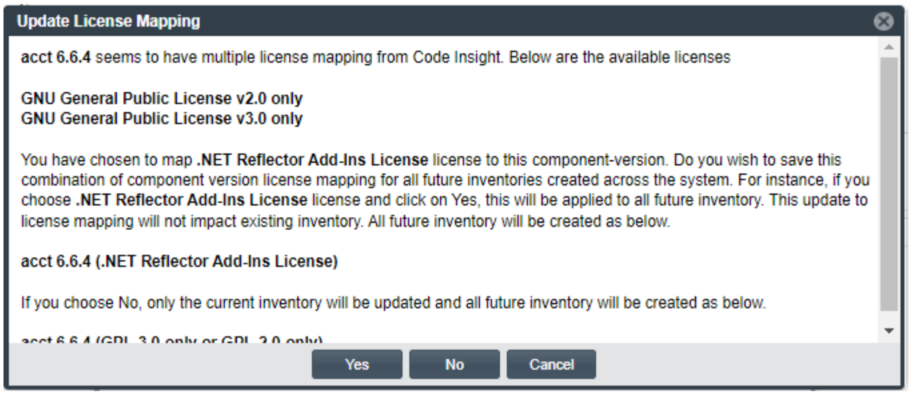
You have chosen to map **.NET Reflector Add-Ins License** license to this component-version. Do you wish to save this combination of component version license mapping for all future inventories created across the system. For instance, if you choose **.NET Reflector Add-Ins License** license and click on Yes, this will be applied to all future inventory. This update to license mapping will not impact existing inventory. All future inventory will be created as below.

cron 3.0pl1-137 (.NET Reflector Add-Ins License)

If you choose No, only the current inventory will be updated and all future inventory will be created as below.

cron 3.0pl1-137 (Unknown License)

The following content is displayed when the component version has multiple license associations in the Code Insight Data Library.



acct 6.6.4 seems to have multiple license mapping from Code Insight. Below are the available licenses

GNU General Public License v2.0 only
GNU General Public License v3.0 only

You have chosen to map **.NET Reflector Add-Ins License** license to this component-version. Do you wish to save this combination of component version license mapping for all future inventories created across the system. For instance, if you choose **.NET Reflector Add-Ins License** license and click on Yes, this will be applied to all future inventory. This update to license mapping will not impact existing inventory. All future inventory will be created as below.

acct 6.6.4 (.NET Reflector Add-Ins License)

If you choose No, only the current inventory will be updated and all future inventory will be created as below.

acct 6.6.4 (GPL-3.0 only or GPL-2.0 only)

After You Save the License Mapping

When you select **Yes** to save the license mapping for the component version at the system level, Code Insight remembers this mapping when creating all future inventory for the component version across projects. This section describes what else occurs when this option is enabled.

General

Because the component-version-license combination is saved to the database, it is made available in other locations as described below.

- The **License** dropdown for the inventory item to which you mapped the license (as well as any subsequent inventory generated by the system for the component version) shows the license as either **User Preferred License** or **User Preferred and System Suggested License**.
- The component-version-license combination is saved to the database and made available in the **Versions for** <**component**> window and in the **Lookup Component** window (as a registered version with a user-preferred-license icon).
- In the **Lookup Component** window, only the instance using the user-preferred license is visible for the given component version. Instances for the same component version but mapped to other licenses are not displayed.

Project Scans

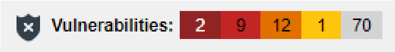
The following describes what happens during project scans when this option is enabled:

- In general, scans on projects should create any new inventory for the component version using only the preferred license.
- During any type of scan in general, user-edited or manually created inventory items are not impacted by the new license mapping.
- Any scan subsequent to the fresh (initial) scan will result in duplicate inventory items for the component version. These items include:
    - The existing inventory item for the component version mapped to the old license.
    - The new inventory item for the same component version mapped to the user-preferred license.
- After a *full* scan (subsequent to a fresh scan), an existing system-generated inventory item that is a duplicate mapped to the old license (see the previous bullet) will no longer have associated files. Only the item having the same component version but mapped to the user-preferred license will have associated files. (As previously noted, edited or manually created inventory items using the same component version stay the same.)
- A fresh (initial) scan does not create duplicate inventory items for the component version.
- When a user-preferred license mapping and a custom detection rule exists for the same component version, the resulting inventory item uses the name from custom rule but is mapped to the user-preferred license.

Working with Security Vulnerabilities

Code Insight uses data from the National Vulnerability Database (NVD) and other advisories such as RubySec to report security vulnerabilities associated with your inventory items. The information from these sources is used to create vulnerability rankings and alerts.

The **Vulnerabilities** bar graph shows the current security-vulnerability counts by severity level for a given inventory item:



Vulnerabilities: 2 9 12 1 70

The graph is shown on the **Analysis Workbench**, **Project Inventory** tab, Inventory View, and Lookup Component Window (for a given component version).

The following sections provide more information about exploring the details for a security vulnerability so that you can better address the vulnerability's impact on your product code and take remedial action if necessary:

- Understanding Severity Levels for Security Vulnerabilities
- Examining Security Vulnerability Details
- Suppressing/Unsuppressing Security Vulnerabilities

Understanding Severity Levels for Security Vulnerabilities

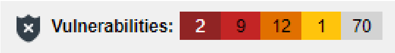
Code Insight obtains the severity level of a security vulnerability from the advisory database used to identify the vulnerability. The severity is based on the vulnerability's CVSS (Common Vulnerability Scoring System) score, which can have two different values depending on the scoring system used to calculate it—CVSS v2.0 or v3.x. Code Insight supports both systems for displaying the scores and severities of security vulnerabilities. The Code Insight System Administrator determines which scoring system your system uses.

CVSS v3.x Scoring System

When Code Insight is configured to report security vulnerabilities using the CVSS v3.x scoring system, the color-coded segments in **Vulnerabilities** bar graph represent the following severity levels:

- **Dark brown**—Critical severity (CVSS score 9.0 - 10.0)
- **Red**—High severity (CVSS score7.0 - 8.9)
- **Gold**—Medium severity (CVSS score 4.0 - 6.9)
- **Yellow**—Low severity (CVSS score 0.1 - 3.9)
- **None**—No severity available (N/A)

The following **Vulnerabilities** bar graph reflects vulnerability counts for an inventory item when CVSS v3.x scoring is used. (The counts are based on vulnerability scores in all CVSS v3 systems supported by Code Insight, currently v3.1 and v3.0. A given vulnerability can have only one v3 score—either a v3.1 or v3.0 score, not both.) This specific graph indicates 13 vulnerabilities of critical severity, 5 of high severity, 3 of medium severity, 0 of low severity, and 5 of unknown severity.



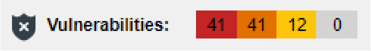
Vulnerabilities: 2 9 12 1 70

CVSS v2.0 Scoring System

When Code Insight is configured to use the CVSS v2.0 scoring system, the color-coded segments in graph represent the following severity levels:

- **Red**—High severity (CVSS score 7.0 - 10.0)
- **Gold**—Medium severity (CVSS score 4.0 - 6.9)
- **Yellow**—Low severity (CVSS score 0.1 - 3.9)
- **Gray**—Unknown severity (N/A)

The following **Vulnerabilities** bar graph reflects vulnerability counts for the same inventory item referenced in the previous section, but in this case CVSS v2.0 scoring is used. Note that the graph shows the same total number of vulnerabilities as the previous graph shows, but the severity distribution is different. In this case, the graph indicates 13 vulnerabilities of high severity, 8 of medium severity, 5 of low severity, and 80 of unknown severity:



Examining Security Vulnerability Details

The following procedure explains how to use the **Vulnerabilities** bar graph to obtain details about the security vulnerabilities associated with a specific inventory item or component version.

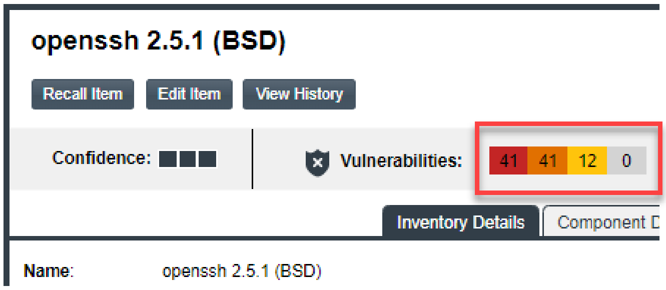
The **Vulnerabilities** bar graph is displayed for any of the following entities if the entity is associated with security vulnerabilities:

- A specific inventory item in the **Analysis Workbench** or in **Project Inventory**
- A component version in an inventory item’s **Lookup Component** window
- A component version in the **Global Component & License Lookup** window

Task ***To view security vulnerabilities for an inventory item, do the following:***

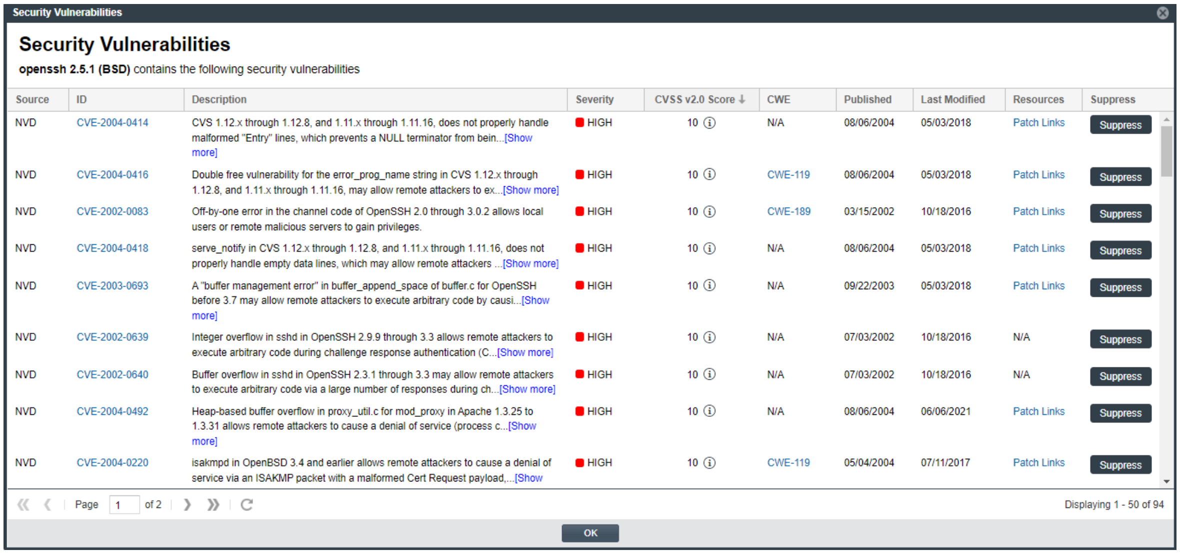
1. Click anywhere on the **Vulnerabilities** bar graph.

   (The graph is displayed only if vulnerabilities exist for the inventory item or component version.)

   Name: openssh 2.5.1 (BSD)

   The **Security Vulnerabilities** window is displayed. (This example uses the CVSS v3.x scoring system.)



Security Vulnerabilities

Security Vulnerabilities

openssh 2.5.1 (BSD) contains the following security vulnerabilities

Source	ID	Description	Severity	CVSS v2.0 Score ↓	CWE	Published	Last Modified	Resources	Suppress
NVD	CVE-2004-0414	CVS 1.12.x through 1.12.8, and 1.11.x through 1.11.16, does not properly handle malformed "Entry" lines, which prevents a NULL terminator from bein...[Show more]	🟥 HIGH	10 ⓘ	N/A	08/06/2004	05/03/2018	Patch Links	Suppress
NVD	CVE-2004-0416	Double free vulnerability for the error_prog_name string in CVS 1.12.x through 1.12.8, and 1.11.x through 1.11.16, may allow remote attackers to ex...[Show more]	🟥 HIGH	10 ⓘ	CWE-119	08/06/2004	05/03/2018	Patch Links	Suppress
NVD	CVE-2002-0083	Off-by-one error in the channel code of OpenSSH 2.0 through 3.0.2 allows local users or remote malicious servers to gain privileges.	🟥 HIGH	10 ⓘ	CWE-189	03/15/2002	10/18/2016	Patch Links	Suppress
NVD	CVE-2004-0418	serve_notify in CVS 1.12.x through 1.12.8, and 1.11.x through 1.11.16, does not properly handle empty data lines, which may allow remote attackers ...[Show more]	🟥 HIGH	10 ⓘ	N/A	08/06/2004	05/03/2018	Patch Links	Suppress
NVD	CVE-2003-0693	A "buffer management error" in buffer_append_space of buffer.c for OpenSSH before 3.7 may allow remote attackers to execute arbitrary code by causi...[Show more]	🟥 HIGH	10 ⓘ	N/A	09/22/2003	05/03/2018	Patch Links	Suppress
NVD	CVE-2002-0639	Integer overflow in sshd in OpenSSH 2.9.9 through 3.3 allows remote attackers to execute arbitrary code during challenge response authentication (C...[Show more]	🟥 HIGH	10 ⓘ	N/A	07/03/2002	10/18/2016	N/A	Suppress
NVD	CVE-2002-0640	Buffer overflow in sshd in OpenSSH 2.3.1 through 3.3 may allow remote attackers to execute arbitrary code via a large number of responses during ch...[Show more]	🟥 HIGH	10 ⓘ	N/A	07/03/2002	10/18/2016	N/A	Suppress
NVD	CVE-2004-0492	Heap-based buffer overflow in proxy_util.c for mod_proxy in Apache 1.3.25 to 1.3.31 allows remote attackers to cause a denial of service (process c...[Show more]	🟥 HIGH	10 ⓘ	N/A	08/06/2004	06/06/2021	Patch Links	Suppress
NVD	CVE-2004-0220	isakmpd in OpenBSD 3.4 and earlier allows remote attackers to cause a denial of service via an ISAKMP packet with a malformed Cert Request payload,...[Show	🟥 HIGH	10 ⓘ	CWE-119	05/04/2004	07/11/2017	Patch Links	Suppress

Note ▪ *When a security vulnerability is suppressed for the component version associated with the current inventory item, the vulnerability is not visible on the* ***Securities Vulnerabilities*** *window, nor is it reflected in the counts on* ***Vulnerabilities*** *bar graph.*

2. Examine the vulnerabilities in the **Security Vulnerabilities** list. For a description of the details shown for each vulnerability entry, see Security Vulnerabilities Window. Note the following about the list itself:

   - Each entry in the **Security Vulnerabilities** list identifies a specific security vulnerability directly associated with the selected inventory item. A vulnerability can be reported by the NVD (National Vulnerability Database) as a CVE (Common Vulnerabilities and Exposures) or by a research organization such as Secunia, Debian, or others. (Such organizations publish well-researched security advisories about CVEs that can include information not found in the NVD descriptions.)

   - By default, the list is sorted on the **CVSS <version> Score** column in descending order.

   - If a CVE is both published by the NVD and referenced in one or more advisories, the vulnerability is counted separately for each location. For example, a CVE that is published by the NVD and referenced in two advisories will have a count of 3 reflected in the vulnerability totals (as displayed on the project dashboards and **Vulnerabilities** bar graphs in the Web UI, as well as shown in API responses and the Project and Audit reports).

   - In cases where the vulnerability score is unknown (and reported as **N/A** in the list), the severity level of the vulnerability is reported as **None** or **Unknown**. (For more information about the vulnerability score and severity, see Security Vulnerabilities Window.)

     Note ▪ *Your feedback is welcome on how Code Insight should handle the severity and scoring of currently unscored vulnerabilities. The Code Insight team will do its best to incorporate the results of this feedback into the Code Insight vulnerability database. Contact Revenera Support with your suggestions.*

3. When you have finished with the **Security Vulnerabilities** window, click **OK** to close the window.

Suppressing/Unsuppressing Security Vulnerabilities

For various reasons, your site might want to suppress—that is, hide—a security vulnerability that is associated with one or more component versions used by your inventory. Once suppressed, the vulnerability is no longer published in reports, counted in vulnerability totals for inventory in projects, or automatically associated with inventory during future project scans in your Code Insight instance. For example, you might choose to suppress a vulnerability if you have taken remedial steps to protect your code against the vulnerability or if the vulnerability has proven to be a “false positive” (that is, is associated with an incorrect component version).

Likewise, you might want to unsuppress a security vulnerability that you have previously suppressed so that it is again visible and counted for inventory in projects.

Only a Code Insight System Administrator can suppress security vulnerabilities, as well as monitor currently suppressed vulnerabilities and unsuppress them as needed.

Any type of security vulnerability can be suppressed (and then unsuppressed when necessary):

- Vulnerabilities retrieved from the Code Insight Data Library during scans or as open alerts
- Custom vulnerabilities

The following sections provide more information about suppressing and unsuppressing security vulnerabilities:

- Suppressing Security Vulnerabilities
- Viewing Suppressed Security Vulnerabilities
- Unsuppressing Security Vulnerabilities

Suppressing Security Vulnerabilities

The System Administrator can suppress a security vulnerability at the Code Insight system level for one or more (or all) component versions associated with the vulnerability. The following provides more details:

- Effects of Suppressing a Security Vulnerability
- Suppressing a Security Vulnerability

Effects of Suppressing a Security Vulnerability

The suppression of a security vulnerability has an impact on your Code Insight system. That is, once a vulnerability is suppressed for a specific component version, it is no longer counted in vulnerability totals or is visibly listed at the project, inventory, and component-version levels. The count reduction is evident on the project dashboards and on the **Vulnerabilities** bar graphs in the Web UI, as well as in subsequently generated API responses and reports (Project and Audit). Likewise, the actual vulnerability is no longer visible in the list of vulnerabilities on the **Security Vulnerabilities** window (which is opened when you click a **Vulnerabilities** bar graph) or in API responses or reports.

Note *▪ The **Vulnerabilities** bar graph in the UI is shown on the **Inventory** view, in the **Lookup Component** window for a specific component version, and in **Inventory Details** for a given inventory item (both on the **Project Inventory** tab and in the **Analysis Workbench**).*

The following describes the impact that a security vulnerability suppressed for a specific component version has on other features of Code Insight:

- **Advanced Search on the Project Inventory tab and Inventory View**—When an inventory search is based the vulnerability name or severity, the results do not show inventory items that are related to the suppressed vulnerability.
- **Alerts**—Any open alerts for the suppressed vulnerability are automatically closed, and the open and closed alert counts are adjusted on the **Project Inventory** tab, in the **Analysis Workbench**, and on the **Inventory** view.
- **Policies**—Once a security vulnerability is suppressed, no changes are initially propagated to those review policies that are based on vulnerabilities. However, each time one of these policies is triggered thereafter (that is, when an inventory item is published), the policy ignores the suppressed vulnerability when determining whether to automatically approve or reject the published inventory item.

  Additionally, a change in policy due to the suppression of a vulnerability does not change the existing approval/rejection status of a published inventory item unless the item is manually recalled and then republished.
- **Subsequent scans and rescans**—Once a vulnerability is suppressed, it is not reflected in the results of subsequent rescans and initial scans, whether incremental or full.

Suppressing a Security Vulnerability

The following procedure is used to suppress a security vulnerability for one or more (or all) versions of an OSS or third-party component associated with your inventory.

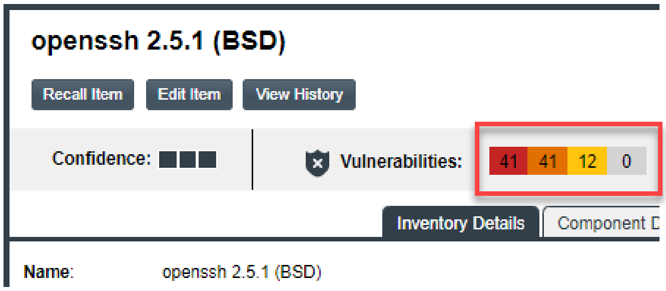
Note ▪ *A System Administrator can also suppress a security vulnerability using the **Suppress vulnerability** REST API. For more information about this API, see the Code Insight Swagger documentation, available from the **Help** > **REST API Guide** option on the Code Insight menu. (To access this menu, click the ☰ icon in the upper right corner of the Code Insight web page.)*

Task ***To suppress a security vulnerability, do the following:***

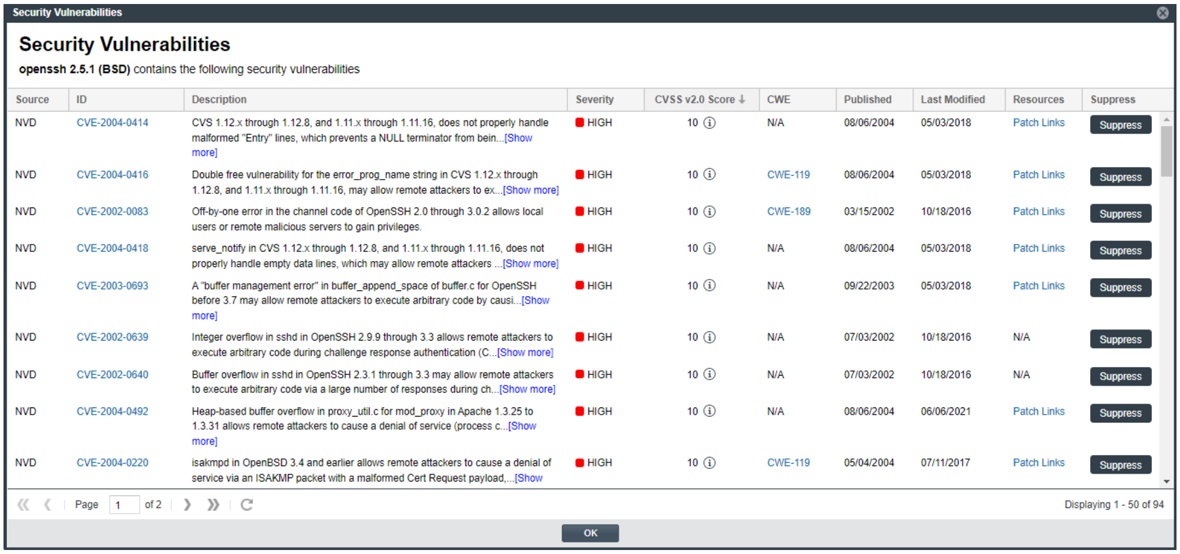
1. Locate the **Vulnerabilities** bar graph within the context of a given inventory item or component version.

   Note ▪ *The **Vulnerabilities** bar graph in the UI is shown on the **Inventory** view, in the **Lookup Component** window for a specific component version, and in **Inventory Details** for a given inventory item (both on the **Project Inventory** tab and in the **Analysis Workbench**).*

   The graph is displayed only if vulnerabilities exist for the inventory item or component version.
2. Click anywhere on the **Vulnerabilities** bar graph.



The **Security Vulnerabilities** window is displayed. (This example uses the CVSS v3.x scoring system.)



Security Vulnerabilities

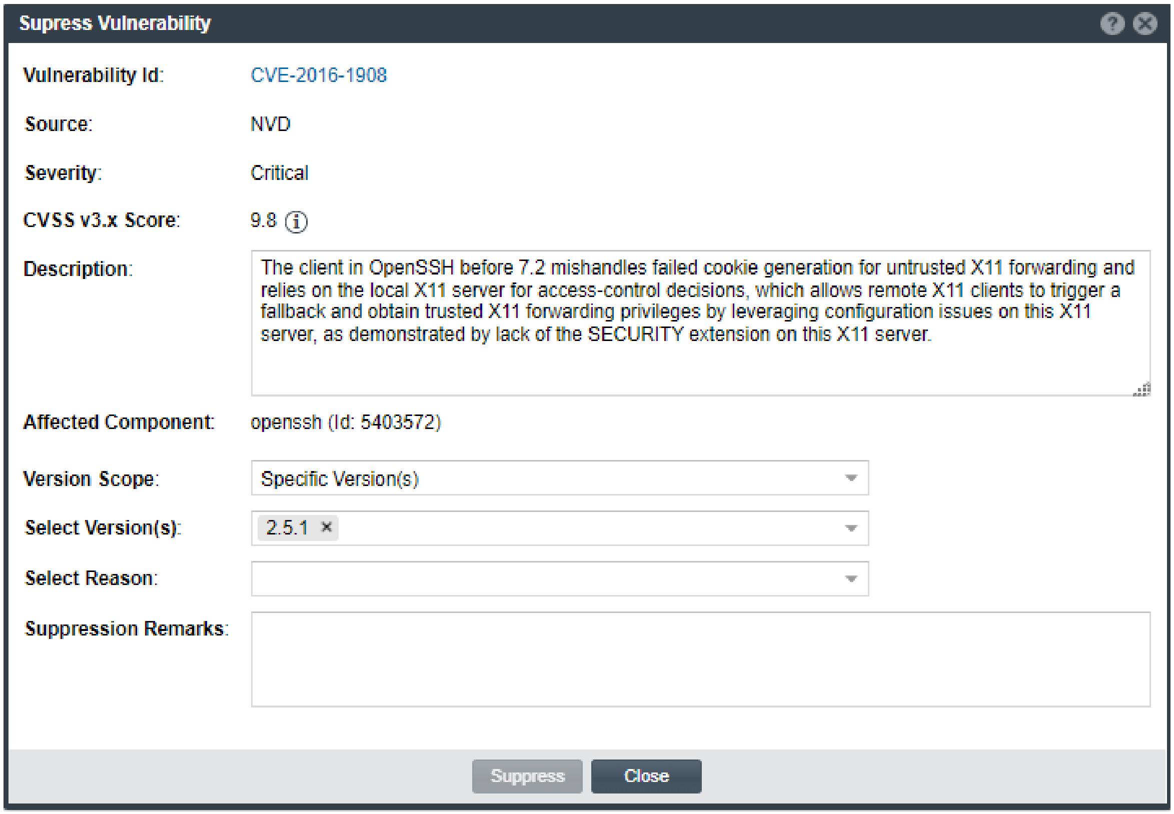
openssh 2.5.1 (BSD) contains the following security vulnerabilities

Source	ID	Description	Severity	CVSS v2.0 Score ↓	CWE	Published	Last Modified	Resources	Suppress
NVD	CVE-2004-0414	CVS 1.12.x through 1.12.8, and 1.11.x through 1.11.16, does not properly handle malformed "Entry" lines, which prevents a NULL terminator from bein...[Show more]	🟥 HIGH	10 ⓘ	N/A	08/06/2004	05/03/2018	Patch Links	Suppress
NVD	CVE-2004-0416	Double free vulnerability for the error_prog_name string in CVS 1.12.x through 1.12.8, and 1.11.x through 1.11.16, may allow remote attackers to ex...[Show more]	🟥 HIGH	10 ⓘ	CWE-119	08/06/2004	05/03/2018	Patch Links	Suppress
NVD	CVE-2002-0083	Off-by-one error in the channel code of OpenSSH 2.0 through 3.0.2 allows local users or remote malicious servers to gain privileges.	🟥 HIGH	10 ⓘ	CWE-189	03/15/2002	10/18/2016	Patch Links	Suppress
NVD	CVE-2004-0418	serve_notify in CVS 1.12.x through 1.12.8, and 1.11.x through 1.11.16, does not properly handle empty data lines, which may allow remote attackers ...[Show more]	🟥 HIGH	10 ⓘ	N/A	08/06/2004	05/03/2018	Patch Links	Suppress
NVD	CVE-2003-0693	A "buffer management error" in buffer_append_space of buffer.c for OpenSSH before 3.7 may allow remote attackers to execute arbitrary code by causi...[Show more]	🟥 HIGH	10 ⓘ	N/A	09/22/2003	05/03/2018	Patch Links	Suppress
NVD	CVE-2002-0639	Integer overflow in sshd in OpenSSH 2.9.9 through 3.3 allows remote attackers to execute arbitrary code during challenge response authentication (C...[Show more]	🟥 HIGH	10 ⓘ	N/A	07/03/2002	10/18/2016	N/A	Suppress
NVD	CVE-2002-0640	Buffer overflow in sshd in OpenSSH 2.3.1 through 3.3 may allow remote attackers to execute arbitrary code via a large number of responses during ch...[Show more]	🟥 HIGH	10 ⓘ	N/A	07/03/2002	10/18/2016	N/A	Suppress
NVD	CVE-2004-0492	Heap-based buffer overflow in proxy_util.c for mod_proxy in Apache 1.3.25 to 1.3.31 allows remote attackers to cause a denial of service (process c...[Show more]	🟥 HIGH	10 ⓘ	N/A	08/06/2004	06/06/2021	Patch Links	Suppress
NVD	CVE-2004-0220	isakmpd in OpenBSD 3.4 and earlier allows remote attackers to cause a denial of service via an ISAKMP packet with a malformed Cert Request payload,...[Show	🟥 HIGH	10 ⓘ	CWE-119	05/04/2004	07/11/2017	Patch Links	Suppress

Displaying 1 - 50 of 94

3. Locate the security vulnerability that you want to suppress, and click the **Suppress** button next its **ID** field.

   The **Suppress Vulnerability** window is displayed.



Vulnerability Id:	CVE-2016-1908
Source:	NVD
Severity:	Critical
CVSS v3.x Score:	9.8
Description:	The client in OpenSSH before 7.2 mishandles failed cookie generation for untrusted X11 forwarding and relies on the local X11 server for access-control decisions, which allows remote X11 clients to trigger a fallback and obtain trusted X11 forwarding privileges by leveraging configuration issues on this X11 server, as demonstrated by lack of the SECURITY extension on this X11 server.
Affected Component:	openssh (Id: 5403572)
Version Scope:	Specific Version(s)
Select Version(s):	2.5.1 ×
Select Reason:	
Suppression Remarks:	

4. Complete all editable fields on the window to define the vulnerability suppression. For a description of these fields, see Suppress Vulnerability Window.

5. Click **Suppress**. Then click **OK** in the pop-up to acknowledge that the vulnerability has been successfully suppressed for the specified component versions.

   If other vulnerabilities exist for the inventory item, you are returned to the **Security Vulnerabilities** window. The suppressed vulnerability should no longer be visible in the window (if you suppressed the vulnerability for the component version of the current inventory item).

   If this vulnerability is the only one for the inventory item, you are returned to the previous context from which you opened the **Security Vulnerabilities** window (for example, the **Inventory Details** tab or the **Lookup Component** window). The **Vulnerabilities** bar graph count should be reduced in the specific severity segment associated with the suppressed vulnerability (if you suppressed the vulnerability for the component version of the current inventory item).

Viewing Suppressed Security Vulnerabilities

The following procedure describes how to obtain a view all currently suppressed security vulnerabilities in your Code Insight instance. Only a Code Insight System Administrator can access this view.

Task ***To obtain a view of all currently suppressed security vulnerabilities in Code Insight, do the following:***

1. Click the following icon in the upper right corner of the Code Insight web page to open the Code Insight main menu:

   ☰

2. Select **DATA LIBRARY** from the menu to open the **Data Library** page.
3. Select **Suppressed Vulnerabilities** tab to view the list of the currently suppressed security vulnerabilities in Code Insight. (This tab is visible to only Code Insight System Administrators.) From this tab, you can do the following:
   - For each suppressed vulnerability, easily review the OSS or third-party component with which the vulnerability is associated and the specific versions of that component for which the vulnerability is currently suppressed. For a description of this tab, see Suppressed Vulnerabilities Tab.
   - View a pop-up that shows details about a given vulnerability by clicking the **Information** icon next to the ID of the suppressed vulnerability in the **Vulnerability Id** column.
   - View a pop-up window that shows the vulnerability-suppression details of every component version for which a given vulnerability is suppressed. To open the pop-up, click the **Information** icon next to the versions listed for the vulnerability in the **Affected Versions** column. The details shown for each of the component versions listed on the pop-up include the user who suppressed the vulnerability for the version, the date and time of the suppression, the reason for the suppression, and additional remarks.
   - To unsuppress a given vulnerability, click its associated **Unsuppress** button. For further details, see Unsuppressing Security Vulnerabilities

Unsuppressing Security Vulnerabilities

The System Administrator can unsuppress a security vulnerability for one, some, or all of the component versions for which it was previously suppressed. The following sections provide more details:

- Effects of Unsuppressing a Security Vulnerability
- Unsuppressing a Security Vulnerability

Effects of Unsuppressing a Security Vulnerability

When you unsuppress a security vulnerability, the effects of the vulnerability's previous suppression are reversed. That is, once a vulnerability is unsuppressed for a specific component version, it is now counted in vulnerability totals or is visibly listed at the project, inventory, and component-version levels. The count increase is evident on the project dashboards and on the **Vulnerabilities** bar graphs in the Web UI, as well as in subsequently generated API responses and reports (Project and Audit). Likewise, the actual vulnerability is now visible in the list of vulnerabilities on the **Security Vulnerabilities** window (which is opened when you click a **Vulnerabilities** bar graph) or in API responses or reports.

Note *▪ The **Vulnerabilities** bar graph in the UI is shown on the **Inventory** view, in the **Lookup Component** window for a specific component version, and in **Inventory Details** for a given inventory item (both on the **Project Inventory** tab and in the **Analysis Workbench**).*

The following describes the impact that unsuppressing a security vulnerability has on other features of Code Insight:

- **Advanced Search on the Project Inventory tab and Inventory View**—When an inventory search is based the vulnerability name or severity, the results now list any inventory items that are associated the unsuppressed vulnerability.
- **Alerts**—Any alerts that were automatically closed due to the previous vulnerability suppression are automatically reopened. Open and closed alert counts are adjusted to reflect the changes on the **Project Inventory** tab, in the **Analysis Workbench**, and on the **Inventory** view.
- **Policies**—Once a security vulnerability is unsuppressed, no changes are initially propagated to those review policies that are based on vulnerabilities. However, each time one of these policies is triggered thereafter (that is, when an inventory item is published), the policy will now consider the vulnerability when determining whether to automatically approve or reject the published inventory item.

  Additionally, a change in policy due to the unsuppression of a vulnerability does not change the existing approval/rejection status of a published inventory item unless the item is manually recalled and then republished.
- **Subsequent scans and rescans**—Once a vulnerability is unsuppressed, it is reflected in the results of subsequent rescans and initial scans, whether incremental or full.

Unsuppressing a Security Vulnerability

The following procedure is used to unsuppress a security vulnerability for one, some, or all of component versions for which it was previously suppressed.

Only a Code Insight System Administrator can perform this operation.

Note *▪ A System Administrator can also unsuppress a security vulnerability using the **UnSuppress vulnerability** REST API. For more information about this API, see the Code Insight Swagger documentation, available from the*

Help *> **REST API Guide** option on the main Code Insight menu. (To access this menu, click the ☰ icon in the upper right corner of the Code Insight web page.)*

Task ***To unsuppress a security vulnerability, do the following:***

1. Open the **Suppressed Vulnerabilities** tab using the steps described in Viewing Suppressed Security Vulnerabilities.
2. In the list of suppressed vulnerabilities (in grid format), locate the vulnerability that you want to unsuppress, and click its associated **Unsuppress** button.

   The **Unsuppress Vulnerability** window is displayed.

3. Complete all editable fields on the window to define the unsuppression process for the vulnerability. For a description of these fields, see Unsuppress Vulnerability Window.
4. Click **Unsuppress**. Then click **OK** in the pop-up to acknowledge that the vulnerability has been successfully unsuppressed for the specified component versions.

   You are returned to the **Suppressed Vulnerabilities** tab. The list of suppressed vulnerabilities on the tab is now modified in one of two ways:

   - If the vulnerability was unsuppressed for one or some of the component versions for which it was previously suppressed, those versions are no longer listed for the vulnerability. The remaining suppressed versions are still listed for the vulnerability.
   - If the vulnerability was unsuppressed for all of the component versions for which it was previously suppressed, the vulnerability is longer shown in the list.

Managing Security Vulnerability Alerts

Code Insight provides the ability to view and clear security vulnerability alerts. When the Electronic Update or Library Refresh process is run, it will generate these alerts for any new security vulnerabilities that are associated with inventory. The alerts allow you to investigate the most recent vulnerabilities and their effect on your project code, if any. Once you have addressed vulnerability impact, either by determining that vulnerability poses no threat to your application or by performing the required remediation to remove the threat, you can close the alert.

***Note** ▪ An alert can be automatically closed when its associated security vulnerability is manually suppressed by a Code Insight System Administrator. See Suppressing/Unsuppressing Security Vulnerabilities for more information.*

When the Electronic Update or Library Refresh generates security vulnerability alerts, an email notification is sent to the Project Contact of each project containing inventory impacted by the alerts. Additionally, remediation tasks can be automatically created for any affected inventory that is subsequently rejected, as dictated by a project's policy profile and remediation options (see Updating Inventory Review and Remediation Settings for a Project).

Users can view the alerts for a given project from the **Inventory Details** pane in the **Analysis Workbench** or from the **Project Inventory** tab (and from the **Inventory** view).

Refer to these topics for more information:

- Accessing Security Vulnerability Alerts
- Using the Alerts Dialog to Manage Alerts

Accessing Security Vulnerability Alerts

The following methods provide access to the **Alerts** dialog, which allows you to view and manage the security vulnerability alerts impacting inventory in a given project:

- From Email Notifications
- From the Analysis Workbench
- From Project Inventory

From Email Notifications

Project Contacts can be alerted via email to any projects and inventory items that contain new security vulnerabilities so that they can be reviewed and acted upon if necessary. For email alerts to be sent, the email server must be enabled and configured. For more information, see “Configuring an Email Server” in the *Installation & Configuration Guide*.

Vulnerability alert emails are sent as part of the Electronic Update and the daily Library Refresh. A vulnerability alert is generated for each new security vulnerability mapped to a published inventory item. While viewing the alert email, click any of the hyperlinked text in the email to open Code Insight or an advisory web site to view additional information about the alert.

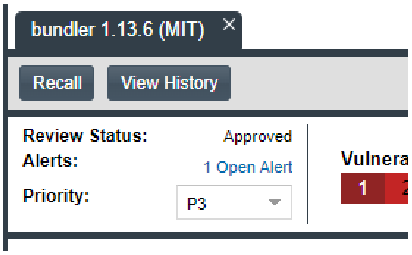
From the Analysis Workbench

This procedure describes how to access security vulnerability alerts from the **Analysis Workbench**.

Task ***To view security vulnerability alerts from the Analysis Workbench, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. From the **Inventory Item** pane on the right, click the inventory item for which you want to check for alerts. The **Inventory Details** tab for the selected item is opened.

   If open alerts exist, the **Alerts** field provides a link to view them. If no alerts exist, the field shows **None**.

   Review Status: Approved
   Alerts: 1 Open Alert
   Priority: P3

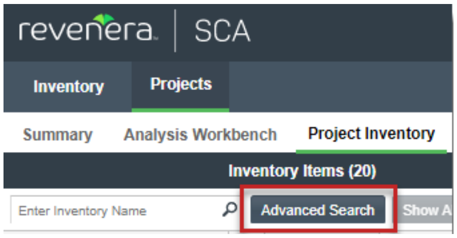
3. Click the link to open the **Alerts** dialog, where you can view the open (and closed) alerts for the inventory item.

From Project Inventory

This procedure describes how to access the security vulnerability alerts for a specific inventory item from the **Project Inventory** tab for a project.

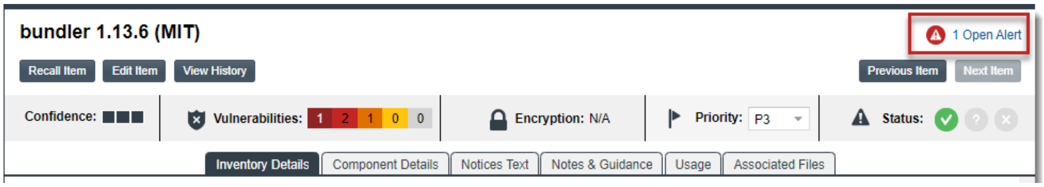
Task ***To view security vulnerability alerts from Project Inventory, do the following:***

1. Open the **Project Inventory** tab for the desired project (see Displaying Project Inventory). The **Inventory Items** list is displayed in the left pane.
2. (Optional) To filter the **Inventory Items** list to show only inventory that have alerts, click **Advanced Search**, select the **Inventory with Open Alerts** option, and click **Apply**.



3. From the **Inventory Items** list results, click the inventory item whose alerts you want to check. Information about the selected inventory item is displayed in the right pane.

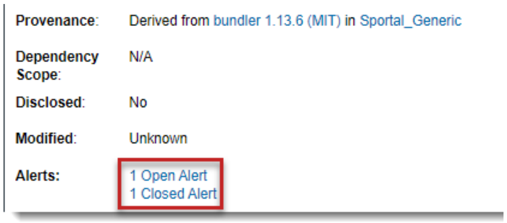
   - For a quick check on any *open* alerts, locate for the ⚠ icon in the header of this page.



bundler 1.13.6 (MIT)

Confidence: ■■■ | Vulnerabilities: 1 2 1 0 0 | Encryption: N/A | Priority: P3 | Status: ✓

   - To view open or closed alerts, open the **Inventory Details** tab in the right pane. If alerts exist, the **Alerts** field on the tab shows separate links to view open or closed alerts, as appropriate.



Provenance:	Derived from bundler 1.13.6 (MIT) in Sportal_Generic
Dependency Scope:	N/A
Disclosed:	No
Modified:	Unknown
Alerts:	1 Open Alert 1 Closed Alert

4. Click the associated link to open the **Alerts** dialog, where you can view details about the alerts.

Using the Alerts Dialog to Manage Alerts

The **Alerts** dialog shows the list of current alerts for a given inventory item in a project. The following describes how to use this dialog to manage security vulnerability alerts for an inventory item:

- Alert Details
- Changing the Priority of an Alert
- Changing the Status of an Alert

Alert Details

The following columns describe each alert entry listed in the **Alert** dialog. To filter the list of alerts, see Filtering Alerts.

Table 2-13 ▪ Alert Details

Column	Description
Type	The alert type. Currently, only **New Vulnerability** alerts are available.
Date	The date that the alert was generated.
Priority	The priority of the alert, which, by default, is based on the official severity level of the security vulnerability associated with the alert, as described here: ● **High**—The default when the vulnerability severity level is Critical, High, or None in the CVSS v3.x scoring system or, in the CVSS v2.0 scoring system, High or Unknown. ● **Medium**—The default when the severity level is Medium in either scoring system. ● **Low**—The default when the severity level is Low in either scoring system. You can change this value as needed. See Changing the Priority of an Alert. For more information about the severity levels for security vulnerabilities, see Viewing Security Vulnerabilities for Project Inventory.
Status	The status of the alert in Code Insight: ● **Open**—The alert needs to be addressed. ● **Closed**—The alert has been addressed (usually because remediation was performed or it was determined to be a false positive for your code or the security vulnerability was suppressed). Hover over the ⓘ icon to see who closed the alert and when. Change this value as needed. See Changing the Status of an Alert.

Table 2-13 ▪ Alert Details (cont.)

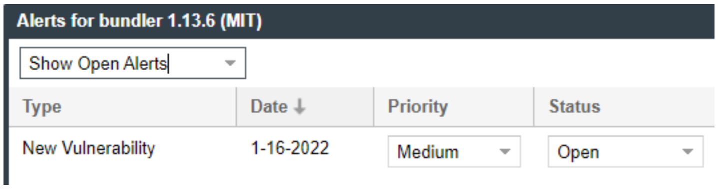
Column	Description
Details	Details about the security vulnerability: ● **Source**—The advisory database in which the Code Insight located the vulnerability, such as **NVD** (National Vulnerability Database), **Secunia** (Secunia Advisories), **Debian Advisories**, or others. ● **ID**—The identification number of the vulnerability associated with the Common Vulnerabilities and Exposures (CVE). If this is a hyperlinked value, click it to go to the actual entry for the vulnerability on the advisory website. ● **CVSS Score**—The score of the vulnerability based on the Common Vulnerability Scoring System (CVSS). The values of the CVSS score range from 0.1 to 10, with 10 being the most serious. If the vulnerability has no score, the value is **N/A**. ● **Description**—The description of the vulnerability as found in the advisory database. Also see Security Vulnerabilities Associated with Inventory.

Filtering Alerts

Use the following procedure to change the view of the list of alerts on the **Alerts** dialog.

Task ***To filter the list of alerts, do the following:***

1. Locate the dropdown list at the top of the **Alerts** dialog:



Alerts for bundler 1.13.6 (MIT)

Show Open Alerts

Type	Date ↓	Priority	Status
New Vulnerability	1-16-2022	Medium	Open

2. Select one of the following options from the list:
   - **Show Open Alerts**—Display only open alerts.
   - **Show Closed Alerts**—Display only closed alerts.
   - **Show All Alerts**—Display both closed and open alerts. This option will only be available if more than one alert is available.

Changing the Priority of an Alert

Use the following procedure to change the **High**, **Medium**, or **Low** priority of an alert. The priority indicates the urgency with which the security vulnerability associated with the alert needs to be addressed. The initial priority value defaults to the severity of the security vulnerability itself, but you can change this priority based on your site's needs. For more information about vulnerability priority, see Alert Details.

Note* ▪ *If the Code Insight System Administrator has switched Code Insight from CVSS v2.0 to CVSS v3.x scoring or vice versa, you might notice a change in the* Severity *and* CVSS Score *for the vulnerability associated with the alert. However, the alert* Priority *should not change from its value current at the time of the switch.

Task ***To change the priority of a security vulnerability alert, do the following:***

1. Open the **Alerts** dialog for a given inventory item in a project, as described in Accessing Security Vulnerability Alerts.
2. In the **Priority** column, select the new priority.

Changing the Status of an Alert

Use the following procedure to change the **Open** or **Closed** status of a security vulnerability alert. Usually, you close an alert because the associated security vulnerability has been addressed in your product through remediation or the alert is a false positive. You might need to reopen an alert because further remediation is required.

For more information about the **Open** and **Closed** statuses, see Alert Details.

Task ***To change the status of a security vulnerability alert, do the following:***

1. Open the **Alerts** dialog for a given inventory item in a project, as described in Accessing Security Vulnerability Alerts.
2. In the **Status** column for a given alert, select either option from the dropdown list:
    - **Open** to reactivate the alert.
    - **Closed** to indicate that the alert has been addressed.

Creating and Editing Custom Components

Code Insight enables you to create custom components that represent OSS or third-party software not found in the Code Insight Data Library or that represent commercial software that you want to track as part of your Bill of Materials. A custom component is created from the **Global Component & License Lookup** tab or within the context of the inventory item with which you want to associate it. The custom component is saved to the Code Insight database (and indexed in the background in the Code Insight Data Library, where it is then available for global use).

Once the custom component is created, an inventory item can be associated with a registered instance of the component—that is, a unique component-version-license combination that you define. The custom component is also available for use by policies and is included in the Notices report. You can also edit any custom component.

The following topics describe how to create and edit custom components:

- Creating a Custom Component
- Editing a Custom Component
- Custom Component Properties
- Supported Forge-URL Domains for Custom Components

Creating a Custom Component

Use the following steps to create a custom component:

- Step 1: Access the “Lookup Component” Window
- Step 2: Create the Custom Component
- Step 3: Associate an Instance of the Custom Component with the Inventory Item

Step 1: Access the “Lookup Component” Window

You can start a Lookup Component search either from the **Lookup Component** window (available when you create or edit inventory) or from the **Components** tab on the **Global Component & License Lookup** tab.

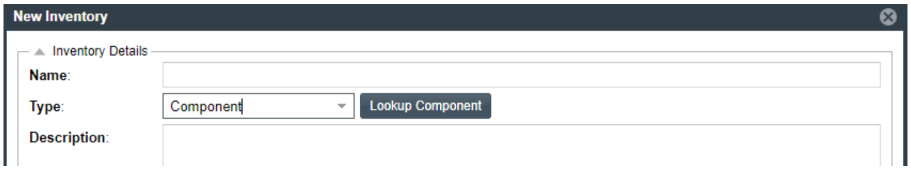
- Accessing the Lookup Component Window
- Accessing the Global Component & License Lookup Tab

Accessing the Lookup Component Window

The **Lookup Component** window in accessed within the context of creating or editing inventory in a project.

Task ***To access the Lookup Component feature, do the following:***

1. For the inventory item that you are currently creating or editing in the **Analysis Workbench** or on the **Project Inventory** tab, ensure that the **Type** field for the inventory item is **Component**.


New Inventory
Inventory Details
Name:
Type: Component

Description:

2. Click **Lookup Component** next to the **Type** field to open the **Lookup Component** window.

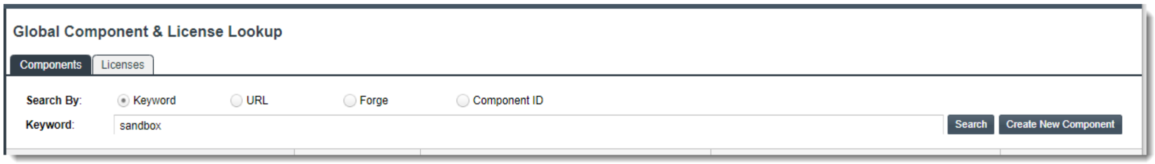
Accessing the Global Component & License Lookup Tab

Follow these steps to open the **Components** tab on the **Global Component & License Lookup** tab.

Task ***To access the Global Component & License Lookup tab, follow these steps:***

1. Open the **Global Component & License Lookup** tab, using the instructions in Accessing the Global Component & License Lookup Feature.

2. Select the **Components** tab.



Global Component & License Lookup

Search By: ● Keyword ○ URL ○ Forge ○ Component ID

Keyword: sandbox

Step 2: Create the Custom Component

Based on the information you have about the custom component you want to create, use one of the following methods to create the component.

- Create the Custom Component Based on a Keyword in Its Name or Title
- Create the Custom Component Based on Its Project or Forge URL
- Create the Custom Component Based on Its Forge
- Create the Custom Component in Free Form

Once a custom component is saved, it is added to the Code Insight database (so that is available for Lookup Component searches for any inventory) and indexed in the Code Insight Data Library (so that it is available for global component searches against the Data Library.) For important information about indexing, see Note About the Indexing Process for Custom Components.

Note About the Indexing Process for Custom Components

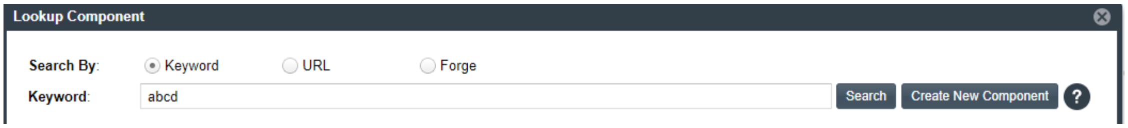
When a custom component is saved, the process of indexing the component in the Code Insight Data Library begins immediately in the background. Indexing can take up to a half minute for a single component. To avoid Code Insight server issues, you should wait until the indexing process is finished before using the Global Component & License Lookup feature (see Exploring Components Globally) or the **Component Search** REST API to search components in the Data Library. You can always check the Code Insight `core.update.log` file or the Tomcat logs to determine the status of the component-indexing process.

Create the Custom Component Based on a Keyword in Its Name or Title

Use the following method to create the custom component based on a keyword in the name or title you intend to give the component.

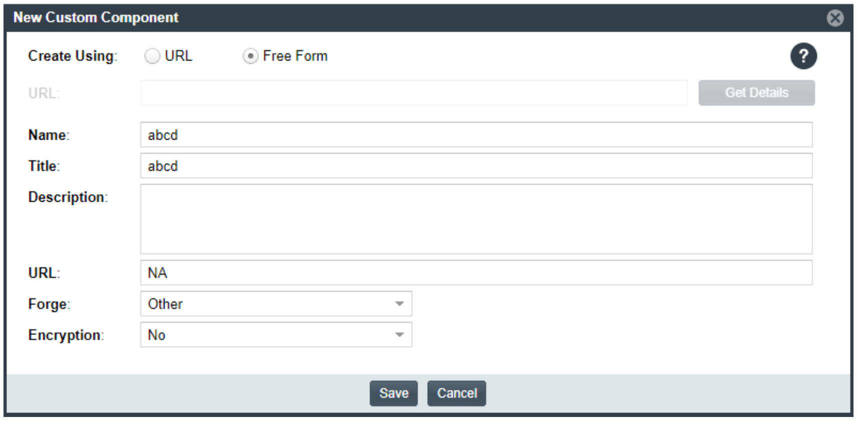
Task ***To create a custom component based on a keyword in its name or title:***

1. After you have performed Step 1: Access the “Lookup Component” Window, select the **Keyword** option either on the **Lookup Component** window or on the **Components** tab on the **Global Component & License Lookup** tab.
2. In the **Keyword** field, enter a string used in the name of the component you are creating.



Search By: ● Keyword ○ URL ○ Forge
Keyword: abcd

3. Click **Create New Component** to open the **New Custom Component** window, showing the **Name** and **Title** fields automatically populated with the keyword you entered.



Create Using: ○ URL ● Free Form
URL:
Name: abcd
Title: abcd
Description:
URL: NA
Forge: Other
Encryption: No

Note that window opens in the Free Form format, enabling you to add missing values and edit pre-populated values for the component as needed.

4. Update the component fields as necessary, noting that the **Name**, **Title**, and **URL** fields are required. (Click ? in the upper part of the window for examples of the standard name, title, and URL conventions used by the **Forge** value you select.) If you do not know the URL, enter **NA**. For more information about these fields, see Custom Component Properties.

   Keep in mind that the **Name** and **Title** of the component you are creating must be unique within the Code Insight Data Library.

5. Click **Save**.
   - If the component is successfully created, it is immediately saved to the Code Insight database and listed in the **Lookup Component** window or on the **Components** tab on the **Global Component & License Lookup** tab. (The **Components** tab automatically filters to the new component by its component ID.)
   - The process of indexing the component in the Code Insight Data Library begins immediately in the background. Once indexed, the custom component is made available for global component searches that users can perform against the Code Insight Data Library. For important information about the indexing process and these searches, see Note About the Indexing Process for Custom Components.

- If the component name and title combination already exists in the Data Library, an error message is displayed. You can edit the custom component details to provide a unique name or title; or (through the **Lookup Component** window) locate the already-existing component and associate it with the inventory item.

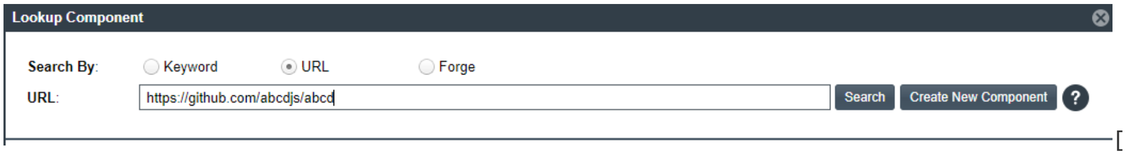
6. (Performed only from the **Lookup Component** window) Continue with Step 3: Associate an Instance of the Custom Component with the Inventory Item.

Create the Custom Component Based on Its Project or Forge URL

Use the following method to create the custom component based on its known project or forge URL.

Task ***To create a custom component based on its URL:***

1. After you have performed Step 1: Access the "Lookup Component" Window, select the **URL** option either on the **Lookup Component** window or on the **Components** tab on the **Global Component & License Lookup** tab.
2. In the **URL** field, enter URL for the component. As you enter the URL, it is checked for an acceptable format (such as `http://example.com`), but not for validity.



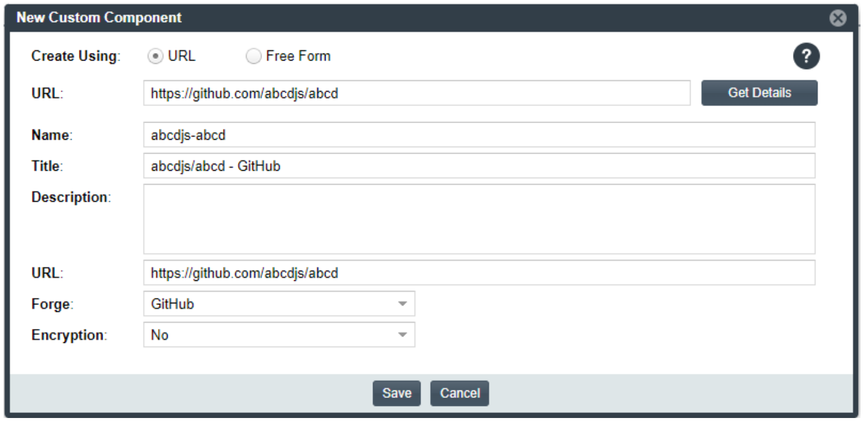
Lookup Component

Search By: ○ Keyword ● URL ○ Forge

URL: https://github.com/abcdjs/abcd

3. Click **Create New Component**.

- If Code Insight recognizes the URL you entered as belonging to one of the forge-URL domains currently supported for custom component creation, the **New Custom Component** window opens, showing component fields—including **Name**, **Title**, **URL**, and **Forge**—automatically populated with values based on domain conventions. (For the list of supported domains, see Supported Forge-URL Domains for Custom Components.)



New Custom Component

Create Using: ● URL ○ Free Form

URL: https://github.com/abcdjs/abcd

Name: abcdjs-abcd

Title: abcdjs/abcd - GitHub

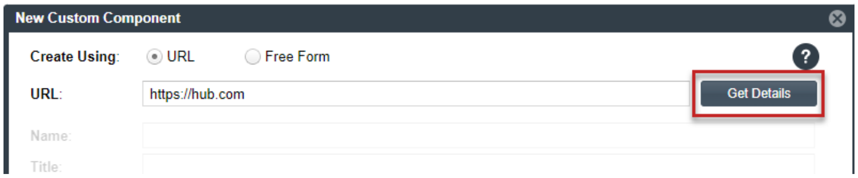
Description:

URL: https://github.com/abcdjs/abcd

Forge: GitHub

Encryption: No

- If Code Insight does not recognize the URL as belonging to a supported domain (that is, it is unable to parse the URL), the **New Custom Component** window opens showing the URL only. You must click **Get Details** to complete the component fields manually in **Free Form** mode. (Click **OK** on the "Unable to parse URL..." message box to proceed to Free Form mode.)



New Custom Component

Create Using: ● URL ○ Free Form

URL: https://hub.com

4. Update the component fields as necessary, noting that the **Name**, **Title**, and **URL** fields are required. For more information about these fields, see Custom Component Properties.

    - If the URL you initially entered (at the top of the window) belongs to a supported domain and you edit that URL, click **Get Details** to update the remaining field values according to forge-URL domain conventions.
    - If you must manually provide field values because the URL you initially entered does not belong to a supported domain, click ? in the upper part of the window for examples of the standard name, title, and URL conventions used by the **Forge** value you select.

5. Click **Save**.

    - If the component is successfully created, it is immediately saved to the Code Insight database and listed in the **Lookup Component** window or on the **Components** tab on the **Global Component & License Lookup** tab. (The **Components** tab automatically filters to the new component by its component ID.)
    - The process of indexing the component in the Code Insight Data Library begins immediately in the background. Once indexed, the custom component is made available for global component searches that users can perform against the Code Insight Data Library. For important information about the indexing process and these searches, see Note About the Indexing Process for Custom Components.
    - If the component name and title combination already exists in the Data Library, an error message is displayed. You can edit the custom component details to provide a unique name or title; or (through the **Lookup Component** window) locate the already-existing component and associate it with the inventory item.

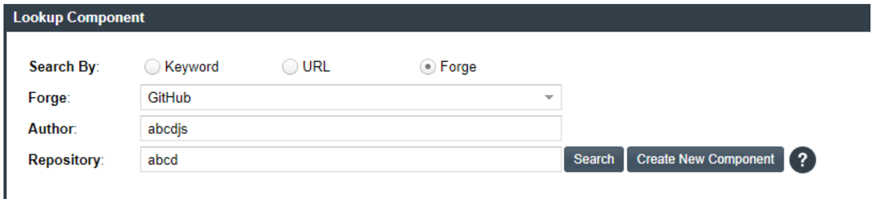
6. (Performed only from the **Lookup Component** window) Continue with Step 3: Associate an Instance of the Custom Component with the Inventory Item.

Create the Custom Component Based on Its Forge

Use the following method to create the custom component if you know the forge (that is, the third-party project repository) of the custom component you are creating.

Task ***To create a custom component based on its forge:***

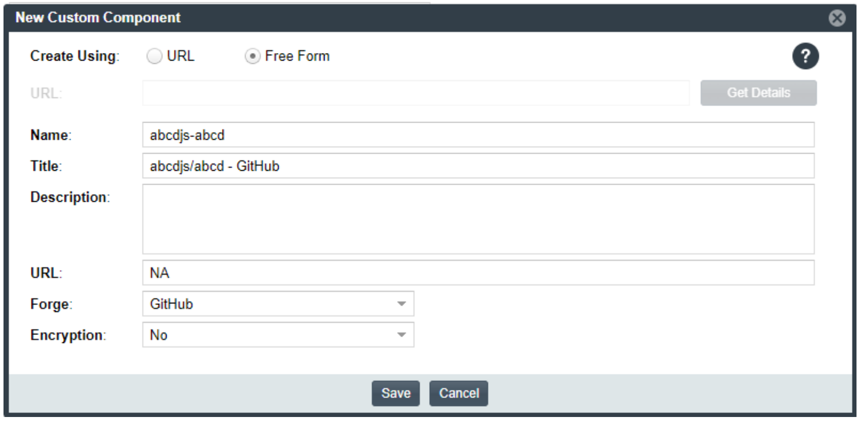
1. After you have performed Step 1: Access the “Lookup Component” Window, select the **Forge** option either on the **Lookup Component** window or on the **Components** tab on the **Global Component & License Lookup** tab.
2. From the **Forge** field, select the forge of the custom component, and enter the information required to identify the forge. (This information varies by forge.)


Lookup Component

Search By: ○ Keyword ○ URL ● Forge
Forge: GitHub
Author: abcdjs
Repository: abcd

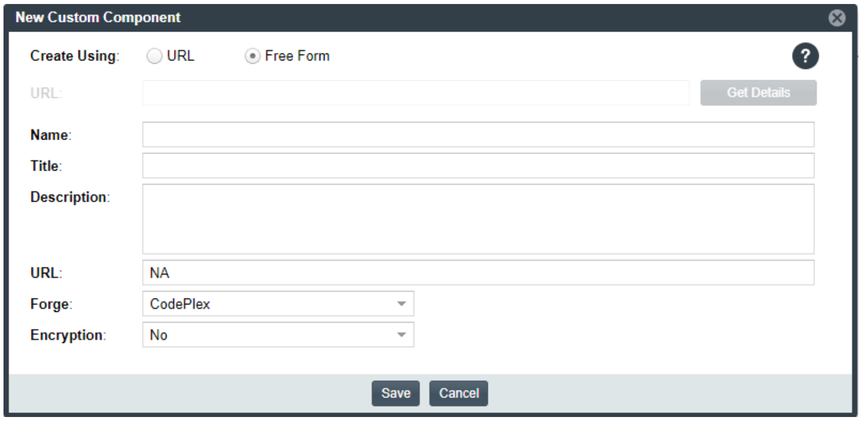
3. Click **Create New Component**.

    - If the forge you entered is one of the forge-URL domains currently supported for custom component creation, the **New Custom Component** window opens, showing component fields—including **Name**, **Title**, and **Forge**—automatically populated with values based on domain conventions. (For the list of supported domains, see Supported Forge-URL Domains for Custom Components.)


New Custom Component

Create Using: ○ URL ● Free Form
URL:
Name: abcdjs-abcd
Title: abcdjs/abcd - GitHub
Description:
URL: NA
Forge: GitHub
Encryption: No

    - If the forge you entered is not one of the supported domains, the **New Custom Component** window opens, showing the **Forge** field automatically populated with the forge type you selected.


New Custom Component

Create Using: ○ URL ● Free Form
URL:
Name:
Title:
Description:
URL: NA
Forge: CodePlex
Encryption: No

Note that **New Custom Component** window opens in the **Free Form** mode, enabling you to add missing values and edit pre-populated values for the component as needed.

4. Update the component fields as necessary, noting that the **Name**, **Title**, and **URL** fields are required. (Click ? in the upper part of the window for examples of the standard name, title, and URL conventions used by the **Forge** value you selected.) If you do not know the URL, enter **NA**. For more information about these fields, see Custom Component Properties.

5. Click **Save**.

   - If the component is successfully created, it is immediately saved to the Code Insight database and listed in the **Lookup Component** window or on the **Components** tab on the **Global Component & License Lookup** tab. (The **Components** tab automatically filters to the new component by its component ID.)

   - The process of indexing the component in the Code Insight Data Library begins immediately in the background. Once indexed, the custom component is made available for global component searches that users can perform against the Code Insight Data Library. For important information about the indexing process and these searches, see Note About the Indexing Process for Custom Components.

   - If the component name and title combination already exists in the Data Library, an error message is displayed. You can edit the custom component details to provide a unique name or title; or (through the **Lookup Component** window) locate the already-existing component and associate it with the inventory item.

6. (Performed only from the **Lookup Component** window) Continue with Step 3: Associate an Instance of the Custom Component with the Inventory Item.

Create the Custom Component in Free Form

Use following method to create the custom component when you do not know the component name (keyword), URL, or forge or when you simply want to provide your own values (for example, when creating a custom component for commercial software you want to track in Code Insight for inclusion in the Bill of Materials).

Task ***To create a custom component in Free Form mode:***

1. After you have performed Step 1: Access the “Lookup Component” Window, click **Create New Component** either on the **Lookup Component** window or on the **Components** tab on the **Global Component & License Lookup** tab.

   The window opens in the **Free Form** form, enabling you to provide your own values for the component fields. (No field values are automatically populated.)

   Optionally, you can switch to the **URL** form on the **New Custom Component** window, enabling you to enter a project or forge URL by which to create the component. Once you enter the URL, you must click **Get Details** to continue:

   - If the URL belongs to a supported forge-URL domain, component fields are automatically populated with values based on domain conventions, as described in Create the Custom Component Based on Its Project or Forge URL.

   - If the URL does not belong to a supported domain (that is, Code Insight is unable to parse the URL), you must click **OK** on the resulting “Unable to parse URL...” message box to proceed to manually complete component fields in **Free Form** mode (continue with the next step 2).

2. Update the component fields, noting that the **Name**, **Title**, and **URL** fields are required. (Click ? in the upper part of the window for examples of the standard name, title, and URL conventions used by the **Forge** value you select.) If you do not know the URL, enter **NA**. For more information about these fields, see Custom Component Properties.

3. Click **Save**.

    - If the component is successfully created, it is immediately saved to the Code Insight database and listed in the **Lookup Component** window or on the **Components** tab on the **Global Component & License Lookup** tab. (The **Components** tab automatically filters to the new component by its component ID.)
    - The process of indexing the component in the Code Insight Data Library begins immediately in the background. Once indexed, the custom component is made available for global component searches that users can perform against the Code Insight Data Library. For important information about the indexing process and these searches, see Note About the Indexing Process for Custom Components.
    - If the component name and title combination already exists in the Data Library, an error message is displayed. You can edit the custom component details to provide a unique name or title; or (through the **Lookup Component** window) locate the already-existing component and associate it with the inventory item.

4. (Performed only from the **Lookup Component** window) Continue with Step 3: Associate an Instance of the Custom Component with the Inventory Item.

Step 3: Associate an Instance of the Custom Component with the Inventory Item

A component instance is a unique component-version-license entity that you can associate with an inventory item. The following procedure creates an instance for the new custom component and associates it with the inventory item you are creating or editing.

You can create multiple instances of the custom component to associate with inventory items. Each instance is saved to the Code Insight Data Library and made available for global use.

Task ***To associate an instance of the new custom component with the inventory item you are creating or editing, follow these steps:***

1. On the **Lookup Component** window showing the custom component you just created, click **Show Instances**.
2. Click **Register New Instance** to create a component instance.
3. Complete the instance registration by selecting **Create Custom Version** to specify a version and then selecting the license to associate with the instance.
4. Click **Use This Instance** next to the new instance to associate it with the inventory item.

You are returned to the inventory item you are creating or updating. The **Name** and **Description** editable fields for the inventory item are automatically populated with information based on the registered instance you selected. Additionally, the **Component** and **License** fields are displayed, showing the component, its version, and the license for the instance.

You can now proceed with completing the inventory creation or update.

Editing a Custom Component

You can edit a custom component either from the **Global Component & License Lookup** tab or within the context of an inventory item with which it is associated. The changes you make are immediately saved to the Code Insight database and made available for Lookup Component searches for any inventory.

Additionally, once changes to a component are saved, the process of indexing these changes in the Code Insight Data Library begins immediately in the background. When the indexing process is complete, the updated custom component is made available for global component searches that users can perform against the Code Insight Data Library. For important information about the indexing process and these searches, see Note About the Indexing Process for Custom Components.

Use either procedure to edit a custom component:

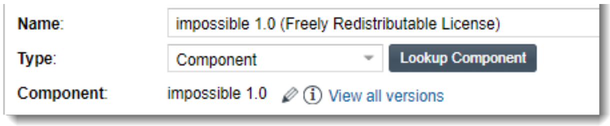
- Editing a Custom Component Within the Context of an Inventory Item
- Editing a Custom Component From the Global Component & License Lookup Tab

Editing a Custom Component Within the Context of an Inventory Item

Use the following procedure to edit a existing custom component within the context of creating or editing an inventory item (in the **Analysis Workbench** or on the **Project Inventory** tab) associated with the component. (See Editing a Custom Component for background information about this editing process.)

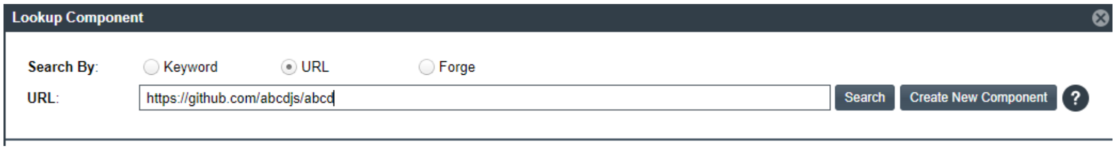
Task ***To edit a custom component within the context of an inventory item associated with the component, do the following:***

1. For the inventory item that you are currently creating or editing in the **Analysis Workbench** or on the **Project Inventory** tab, ensure that the **Type** field for the inventory item is **Component**.



Name: impossible 1.0 (Freely Redistributable License)
Type: Component
Component: impossible 1.0

2. Click **Lookup Component** next to the field.
3. If necessary, search for the custom component by keyword, URL, or forge. For search guidelines, see Guidelines for Lookup Component Searches.



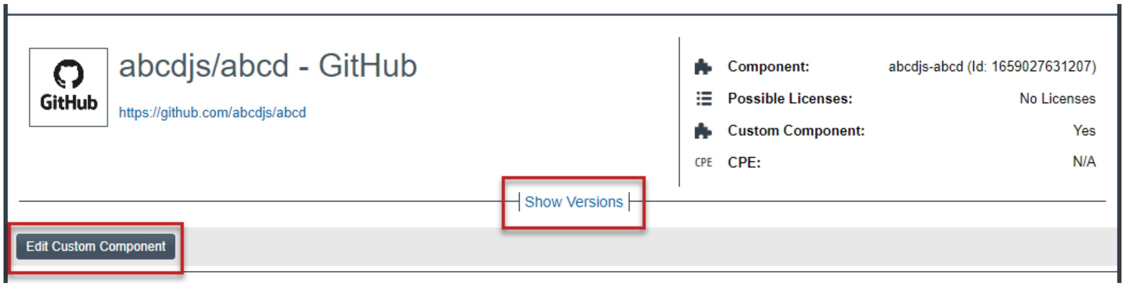
Lookup Component

Search By: ○ Keyword ● URL ○ Forge
URL: https://github.com/abcdjs/abcd

Note ▪ *A custom component listed in the **Lookup Component** window also has an **Edit Custom Component** button associated with it.*

4. Once you have located the custom component, edit it using any one or both of these methods:

    - Click **Edit Custom Component** to open the **Edit Custom Component** window, update the component properties, and save them. See Custom Component Properties for field descriptions.
    - Click **Show Versions** to create one or more component-version-license instances for the component.



abcdjs/abcd - GitHub

https://github.com/abcdjs/abcd

Component:	abcdjs-abcd (Id: 1659027631207)
Possible Licenses:	No Licenses
Custom Component:	Yes
CPE:	N/A

5. Click **Use This Instance** next to a component-version-license instance to associate it with the inventory item or click **Cancel** to close the **Lookup Component** window. (If you click **Cancel**, edits to the custom component are still saved.)
6. If necessary, proceed with completing any other updates to the inventory item.

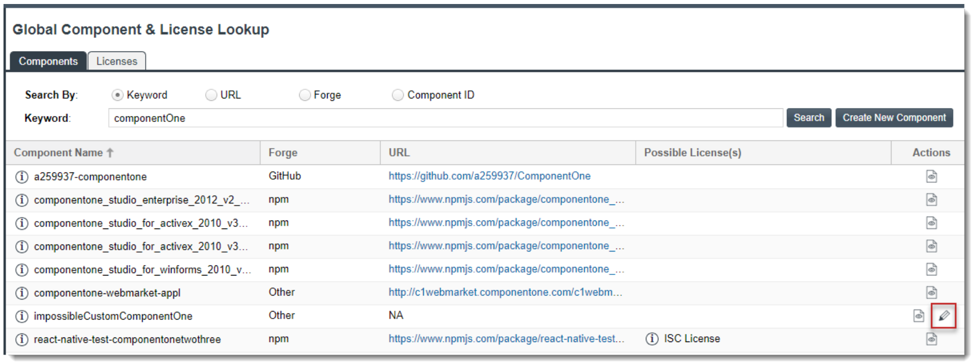
Editing a Custom Component From the Global Component & License Lookup Tab

The following procedure describes how to edit a custom component using the Global Component & License Lookup feature. (See Editing a Custom Component for background information about this editing process.)

Task ***To edit a custom component from the Global Component & License Lookup tab, do the following:***

1. Open the **Global Component & License Lookup** tab, using the instructions in Accessing the Global Component & License Lookup Feature.
2. Select the **Components** tab to view a list of OSS and third-party components—both standard and custom—in the Code Insight Data Library.
3. To locate the custom component that you want to update, filter the list by component keyword, URL, forge, or ID. For further information on how to filter the list, see Setting Up a Global Component Search.

    In the search results, only custom components are listed with an **Edit Component** icon in the **Actions** column.



Global Component & License Lookup

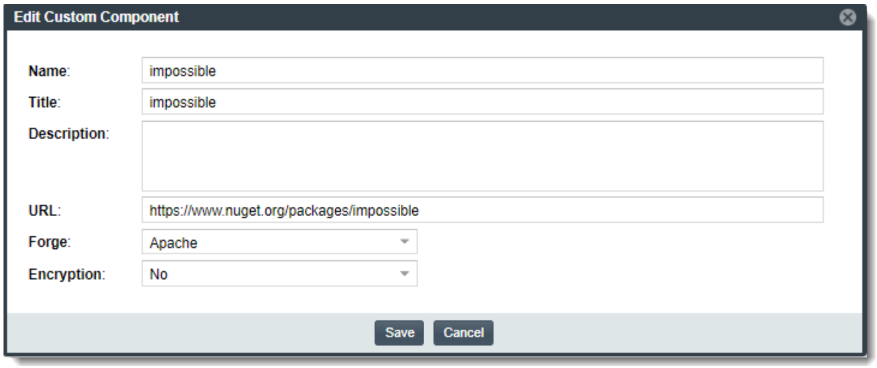
Search By: ● Keyword ○ URL ○ Forge ○ Component ID

Keyword: componentOne

Component Name ↑	Forge	URL	Possible License(s)	Actions
ⓘ a259937-componentone	GitHub	https://github.com/a259937/ComponentOne		
ⓘ componentone_studio_enterprise_2012_v2_...	npm	https://www.npmjs.com/package/componentone_...		
ⓘ componentone_studio_for_activex_2010_v3...	npm	https://www.npmjs.com/package/componentone_...		
ⓘ componentone_studio_for_activex_2010_v3...	npm	https://www.npmjs.com/package/componentone_...		
ⓘ componentone_studio_for_winforms_2010_v...	npm	https://www.npmjs.com/package/componentone_...		
ⓘ componentone-webmarket-appl	Other	http://c1webmarket.componentone.com/c1webm...		
ⓘ impossibleCustomComponentOne	Other	NA		
ⓘ react-native-test-componentonetwothree	npm	https://www.npmjs.com/package/react-native-test...	ⓘ ISC License	

4. Click the **Edit Component** icon in the **Actions** column for the custom component you want to update.

5. On the **Edit Custom Component** window, update the component properties as needed. See Custom Component Properties for field descriptions.



Edit Custom Component

Name:	impossible
Title:	impossible
Description:	
URL:	https://www.nuget.org/packages/impossible
Forge:	Apache
Encryption:	No

6. Click **Save**.

   The **Components** tab automatically filters to the updated custom component by its component ID.

Custom Component Properties

The following describes the fields on the **New Custom Component** and **Edit Custom Component** windows used to define a custom component. When you are creating a custom component, certain fields might be automatically populated based on information entered on the **Lookup Component** window. However, any field can be edited.

Table 2-14 ▪ Fields to Define a Custom Component

Component Field	Description
Create Using	(Available when creating a component) The “form” mode used to create the custom component. The information you entered on the **Lookup Component** window (or on the **Components** tab on the **Global Component & License Lookup** tab) initially determines the mode used, but you can switch modes as needed. ● **URL**—This mode is used when creating the component based on the URL for its project or forge (see Create the Custom Component Based on Its Project or Forge URL). This form includes the actual **URL** value, located beneath the **Create Using** section, which is either pre-populated from the **Lookup Component** window or manually entered here. If necessary, click **Get Details** to view component fields to complete or update them. If the URL is recognized as belonging to a supported forge-URL domain supported for custom component creation, required component fields are automatically populated on the **Lookup Component** window. See the **URL** field description next in this table for more information. ● **Free Form**—This mode is used to define or update component fields manually when creating the custom component. Certain fields might be automatically populated based on the information entered previously on the **Lookup Component** window.
URL	(Available under **Create Using** during component creation when the **URL** form is selected) The project or forge URL for which you are creating the custom component. If the URL is recognized as belonging to a supported forge-URL domain supported for custom component creation, required component fields are automatically populated according to domain conventions. Any change made to this **URL** is automatically updated to the other field values when you click **Get Details**. If the URL does not belong to a supported domain, you must manually provide all remaining necessary component details. (Click **Get details** to display the fields.) For information about supported forge-URL domains, see Supported Forge-URL Domains for Custom Components.

Table 2-14 ▪ Fields to Define a Custom Component (cont.)

Component Field	Description
Name	(Required) The name of the custom component. During component creation, this field might be automatically populated based on information entered on the **Lookup Component** window. To help you provide this value in an appropriate format when creating a custom component, click ? in the upper part of the window for examples of the standard name, title, and URL conventions used by different forge-URL domains. ***Note ▪*** *The **Name** and **Title** combination of the new component must be unique in the Data Library.*
Title	(Required) The component title. During component creation, this field might be automatically populated based on information entered on the **Lookup Component** window. To help you provide this value in an appropriate format when creating a custom component, click ? in the upper part of the window for examples of the standard name, title, and URL conventions used by different forge-URL domains. ***Note ▪*** *The **Name** and **Title** combination of the custom component must be unique in the Data Library.*
Description	A description of the custom component to provide any additional meaningful information about the component.
URL	(Required) The URL for the project or forge of the custom component. During component creation, this field is pre-populated with the same **URL** value provided in the URL form (see the **Create Using** field) when that URL is recognized as belonging to a supported forge-URL domain. Otherwise, to help you provide this value in an appropriate format when creating a custom component, click ? in the upper part of the window for examples of the standard name, title, and URL conventions used by different forge-URL domains. If you do not know the URL for the component, enter **NA**.

Table 2-14 ▪ Fields to Define a Custom Component (cont.)

Component Field	Description
Forge	The third-party project repository used by the custom component. During component creation, this field is automatically populated with an appropriate value when you are creating the component based on either of the following: ● Its forge (see Create the Custom Component Based on Its Forge). ● The URL for its project or forge *and* the URL belongs to a forge-URL domain supported by the custom-component creation process. For more information, see Create the Custom Component Based on Its Project or Forge URL and Supported Forge-URL Domains for Custom Components. Otherwise, the default **Other** is displayed. However, you can select any other forge from the dropdown list.
Encryption	**Yes** or **No** value depicting whether this component supports encryption. The default is **No**.

Supported Forge-URL Domains for Custom Components

When creating or editing a custom component, you can select any forge supported by Code Insight and provide free-form component details. However, the forge might require that the URL and other component details be in a certain format. During the component-creation process (not the editing process), if the user initiates component creation by providing a URL or forge that the creation process recognizes as belonging to a supported domain, it automatically populates component fields with values formatted according to domain conventions.

The following are the forge-URL domains currently supported by the custom-component creation process:

- NuGet Gallery
- npm
- SourceForge
- RubyGems

For other forges that you might use to create custom components, you can click ? on the **Lookup Component** and **New Custom Component** windows for guidance on how to format the URL and the component name and title according to forge conventions.

Creating Custom Component Versions

Code Insight lets you create custom versions for existing components easily from the **Versions for** <**component**> window, accessed from the **Components** tab in **Global Component & License Lookup** or within the context of the inventory item are editing in the **Analysis Workbench** or on the **Project Inventory** tab. When defining a new version, you are required to associate it with a license. You are then given the option to have the system “remember” the new component-version-license combination so that all future inventory generated by the system for the new component version is automatically mapped to this license.

- Step 1: Accessing the Versions Window
- Step 2: Creating the Component Version

Step 1: Accessing the Versions Window

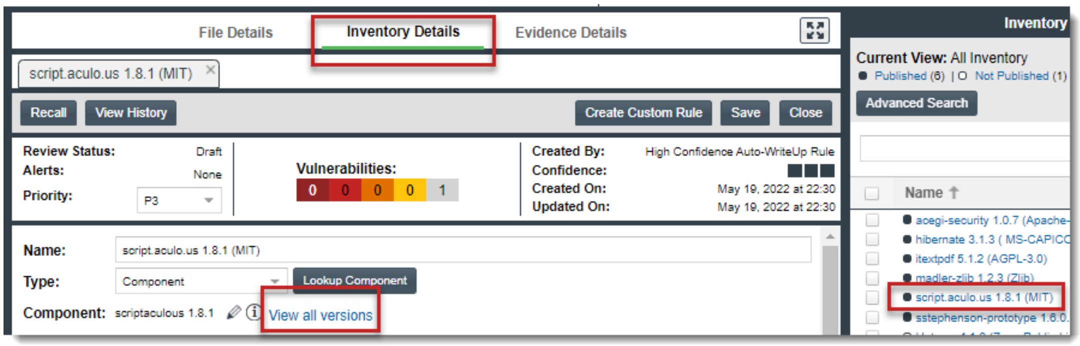
Use one of these methods to access the **Versions for** <**component**> window to add a new component version.

When Editing an Inventory Item in the Analysis Workbench

Follow these steps to access the **Versions for** <**component**> window within the context of editing an inventory item in the **Analysis Workbench** for a project.

Task ***To access the Versions windows when editing an inventory item in the Analysis Workbench, do the following:***

1. Open the **Analysis Workbench** for the desired project. (For instructions, see Opening the Analysis Workbench.)
2. In the **Inventory Items** pane on the right, select the inventory item you want to edit. The **Inventory Details** tab in the middle pane is refreshed with details for the selected item.
3. On the **Inventory Details** tab, locate the **Component** field and, next to the item’s component name and version, click **View all versions**.



script.aculo.us 1.8.1 (MIT)

Review Status: Draft
Alerts: None
Priority: P3

Vulnerabilities: 0 0 0 0 1

Created By: High Confidence Auto-WriteUp Rule
Confidence: ■■■
Created On: May 19, 2022 at 22:30
Updated On: May 19, 2022 at 22:30

Name: script.aculo.us 1.8.1 (MIT)
Type: Component
Component: scriptaculous 1.8.1

Current View: All Inventory
● Published (6) | ○ Not Published (1)

When Editing an Inventory Item in Project Inventory

Follow these steps to access the **Versions for** <**component**> window within the context of editing an inventory item on the **Project Inventory** tab for a project.

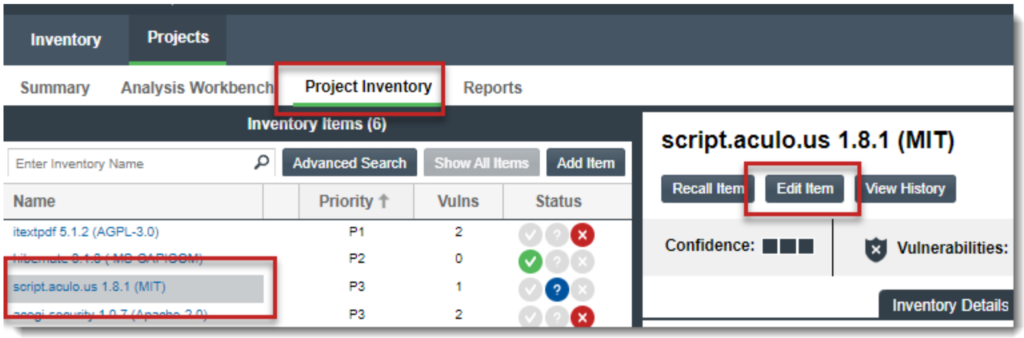
Task ***To access the Versions windows when editing an inventory item on the Project Inventory tab, do the following:***

1. Open the project containing the inventory whose published inventory item you want to edit. (For instructions, see Opening a Project.)

2. Select the **Project Inventory** tab for the project.

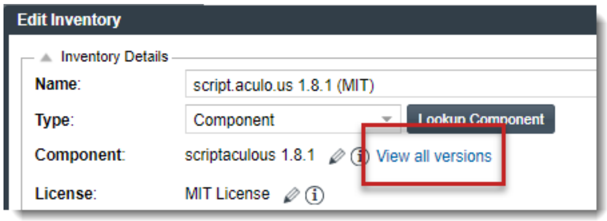
3. From the **Inventory Items** list on the left, select the inventory item you want to edit. Details for the selected inventory item populate the **Project Inventory Details** pane on the right.

4. Click the **Edit Item** button in the header of the **Project Inventory Details** pane.



Name	Priority	Vulns	Status
itextpdf 5.1.2 (AGPL-3.0)	P1	2	✓ ? ✗
hibernate 3.1.3 (MS-CAPICOM)	P2	0	✓ ? ✗
script.aculo.us 1.8.1 (MIT)	P3	1	✓ ? ✗
osgi-security 1.0.7 (Apache 2.0)	P3	2	✓ ? ✗

5. In the **Edit Inventory** window for the inventory item, locate the **Component** field and, next to the item's component name and version, click **View all versions**.



Name: script.aculo.us 1.8.1 (MIT)
Type: Component
Component: scriptaculous 1.8.1
License: MIT License

When Exploring Components with the Global Component & License Lookup Feature

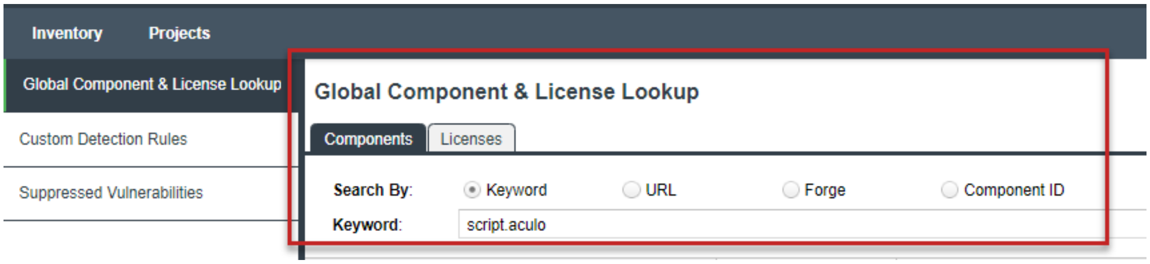
Follow these steps to access **Versions for** <**component**> window from the **Components** tab on the **Global Component & License Lookup** tab.

Task ***To access the Global Component & License Lookup tab, follow these steps:***

1. Open the **Global Component & License Lookup** tab, using the instructions in Accessing the Global Component & License Lookup Feature.

2. Select the **Components** tab to view a list of OSS and third-party components that are standard to the Code Insight Data Library and those that are custom.

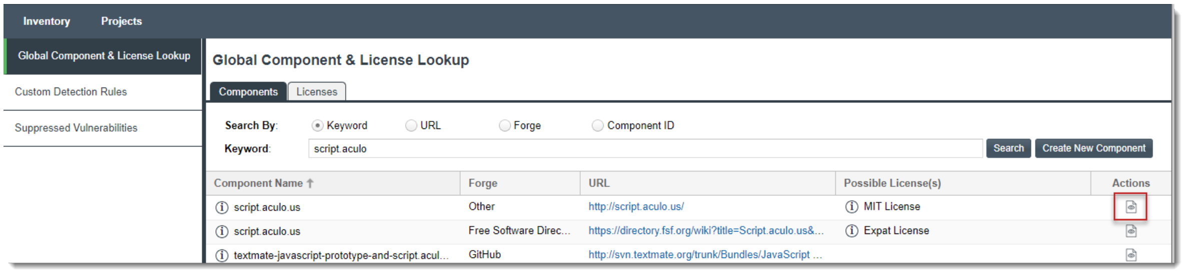
3. To locate the component to which you want to add a version, filter the list by component keyword, URL, forge, or ID. For further information on how to filter the list, see Setting Up a Global Component Search.


Inventory
Projects
Global Component & License Lookup
Custom Detection Rules
Suppressed Vulnerabilities
Global Component & License Lookup
Components
Licenses

Search By: ● Keyword ○ URL ○ Forge ○ Component ID

Keyword: script.aculo

4. In the row for the component, click the **View Versions** icon in the **Actions** column.


Inventory
Projects
Global Component & License Lookup
Custom Detection Rules
Suppressed Vulnerabilities
Global Component & License Lookup
Components
Licenses
Search
Create New Component

Search By: ● Keyword ○ URL ○ Forge ○ Component ID

Keyword: script.aculo

Component Name ↑	Forge	URL	Possible License(s)	Actions
ⓘ script.aculo.us	Other	http://script.aculo.us/	ⓘ MIT License	🗎
ⓘ script.aculo.us	Free Software Direc...	https://directory.fsf.org/wiki?title=Script.aculo.us&...	ⓘ Expat License	🗎
ⓘ textmate-javascript-prototype-and-script.acul...	GitHub	http://svn.textmate.org/trunk/Bundles/JavaScript ...		🗎

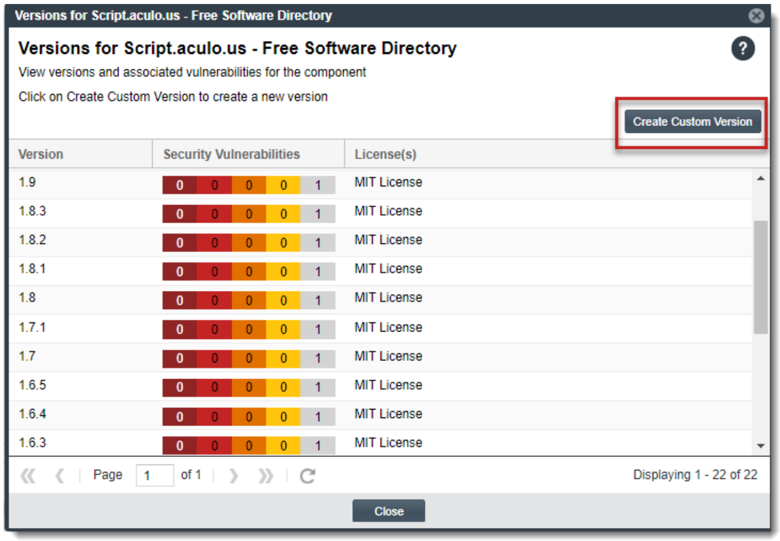
The **Versions for** <**component**> window is opened.

Step 2: Creating the Component Version

Once you have opened the **Versions for** <**component**> window, use this procedure to create a new component version.

Task ***To create a new component version, do the following:***

1. On the **Versions for** <**component**> window, click the **Create Custom Version** button.



Versions for Script.aculo.us - Free Software Directory

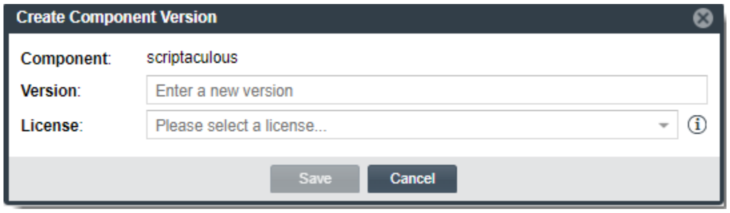
View versions and associated vulnerabilities for the component

Click on Create Custom Version to create a new version

Version	Security Vulnerabilities	License(s)
1.9	0 0 0 0 1	MIT License
1.8.3	0 0 0 0 1	MIT License
1.8.2	0 0 0 0 1	MIT License
1.8.1	0 0 0 0 1	MIT License
1.8	0 0 0 0 1	MIT License
1.7.1	0 0 0 0 1	MIT License
1.7	0 0 0 0 1	MIT License
1.6.5	0 0 0 0 1	MIT License
1.6.4	0 0 0 0 1	MIT License
1.6.3	0 0 0 0 1	MIT License

Displaying 1 - 22 of 22

The **Create Custom Version** window is displayed, showing the component name (which is not editable) and the fields required to define the version.



Create Component Version

Component: scriptaculous

Version: Enter a new version

License: Please select a license...

2. Complete the fields:

Field	Description
Version	Enter the new version string.
License	From the dropdown, select the license to associate with the new version. The licenses are organized in two categories on the dropdown: **System Suggested License** (available only if applicable to the component version) and **Other Licenses**. (For a description of these license categories, see License Categories in the License Dropdown.) You can click the ⓘ icon next to the selected license to view its details and text. For a description of the details, see License Details from the Code Insight Data Library.

3. Click **Save**.

   - If you have selected the license from the **Other Licenses** category or from multiple licenses in the **System Suggested Licenses** category, the **Update License Mapping** window is displayed. This window provides the option to have the system “remember” this license mapping for all future inventory that the system generates for the new component version across projects. (For more information about this window and the option to save your license mapping at the system level, see Specifying a User-Preferred License Mapping.) Proceed to step 4.

   - If you have a selected the only license in the **System Suggested License** category, the new version is created. Click **Close** to return to the previous window.

4. Select the appropriate option on the **Update License Mapping** window:

   - **Yes**—All future inventory items that the system generates for the version across projects are automatically mapped to the license you selected. This component-version-license combination is saved to the database and made available in the **Versions for** <**component**> window and in the **Lookup**

     Component window (as a registered version with a user-preferred-license icon 👤✓). The license is also listed in the **License** dropdown as the **User Preferred License** when you edit an inventory item for the component version.

     You are returned to the **Versions for** <**component**> window.

   - **No**—This component-version-license combination is saved to the database and made available in the **Versions for** <**component**> window and in **Lookup Component** window (as a registered instance).

However, any future system-generated inventory for this component version will be mapped to the license associated with this version in Code Insight Data Library.

You are returned to the **Versions for** <**component**> window.

- **Cancel**—Return to the **Create Component Version** to revisit the process of creating the new version.

Creating and Editing Custom Licenses

Code Insight enables you to create custom licenses that represent licenses not found in the Code Insight Data Library or commercial EULAs that are typically not included in the Data Library. The opportunity to create or edit a custom license is made available during certain processes in which you have an option to associate a license with an inventory item. You can also create a custom license while exploring licenses in the Code Insight Data Library from the **Global Component & License Lookup** tab.

The custom licenses are saved to the Code Insight database and made available for immediate license lookups during inventory creation or editing processes and from the **Global Component & License Lookup** tab.

The following topics describe how to create and edit custom licenses:

- Creating a Custom License
- Editing a Custom License
- Custom License Properties

Creating a Custom License

Use the following steps to create a custom license:

- Step 1: Initiate the Creation of a Custom License
- Step 2: Create the Custom License

Step 1: Initiate the Creation of a Custom License

You have the option to create a custom license within the context of any of the following:

- When Creating or Editing a “Component” Inventory Item
- When Creating or Editing a “License Only” Inventory Item
- When Searching Licenses with the Global Component & License Lookup Feature
- When Creating or Updating a License Policy

When Creating or Editing a “Component” Inventory Item

You can create a custom license within the context of creating or editing an inventory item of the type **Component** in the **Analysis Workbench** or from the **Project Inventory** tab.

Task ***To create a custom license when creating a “component” inventory item, do the following:***

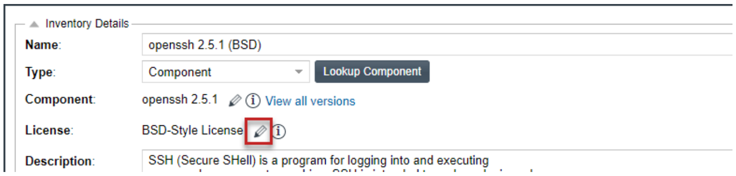
1. Start the appropriate procedure for creating or editing an inventory item with **Component** selected for its **Type** field. See References to Full Instructions for Creating or Editing Inventory for a quick reference to documentation links on how to create or edit inventory.

2. Proceed as follows:

   - **If you editing the properties of an existing inventory item without performing a Lookup Component procedure**—Proceed to the step 3 at any time.

   - **If you are creating an inventory item *or* selecting a new component-version-license instance for an existing item**—Proceed to the next step after your have performed the Look Component procedure and have returned to the tab or dialog from which are you creating or editing the inventory item. (The Lookup Component feature does not allow you create a custom license for a component-version-license instance.)

3. From the tab or dialog from which you are creating or editing the inventory item, click ✎ next to the **License** or **Component** field.

   Inventory Details

   Name: openssh 2.5.1 (BSD)

   Type: Component

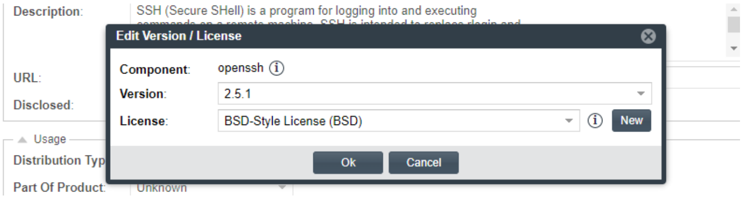
   Component: openssh 2.5.1 ✎ ⓘ View all versions

   License: BSD-Style License ✎ ⓘ

   Description: SSH (Secure SHell) is a program for logging into and executing

   The **Edit Version/License** dialog is displayed.

4. To the right of the **License** dropdown list, click **New** to display the **Create Custom License** window.

   Description: SSH (Secure SHell) is a program for logging into and executing

   URL:

   Disclosed:

   Usage

   Distribution Typ

   Part Of Product: Unknown

   Edit Version / License

   Component: openssh ⓘ

   Version: 2.5.1

   License: BSD-Style License (BSD) ⓘ

5. Continue with Step 2: Create the Custom License for details on how to create the license.

   Once the custom license is successfully saved, it is added to the Code Insight database and to the **License** dropdown list. You are returned to the **Edit Version/License** dialog.

6. Click **OK** on the **Edit Version/License** dialog.

   The **Update License Mapping** window is displayed. Its contents explain what happens if you accept this license mapping for all future inventory automatically created for the component version. For more information about the window contents, see About the Update License Mapping Window.

7. Choose the appropriate option:

   - **Yes**—All future inventory items that the system generates for the component version across projects are automatically mapped to the license you selected. For more information about what happens when you set up a user-preferred license, see After You Save the License Mapping.

You are returned to the tab or dialog from which you were creating or editing the inventory item.

- **No**—This component-version-license combination is saved for the specific inventory in the database and made available “behind the scenes” as an instance in the **Lookup Component** window and the **Versions for** <**component**> window. However, any future system-generated inventory for this component version will be mapped to the license that the Code Insight Data Library commonly maps to the version.

  You are returned to the tab or dialog from which you were creating or editing the inventory item.

- **Cancel**—Return to the **Edit Version/License** window.

8. Complete the process of creating or editing the inventory item. (See References to Full Instructions for Creating or Editing Inventory.) Once the inventory item is saved, the custom license is mapped to item.

References to Full Instructions for Creating or Editing Inventory

The following references provide complete instructions on how to create or edit inventory with **Component** or **License** for its **Type** field. You can refer to the appropriate procedure so that you have context when creating or editing a custom license:

In the Analysis Workbench

- Creating Inventory From the Inventory Items List (performed from the <**inventory name**> tab)
- Editing Inventory from the Analysis Workbench (performed from a **New Inventory Item** tab)

On the Project Inventory Tab

- Creating Inventory from the Project Inventory Tab (performed from the **New Inventory** dialog)
- Editing Inventory from the Project Inventory Tab (performed from the **Edit Inventory** dialog)

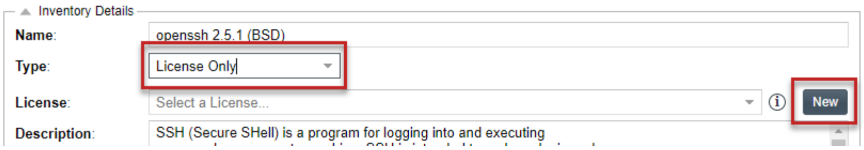
When Creating or Editing a “License Only” Inventory Item

You can create a custom license within the context of creating or editing an inventory item of the type **License Only** in the **Analysis Workbench** or from the **Project Inventory** tab.

Note ▪ *Generally you create a **License Only** inventory item if you know the license for the third-party code or artifact but do not know the component. (You can later edit this inventory item to convert it to one of the other inventory types.) This type of inventory is typically used for groups of files of unknown origin that are governed by a specific license. When you select the license, the inventory name is automatically generated as **Files under <LICENSE NAME> License**.*

Task ***To create a custom license when creating or editing a “License Only” inventory item, do the following:***

1. Start the appropriate procedure for creating or editing an inventory item with **License Only** selected for its **Type** field. See References to Full Instructions for Creating or Editing Inventory for a quick reference to documentation links on how to create or edit inventory.
2. To the right of the **License** dropdown list, click **New** to display the **Create Custom License** window.



Inventory Details

Name: openssh 2.5.1 (BSD)

Type: License Only

License: Select a License...

Description: SSH (Secure SHell) is a program for logging into and executing

3. Continue with Step 2: Create the Custom License for details on how to create the license.

   Once the custom license is successfully saved, it is added to the Code Insight database and to the **License** dropdown list. You are returned to the tab or dialog from which you were creating or editing the inventory item.

4. Complete the process of creating or editing the inventory item. (See References to Full Instructions for Creating or Editing Inventory.) Once the inventory item is saved, the custom license is mapped to item.

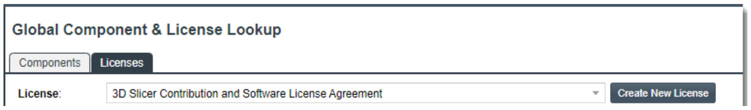
When Searching Licenses with the Global Component & License Lookup Feature

When using the Global Component & License Lookup feature to search licenses in the Code Insight Data Library, you can create a custom license to add to the Data Library should you find the license missing from the library.

Task ***To create a custom license when using Global Component & License Lookup, do the following:***

1. Open the **Global Component & License Lookup** tab, using the instructions in Accessing the Global Component & License Lookup Feature.

2. Select the **Licenses** tab on the **Global Component & License Lookup** tab.



Global Component & License Lookup

License: 3D Slicer Contribution and Software License Agreement

3. Click **Create New License** to open the **Create Custom License** window.

4. Continue with Step 2: Create the Custom License for details on how to create the license.

   Once you successfully save the custom license, it is added to the Code Insight database and automatically selected in the **License** dropdown list on the **Licenses** tab so that its attributes populate the fields on the tab.

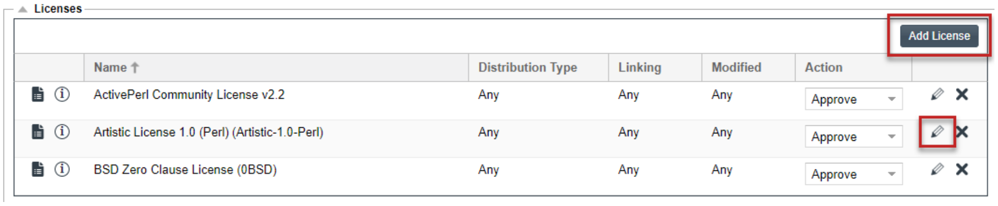
When Creating or Updating a License Policy

Policies are used by Code Insight to automate the inventory review process—that is, automatically mark published inventory items as approved or rejected—without the need for a manual review. A policy's criteria is based on OSS or third-party component versions, license attributes, or security vulnerability score and severities. For complete details, see Managing Policies to Automatically Review Inventory.

As an alternative to selecting an existing license when you create or edit a policy based on a license, you can create a custom license to assign to the policy.

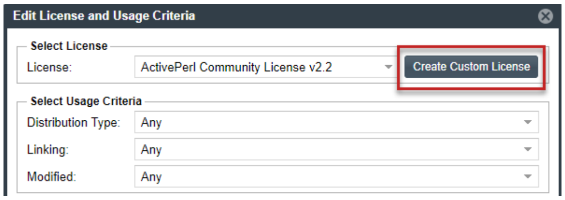
Task ***To create a custom license to assign to a policy as you create or edit the policy, do the following:***

1. Open the policy profile for which you want to create or edit a license policy. See Adding or Editing a Policy Profile for details.
2. Navigate to the **Licenses** section in the policy profile and do either:
   - To edit an existing license policy, click the **Edit** icon ✎ at the end of that policy's row.
   - To create a new license policy, click **Add License**.

	Name ↑	Distribution Type	Linking	Modified	Action	
	ActivePerl Community License v2.2	Any	Any	Any	Approve	✎ ✕
	Artistic License 1.0 (Perl) (Artistic-1.0-Perl)	Any	Any	Any	Approve	✎ ✕
	BSD Zero Clause License (0BSD)	Any	Any	Any	Approve	✎ ✕

   The **Edit** (or **Add**) **License and Usage Criteria** window is displayed.

   Select License
   License: ActivePerl Community License v2.2

   Select Usage Criteria
   Distribution Type: Any
   Linking: Any
   Modified: Any

3. Click **Create Custom License** to open the **Create Custom License** window.
4. Continue with Step 2: Create the Custom License for details on how to create the license.

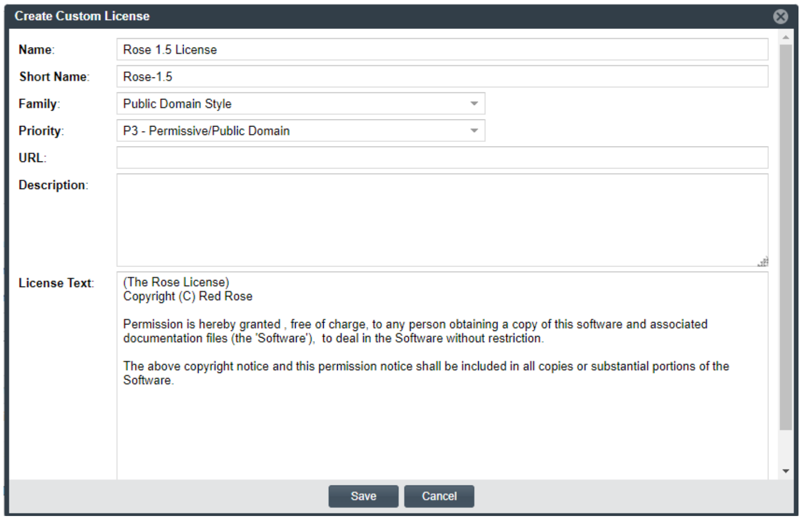
   Once you successfully save the custom license, it is added to the Code Insight database. Additionally, it is automatically added to the **License** dropdown list on the **Edit** (or **Add**) **License and Usage Criteria** window and is in focus for your immediate selection. For more details about fields on this window, see Fields Specific to Maintaining License Policies.
5. Click **Save** on the **Edit** (or **Add**) **License and Usage Criteria** window to return to the **Policy Details** window showing the policy profile. From here you can save the profile or continue to edit it.

Step 2: Create the Custom License

Once you performed one of the procedures in Step 1: Initiate the Creation of a Custom License to open the **Create Custom License** window, use these steps to create the custom license.

Task ***To create the custom license, follow these steps:***

1. From the **Create Custom License** window, provide the license properties. **Name**, **Short Name**, and **License Text** are required fields. For a description of the properties, see Custom License Properties.



Name: Rose 1.5 License
Short Name: Rose-1.5
Family: Public Domain Style
Priority: P3 - Permissive/Public Domain
URL:
Description:
License Text:
(The Rose License)
Copyright (C) Red Rose

Permission is hereby granted , free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

2. Click **Save** to save the license to the Code Insight database.

3. Click **OK** on the confirmation pop-up.

   You are returned to the previous window.

4. Refer to the previous sections in Step 1: Initiate the Creation of a Custom License for a description of the specific “save” behavior related to the context in which the license was created.

Editing a Custom License

You can edit only custom licenses. When a custom license is selected in any **License** dropdown list, the **Edit** button is displayed next to the **New** button, enabling you to update properties for the license. The updates you make to the custom license at the inventory-item level are saved to the license in the Code Insight database and eventually to Code Insight Data Library.

Use the following steps to edit a custom license:

- Step 1: Initiate the Custom-License Editing Process
- Step 2: Edit the Custom License

Step 1: Initiate the Custom-License Editing Process

You have the option to create a custom license within the context of any of the following:

- When Creating or Editing a “Component” Inventory Item
- Creating or Editing a “License Only” Inventory Item

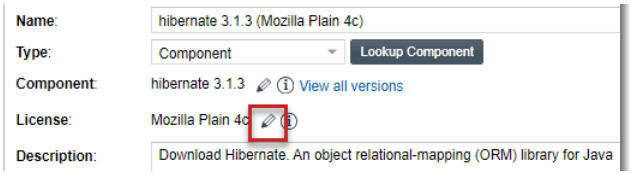
- When Searching Licenses with the Global Component & License Lookup Feature

When Creating or Editing a "Component" Inventory Item

You can edit a custom license within the context of creating or editing an inventory item of the type **Component** in the **Analysis Workbench** or from the **Project Inventory** tab.

Task ***To edit a custom license when creating a "component" inventory item, do the following:***

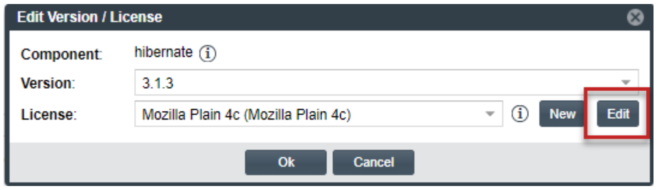
1. Start the appropriate procedure for creating or editing an inventory item with **Component** selected for its **Type** field. See References to Full Instructions for Creating or Editing Inventory for a quick reference to documentation links on how to create or edit inventory.
2. Proceed as follows:
   - **If you editing the properties of an existing inventory item without performing a Lookup Component procedure**—Proceed to the step 3 at any time.
   - **If you are creating an inventory item *or* selecting a new component-version-license instance for an existing item**—Proceed to the next step after your have performed the Look Component procedure and have returned to the tab or dialog from which are you creating or editing the inventory item. (The Lookup Component feature does not allow you create a custom license for a component-version-license instance.)
3. From the tab or dialog from which you are creating or editing the inventory item, click ✎ next to the **License** field.



Name: hibernate 3.1.3 (Mozilla Plain 4c)
Type: Component
Component: hibernate 3.1.3
License: Mozilla Plain 4c
Description: Download Hibernate. An object relational-mapping (ORM) library for Java

   The **Edit Version/License** dialog is displayed.

4. If necessary, from the **License** dropdown, select the custom license that you are associating with the inventory item. (The **Edit** button is available only when a custom license selected.)



Edit Version / License

Component: hibernate
Version: 3.1.3
License: Mozilla Plain 4c (Mozilla Plain 4c)

5. Click **Edit** to display the **Edit Custom License** window.
6. Continue with Step 2: Edit the Custom License for details on how to edit the license.

   Once the edits to the custom license are successfully saved, they added to the Code Insight database and changes to the license name are reflected in the **License** dropdown list. You are returned to the **Edit Version/ License** dialog.

7. Click **OK** to close the **Edit Version/License** dialog.

   If you have not selected a new license from the **License** dropdown, proceed to step 9.

   If you have selected a new license from the **License** dropdown, the **Update License Mapping** window is displayed. Its contents explain what happens if you accept this license mapping for all future inventory automatically created for the component version. For more information about the window contents, see About the Update License Mapping Window.

8. Choose the appropriate option:

   - **Yes**—All future inventory items that the system generates for the component version across projects are automatically mapped to the license you selected. For more information about what happens when you set up a user-preferred license, see After You Save the License Mapping.

     You are returned to the tab or dialog from which you were creating or editing the inventory item.

   - **No**—This component-version-license combination is saved for the specific inventory in the database and made available “behind the scenes” as an instance in the **Lookup Component** window and the **Versions for** <**component**> window. However, any future system-generated inventory for this component version will be mapped to the license that the Code Insight Data Library commonly maps to the version.

     You are returned to the tab or dialog from which you were creating or editing the inventory item.

   - **Cancel**—Return to the **Edit Version/License** window.

9. Complete the process of creating or editing the inventory item. (See References to Full Instructions for Creating or Editing Inventory.) Once the inventory item is saved, the custom license is mapped to item.

   Note *▪ If this inventory item is currently published, the save can trigger an automatic review of the inventory item. See Automatic Review Triggered When Saving Existing Inventory in the Analysis Workbench or Automatic Review When Saving Existing Inventory on the Project Inventory Tab for details.*

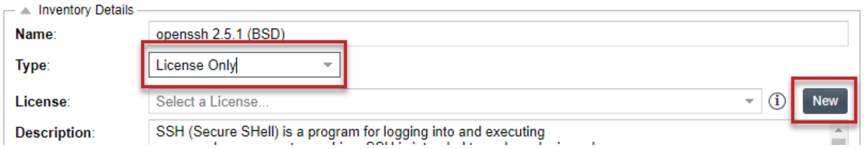
Creating or Editing a “License Only” Inventory Item

You can edit a custom license associated with an inventory item of the type **License Only** within the context of creating or editing the inventory item from the **Project Inventory** tab or in the **Analysis Workbench**.

Note *▪ Generally you create a* ***License Only*** *inventory item if you know the license for the third-party code or artifact but do not know the component. (You can later edit this inventory item to convert it to one of the other inventory types.) This type of inventory is typically used for groups of files of unknown origin that are governed by a specific license. When you select the license, the inventory name is automatically generated as* ***Files under <LICENSE NAME> License****.*

Task ***To edit a custom license when creating or editing a "License Only" inventory item, do the following:***

1. Start the appropriate procedure for creating or editing an inventory item with **License Only** selected for its **Type** field. See References to Full Instructions for Creating or Editing Inventory for a quick reference to documentation links on how to create or edit inventory.
2. To the right of the **License** dropdown list, click **New** to display the **Create Custom License** window.


Inventory Details
Name: openssh 2.5.1 (BSD)
Type: License Only
License: Select a License...
Description: SSH (Secure SHell) is a program for logging into and executing

3. Continue with Step 2: Create the Custom License for details on how to create the license.

   Once the custom license is successfully saved, it is added to the Code Insight database and to the **License** dropdown list. You are returned to the tab or dialog from which you were creating or editing the inventory item.
4. Complete the process of creating or editing the inventory item. (See References to Full Instructions for Creating or Editing Inventory.) Once the inventory item is saved, the custom license is mapped to item.

   Note ▪ *If this inventory item is currently published, the save can trigger an automatic review of the inventory item. See Automatic Review Triggered When Saving Existing Inventory in the Analysis Workbench or Automatic Review When Saving Existing Inventory on the Project Inventory Tab for details.*

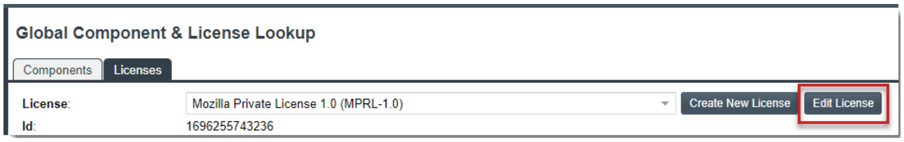
When Searching Licenses with the Global Component & License Lookup Feature

When using the Global Component & License Lookup feature to search licenses in the Code Insight Data Library, you can edit a custom license that was previously added to the Data Library.

Task ***To edit a custom license when using Global Component & License Lookup, do the following:***

1. Using the instructions in Accessing the Global Component & License Lookup Feature, open the **Global Component & License Lookup** tab.
2. Select the **Licenses** tab.
3. From the **License** dropdown, select the custom license you want to edit.

   The **Edit** button is automatically displayed next to the selection. (The **Edit** button is available only when a custom license selected.)


Global Component & License Lookup
License: Mozilla Private License 1.0 (MPRL-1.0)
Id: 1696255743236

4. Continue with Step 2: Edit the Custom License for details on how to create the license.

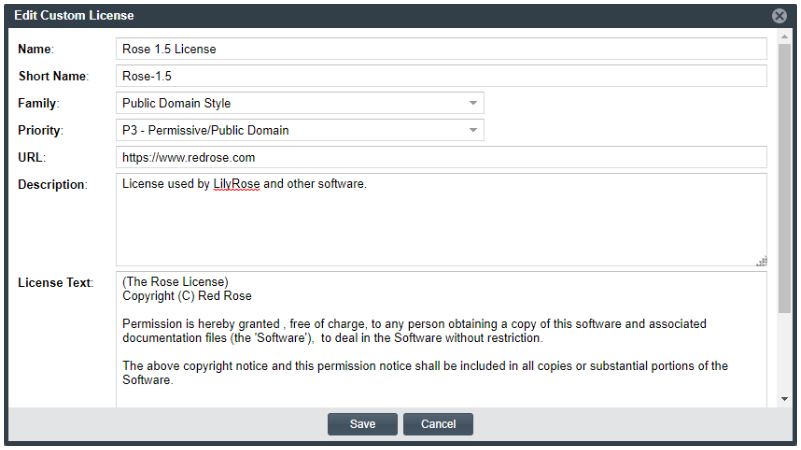
Once you successfully saved the custom license, it is added to the Code Insight database and automatically selected in the **License** dropdown list on the **Licenses** tab so that its attributes populate the fields on the tab.

Step 2: Edit the Custom License

Once you performed one of the procedures in Step 1: Initiate the Custom-License Editing Process to open the **Create Custom License** window, use these steps to create the custom license.

Task ***To edit a custom license, follow these steps:***

1. From the **Edit Custom License** window, update the license properties as needed. For property descriptions, see Custom License Properties.



Edit Custom License

Name: Rose 1.5 License
Short Name: Rose-1.5
Family: Public Domain Style
Priority: P3 - Permissive/Public Domain
URL: https://www.redrose.com
Description: License used by LilyRose and other software.
License Text: (The Rose License)
Copyright (C) Red Rose

Permission is hereby granted , free of charge, to any person obtaining a copy of this software and associated documentation files (the 'Software'), to deal in the Software without restriction.

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

2. Click **Save** to save the license updates to the Code Insight database.
3. Click **OK** on the confirmation pop-up.

   You are returned to the previous window.
4. Refer to the previous sections in Step 1: Initiate the Custom-License Editing Process for a description of the specific “save” behavior related to the context in which the license was created.

Custom License Properties

The following describes the fields on the **Create** (or **Edit**) **Custom License** window used to define a custom license.

Table 2-15 ▪ Fields to Define a Custom License

Component Field	Description
Name	(Required) The full name of the license (for example, `The Rose License 1.5`).
Short Name	(Required) A unique shorthand representation of the license. This is usually the license SPDX short identifier (for example, `Rose-1.5`).

Table 2-15 ▪ Fields to Define a Custom License (cont.)

Component Field	Description
Family	A license category that spans multiple license instances (for example, MIT, Public Domain, BSD-3 Clause, and others). The family designation is helpful to a legal reviewer to understand the “type” of license prior to investing the time to analyze the complete license text.
URL	(Required) The URL used to access the license on the Internet. This value should start with `http://` or `https://`.
Priority	The level of importance in investigating this license in terms of its possible business impact on your organization. The higher the level, the greater need to investigate the license: ● **P1**—Viral, strong copyleft license (requires immediate attention). ● **P2**—Weak copyleft or commercial or uncommon license (requires legal review). ● **P3**—Permissive or public domain license (generally allowed because of its minimal business impact). This is the default if no priority is specified. For details about each priority level, see Auditing Scan Results in the Analysis Workbench.
Description	Meaningful information about the license for your reference.
License Text	(Required) The complete text of the license. Be sure to encode any HTML characters.

Managing Custom Detection Rules

During a Code Insight project scan, the Automated Analysis component of the Scan Server uses a set of internal detection rules, stored in the Code Insight Data Library, to automatically generate inventory items.

During your manual analysis, you might find that one or more codebase files that indicate the presence of specific third-party or OSS component are not being associated with any inventory in your projects. Either the file is missing from an existing inventory item based on the component, or the inventory item is missing from your projects altogether.

Code Insight enables you to create a custom detection rule based on your findings to ensure that, when the rule’s file criteria for detecting a specific component match files in the codebase, the appropriate inventory item is generated during a scan. These rules are saved to the Code Insight Data Library for global use by Automated Analysis during subsequent scans and rescans.

The following sections provide more information about managing custom detection rules:

- Creating a Custom Detection Rule
- Viewing All Current Custom Detection Rules

- Editing a Custom Detection Rule
- Deleting a Custom Detection Rule
- Rule-Processing Considerations

Creating a Custom Detection Rule

Code Insight provides two methods for creating custom detection rules—either from scratch or within the context of an inventory item that you had to manually create or update because no current rule automatically identified the component by its associated files. Refer to the following sections:

- Creating a Custom Detection Rule Within Context of an Inventory Item
- Creating a Custom Detection Rule from Scratch

Creating a Custom Detection Rule Within Context of an Inventory Item

This method for creating a custom detection rule requires access to the **Analysis Workbench** for the project with the missing inventory item or the inventory item with a missing associated file.

During codebase analysis in the **Analysis Workbench** for a project, you might find that one or more codebase files that are evidence of a specific third-party or OSS component are not being associated with inventory in your project. You must manually fix the situation—either by updating the existing inventory item to include the associated files or by creating the missing inventory item associated with the files.

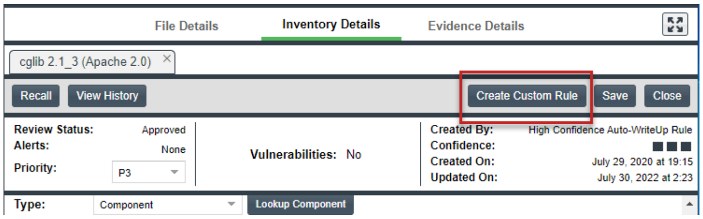
Code Insight enables you to create a custom detection rule based on the file criteria of the inventory item that you had to create or update. Because you are creating this rule within the context of an existing inventory item, most of the fields that define the rule are pre-populated with details from the item, including the MD5 value for each file currently associated with the inventory item.

Refer to the following procedure for instructions.

***Note** ▪ To create the rule based on an existing inventory item, the inventory item **Type** must be **Component**.*

Task ***To create a custom detection rule within the context of an inventory item that you manually created or updated with the associated files, do the following:***

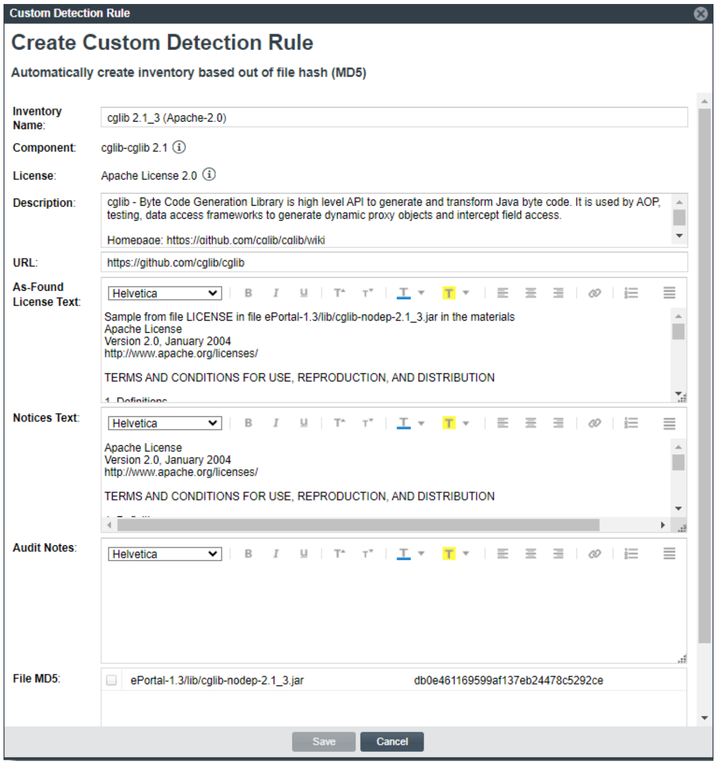
1. In the **Analysis Workbench** for the desired project (see Opening the Analysis Workbench), navigate to the **Inventory Items** pane and select the manually updated or created inventory item from which you want to create the custom detection rule. The **Inventory Details** tab for inventory item is opened.



Review Status: Approved
Alerts: None
Priority: P3
Vulnerabilities: No
Created By: High Confidence Auto-WriteUp Rule
Confidence: ■■■
Created On: July 29, 2020 at 19:15
Updated On: July 30, 2022 at 2:23
Type: Component

2. Click the **Create Custom Rule** button to open the **Custom Detection Rule** dialog. For a description of the fields in this dialog, refer to Create Custom Detection Rule Dialog.

   Note that the fields are pre-populated with information already defined for the inventory item on which you are basing the rule.



Create Custom Detection Rule

Automatically create inventory based out of file hash (MD5)

Inventory Name: cglib 2.1_3 (Apache-2.0)

Component: cglib-cglib 2.1

License: Apache License 2.0

Description: cglib - Byte Code Generation Library is high level API to generate and transform Java byte code. It is used by AOP, testing, data access frameworks to generate dynamic proxy objects and intercept field access.

Homepage: https://github.com/cglib/cglib/wiki

URL: https://github.com/cglib/cglib

As-Found License Text: Sample from file LICENSE in file ePortal-1.3/lib/cglib-nodep-2.1_3.jar in the materials
Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. Definitions

Notices Text: Apache License
Version 2.0, January 2004
http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

Audit Notes:

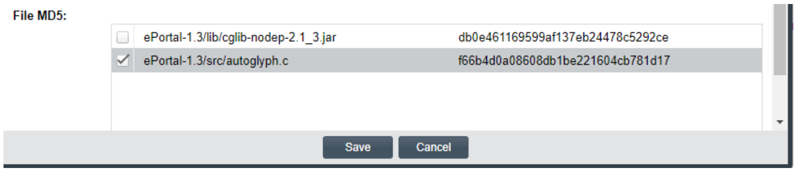
File MD5: □ ePortal-1.3/lib/cglib-nodep-2.1_3.jar db0e461169599af137eb24478c5292ce

3. Edit the inventory-related fields as needed (with the exception of the **Component** and **License** fields, which are not editable). These fields are used to define inventory that is automatically created or updated by this rule during future scans.

   Note that you can use the **Inventory Name** field to overwrite the default name *component version (license)* otherwise assigned to inventory items created by this rule.

4. Scroll down to the **File MD5** pane, which is pre-populated with the list of codebase files associated with the inventory item you created. The MD5 value for each file is provided.

5. Select one or more files to add to the rule. The MD5 value for each selected file becomes a criterion used to detect the component in the codebase.



File MD5:

□	ePortal-1.3/lib/cglib-nodep-2.1_3.jar	db0e461169599af137eb24478c5292ce
☑	ePortal-1.3/src/autoglyph.c	f66b4d0a08608db1be221604cb781d17

6. Click **Save** and then click **Yes** to confirm that you want to proceed with creating the rule. It is then added to the Code Insight Data Library, where it will be available for global use.

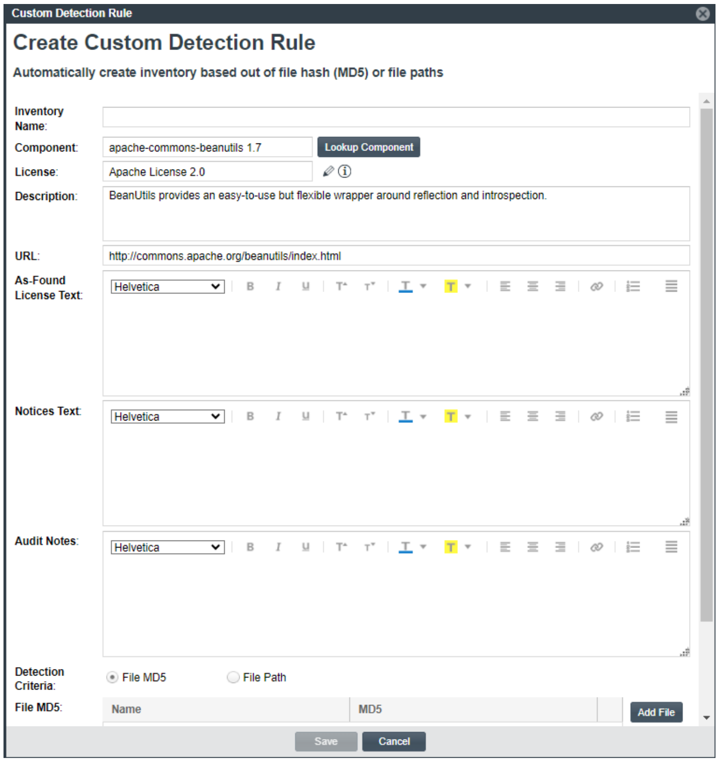
Creating a Custom Detection Rule from Scratch

When you want to create a custom detection rule but do not have access to the **Analysis Workbench**, you must create the rule from scratch, providing all the necessary details that define the rule. These details include, along with other information, the set of file criteria used to detect the third-party or OSS component and create the associated inventory. The set of criteria can be based on either the file path or MD5 value of the files.

Refer to the following procedure for creating the custom detection rule from scratch.

Task ***To create a custom detection rule from scratch, do the following:***

1. Open the **Custom Detection Rules** tab, using the procedure in Viewing All Current Custom Detection Rules.
2. Click **Create Custom Rule** to open the **Custom Detection Rule** dialog.
3. Use **Lookup Component** to select or create a component for the rule. Once the component instance is selected, its information is populated in the dialog.


Custom Detection Rule

Create Custom Detection Rule

Automatically create inventory based out of file hash (MD5) or file paths

Inventory Name:

Component: apache-commons-beanutils 1.7

License: Apache License 2.0

Description: BeanUtils provides an easy-to-use but flexible wrapper around reflection and introspection.

URL: http://commons.apache.org/beanutils/index.html

As-Found License Text: Helvetica

Notices Text: Helvetica

Audit Notes: Helvetica

Detection Criteria: ● File MD5 ○ File Path

File MD5:

Name	MD5

4. Update the populated information if necessary and provide values for the blank **As-Found License Text**, **Notices Text**, and **Audit Notes** fields as needed. (For a description of each field, refer to Create Custom Detection Rule Dialog.) The information provided in these fields defines the inventory created or updated by this rule in future scans.

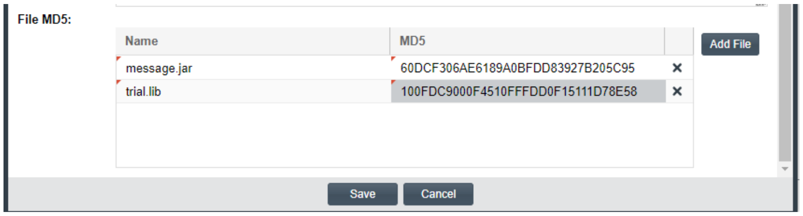
   Note that you can use the **Inventory Name** field to overwrite the default name *component version (license)* otherwise assigned to inventory items created by this rule.

5. In the **Detection Criteria** field, select the type of file criteria you are specifying to detect the presence of the component—file MD5 values or file paths. (The set of files can use only one criteria type. The default type is **File MD5**.)

***Note** ▪ If you attempt to set up detection criteria for both types (**File MD5** or **File Path**), keep in mind that you lose the criteria for the type that is currently not selected for **Detection Criteria** when you save the rule. A custom detection rule allows only a single set of criteria to exist at any one time.*

- When you select **File MD5**, the **File MD5** grid is displayed.

  For each file criterion you want to add for detecting the component, click the **Add File** button and provide the file's name and MD5 value in the new row in the grid.

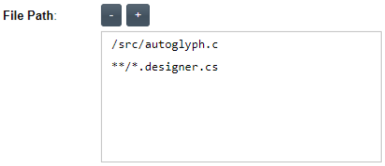
  

Name	MD5	
message.jar	60DCF306AE6189A0BFDD83927B205C95	✕
trial.lib	100FDC9000F4510FFFDD0F15111D78E58	✕

  To remove a file from the grid, click ✕ to the right of its row.

- When you select **File Path**, the **File Path** text box is displayed.

  For each file criterion you want to add for detecting the component, click the Add icon + and enter the file's path. You can provide the file's absolute or relative path or enter a path pattern.

  ```
  /src/autoglyph.c
  **/*.designer.cs
  ```

  A *path pattern* consists of the asterisk symbol * within the path, denoting any number of directories or files. For example, the path pattern used in the screenshot above indicates that any file whose file name ends with `.designer` and has a `.cs` extension will be considered detection criteria for the rule.

  To remove a path, click the Remove icon -.

6. Click **Save** and then click **Yes** to confirm that you want to proceed with creating the rule and adding it to the Code Insight Data Library, where it will be available for global use.

Viewing All Current Custom Detection Rules

Use the following the procedure to access the **Custom Detection Rules** tab, from which you can view all currently defined custom detection rules and act on them as needed.

Task ***To view all current custom detection rules created for your Code Insight system, do the following:***

1. Click the following icon in the upper right corner of the Code Insight web page to open the Code Insight main menu:

   ☰

2. Select **DATA LIBRARY** from the menu to open the **Data Library** page.
3. Select the **Custom Detection Rules** tab, showing the list of current custom detection rules. (For a description of the information shown each rule, see Custom Detection Rules Tab.)

   From this tab, you can do the following:

   - View the component information (name, version, license, and forge URL) on which a given rule is based.
   - Create a custom detection rule from scratch (see Creating a Custom Detection Rule from Scratch).
   - Edit a custom detection rule or remove it from the Code Insight system.

Editing a Custom Detection Rule

Use the following procedure to edit a specific custom detection rule. The changed rule is applied to all future scans or rescans on projects in the Code Insight.

Task ***To edit an existing custom detection rule, do the following:***

1. Open the **Custom Detection Rules** tab, following the procedure in Viewing All Current Custom Detection Rules.
2. In the **Actions** column for the component whose detection criteria you want to update, click the ✎ icon. The **Edit Custom Rule** dialog opens.
3. Edit fields as needed. See Edit Custom Rule Dialog for a description of each field.

   Note that you can use the **Inventory Name** field to overwrite the default name *component version (license)* otherwise assigned to inventory items created by this rule.

4. Manage the detection criteria for component. The type of criteria available depends on the **Detection Criteria** value assigned to the rule—**File MD5**or **File Path**. At least one criterion for the rule’s specified criteria type is required.

   - When the criteria type is **File MD5**, the files associated with the component are identified by their MD5 value and listed in the **File MD5** grid. To edit this criteria in the grid, do any of the following:
     - Add a criterion by clicking the **Add File** button and providing the file’s name and MD5 value.
     - Edit a criterion by clicking within its **Name** or **MD5** field and making textual changes.
     - Remove a file from the grid by clicking the ✖ icon next to the criterion.

- When the criteria type is **File Path**, the files associated with the component are identified by their file paths and listed in the **File Path** text box. To edit this criteria in the text box, do any of the following:
    - Add a criterion by clicking the Add icon + and entering file’s path. (You can provide the file’s absolute path or relative path, or you can provide a path pattern.)
    - Edit a criterion by clicking within the path field and make textual changes.
    - Remove a file from the text box by clicking the Remove icon -.

Note ▪ *If you want to switch the current **Detection Criteria** type from **File MD5** to **File Path** or vice versa, know that once you enter the new set of criteria and save the rule, the criteria for the type currently not selected for **Detection Criteria** is automatically deleted. A custom detection rule allows only a single set of criteria to exist at any one time.*

5. Click **Save** to save the detection rule changes to the Code Insight Data Library. You will be asked for confirmation to proceed with the updates.

Deleting a Custom Detection Rule

Use the following procedure to remove a specific custom detection rule from the Code Insight Data Library. The rule will no longer be applied to any future scan in your Code Insight system.

Task ***To remove a custom detection rule from the Code Insight Data Library, do the following:***

1. Open the **Custom Detection Rules** tab, following the procedure in Viewing All Current Custom Detection Rules.
2. In the **Actions** column for the entry you want to remove, click the ✖ icon. You are asked to confirm the deletion.

Rule-Processing Considerations

As you manage custom detection rules, consider how the rules are processed under certain circumstances:

- If the custom detection rule is associated with more than one file, the scan uses OR logic when processing the files against the target codebase. Consequently, only one file match between codebase and the rule is required to automatically create an inventory item.
- If two rules are created with identical details and codebase files, a single inventory item is generated during a scan when both rules are applied.
- If two rules are created with the same **Component** and **License** details but have different **Inventory Name** values, the rule created more recently is applied.

- If two rules are created using the same **Inventory Name**, **Component**, and **License** details and the same codebase files, but have a different **Description**, **URL**, **Audit Notes**, **As-Found License Text**, or **Notices Text** value, a single inventory item is generated during a scan when both rules are applied. In the inventory item, values that differ between the rules for a given field are separated (shown on separate lines or with a separator) within the field.
- If two rules with are created with the same codebase files but use a different **Component** value, two inventory items are generated during the scan.
- If a rule has a **File MD5** criterion for a given codebase file and another rule has a **File Path** criterion for the same codebase file but is set up for a different component, only the rule with the **File MD5** file criterion will be processed. (In such cases, a rule with **File MD5** detection criteria is given precedence over a rule with **File Path** detection criteria.)

Finalizing the Notices Text for the Notices Report

You can use the **Notices Text** field on the **Notices Text** tab for an inventory item to provide the exact license content to include for that item in the Notices report. For example, you can edit any license text previously saved to this field or add your own license text, such as license information for rules that you developed during your manual research on the inventory item. You can also copy the **As-Found License Text** content, if it exists, to the **Notices Text** field and modify it as needed. (The **As-Found License Text** content, which is not editable, consists of the license text or license references found in the scanned codebase). As a third option, you can pull a copy of the current text for the license from the Revenera Data Library into the **Notices Text** field and modify it as needed.

***Note** ▪ Code Insight also provides the option to update the **Notices Text** field automatically across all inventory in the project by retrieving the appropriate notices text from the Revenera Data Library. For more information, see Updating Third-Party Notices Across Inventory for a Project.*

If the **Notices Text** field item field contains information when the Notices report is run, the content of this field alone is pulled into the report. If the **Notices Text** field is empty, the content of the **As-Found License Text** field is used in the report. If both fields are empty, the report uses the license content from Revenera Data Library (see License Details from the Code Insight Data Library).

For more information about these fields, see Reporting of Detected License Text through the As-Found Text Inventory Field and Notices Text Field.

***Note** ▪ If the **As-Found License Text** field contains content populated by the Scan Server, best practice is to leave the **Notices Text** field empty (as long as custom information or edits are not required) so that the report is forced to use the license information found in the **As-Found License Text** field.*

Refer to the next section for instructions on how to manage the license content for an inventory item in the Notices report.

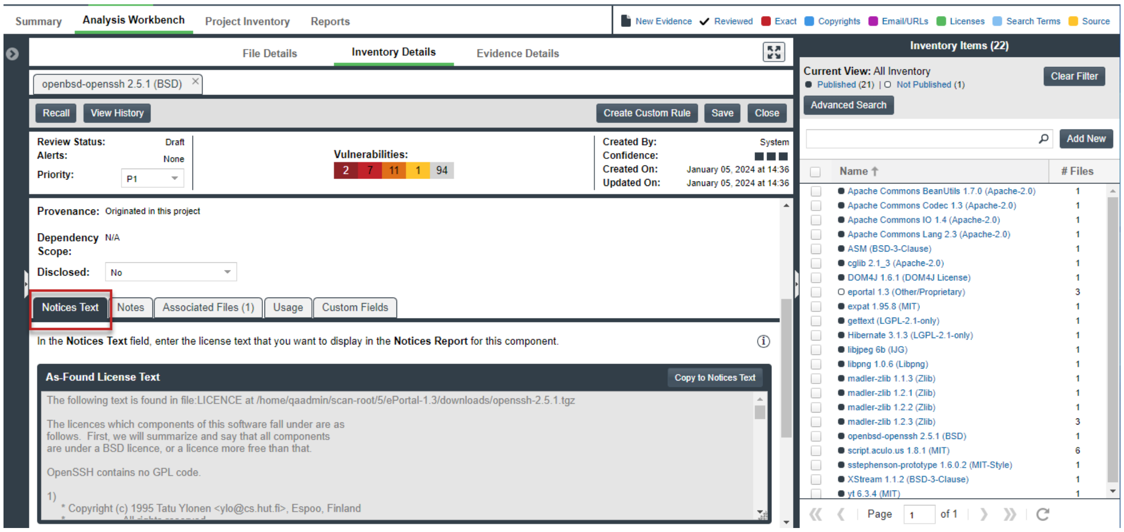
Finalizing License Content

Use the following procedure to finalize the license content that is reported for a given inventory item in the Notices report.

Task ***To finalize license content that is reported for an inventory item in the Notices report, do the following:***

1. Navigate to one of these locations:

   In the **Analysis Workbench**, on the **Inventory Details** tab for a specific inventory item, open the **Notices Text** tab. If necessary, see Opening the Analysis Workbench.



Review Status: Draft
Alerts: None
Priority: P1
Vulnerabilities: 2 7 11 1 94
Created By: System
Confidence:
Created On: January 05, 2024 at 14:36
Updated On: January 05, 2024 at 14:36
Provenance: Originated in this project
Dependency Scope: N/A
Disclosed: No

In the **Notices Text** field, enter the license text that you want to display in the **Notices Report** for this component.

As-Found License Text

The following text is found in file:LICENCE at /home/qaadmin/scan-root/5/ePortal-1.3/downloads/openssh-2.5.1.tgz

The licences which components of this software fall under are as follows. First, we will summarize and say that all components are under a BSD licence, or a licence more free than that.

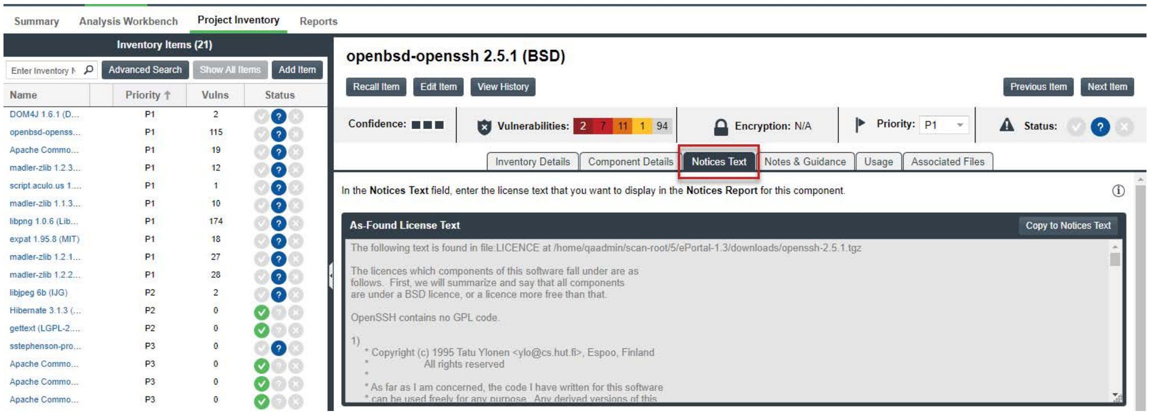
OpenSSH contains no GPL code.

1)
 * Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland

Current View: All Inventory
● Published (21) | ○ Not Published (1)

Name	# Files
Apache Commons BeanUtils 1.7.0 (Apache-2.0)	1
Apache Commons Codec 1.3 (Apache-2.0)	1
Apache Commons IO 1.4 (Apache-2.0)	1
Apache Commons Lang 2.3 (Apache-2.0)	1
ASM (BSD-3-Clause)	1
cglib 2.1_3 (Apache-2.0)	1
DOM4J 1.6.1 (DOM4J License)	1
eportal 1.3 (Other/Proprietary)	3
expat 1.95.8 (MIT)	1
gettext (LGPL-2.1-only)	1
Hibernate 3.1.3 (LGPL-2.1-only)	1
libjpeg 6b (IJG)	1
libpng 1.0.6 (Libpng)	1
madler-zlib 1.1.3 (Zlib)	1
madler-zlib 1.2.1 (Zlib)	1
madler-zlib 1.2.2 (Zlib)	1
madler-zlib 1.2.3 (Zlib)	3
openbsd-openssh 2.5.1 (BSD)	1
script.aculo.us 1.8.1 (MIT)	6
sstephenson-prototype 1.6.0.2 (MIT-Style)	1
XStream 1.1.2 (BSD-3-Clause)	1
yt 6.3.4 (MIT)	1

   From **Project Inventory**, select an inventory item and open the **Notices Text** tab. (If necessary, see Displaying Project Inventory.)



Name	Priority	Vulns	Status
DOM4J 1.6.1 (D...	P1	2	
openbsd-openss...	P1	115	
Apache Commo...	P1	19	
madler-zlib 1.2.3...	P1	12	
script.aculo.us 1....	P1	1	
madler-zlib 1.1.3...	P1	10	
libpng 1.0.6 (Lib...	P1	174	
expat 1.95.8 (MIT)	P1	18	
madler-zlib 1.2.1...	P1	27	
madler-zlib 1.2.2...	P1	28	
libjpeg 6b (IJG)	P2	2	
Hibernate 3.1.3 (...	P2	0	✓
gettext (LGPL-2....	P2	0	✓
sstephenson-pro...	P3	0	
Apache Commo...	P3	0	✓
Apache Commo...	P3	0	✓
Apache Commo...	P3	0	✓

openbsd-openssh 2.5.1 (BSD)

Confidence: Vulnerabilities: 2 7 11 1 94 Encryption: N/A Priority: P1 Status:

In the **Notices Text** field, enter the license text that you want to display in the **Notices Report** for this component.

As-Found License Text

The following text is found in file:LICENCE at /home/qaadmin/scan-root/5/ePortal-1.3/downloads/openssh-2.5.1.tgz

The licences which components of this software fall under are as follows. First, we will summarize and say that all components are under a BSD licence, or a licence more free than that.

OpenSSH contains no GPL code.

1)
 * Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland
 *   All rights reserved
 *
 * As far as I am concerned, the code I have written for this software
 * can be used freely for any purpose. Any derived versions of this

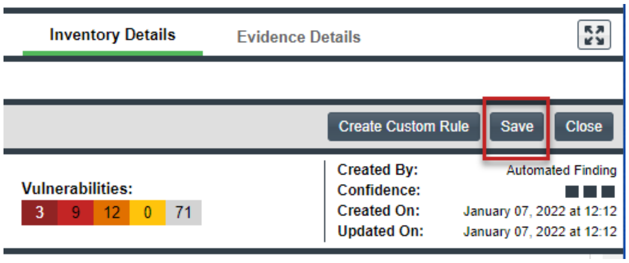
2. Do one of the following:

   - In the **Notices Text** field, add new license content for the inventory item or modify existing content. The text and its format should look exactly as you want it to appear in the Notices report.

- If **As-Found License Text** content exists and you want to use it in the Notices report, copy the content to the **Notices Text** field and modify it as needed. You can choose to append this content to existing any existing content in the **Notices Text** field or simply overwrite the existing content. See Using As-Found License Text in the Notices Report for details.
- Copy the current content for the license from the Revenera Data Library into the **Notices Text**. (This operation overwrites any existing text in the field.) You can modify this content as needed. See Using License Text from the Revenera Data Library in the Notices Report for details.
- Do nothing. If the **Notices Text** field item field contains information when the Notices report is run, the content of this field alone is pulled into the report for the inventory item. If the **Notices Text** field is empty, the content of the **As-Found License Text** field is used in the report. If both fields are empty, the report uses the license content from Revenera Data Library.

3. If you have updated the **Notices Text** field, save the changes to the current inventory item:

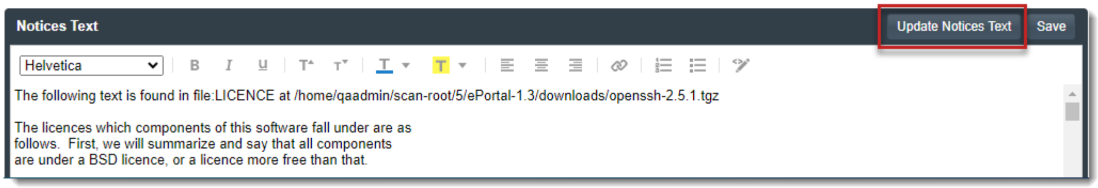
   - If you are in the **Analysis Workbench**, click the **Save** button at the top of the **Inventory Details** tab. (Alternatively, click **Close** to shut down the tab for the current inventory item. You are prompted to save the inventory changes before the tab closes.)



Vulnerabilities: 3 9 12 0 71

Created By: Automated Finding
Confidence: ■■■
Created On: January 07, 2022 at 12:12
Updated On: January 07, 2022 at 12:12

   - If you are in **Project Inventory**, click the **Save** button at the top of the **Notices Text** field.



The following text is found in file:LICENCE at /home/qaadmin/scan-root/5/ePortal-1.3/downloads/openssh-2.5.1.tgz

The licences which components of this software fall under are as
follows. First, we will summarize and say that all components
are under a BSD licence, or a licence more free than that.

When the Notices report is run, the content from the **Notices Text** pane (or the default content) is used as the "notices" information for the inventory item in the report.

Using As-Found License Text in the Notices Report

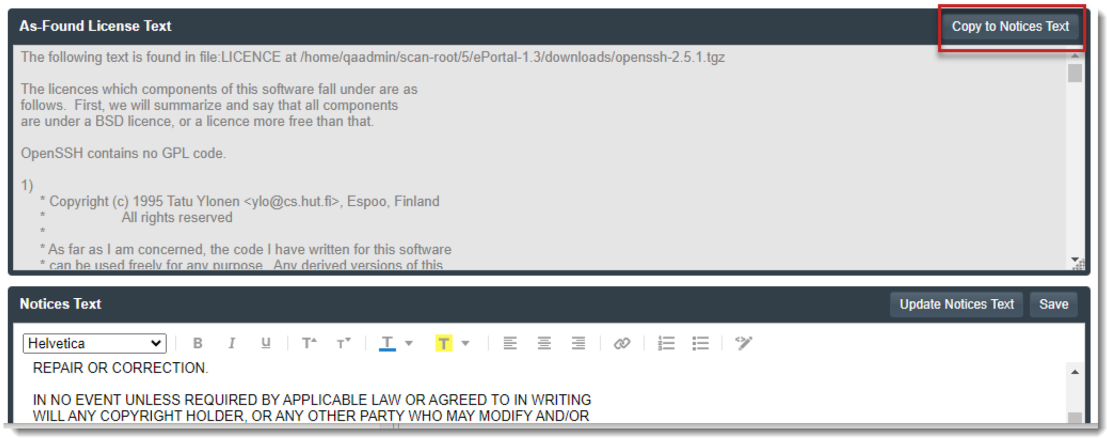
If **As-Found License Text** content exists and you want to use a modified version of it in the Notices report or you want to use this content to replace or add to text already in the **Notices Text** field, you must copy the **As-Found License Text** content to the **Notices Text** field.

Note *▪ If no content exists in the* ***Notices Text*** *field and you want the Notices report to use the* ***As-Found License Text*** *content without any modification, you do not need to perform this procedure. By default, if the* ***Notices Text*** *field is empty and the* ***As-Found License Text*** *field is populated, the Notices report automatically pulls in the license content from the* ***As-Found License Text*** *field for the inventory item.*

Task ***To copy the As-Found License Text content to the Notices Text field, do the following:***

1. In the **Notices Text** tab for the inventory item, click the **Copy to Notices Text** button in the top right corner of the **As-Found License Text** field.

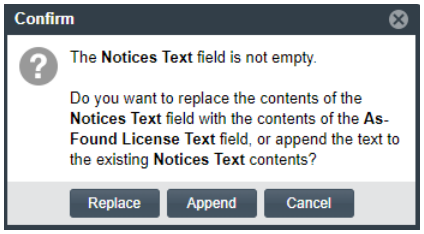
    If the **Notices Text** field is empty, the **As-Found License Text** content is copied to the **Notices Text** field, with the formatting preserved.

    ```
    The following text is found in file:LICENCE at /home/qaadmin/scan-root/5/ePortal-1.3/downloads/openssh-2.5.1.tgz

    The licences which components of this software fall under are as
    follows.  First, we will summarize and say that all components
    are under a BSD licence, or a licence more free than that.

    OpenSSH contains no GPL code.

    1)
       * Copyright (c) 1995 Tatu Ylonen <ylo@cs.hut.fi>, Espoo, Finland
       *                    All rights reserved
       *
       * As far as I am concerned, the code I have written for this software
       * can be used freely for any purpose.  Any derived versions of this
    ```

    ```
    REPAIR OR CORRECTION.

    IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING
    WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR
    ```

    If the **Notices Text** field is *not* empty, you are given the option either to append the **As-Found License Text** content to the existing **Notices Text** content or to replace all the existing **Notices Text** content with the **As-Found License Text** content.

    The **Notices Text** field is not empty.

    Do you want to replace the contents of the **Notices Text** field with the contents of the **As-Found License Text** field, or append the text to the existing **Notices Text** contents?

    If you select **Append**, the appended content is added to the **Notices Text** field, starting on a new line after the existing content. If you select **Replace**, the existing **Notices Text** content is replaced.

2. Modify and format the **Notices Text** content if needed.

3. Proceed with step 3 in the previous section to complete the preparation of the license text for the Notices report.

Using License Text from the Revenera Data Library in the Notices Report

If either the **As-Found License Text** or **Notices Text** field are populated and you want the Notices report to use the license content from the Revenera Data Library, use this procedure to copy the license content from the library to the **Notices Text** field. Any existing text in the **Notices Text** field is overwritten. You can modify the retrieved license content as needed.

Note that the **Update Notices Text** button required to perform this operation is enabled only under the following conditions:

- You have Analyst or Reviewer permissions on the project.
- The inventory item is defined as a “Component” type. (The content cannot be retrieved for a “Work in Progress” or License Only” inventory item.)

Additionally, the operation will fail if the component version for the inventory item is custom or if a version has not been selected for the item.

Note ▪ *If both the **As-Found License Text** and **Notices Text** fields are empty and you want to use the license content from the Revenera Data Library without any modifications, you do not need to perform this procedure. By default, if both fields are empty, the Notices report automatically pulls in the license content from the library for the inventory item.*

Task ***To copy the license content from the Revenera Data Library to the Notices Text field, do the following:***

1. Click the **Update Notices Text** button in the top right corner of the **Notices Text** field.

   If the operation completes successfully, the license content from the Revenera Data Library is copied to the **Notices Text** field. (The library license information overwrites any existing information in the field.) Skip to step 3.

   If operation encounters any special conditions, proceed to step 2.

2. Address special conditions accordingly:

   - If the inventory item does not have a component version associated with it or the component version is custom, an error occurs and the operation is ended. Before retrying this procedure, you must select a non-custom version for the inventory item.
   - If the **Notices Text** field already contains content, you asked whether to overwrite the content. If you select **No**, the operation is ended. If you select **Yes**, the operation proceeds.
   - If no content for the license is found in the Revenera Data Library, the operation is ended with the message “Notices Text not found in our collection. Manual review required.”

3. If license content has been successfully copied from the Data Library to the **Notices Text** field, modify and format the content as needed.

4. Proceed with step 3 in the previous section to complete the preparation of the license text for the Notices report.

Generating Reports for a Project

The **Reports** tab enables users to generate Code Insight reports that present different views of the data collected for a given project. The following sections describe the different reports available for any project and how to generate these reports:

- About the Standard Reports for Projects

- About Custom Reports for Projects
- Generating a Report for a Project

About the Standard Reports for Projects

The following describes the reports that come standard with Code Insight and are available for any project:

- Project Report
- Audit Report
- Notices Report

Project Report

The Project report summarizes the inventory, security vulnerabilities, remaining scan evidence, and review and remediation tasks for a selected project. It produces output in JSON and Excel format. This report is useful in understanding the existing project's legal and security risks based on identified inventory items, as well as the additional potential risk based on the file-based scan results known as third-party indicators.

Note the following:

- The metrics and statistics in this report are based on the results of the most recent server scan and remote scan(s) associated with the project.
- Currently, Code Insight is able to report license evidence found in remote files scanned by a scan agent. This evidence is reflected (along with evidence detected by the Scan Server) in the charts and data in the following locations:
    - **Additional Evidence** section of the **Summary** sheet
    - **Files with License** sheet (with an **Alias** column to help you determine which files are remote)
    - **All Scanned Files** sheet
- When the report lists codebase files, an alias and file path can be included with each file name in the format <alias>:<filePath> (or as separate properties). The alias is a unique descriptive name representing the scan-root path for the Scan Server or remote scan agent, and the file path is relative to scan root. (The actual absolute scan-root path for each scanner associated with the project is available on the project's **Summary** sheet.)
- The security vulnerability information in the report is based on the CVSS version (v3.x or v2.0) currently used by your Code Insight system for reporting purposes. If CVSS 3.x is used, vulnerability counts and information in the report are based on data from all CVSS v3 systems supported by Code Insight, currently v3.1 and v3.0. (A given vulnerability can have only one v3 score—either a v3.1 or v3.0 score, not both.)
- Suppressed security vulnerabilities are not shown in this report, and total counts for vulnerabilities do not include suppressed vulnerabilities.

Audit Report

Audit reports provide another way to distribute your research and findings to others in your organization. Only published inventory items appear in Audit reports.

Note the following:

- The metrics and statistics in this report are based on the results of the most recent server scan and remote scan(s) associated with the project.
- When the report lists codebase files, an alias and file path can be included with each file name in the format <alias>:<filePath>. The alias is a unique, descriptive name representing the scan-root path for the Scan Server or remote scan agent, and the file path is relative to scan root. (The actual scan root for each scanner associated with a project is available on the project's **Summary** sheet.)
- The total lines of code listed on the **Summary** sheet is based on the server-side codebase only; the total does not include lines of code in the remote codebase(s).
- The security vulnerability information in the report is based on the CVSS version (v3.x or v2.0) currently used by your Code Insight system for reporting purposes. If CVSS 3.x is used, vulnerability counts and information in the report are based on data from all CVSS v3 systems supported by Code Insight, currently v3.1 and v3.0. (A given vulnerability can have only one v3 score—either a v3.1 or v3.0 score, not both.)
- Suppressed security vulnerabilities are not shown in this report, and total counts for vulnerabilities do not include suppressed vulnerabilities.

Notices Report

Code Insight provides the ability to produce a Notices report to satisfy the attribution requirements of most open source licenses. The report is created in text format.

After Engineering has completed the remediation plan, resolving all rejected inventory items, the codebase is rescanned until it is approved for release. When the codebase is approved for release, you need to generate a Notices report to accompany the software application. This report is a compilation of all the open source/third-party components contained in the product and their license content (notices).

The Notices report shows only published inventory. The inventory can be system-generated or custom and of any type—**Work in Progress**, **Component**, or **License**.

The following items can appear in the Notices report for each inventory item:

- **Inventory name**—The entry in this field is based on naming conventions, which is usually the component name, version, and governing license name.
- **Inventory URL**—If the inventory URL is not available, Code Insight uses the associated component URL. If both are unavailable, no URL will appear in the report.
- **Inventory Notices Text**— The final "notices" text associated with the inventory item. It is pulled from the **Notices Text** field on the **Notices Text** tab for a selected inventory item in the **Analysis Workbench** or in **Project Inventory**. If this field is empty, Code Insight uses the content in the **As-Found License Text** field (also on the **Notices Text** tab), which shows the verbatim text license text found in the codebase by the system. If no **As-Found License Text** or **Notices Text** information is available, the text pulled from the Revenera Data Library for the selected license is used in the Notices report. For more information, see Finalizing the Notices Text for the Notices Report

About Custom Reports for Projects

Code Insight provides a Custom Reports Framework that enables users to develop and register custom reports that show project data curtailed for the needs of one's site. For complete details on developing and registering a custom report, refer to the Custom Reports Framework in Code Insight article in the Revenera Community.

A custom report can be defined to output in one or more desired formats. Once a custom report is developed and registered through the Framework, it is available to any project as a selection along with the standard reports on the **Reports** tab. Users can then generate the custom report using the same procedure used to generate a standard report.

Depending on how the custom report is defined, users might be prompted for additional information before they can generate the report. For example, they might be required to select a second project so that data from the current project and second project can be compared or combined into a single report. Or they might be prompted to enter other information such as values to filter report content.

Any user can set up custom reports for a project. (For private projects, any user assigned to a role for the private project can set up custom reports for the project.) Code Insight also provides a set of example custom reports that can serve as a basis for creating your own reports. See the next section Example Custom Reports.

Example Custom Reports

To assist in creating your own custom reports, Code Insight provides the following collection of example custom reports (located in Revenera's public GitHub report repositories):

- Project Inventory Report
- Evidence Report
- Project Comparison Report
- Claimed Evidence Report
- Vulnerabilities Report
- Third-Party Notices Report

The example reports can be registered by following the instructions in sca-codeinsight-reports-installer READme. If your Code Insight server uses a self-signed certificate, you must download and register these reports manually, just as you would register your own custom reports (see the article referenced earlier in About Custom Reports for Projects).

Once the reports are registered, you can modify them as your own or use them as a basis for creating other custom reports.

Disclaimer for Using the Example Custom Reports

These report scripts are being provided solely as examples. They are external to, and not an official part of, the Code Insight product.

THE REPORT SCRIPTS ARE PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SCRIPT OR THE USE OR OTHER DEALINGS IN THE REPORT SCRIPTS.

Project Inventory Report

This report provides an easy, quick method for obtaining a high-level summary of the inventory items within a project.

If you have designated a parent-child hierarchy for your projects to better represent your company offerings, this report can be configured to pull in all child projects (recursively) for the current project and roll up the associated inventory information on a project, as well as an application, basis. Including child projects in the report is useful for keeping track of your software Bill of Materials (SBOM). The report can be further customized to report on other inventory attributes, such as third-party notices, which in turn would capture the notices for all the third-party components included in the report scope.

The report is available in the sca-codeinsight-reports-project-inventory repository in Github.

Evidence Report

This report allows you to report on the following types evidence found in the project:

- Copyrights
- Licenses
- Emails and URLs
- Search terms
- Exact-file matches
- Source-code matches (snippets)

The report is available in the sca-codeinsight-reports-third-party-evidence repository in GitHub.

Project Comparison Report

This report compares the inventory of two projects or two project versions, enabling you to identify inventory differences and commonalities.

The report is available in the sca-codeinsight-reports-project-comparison repository in GitHub.

Claimed Evidence Report

This report allows you to determine which files in a project contain only evidence that is claimable based on string comparisons to the follow evidence types:

- Copyrights
- Emails/URLs

Additionally, you can configure the report so that scanned files that contain only the evidence for the specified claimable values are marked as reviewed and associated with the appropriate inventory items.

The report is available in the sca-codeinsight-reports-claim-files repository in GitHub.

Vulnerabilities Report

This security-focused report calls out all vulnerable project inventory items and lists their associated security vulnerabilities. Use this report to easily collect and review security issues or to share data with your Security team. The report supports searches and enables you to click-through to the actual vulnerable inventory in Code Insight for additional information.

The report is available in the sca-codeinsight-project-vulnerabilities repository in GitHub.

Third-Party Notices Report

This report provides the Notices text for the licenses associated with inventory in the project. This report will automatically include licenses with attribution data if available, thus satisfying the attribution requirement of third-party licenses.

The report will also update an inventory item’s **Notices Text** field for with the attributed license text when possible, based on report options. When multiple licenses are found, all variants will be included, in which case you might need to manually inspect and modify this content for the appropriate inventory in the project.

The report is available in the sca-codeinsight-reports-project-vulnerabilities repository in GitHub.

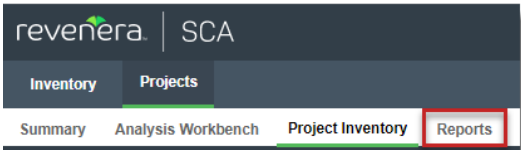
Generating a Report for a Project

Use the following procedure for generating a report for a given project from its **Reports** tab. For more information about the details available on this tab, see Reports Tab. Any Code Insight user can generate reports.

Note* ▪ *If a scan or rescan on the project is currently in queue or in progress, the* **Generate Selected Report** *button on the* **Reports** *tab is disabled to prevent you from generating any report, standard or custom. Once the scan is complete, the button is re-enabled and you can proceed with running the report.

Task ***To generate a report for a project, do the following:***

1. Open a project in the **Projects** view. (For instructions, see Opening a Project.)
2. Click the **Reports** button at the top of the view to open the **Reports** tab.



   The tab opens, showing the list of standard and custom reports available for the project.

3. Select a report from the report list.
4. Click **Generate Selected Report**.
   - If additional information is needed for the report, a pop-up window is displayed, prompting you for the information. See step 5. (Only custom reports can request additional information.)
   - If no additional information is needed, skip to step 6.

5. From the pop-up window requesting additional information to run the report, complete the fields:

    - If the **Include data from Second Project** field is displayed, enter the name of the second project whose data will be included along with the data from the current project for comparison purposes. As you type a string, project names containing that string are listed in a dropdown list from which you can then select the desired project name. (This is a required field.)
    - If other fields are displayed, enter the requested values in those fields. Default values can be overwritten. Click the ? icon next to a field for more information about its purpose and possible values. The **Generate Report** button on the pop-up remains disabled until all required fields are completed. (Required fields left blank are outlined in red.)

    When these additional fields have been properly completed, click **Generate Report** on the pop-up window.

6. Click **OK** from the message box that is displayed, stating that the report will run in the background.

    You can monitor the report's progress on the **Jobs** queue. For more information, see Monitoring the Code Insight Jobs Queue.

7. (Optional) While a given report is generating, repeat steps 3 through 6 to generate another report. You can generate multiple different reports simultaneously. (While a report is generating, the **Generate Selected Report** button is disabled for that report, but *is* enabled for any other report not being generated.)

8. Once the generation of the report has successfully completed, links are displayed in the **View Report** and **Download Report** columns for the report. Select either or both options:

    - To view the report in your browser, click **View**.
    - To download the report, click **Download**. A `.zip` file is downloaded to your system's default location. The archive contains the report in one or more of these formats:
        - **JSON**—The report data can be processed programmatically to integrate with other applications.
        - **XLSX**—The report can be viewed in Microsoft Excel.
        - **TXT**—The report data is saved as text so that you can reformat the report data as desired.

9. Navigate to the folder where you saved the report `.zip` file, unzip the file, and open the report in the desired format.

Post Report Generation

After the successful generation of a given report, its **View** and **Download** links continue to display, along with the date and time of the report generation, on the **Reports** tab until you regenerate the report.

Report Generation Failure

If a report fails to generate, the message "Report generation failed, please refer to the logs for details" is displayed for the report on the **Reports** tab. A Code Insight System Administrator can review the contents of the `core.log` to determine the reason for the report failure and relay the information to the appropriate contacts to fix the issue. The message remains for the report on the **Reports** tab until another attempt to generate the report is made.

Managing Scan Queues Across All Scan Servers

Code Insight provides a global Scan Queue that lets you manage the current scan queues for all active Scan Servers from a single location. You can monitor the queue for any Scan Server; and, if you have sufficient project permissions, you can stop the scan currently running on a given Scan Sever or remove any scans from that server’s scan queue.

- Monitoring Scan Queues
- Stopping a Scan Currently Running on a Scan Server
- Removing a Scan from the Scan Queue for a Scan Server

This feature does not monitor the scan queues for Code Insight remote scan agents.

Monitoring Scan Queues

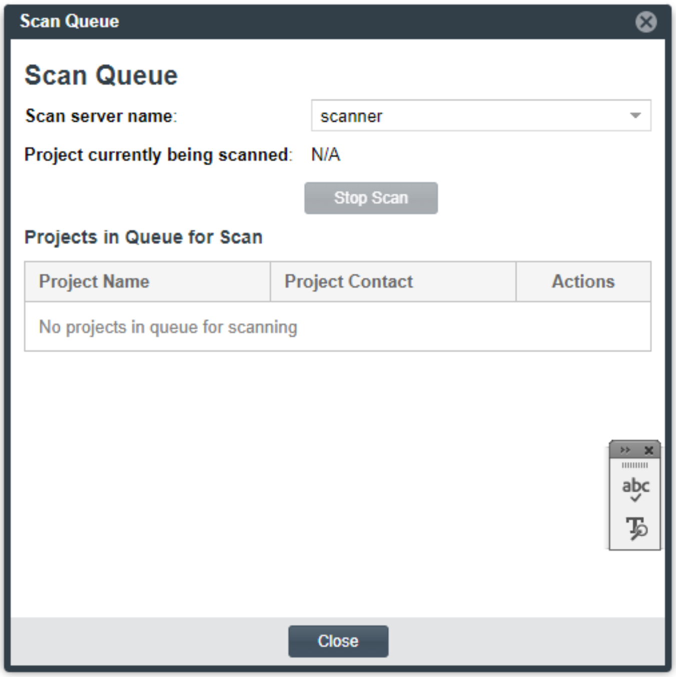
Use the following procedure to open the scan queue for a specific Scan Server to monitor its queue progress. You can then toggle between other Scan Servers to monitor their queues.

Task ***To access a global view of scan queues for all active Scan Servers, do the following:***

1. Click the following icon in the upper right corner of the Code Insight web page to open the Code Insight main menu:

   ☰[

2. Select **SCAN QUEUE** from the menu.
3. From the **Scan Queue** dialog, use the **Scan server name** dropdown list to select the Scan Server whose queue you want to view. (By default, this field initially displays the Scan Server selected by the System Administrator as the global project default.)



Scan Queue

Scan Queue

Scan server name: scanner

Project currently being scanned: N/A

Project Name	Project Contact	Actions
No projects in queue for scanning		

The **Scan Queue** dialog shows the following information about the scan queue for the selected Scan Server:

Field	Description
Project currently being scanned	A hyperlinked value in **<projectName> (<projectContact>)** format, showing the project currently being scanned by the selected Scan Server. Optionally, click the project name to open to the project's **Summary** tab; or click the name of the Project Contact to create and send an email to this contact. These options can be useful for checking the scan's progress or notifying the Project Contact of an issue, especially if the scan if taking an excessive amount of time to execute. If no project is being scanned, this field shows **N/A**. If you have appropriate project permissions, a **Stop Scan** button is enabled to stop the currently running project. See Stopping a Scan Currently Running on a Scan Server for details.
Projects in Queue for Scan	The list of projects (in queue order) waiting to be scanned. Each project entry shows a hyperlinked project name and contact. Optionally, click the name of a given project to open to the project's **Summary** tab; or click the name of a Project Contact to create and send an mail to this contact. These options can be useful if an issue exists with a project in the queue and notifications need to be sent. If you have appropriate project permissions for a given project in the scan queue, an **X** icon is enabled in the **Actions** column to let you remove that project scan from the queue. See Removing a Scan from the Scan Queue for a Scan Server.

4. Use the **Scan server name** dropdown list to toggle between other Scan Server queues, or click **Close**.

Stopping a Scan Currently Running on a Scan Server

Use this procedure to stop the scan currently running on a given Scan Server.

You can stop a scan only if you are an Analyst or a Project Administrator for the project associated with the scan.

Task ***To stop the scan currently running on a given Scan Server, do the following:***

1. Click the following icon in the upper right corner of the Code Insight web page to open the Code Insight main menu:

   ☰

2. Select **SCAN QUEUE** from the menu.
3. From the **Scan Queue** dialog, use the **Scan server name** dropdown list to select the Scan Server on which the scan you want to stop is running.

   Once you select a server, the **Stop Scan** button required to stop the scan will be enabled only if a scan is currently in progress on the Scan Server *and* you are an Analyst or a Project Administrator for the project on which the scan is running.

4. Ensure that the scan you are stopping is the desired scan; and, if so, click **Stop Scan**.

   You are asked to confirm that you want to stop the scan.

5. Click **Yes**.

   Once the scan stops, the scan for the next project in the queue begins.

Removing a Scan from the Scan Queue for a Scan Server

Use this procedure to remove a scan from the scan queue for a given Scan Server.

You can remove a queued scan only if you are an Analyst or a Project Administrator for the project associated with the scan.

Task ***To remove a scan from the scan queue for a given Scan Server, do the following:***

1. Click the following icon in the upper right corner of the Code Insight web page to open the Code Insight main menu:

   ☰

2. Select **SCAN QUEUE** from the menu.
3. From the **Scan Queue** dialog, use the **Scan server name** dropdown list to select the Scan Server from whose scan queue you want to remove one or more scans.

4. In the scan queue, locate the project associated with the scan you want to remove, and click the **X** in the **Actions** column for that scan.

   ***Note** ▪ The **X** icon is enabled only if you are an Analyst or a Project Administrator for the project associated with the scan.*

   You are asked to confirm that you want to remove the scan from the scan queue.

5. Click **Yes** to remove the scan from the queue.

6. Repeat steps 4 and 5 to remove any additional scans in queue.

Monitoring the Code Insight Jobs Queue

The Code Insight **Jobs** queue enables you to monitor all scheduled and active jobs and review historical (completed, failed, or otherwise) jobs in your Code Insight system. Job types in the queue range from project scans and rescans to system-wide Electronic Updates and Library Refreshes and the myriad of all project and global operations in between. The grid format of the queue enables you to easily read the details for a given job. Additionally, search mechanisms are available that help you locate the jobs you want to monitor or review. You can refresh the queue at anytime to obtain the latest job information.

These sections provide the information needed to use the **Jobs** queue:

- Opening the Jobs Queue
- Searching the Jobs Queue
- Refreshing the Jobs Queue
- Job Queue REST Interface

Opening the Jobs Queue

The Code Insight **Jobs** queue is accessed from the Code Insight main menu.

Task ***To access the Jobs queue, do the following:***

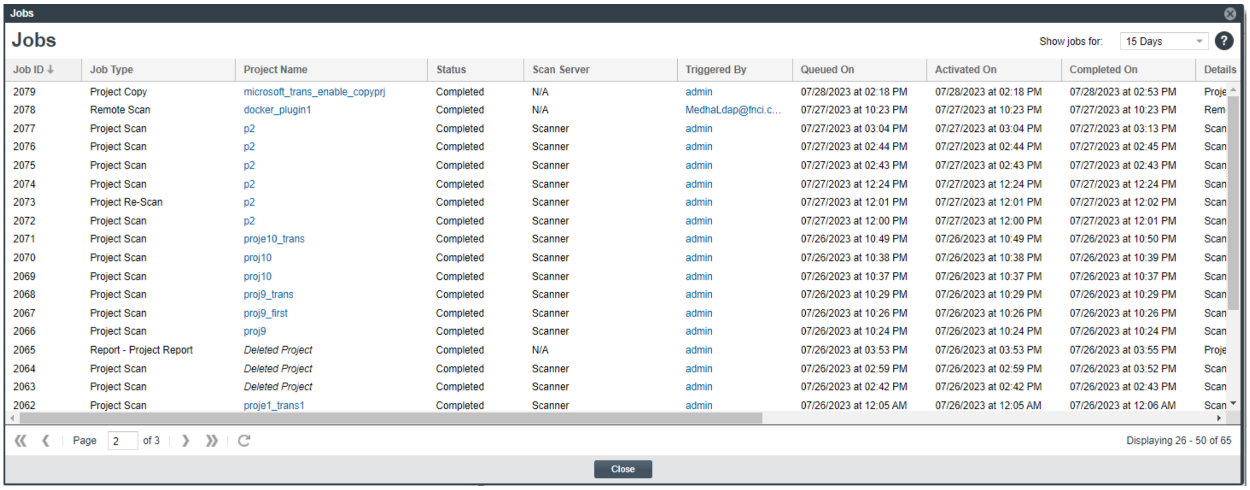
1. Click the following icon in the upper right corner of the Code Insight web page to open the Code Insight main menu:

   ☰

2. Select **JOBS** from the menu.

The **Jobs** queue is displayed (within the **Jobs** window). For a detailed description of the columns that provide details for each job in the **Jobs** queue, see Jobs Queue.

By default, the list shows all jobs queued in the last 15 days and sorted in descending order by the **Job ID** column. (See Searching the Jobs Queue for instructions on how to adjust this default view to help you locate jobs.)



Jobs

Show jobs for: 15 Days

Job ID ↓	Job Type	Project Name	Status	Scan Server	Triggered By	Queued On	Activated On	Completed On	Details
2079	Project Copy	microsoft_trans_enable_copyprj	Completed	N/A	admin	07/28/2023 at 02:18 PM	07/28/2023 at 02:18 PM	07/28/2023 at 02:53 PM	Proje
2078	Remote Scan	docker_plugin1	Completed	N/A	MedhaLdap@fnci.c...	07/27/2023 at 10:23 PM	07/27/2023 at 10:23 PM	07/27/2023 at 10:23 PM	Rem
2077	Project Scan	p2	Completed	Scanner	admin	07/27/2023 at 03:04 PM	07/27/2023 at 03:04 PM	07/27/2023 at 03:13 PM	Scan
2076	Project Scan	p2	Completed	Scanner	admin	07/27/2023 at 02:44 PM	07/27/2023 at 02:44 PM	07/27/2023 at 02:45 PM	Scan
2075	Project Scan	p2	Completed	Scanner	admin	07/27/2023 at 02:43 PM	07/27/2023 at 02:43 PM	07/27/2023 at 02:43 PM	Scan
2074	Project Scan	p2	Completed	Scanner	admin	07/27/2023 at 12:24 PM	07/27/2023 at 12:24 PM	07/27/2023 at 12:24 PM	Scan
2073	Project Re-Scan	p2	Completed	Scanner	admin	07/27/2023 at 12:01 PM	07/27/2023 at 12:01 PM	07/27/2023 at 12:02 PM	Scan
2072	Project Scan	p2	Completed	Scanner	admin	07/27/2023 at 12:00 PM	07/27/2023 at 12:00 PM	07/27/2023 at 12:01 PM	Scan
2071	Project Scan	proje10_trans	Completed	Scanner	admin	07/26/2023 at 10:49 PM	07/26/2023 at 10:49 PM	07/26/2023 at 10:50 PM	Scan
2070	Project Scan	proj10	Completed	Scanner	admin	07/26/2023 at 10:38 PM	07/26/2023 at 10:38 PM	07/26/2023 at 10:39 PM	Scan
2069	Project Scan	proj10	Completed	Scanner	admin	07/26/2023 at 10:37 PM	07/26/2023 at 10:37 PM	07/26/2023 at 10:37 PM	Scan
2068	Project Scan	proj9_trans	Completed	Scanner	admin	07/26/2023 at 10:29 PM	07/26/2023 at 10:29 PM	07/26/2023 at 10:29 PM	Scan
2067	Project Scan	proj9_first	Completed	Scanner	admin	07/26/2023 at 10:26 PM	07/26/2023 at 10:26 PM	07/26/2023 at 10:26 PM	Scan
2066	Project Scan	proj9	Completed	Scanner	admin	07/26/2023 at 10:24 PM	07/26/2023 at 10:24 PM	07/26/2023 at 10:24 PM	Scan
2065	Report - Project Report	*Deleted Project*	Completed	N/A	admin	07/26/2023 at 03:53 PM	07/26/2023 at 03:53 PM	07/26/2023 at 03:55 PM	Proje
2064	Project Scan	*Deleted Project*	Completed	Scanner	admin	07/26/2023 at 02:59 PM	07/26/2023 at 02:59 PM	07/26/2023 at 03:52 PM	Scan
2063	Project Scan	*Deleted Project*	Completed	Scanner	admin	07/26/2023 at 02:42 PM	07/26/2023 at 02:42 PM	07/26/2023 at 02:43 PM	Scan
2062	Project Scan	proje1_trans1	Completed	Scanner	admin	07/26/2023 at 12:05 AM	07/26/2023 at 12:05 AM	07/26/2023 at 12:06 AM	Scan

Displaying 26 - 50 of 65

Searching the Jobs Queue

The following methods help you search the **Jobs** queue to locate jobs and their details more easily.

- Filtering the Jobs Queue by Column
- Changing the “Show jobs for” Filter
- Sorting the Jobs Queue
- Managing Column Visibility
- Navigating Pages in the Jobs Queue

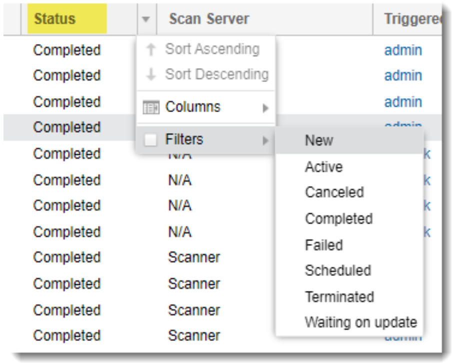
The search methods that you apply to the queue persist only while the **Jobs** window is open. Once you close the window and reopen it, the queue is displayed in its default view.

Filtering the Jobs Queue by Column

You can filter the **Jobs** queue on the **Job Type**, **Project Name**, **Status**, **Scan Server**, and **Triggered By** columns. If you filter by multiple columns, a job must meet the criteria of all the column filters in order to be included in the resulting list.

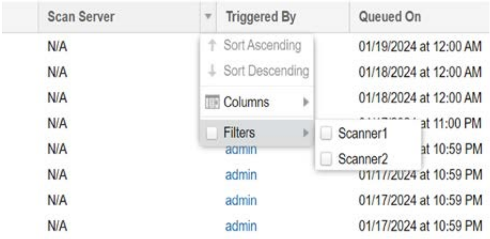
Task ***To define a filter on a column, do the following:***

1. Hover over the column header, and click the down arrow in the header to open a dropdown menu.
2. Move the cursor to the **Filters** checkbox in the menu. The field (a list or text box) used to identify the filter value is displayed to the right of **Filters**.
3. Provide the filter value:
    - For the **Job Type** or **Status** column, select a value from the list. (Currently, you can make only one selection.) Once you select a value, the **Filters** checkbox is automatically selected and the queue is filtered to that value.

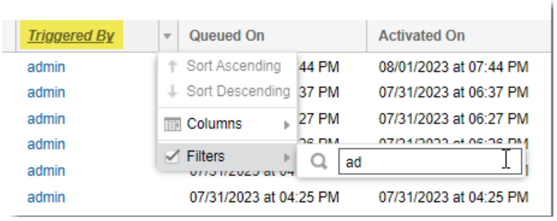


- For **Scan Server**, select one or more Scan Server aliases from the list. (The aliases for all current Scan Servers, including disabled ones, are available in the list.) Once you select the aliases, the **Filters** checkbox is automatically selected and the queue is filtered to those aliases.

  ***Note** ▪ You can not filter by the **N/A** value (representing the Core Server).*



- For **Project Name** or **Triggered By,** enter the name (or a string within the name) of the project or of the user who triggered the job, respectively. When you begin typing, the **Filters** checkbox is automatically selected, and the queue is filtered as you type.



The column name is italicized to remind you that the queue is currently filtered on this column.

4. (Optional) Repeat the previous steps to define a filter on another column.

Clearing a Column Filter

Use these steps to clear the filter on a column in the **Jobs** queue.

Task ***To clear a column filter, do the following:***

1. Hover over the header for the column on which the filter is defined, and click the down arrow in the header.

   Note ▪ *The headers of those columns on which the queue is currently filtered are italicized to help you locate the column.*

2. From the dropdown menu, click the **Filters** checkbox to “uncheck” it and disable the filter. The queue is automatically updated to reflect the removal of the filter.

Changing the “Show jobs for” Filter

By default, the **Jobs** queue shows only those jobs queued in the last 15 days so that you have a manageable number of jobs to monitor. You can adjust this filter by updating the number of days in the **Show jobs for** dropdown.

Task ***To change the “Show jobs for” value, do the following:***

In the **Show jobs for** dropdown in the upper right corner of the **Jobs** window, select a value:

- **15 Days**—Show only jobs queued in the last 15 days. (Default)
- **30 Days**—Show only jobs queued in the last 30 days.
- **All**—Show all jobs in the queue.

Once you select a value, the **Jobs** queue is immediately adjusted to reflect your selection.

Sorting the Jobs Queue

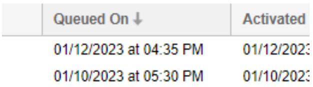
By default, the **Jobs** queue is sorted by the **Jobs ID** column in descending order. However, you can choose to sort by this column in ascending order or sort by the **Queued On** or **Completed On** column instead.

Task ***To change the current sorting configuration, do the following:***

1. Hover over the header of the column on which you want to sort the queue, and click the down arrow in the header.

2. In the dropdown menu, click **Sort Ascending** or **Sort Descending**.

   An up or down arrow is displayed in the column header to indicate that the **Jobs** queue is currently sorted on this column in ascending or descending order, respectively.



Queued On ↓	Activated
01/12/2023 at 04:35 PM	01/12/2023
01/10/2023 at 05:30 PM	01/10/2023

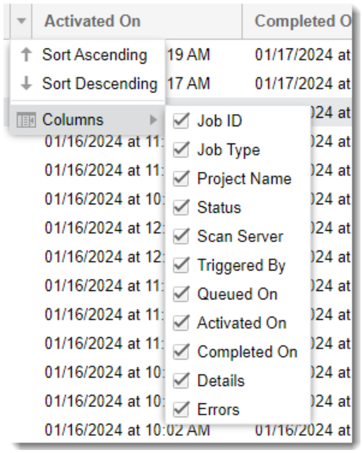
Managing Column Visibility

To focus on specific information in the **Jobs** queue, you can hide or redisplay individual columns in the grid.

Task ***To manage column visibility in the Jobs queue, do the following:***

1. Hover over the header of any column, and click the down arrow in the header to open a dropdown menu.
2. Move the cursor to the **Columns** option in the menu. The list of all columns available in the **Jobs** queue is displayed to right of **Columns**.

   Note ▪ *Each time you open the **Jobs** window, all columns are visible by default. Thus, all items in the list are initially "checked".*

3. From this list, manage column visibility in the queue.
   - Select a "checked" column name to hide it in the queue.
   - Select an "unchecked" column name make it visible in the queue.

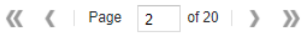
   As you make selections to hide or restore columns, the **Jobs** queue automatically reflects the changes.

Navigating Pages in the Jobs Queue

The **Jobs** queue is paginated, showing 25 jobs per page. Page controls are available to help you navigate between pages to locate jobs.

Task ***To navigate between pages in the Jobs queue, do the following:***

Use the controls at the bottom of any page in the **Jobs** queue page to move to the next or previous page, to the first or last page in the queue, or to a specific page number.



Refreshing the Jobs Queue

You can refresh the **Jobs** queue as often as needed to monitor the latest job information. If you have applied search methods to the queue (see Searching the Jobs Queue), your current view is retained when you refresh the queue.

Task ***To refresh the Jobs queue, do the following:***

Click the **Refresh** icon at the bottom of the queue.

↻

Job Queue REST Interface

Code Insight also provides the following REST APIs that retrieve job information so that you can monitor current jobs and review historical ones.

- **Get jobs details based on filters** (`/jobs`)—Retrieves the details for all jobs or for those jobs based on the filters you specify.
- **Get job details based on the jobid provided** (`/jobs/{jobId}`)—Retrieves details for a specific job.

For more information about using these APIs, see the following references:

- Code Insight Swagger *REST API Guide* (accessed through the **HELP** option on the Code Insight main menu)
- *Code Insight REST API Documentation* (accessed from the Revenera Documentation site)

Viewing Inventory Across All Projects

Code Insight enables you to view published inventory of open-source software (OSS) and other third-party components found across the projects in your Code Insight system. This inventory, displayed in a single scrollable window called the **Inventory** view, provides the means to make overall assessments of the OSS or third-party code used in your company’s software.

The **Inventory** view can be filtered and refined as needed to focus on the inventory details and trends that most concern you and that are needed to make sound business decisions.

For example, you might need to determine which open-source or third-party components are putting your software deliverables at security risk due to security vulnerabilities above a certain severity or CVSS score, or which are posing threats to your intellectual property due to non-compliant licensing per your corporate policies. You might want to filter to inventory where security, legal, and development resources are most needed to complete the review or remediation work required to ensure that OSS or third-party code is properly and safely integrated in all your software projects.

The **Inventory** view also provides direct links to inventory items and the projects associated with these items so that you can investigate or manage the inventory and projects as needed.

The following topics provide the procedures needed to access and use the **Inventory** view:

- Opening the Inventory View

- Switching the Context of the Inventory View
- Including the Inventory of Child Projects on the Inventory View
- Refining the Inventory View
- Viewing Inventory Properties and Linking to Additional Information
- Opening the Project Associated with Inventory from the Inventory View

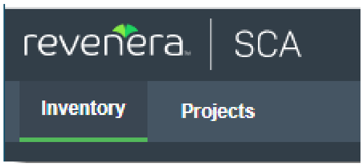
Opening the Inventory View

Use the following procedure to access the **Inventory** view.

Task ***To access the Inventory view, do the following:***

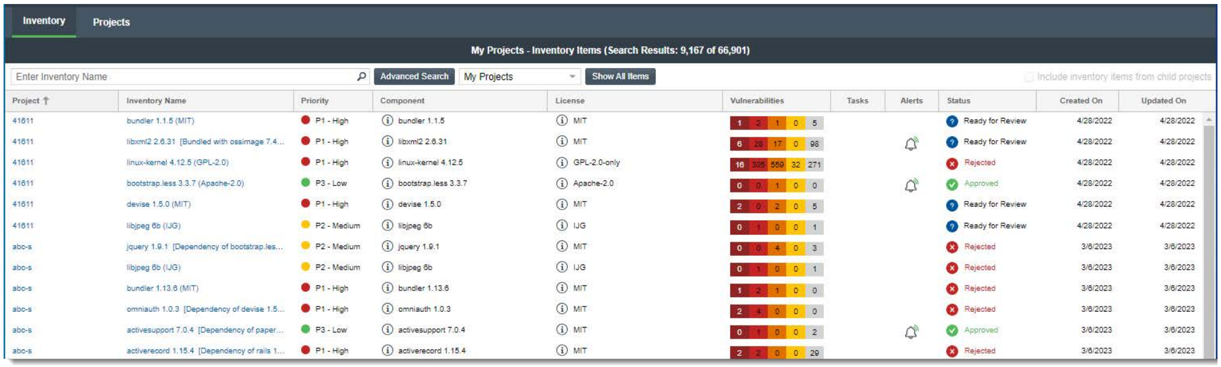
Open the **Inventory** view using one of these methods:

- From the Code Insight dashboard (which is displayed when you start Code Insight), click **view inventory**. See Opening Code Insight for details on accessing this dashboard.
- From any location in the Code Insight Web UI, click the **Inventory** button under the Code Insight logo.



- Click the ☰ icon in the upper right corner of the Code Insight web page to open the Code Insight main menu. Select **INVENTORY** from the menu.

The **Inventory** view is displayed. For a description of all the columns, fields, and buttons available on the view, see Inventory View.



My Projects - Inventory Items (Search Results: 9,167 of 66,901)

Project ↑	Inventory Name	Priority	Component	License	Vulnerabilities	Tasks	Alerts	Status	Created On	Updated On
41611	bundler 1.1.5 (MIT)	P1 - High	bundler 1.1.5	MIT	1 2 1 0 5			Ready for Review	4/28/2022	4/28/2022
41611	libxml2 2.6.31 [Bundled with ossimage 7.4...	P1 - High	libxml2 2.6.31	MIT	6 28 17 0 98		🔔	Ready for Review	4/28/2022	4/28/2022
41611	linux-kernel 4.12.5 (GPL-2.0)	P1 - High	linux-kernel 4.12.5	GPL-2.0-only	16 305 559 32 271			Rejected	4/28/2022	4/28/2022
41611	bootstrap.less 3.3.7 (Apache-2.0)	P3 - Low	bootstrap.less 3.3.7	Apache-2.0	0 0 1 0 0		🔔	Approved	4/28/2022	4/28/2022
41611	devise 1.5.0 (MIT)	P1 - High	devise 1.5.0	MIT	2 0 2 0 5			Ready for Review	4/28/2022	4/28/2022
41611	libjpeg 6b (IJG)	P2 - Medium	libjpeg 6b	IJG	0 1 0 0 1			Ready for Review	4/28/2022	4/28/2022
abc-s	jquery 1.9.1 [Dependency of bootstrap.les...	P2 - Medium	jquery 1.9.1	MIT	0 0 4 0 3			Rejected	3/6/2023	3/6/2023
abc-s	libjpeg 6b (IJG)	P2 - Medium	libjpeg 6b	IJG	0 1 0 0 1			Rejected	3/6/2023	3/6/2023
abc-s	bundler 1.13.6 (MIT)	P1 - High	bundler 1.13.6	MIT	1 2 1 0 0			Rejected	3/6/2023	3/6/2023
abc-s	omniauth 1.0.3 [Dependency of devise 1.5...	P1 - High	omniauth 1.0.3	MIT	2 4 0 0 0			Rejected	3/6/2023	3/6/2023
abc-s	activesupport 7.0.4 [Dependency of paper...	P3 - Low	activesupport 7.0.4	MIT	0 1 0 0 2		🔔	Approved	3/6/2023	3/6/2023
abc-s	activerecord 1.15.4 [Dependency of rails 1...	P1 - High	activerecord 1.15.4	MIT	2 2 0 0 29			Rejected	3/6/2023	3/6/2023

Switching the Context of the Inventory View

When you open the **Inventory** view, you see the inventory associated with only your Code Insight projects. However, you can switch the context of inventory items as described in these sections:

- About the Available Contexts for the View
- Switching the Context of the View

About the Available Contexts for the View

The following describe the three major contexts that are available for the **Inventory** view. You can switch between contexts as needed:

Table 2-16 ▪ Contexts of the Inventory View

Context	Description
My Projects	Shows all published inventory across those Code Insight projects in which you are assigned a role. You might use this context to show areas where you need to provide review or remedial work, or you might want to review the overall state of inventory found in your projects. This is the context enabled by default when you open the **Inventory** view.
All Projects	Shows all published inventory across all projects in your Code Insight system. This context is helpful in visualizing trends in your company’s use of open- source and third-party code in its software projects.
Selected Project	Shows all published inventory for a selected Code Insight project. Since projects represent versions of a particular software product, this view allows you to see all inventory items for that product. Furthermore, you can also opt to list the inventory for all child projects of the selected project. These child projects represent modules used by your top-level product. You can directly link to the inventory item of the child project or to the child project itself. You can also view the parent hierarchy of the child project to understand the provenance of the inventory items. See Including the Inventory of Child Projects on the Inventory View for details.

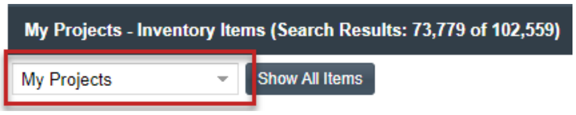
When you switch to a different context, any filtering criteria that you have selected on the **Advanced Inventory Search** dialog remains in effect. See Filtering by Inventory Details for more information about setting this criteria.

Switching the Context of the View

Use the following procedure to switch the major context of the **Inventory** view.

Task ***To switch the context of the Inventory view, do the following:***

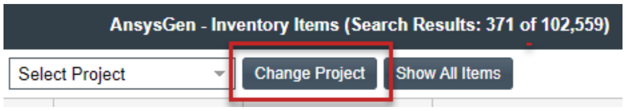
1. Open the **Inventory** view. See Opening the Inventory View.
2. In the dropdown list at the top of the view, select the context to which you want to switch the view.



My Projects - Inventory Items (Search Results: 73,779 of 102,559)

- If you select **All Projects** or **My Projects**, the list is refreshed with the inventory for the appropriate Code Insight projects.
- If you choose **Select Project**, a dialog is displayed from which to select a project. Once you choose the project, the inventory list is refreshed with the inventory for that project only.

  To switch to a different project, click the **Change Project** button that is displayed next to the dropdown list, and choose another project.



AnsysGen - Inventory Items (Search Results: 371 of 102,559)

  When select to focus the **Inventory** view on a specific project, you have the option to include the inventory of that project's child projects, if any exist. See Including the Inventory of Child Projects on the Inventory View.

Including the Inventory of Child Projects on the Inventory View

When you have selected to focus on a specific Code Insight project in the **Inventory** view, you have the option to include the inventory for the child projects of the selected project.

A *child project* is a project dependent on another project. For example, the project for an application that your company is developing might be associated with modules for which you have created separate projects to track the specific OSS and third-party components found in their codebases. To help you keep track of the project relationships, Code Insight lets you identify these dependent projects, such as those for the modules, as child projects to the main application project (called a *parent project*). For more information about creating a project hierarchy, see Identifying Child Projects for a Project.

If you have identified child projects for the project currently selected for the **Inventory** view, you can include the inventory for the child projects in the view. In this way, you can examine the inventory found across the project codebases for all parts of your software project, including its dependencies and sub-modules.

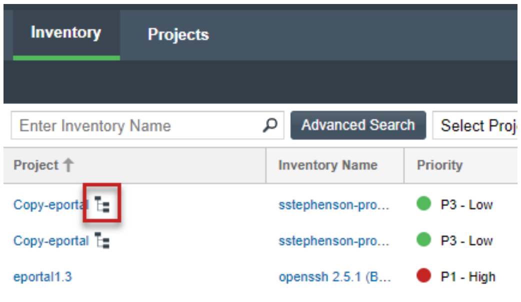
Note that by selecting to include inventory from child projects, all child projects associated with the current top-level project will be recursively included in your inventory items list.

Task ***To include inventory for child projects of the project selected for the Inventory view, do the following:***

1. Open the **Inventory** view. See Opening the Inventory View.
2. In the dropdown list at the top of the view, choose **Select Project** and then choose the project on whose inventory you want to focus the view.
3. Select **Include Inventory items for child projects** in the top right of the **Inventory** view.

   The view is refreshed to include the inventory of all child projects (recursively) of the project in focus.

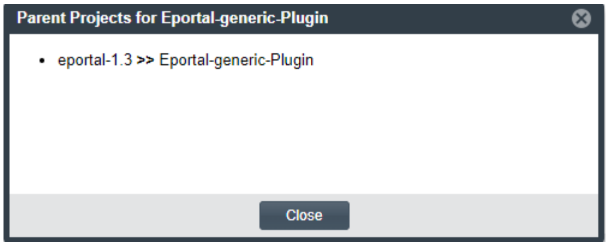
   In the entry for each inventory item of a child project, the hierarchy icon is displayed next to the project link in the **Project** column.



Project	Inventory Name	Priority
Copy-eportal	sstephenson-pro...	● P3 - Low
Copy-eportal	sstephenson-pro...	● P3 - Low
eportal1.3	openssh 2.5.1 (B...	● P1 - High

4. To view a hierarchy description of each parent of the child project, click .

   The following shows an example of the hierarchy description for child project with two parents.



- eportal-1.3 >> Eportal-generic-Plugin

Refining the Inventory View

You can use the following methods to refine the inventory on the **Inventory** view to pinpoint the data you want to examine:

- Filtering by Inventory Name
- Filtering by Inventory Details
- Focusing Column Content
- Removing All Filters

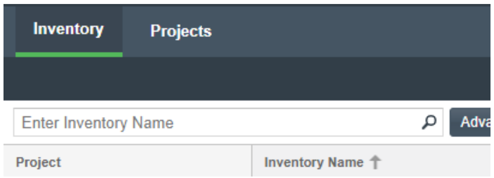
For instructions on accessing the **Inventory** view, see Opening the Inventory View.

Filtering by Inventory Name

You can filter the inventory on the **Inventory** view by the name of the inventory item.

Task ***To filter inventory by the inventory name, do the following:***

In the **Enter Inventory Name** field at the top of the **Inventory** view, enter a string by which to search inventory names.



If necessary, click the search icon next to the field to initiate search.

To remove the string and restore the full list of inventory items, click the **X** in the field.

Filtering by Inventory Details

The **Inventory** view enables you to perform an advanced search that filters inventory by details such as:

- Inventory attributes such as inventory name, priority, age, review status, confidence level, and notifications
- Attributes of the tasks and licenses associated with inventory
- Attributes of security vulnerabilities associated with inventory

The results of the advanced search on the **Inventory** view remain within the current context of the view (**My Projects**, **All Projects**, or a selected project). Likewise, the search criteria persists on the **Inventory** view even if you change the context of the **Inventory** view. (See Switching the Context of the Inventory View.)

Task ***To filter the inventory by inventory details, do the following:***

1. Click the **Advanced Search** button at the top of the **Inventory** view. The **Advanced Inventory Search** dialog is displayed.
2. From this dialog, select search criteria as needed from the following categories. For a detailed description of the search criteria, see Advanced Inventory Search Dialog.

   - **Inventory Items**—Search for inventory items of a certain name (or string), review status, priority, type, creator, dependency scope, age, usage, or confidence level or that have open vulnerability alerts and work items. (For details on alerts and work items, see Managing Security Vulnerability Alerts and Creating and Viewing External Work Items for a Project Inventory Task.)

     If you entered a name filter in the **Inventory Name** field at the top of the **Inventory** view, it is automatically displayed for the **Inventory Name** filter on the **Advanced Inventory Search** dialog. (Likewise, if you enter a name filter on the **Advanced Inventory Search** dialog, it is copied to the **Inventory** view.) This behavior enables you to keep the name filter persistent. However, you can remove or replace this filter as needed in either location.

- **Inventory Tasks**—Search for inventory items that have been assigned tasks. You can refine the search to locate inventory with open or closed tasks, tasks of a certain age or type (such as manual reviews or source-code remediation), or tasks assigned to a specific user.
- **Inventory Custom Fields**—Search for inventory whose custom inventory fields contain the value you specify as criteria (or contain no value). Custom inventory fields are defined specifically for your site. If no such fields have been defined this section is not visible.
- **Security Vulnerabilities**—Search for inventory items that have vulnerabilities of a certain vulnerability ID, CVSS severity, or age. (Note that list of available severities for **Security Vulnerability Severity** varies depending on the CVSS version being used by Code Insight. The picture above shows the severities for CVSS v3.x. See Working with Security Vulnerabilities for details.)
- **Licenses**—Search for inventory items that have licenses of a certain of a certain name or license priority or licenses with no associated version.

***Note** ▪ If you select to filter by license or security-vulnerability criteria on the **Advanced Inventory Search** dialog, the filtering process might take longer than usual.*

3. In **Apply Criteria** field, select the boolean operator to apply to the criteria:
    - **Or**—To be included in the search results, an inventory item must contain at least one of the criteria you selected on this dialog.
    - **And**—To be included in the search results, an inventory item must meet all the criteria across the advanced search, as selected in this dialog. (This is the default operator.)
4. Click **Apply** to filter the inventory to display only those inventory items that meet the selected criteria.
5. To refresh the **Inventory** view to remove all advanced search filters, do either:
    - To remove the advanced-search filters only, return to the **Advanced Inventory Search** dialog, and click **Clear Form** and then **Close**.
    - To remove all filters, see Removing All Filters.

Focusing Column Content

Focus the column content as needed on the **Inventory** view to see the values most pertinent to you.

Task ***To focus column content, do the following:***

1. Hover over the right side of any column header, and click its dropdown menu.
2. From the menu, do any of the following:
    - Re-sort the **Inventory** view by column in ascending or descending order. By default, the view is sorted alphabetically by **Inventory Name** in ascending order.

      ***Note** ▪ Currently you can re-sort the view by the **Project**, **Inventory Name**, **Priority**, **#Files** (hidden by default), **Status**, **Created By** (hidden by default), **Created On**, or **Updated On** column.*

- Click **Columns** to select specific columns to display or hide in the view. (The **# Files** and **Created By** columns are hidden by default.)

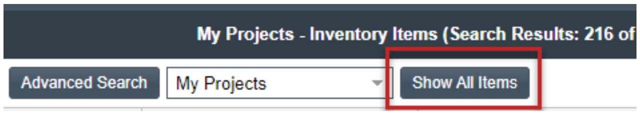
Your column configuration persists when you change the major context of the **Inventory** view (see Switching the Context of the Inventory View).

Removing All Filters

The following procedure removes all current criteria configured on the **Advanced Inventory Search** dialog and switches the context of the **Inventory** view to show all projects.

Task ***To remove all filters and switch to the All Projects context, do the following:***

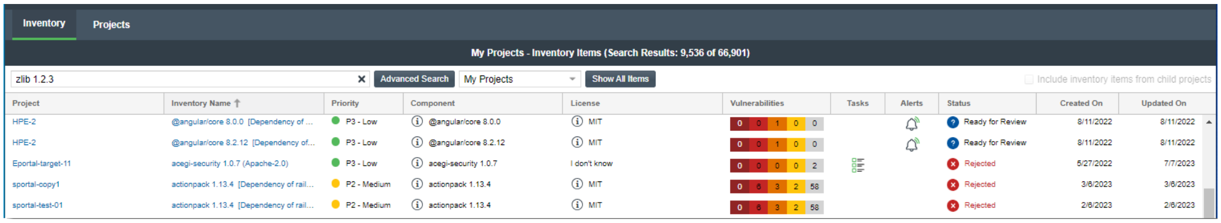
Click the **Show All Items** button at the top of the **Inventory** view.



Note ▪ *This button does not display if the **Inventory** view is already using the **All Projects** context.*

Viewing Inventory Properties and Linking to Additional Information

The **Inventory** view highlights the important properties for the inventory—the project to which the inventory is associated, the number of known security vulnerabilities, inventory review status, whether the inventory has open task or alerts, and more—enabling you to quickly review the listed inventory items for areas concern or interest. Some properties include links to additional information, enabling you to explore an inventory item in more depth.



My Projects - Inventory Items (Search Results: 9,536 of 66,901)

zlib 1.2.3 | My Projects | □ Include inventory items from child projects

Project	Inventory Name ↑	Priority	Component	License	Vulnerabilities	Tasks	Alerts	Status	Created On	Updated On
HPE-2	@angular/core 8.0.0 [Dependency of ...	● P3 - Low	ⓘ @angular/core 8.0.0	ⓘ MIT	0 0 1 0 0		🔔	? Ready for Review	8/11/2022	8/11/2022
HPE-2	@angular/core 8.2.12 [Dependency of...	● P3 - Low	ⓘ @angular/core 8.2.12	ⓘ MIT	0 0 1 0 0		🔔	? Ready for Review	8/11/2022	8/11/2022
Eportal-target-11	acegi-security 1.0.7 (Apache-2.0)	● P3 - Low	ⓘ acegi-security 1.0.7	I don't know	0 0 0 0 2	☰		✕ Rejected	5/27/2022	7/7/2023
sportal-copy1	actionpack 1.13.4 [Dependency of rail...	● P2 - Medium	ⓘ actionpack 1.13.4	ⓘ MIT	0 6 3 2 58			✕ Rejected	3/6/2023	3/6/2023
sportal-test-01	actionpack 1.13.4 [Dependency of rail...	● P2 - Medium	ⓘ actionpack 1.13.4	ⓘ MIT	0 6 3 2 58			✕ Rejected	2/6/2023	2/6/2023

For example, the **Tasks** property provides a link to view and edit any open tasks for an inventory item. The **Component** and **License** properties can link to more details about the component or license as found in the Code Insight Data Library of third-party and OSS component information. The **Vulnerabilities** and **Alerts** properties can link directly to the CVSS information pertaining to any security issues associated with an inventory item.

Additionally, for a more comprehensive information of a given inventory item, you can click within the row for a given inventory item on the **Inventory** view to open a read-only slide-out panel of the item's details. Alternatively, links are provided to directly access the project with which an inventory item is associated. Then, from within the project, you can explore information gathered for the specific inventory item (and for all inventory in the project) and edit this information as necessary according to your permissions.

For more information, see the following topics:

- Opening a Read-Only Version of Inventory Details on the Inventory View
- Opening the Project Associated with Inventory from the Inventory View

For a complete description of the inventory properties and associated links available on the **Inventory** view, see Inventory View.

Opening a Read-Only Version of Inventory Details on the Inventory View

If you want to examine a read-only version of the details for a given inventory item on the **Inventory** view, use the procedure described in this section. The inventory details are displayed on a slide-out panel within the **Inventory** view, providing an easy means of obtaining information about the inventory item without having to open the project link on the view (although links are available on the slide-out, enabling you to access the project if necessary).

Alternatively, you can open the links to the project directly from the **Inventory** view to view and edit inventory, as described in Opening the Project Associated with Inventory from the Inventory View.

Task ***To open a read-only view of the details for a given inventory item, do the following:***

1. Open the **Inventory** view. For instructions, see Opening the Inventory View.
2. Click within the row of the inventory item whose details you want to view. (Click anywhere within the row except on linked text or a linked icon.)

   A slide-out panel is displayed, showing most of the inventory tabs and details that are also available for the inventory item on its **Project Inventory Details** pane in the actual project. (Note that the **Component Details** tab is not available on the slide-out.) Unlike the **Project Inventory Details** pane, you cannot edit detail values on the slide-out.

   While the slide-out details are read-only, certain values are hyperlinked, enabling you to still explore and maintain the inventory item (as your permissions allow). For example, you can click the **Project** name link to open the **Project Inventory** tab in the actual project, where you can access and edit any inventory in the project. Or you can click the inventory item’s **Name** link to open the item’s **Project Inventory Details** pane in project, where you can actually edit the inventory item. Links are also available to open and maintain existing tasks and alerts for the inventory item and to access the item’s **Provenance**, **Workflow**, and component **URL** sites.

   For a description of all available inventory details, see Project Inventory Details Pane.
3. Once you have completed examining the details, click in window area outside the slide-out (or click ✕ in the upper-right corner of the slide-out) to close it.

Opening the Project Associated with Inventory from the Inventory View

From the **Inventory** view, you can open the project to which a given inventory item on the **Inventory** view is associated. Within the project, you can then view and edit details for the selected inventory item, edit other inventory, and perform project-related functions as your permissions allow. To open a project, use either procedure:

- Open the Associated Project Directly to the Details for a Given Inventory Item
- Open the Associated Project to the List of All Inventory in the Project

The project opens to the appropriate location on its **Project Inventory** tab. However, once on the **Project Inventory** tab, you can navigate anywhere in the project and perform any function for which you have user permissions.

Alternatively, if you simply want to examine the details of a given inventory item on the **Inventory** view, you can open a read-only version of the details within the view instead of opening the associated project. See Opening a Read-Only Version of Inventory Details on the Inventory View.

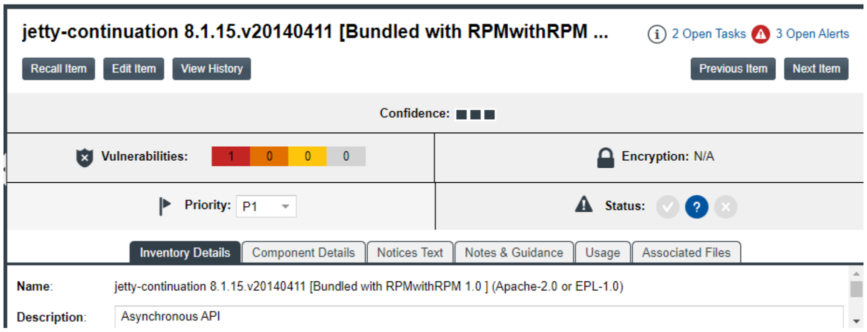
Open the Associated Project Directly to the Details for a Given Inventory Item

For a given inventory item on the **Inventory** view, you can open its associated project directly to the **Project Inventory Details** pane for that inventory item, where you can then examine and edit the item's details.

Task ***To open the project directly to the details for a given inventory item, do the following:***

1. Open the **Inventory** view. For instructions, see Opening the Inventory View.
2. For the given inventory item on the **Inventory** view, click its hyperlinked inventory name (in the **Inventory Name** column).

   The project opens to the **Project Inventory Details** pane for the specific inventory item (see the following example) on the **Project Inventory** tab, providing access to all information for the item and enabling you update this information according to your user permissions. For more information, see Project Inventory Details Pane.



jetty-continuation 8.1.15.v20140411 [Bundled with RPMwithRPM ...

Confidence: ■■■

Vulnerabilities: 1 0 0 0

Encryption: N/A

Priority: P1

Status:

Name: jetty-continuation 8.1.15.v20140411 [Bundled with RPMwithRPM 1.0] (Apache-2.0 or EPL-1.0)

Description: Asynchronous API

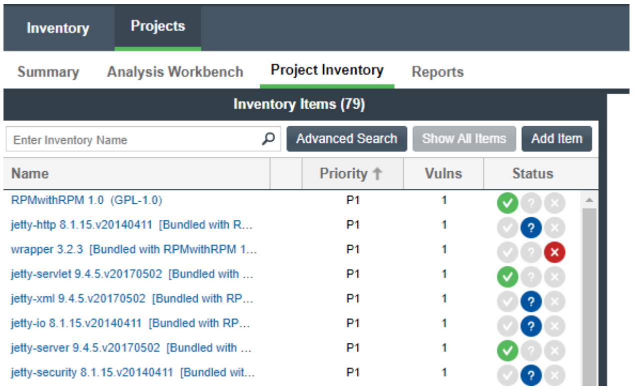
3. To return to the **Inventory** view, click the **Inventory** button under the Code Insight logo on the screen.

Open the Associated Project to the List of All Inventory in the Project

For a given inventory item on the **Inventory** view, you can open its associated project directly to the inventory list on the **Project Inventory** tab, where you have access to all published inventory in the project, including the given inventory item, and to the project and its functionality. From here, you can edit any inventory and the project as your permissions allow.

Task ***To open the project associated with a given inventory item to access information for all published inventory in the project, do the following:***

1. Open the **Inventory** view. For instructions, see Opening the Inventory View.
2. For the inventory item on the **Inventory** view, click its hyperlinked project name (in the **Project** column). The project opens directly to the **Inventory Items** list on the **Project Inventory** tab (see the following example), enabling you to access and update details for any published inventory item in the project according to your user permissions. You also have access to the entire project to perform any project-related function as needed. For more information, see Project Inventory Tab.



Name		Priority ↑	Vulns	Status
RPMwithRPM 1.0 (GPL-1.0)		P1	1	✓
jetty-http 8.1.15.v20140411 [Bundled with R...		P1	1	?
wrapper 3.2.3 [Bundled with RPMwithRPM 1...		P1	1	✗
jetty-servlet 9.4.5.v20170502 [Bundled with ...		P1	1	✓
jetty-xml 9.4.5.v20170502 [Bundled with RP...		P1	1	?
jetty-io 8.1.15.v20140411 [Bundled with RP...		P1	1	?
jetty-server 9.4.5.v20170502 [Bundled with ...		P1	1	✓
jetty-security 8.1.15.v20140411 [Bundled wit...		P1	1	?

3. To return to the **Inventory** view, click the **Inventory** button under the Code Insight logo on the screen.

Exploring Components and Licenses in the Data Library

Users can look up components and search license details as found in the Code Insight Data Library when they create and edit inventory within a project. However, Code Insight also provides the Global Component & License Lookup feature, which enables users to explore components and licenses in the Data Library outside the context of project inventory.

This type of search might be useful, for example, when a current inventory component is associated with security vulnerabilities. Users can perform a global search on the Data Library to look for components and their versions that might be associated with less severe or no vulnerabilities, thereby helping the user to decide whether to replace the current inventory component with another more secure one.

Global Component & License Lookup feature also enables users to create custom components and licenses that are missing from the Data Library, as well as update these custom elements.

The following sections provide information about this feature:

- Accessing the Global Component & License Lookup Feature
- Exploring Components Globally
- Exploring Licenses Globally

Accessing the Global Component & License Lookup Feature

You initiate a Global Component & License Lookup session from the **Global Component & License Lookup** tab.

Task ***To access the Global Component & License Lookup tab, do the following:***

1. Click the following icon in the upper right corner of the Code Insight web page to open the Code Insight main menu:

   ☰

2. Select **DATA LIBRARY** from the menu to open the **Data Library** page.
3. Select the **Global Component & License Lookup** tab. From this tab, you can search for components and licenses in the Code Insight Data Library (on the **Components** and **Licenses** tabs, respectively) and then explore your search results. See the following sections for more instructions:

   - Exploring Components Globally
   - Exploring Licenses Globally

   Note *▪ By default, the* ***Global Component & License Lookup*** *tab (with the* ***Components*** *tab in focus) is opened when you access the* ***Data Library*** *page.*

Exploring Components Globally

From the **Components** tab on the **Global Component & License Lookup** tab, users can explore the various OSS and third-party components that are standard to the Code Insight Data Library and those that are custom. You can also create a custom component for any component that you discover missing from the Data Library.

Refer to the following sections for more information:

- Setting Up a Global Component Search
- Results of a Global Component Search
- Viewing Information About Individual Components
- Creating a Custom Component to Add to the Data Library
- Adding Changes to a Custom Component in the Data Library

***Note** ▪ If you have just created or edited a custom component, consider waiting a half a minute before performing a component search to ensure that component indexing in the Data Library is complete. This practice can help avoid Code Insight server issues during the search. For more information, see Note About the Indexing Process for Custom Components.*

In addition to the descriptions provided in these sections, see Components Tab for descriptions of all the fields available on **Component** tab.

Setting Up a Global Component Search

From the **Global Component & License Lookup** > **Components** tab (see Accessing the Global Component & License Lookup Feature), use any one of these search mechanisms by which to search for specific OSS and third-party components:

- Keyword
- URL
- Forge
- Component ID

Use the **Forge**, **URL**, or **Component ID** search to obtain the most targeted search results. The **Keyword** search can provide a broader set of results to explore.

If the search finds no results that meet your criterion, a pop-up message is displayed, stating that no results are found.

Keyword

Use the **Keyword** option to search components by their name in the Code Insight Data Library. For the search criterion, enter one or more character strings found in a component name. Multiple strings must be separated with spaces (not commas) and can be listed in any order, as in the following example:

> **tomcat apache**

The search will filter to only those component names that contain *all* the strings entered for the **Keyword** field.

***Note** ▪ The search is case-insensitive, so it filters to all such components, no matter the upper or lower case used in the strings in the **Keyword** field or in the actual component name.*

In general, the name of a component is a unique identifier that can be based on the project, package, or gem name of the component or on another convention such as the component's author or repository. For your reference, the following shows the common conventions used for component names in the various forges:

- **Apache**— <PROJECT_NAME>, for example "apache-batik"
- **Debian**—<PACKAGE_NAME>, for example "0ad"
- **GitLab**—<AUTHOR/ORGANIZATION>-<REPOSITORY_NAME>, for example:
    - "cryptsetup-cryptsetup" (as found in component URL: `https://gitlab.com/`**`cryptsetup`**`/`**`cryptsetup`**)

  - “redhat-bison” (as found in component URL: `https://gitlab.com/redhat/centos-stream/rpms/bison`)

- **NuGet Gallery**— <PACKAGE_NAME>, for example “newtonsoft.json”
- **Pypi**—<PACKAGE_NAME>, for example “hash_ring”
- **RubyGems**— <GEM_NAME>, for example “x-editable-rails”
- **Other**— <PROJECT_NAME>, for example “openssl”

Task ***To search for components by a keyword, do the following:***

1. On the **Global Component & License Lookup** > **Components** tab, select the **Keyword** option.
2. In the **Keyword** field, enter one or more character strings found in the name of the component(s) for which you are searching. For more information about this value, see the introductory content (just above) in this section.
3. Click **Search**.

Note ▪ *If a component that you want to explore is not available with the keyword search, try the URL, forge, or component ID search. If you are still unable to locate the component, the component might not exist in the Code Insight Data Library nor be saved as a custom component.*

URL

If you know the URL of the forge containing the components you want to locate, you can use the **URL** search option. For the **URL** value, you can enter the complete path for the forge, such as **https://github.com/jquery/jquery**, or a string in the path, such as **jquery**.

Task ***To search for components by the forge path, do the following:***

1. On the **Global Component & License Lookup** > **Components** tab, select the **URL** option.
2. In the **URL** field, enter the forge path (or a partial string in the forge path) for the component(s) you want to locate.

   Note ▪ *The search is case-insensitive, so the results will include all components with a matching forge path or path string (whichever criterion you entered in the **URL** field), no matter the upper or lower case used in the criterion or in the actual component path.*

3. Click **Search**.

Forge

Use the **Forge** option to search for a specific component if you know the exact name of its forge and third-party project/repository.

Task ***To search for a specific component by the name of its forge and project/repository, do the following:***

1. On the **Global Component & License Lookup** > **Components** tab, select the **Forge** option.
2. From the **Forge** dropdown list, select the forge that contains the component for which you are searching.
3. In the additional field(s) associated with the forge, identify component’s third-party project/repository.
4. Click **Search**.

Component ID

Use the **Component ID** option to search for a component by its unique ID within Code Insight.

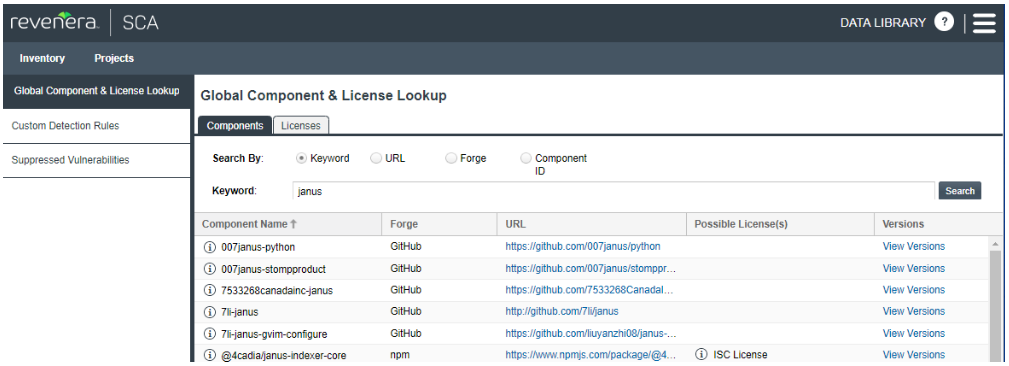
Task ***To search for a specific component by its ID, do the following:***

1. On the **Global Component & License Lookup** > **Components** tab, select the **Component ID** option.
2. In the **Component ID** field, enter the complete ID of the component for which you are searching.

   If you enter a non-integer value, a negative integer, or a value greater than 19 characters, the field is bordered in red and the **Search** button is disabled. You can hover over the value to determine the type of error and then try another value if you want.
3. Click **Search**.

Results of a Global Component Search

Once you have performed a successful search of OSS and third-party components (the setup of which is described in Setting Up a Global Component Search), the results are listed in a grid below your search criterion on the **Global Component & License Lookup** > **Components** tab. (For a complete description of the details shown for the components in these results, see Components Tab.)

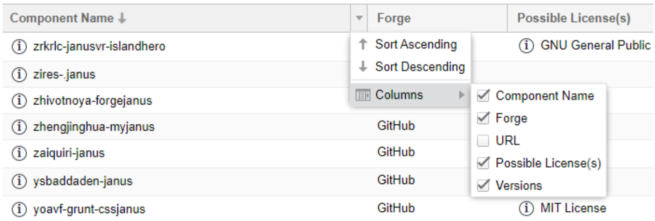

revenera | SCA
DATA LIBRARY
?
Inventory
Projects
Global Component & License Lookup
Custom Detection Rules
Suppressed Vulnerabilities
Global Component & License Lookup
Components
Licenses
Search

Search By: ● Keyword ○ URL ○ Forge ○ Component ID

Keyword: janus

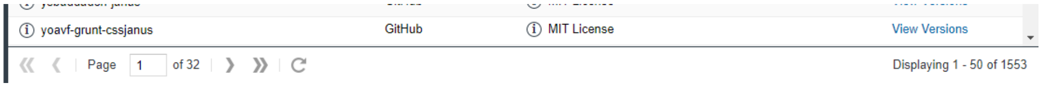
Component Name ↑	Forge	URL	Possible License(s)	Versions
ⓘ 007janus-python	GitHub	https://github.com/007janus/python		View Versions
ⓘ 007janus-stompproduct	GitHub	https://github.com/007janus/stomppr...		View Versions
ⓘ 7533268canadainc-janus	GitHub	https://github.com/7533268Canadal...		View Versions
ⓘ 7li-janus	GitHub	http://github.com/7li/janus		View Versions
ⓘ 7li-janus-gvim-configure	GitHub	https://github.com/liuyanzhi08/janus-...		View Versions
ⓘ @4cadia/janus-indexer-core	npm	https://www.npmjs.com/package/@4...	ⓘ ISC License	View Versions

Note that the components are listed alphabetically by name. To toggle between ascending and descending alphabetic order, click the up ↑ (or down) arrow in the **Component Name** column header (or select the appropriate sorting order from the header’s dropdown list). Also, from the dropdown list located in any column header, you can select columns you want to display or hide in the grid.



Component Name	Forge	Possible License(s)
ⓘ zrkrlc-janusvr-islandhero		ⓘ GNU General Public
ⓘ zires- janus		
ⓘ zhivotnoya-forgejanus		
ⓘ zhengjinghua-myjanus	GitHub	
ⓘ zaiquiri-janus	GitHub	
ⓘ ysbaddaden-janus	GitHub	
ⓘ yoavf-grunt-cssjanus	GitHub	ⓘ MIT License

The list is paginated, enabling you to navigate to the results shown on the next or previous pages or on a specific page number. You can also refresh the entire list to keep it current.



ⓘ yoavf-grunt-cssjanus	GitHub	ⓘ MIT License	View Versions

From this list, you can obtain details about any of the individual components listed, as described in Viewing Information About Individual Components.

Viewing Information About Individual Components

From the list of components resulting from your global search (see Results of a Global Component Search) on the **Global Component & License Lookup** > **Components** tab, you can explore details for any of the individual components shown. Refer to the following sections for ways to mine details for a given component:

- Viewing Details of a Component
- Accessing the Component's Project/Repository Web Page
- Viewing Details of Licenses Associated with the Component
- Viewing Component Versions and Associated Security Vulnerabilities

Viewing Details of a Component

In the search results, you can immediately view important information about each component in the list (see Components Tab for a description of the component information shown). However, you can easily access additional details for a given component, such as its internal Code Insight ID (for executing APIs) and whether the component is custom, supports your product's encryption capabilities, or has associated security vulnerabilities.

Task ***To view the details of a given component, do the following:***

From the list of components resulting from your global search **Global Component & License Lookup** >

Components tab, click the ⓘ icon next to the name of the component you want to research.

The **Component Details** window opens, showing publicly available information about the component. For a description of the component attributes, see Component Details Window.

You can also examine all versions currently available for the component in the Code Insight Data Library and their associated licenses and security vulnerabilities. See Viewing Component Versions and Associated Security Vulnerabilities.

Accessing the Component’s Project/Repository Web Page

You can visit the web page of a component’s third-party project or repository within the forge.

Task ***To access a component’s project/repository web page, do the following:***

From the list of components resulting from your global search on the **Global Component & License Lookup** > **Components** tab, click the hyperlinked URL (in the **URL** column) for a component to access the web page of its third-party project or repository within the forge.

This external web page is opened in a separate browser tab.

Viewing Details of Licenses Associated with the Component

You can view details about any of the licenses that can be associated with the given component.

Task ***To access the possible licenses for a component, do the following:***

From the list of components resulting from your global search on the **Global Component & License Lookup** > **Components** tab, click the ⓘ icon next to a license in the **Possible Licenses** column for a given component.

The **License Details** window for the license is displayed, showing information about the license. For a description of the license attributes, see License Details Window.

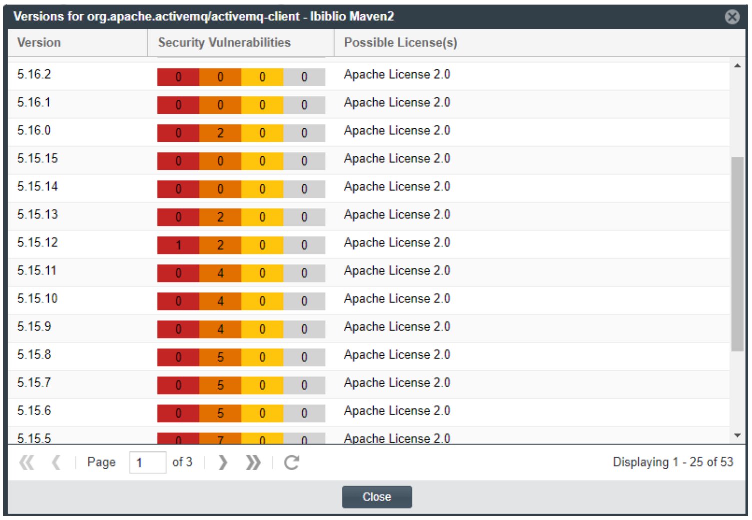
Viewing Component Versions and Associated Security Vulnerabilities

You can view all versions currently stored in the Code Insight Data Library for a given component, along with the possible licenses and security vulnerability totals (by severity) associated with each version. You can also examine details for the associated vulnerabilities and, if necessary, suppress a vulnerability for the version.

Task ***To view the versions of a given component and any security vulnerabilities associated with the versions, do the following:***

From the list of components resulting from your global search on the **Global Component & License Lookup** > **Components** tab, click the **View Versions** icon in the **Actions** columns for a given component.

The **Versions for** <**component**> window is displayed.



Versions for org.apache.activemq/activemq-client - Ibiblio Maven2

Version	Security Vulnerabilities				Possible License(s)
5.16.2	0	0	0	0	Apache License 2.0
5.16.1	0	0	0	0	Apache License 2.0
5.16.0	0	2	0	0	Apache License 2.0
5.15.15	0	0	0	0	Apache License 2.0
5.15.14	0	0	0	0	Apache License 2.0
5.15.13	0	2	0	0	Apache License 2.0
5.15.12	1	2	0	0	Apache License 2.0
5.15.11	0	4	0	0	Apache License 2.0
5.15.10	0	4	0	0	Apache License 2.0
5.15.9	0	4	0	0	Apache License 2.0
5.15.8	0	5	0	0	Apache License 2.0
5.15.7	0	5	0	0	Apache License 2.0
5.15.6	0	5	0	0	Apache License 2.0
5.15.5	0	7	0	0	Apache License 2.0

This window lists each version for the component, along with the version’s possible licenses and security vulnerability totals. For each version that has no associated vulnerabilities, the value **None** is displayed in the **Security Vulnerabilities** column. However, if a version *is* associated with vulnerabilities, a **Vulnerabilities** bar graph is displayed in the column, showing the version’s vulnerability totals by severity. For instructions about interacting with this graph to examine details about each associated vulnerability and, if needed, to suppress vulnerabilities for the version, see Examining Security Vulnerability Details.

For additional information about this window, see Versions for <component> Window.

Creating a Custom Component to Add to the Data Library

While using the **Global Component & License Lookup** tab to explore the third-party or OSS components in the Code Insight Data Library, you might discover that a specific component is not found in the Data Library. You can use the **Create New Component** button on the **Components** tab to create a custom component for the missing component, which in turn adds it to the Code Insight database and, as a background process, adds it to the Code Insight Data Library. For complete instructions, refer to Creating a Custom Component.

Adding Changes to a Custom Component in the Data Library

From the **Components** tab the **Global Component & License Lookup** tab, you can edit any component that has been manually created. (Such a component is displayed with an **Edit Component** icon in the **Actions** column.) For complete instructions, refer to Editing a Custom Component.

Exploring Licenses Globally

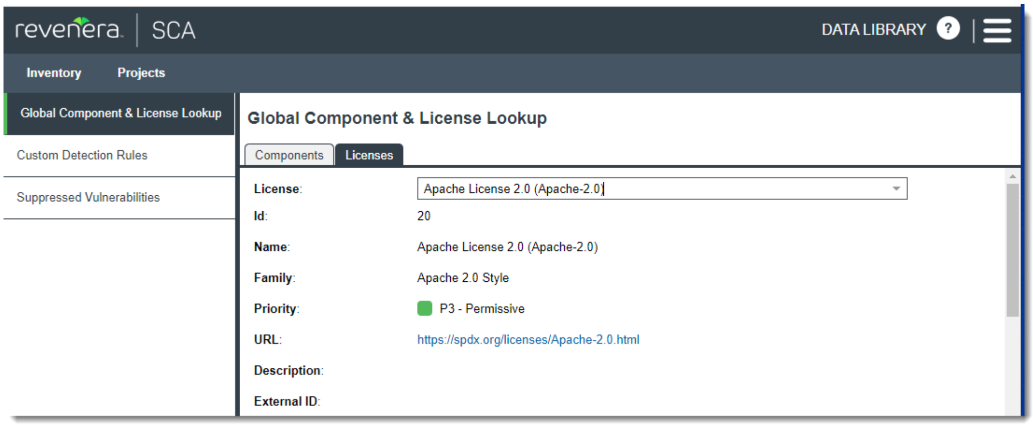
The **Global Component & License Lookup** > **Licenses** tab enables users to search for a specific license that is standard to the Code Insight Data Library or that is custom.

From the **Licenses** tab, you can also create a custom license for any license that you discover missing from the Data Library (see Creating a Custom License to Add to the Data Library).

Task ***To look up a license in the Data Library, do the following:***

1. Open the **Global Component & License Lookup** > **Licenses** tab. See Accessing the Global Component & License Lookup Feature.
2. Select the desired license from the **License** dropdown list. Alternatively, enter a string found in the license name to filter the list, and then select the license from the filtered list.

   The **Licenses** tab is populated with the attributes for the selected license.


revenera | SCA
DATA LIBRARY
Inventory
Projects
Global Component & License Lookup
Custom Detection Rules
Suppressed Vulnerabilities
Global Component & License Lookup
Components
Licenses

License: Apache License 2.0 (Apache-2.0)
Id: 20
Name: Apache License 2.0 (Apache-2.0)
Family: Apache 2.0 Style
Priority: 🟩 P3 - Permissive
URL: https://spdx.org/licenses/Apache-2.0.html
Description:
External ID:

3. Scroll down to view all attributes for the license, including the license text. For a description of these attributes, see Licenses Tab.
4. To research another license, repeat steps 2 and 3.

Creating a Custom License to Add to the Data Library

During during your exploration of the third-party or OSS licenses in the Code Insight Data Library, you might discover that a specific license is not found in the Data Library. You can use the **Create New License** button on the **Licenses** tab to create a custom license for the missing license. Once saved, the custom license is added to the Code Insight database. For complete instructions, refer to Creating a Custom License.

Accessing Projects in Code Insight

This section describes the various ways you can access projects in your Code Insight system in order to view their details and manage them:

- Opening the Projects View
- Showing Only Your Projects
- Searching Across All Projects in the System
- Using the Project Dashboard
- Opening a Project
- Managing Items in the Projects Display

Opening the Projects View

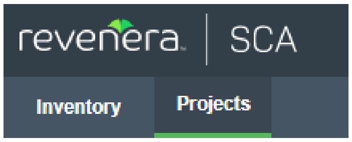
All Code Insight projects are created, accessed, and managed in the **Projects** view. This view shows a manageable list of the projects currently available in the system. From this list you open individual projects to assess scan their results, edit their details, and finalize their inventory of open-source and third-party software.

Use the following procedure to open the **Projects** view. The procedure assumes that you have logged into Code Insight.

Task ***To open the Projects view, do the following:***

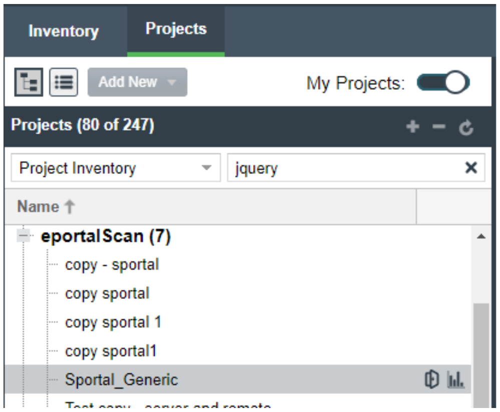
Open the **Projects** view using one of these methods:

- From the Code Insight dashboard (which is displayed when you start Code Insight), click **go to project**. See Opening Code Insight for details on accessing this dashboard.
- On the Code Insight web page, click the **Projects** button under the Code Insight logo:



- Click the ☰ icon in the upper right corner of the Code Insight web page to open the Code Insight main menu. Select **PROJECTS** from the menu.

The **Projects** pane on the left side of the **Projects** view lists the projects in Code Insight. (The projects are listed in a tree or list format, depending on your configuration. To toggle to the other format, see Managing Items in the Projects Display.)



From this view you can do the following (depending on your project role in some cases):

- Filter the projects by those with which you are associated as Project Contact or through a project role (see Showing Only Your Projects).
- Filter the projects by project name, name of associated inventory, or security vulnerability, as described in Searching Across All Projects in the System.
- View the Code Insight scan statistics for a specific project (see Using the Project Dashboard).

- Create or delete projects (Creating a Project or Deleting a Project).
- Toggle the **Projects** pane between tree view and list view (see Managing Items in the Projects Display).
- Move projects to different folders (Managing Items in the Projects Display).
- Create, delete, and move folders (Managing Items in the Projects Display).
- Open a project to manage it, assess scan results, and finalize its inventory (see Opening a Project).

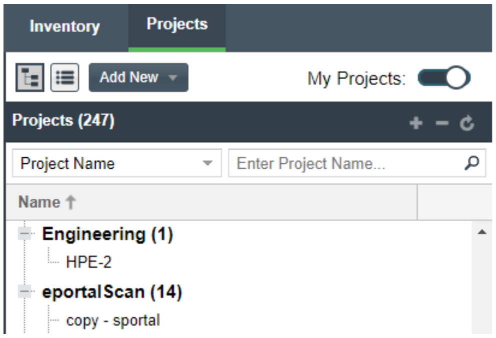
Showing Only Your Projects

Code Insight provides the option to filter the list of projects to show only those projects with which the current user is associated as either Project Contact, Project Administrator, Analyst, Reviewer, or Observer. (For a description of these roles, see Assigning and Removing Project Users.)

Additionally, this filter can work in conjunction with the system filters described in Searching Across All Projects in the System to display only your projects that have specific project or inventory attributes.

Task ***To show your projects only, do the following:***

1. Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.)
2. At the top of the **Projects** pane, click the **My Projects** toggle.



3. Select a project from the filtered list of projects to open its dashboard. See Showing Only Your Projects for details.

   Alternatively, open the project to view its inventory on the **Project Inventory** tab. See Opening a Project for details.
4. (Optional) To turn off this filter, click the **My Projects** toggle again.

Searching Across All Projects in the System

Using the search filters available in the **Projects** pane, you can search projects across the Code Insight system by name, inventory, components and versions, licenses, and security vulnerabilities. You can perform multiple types of these global searches, each one filtering deeper into the current search results. Then, at the project level, you can continue to narrow your search results to specific inventory items for the given project.

The following topics describe these search methods:

- Available Filters for Searching Across Projects
- Searching for Projects by Project Name
- Searching for Projects That Include Inventory Based on a Specific Component or Component Version
- Searching for Projects That Include Inventory Associated with a Specific License
- Searching for Projects That Include Inventory Impacted by a Specific Security Vulnerability
- Restoring the Full Project Tree or List

Available Filters for Searching Across Projects

The following filters are available from the **Projects** pane to perform searches across all projects in your Code Insight system.

Table 2-17 ▪ Available Filters for Searching Across Projects

To search across all projects for...	...use this filter	Filter criterion	Criterion example	Refer to...
Projects with a name containing a specific string	Project Name	Project name	`MyProject`	Searching for Projects by Project Name
Inventory based on a specific component and version	Project Inventory	Component and version as it appears in the **Inventory Name** value	`Apache Struts 2.3.14.3`	Searching for Projects That Include Inventory Based on a Specific Component or Component Version
		Component name as it appears on the **Component Details** tab for the inventory item	`struts2-core`	
Inventory associated with a specific license	Project Inventory	The **Selected License** value as it appears on the **Component Details** tab for the inventory item	`GNU General Public License v2.0`	Searching for Projects That Include Inventory Associated with a Specific License
		The license name as it appears in the **Inventory Name** value	`GNU General Public License` or `GPL-2.0+`	
		The SPDX short identifier for the license	`GPL-2.0+`	

Table 2-17 ▪ Available Filters for Searching Across Projects (cont.)

To search across all projects for...	...use this filter	Filter criterion	Criterion example	Refer to...
Inventory impacted by a specific security vulnerability	Security Vulnerability	The complete ID of the security vulnerability	`CVE-2018-11776` for an NVD vulnerability `SA40575` for a Secunia advisory `DSA-4315` for a Debian advisory	Searching for Projects That Include Inventory Impacted by a Specific Security Vulnerability

Searching for Projects by Project Name

You can use the **Project Name** search filter available in the **Projects** pane to search for projects by a full or partial project name.

Rules When Performing This Search

When you search for projects by project name, the following rules apply:

- The name string value you enter is case-insensitive.
- All characters in the search string must be consecutive.
- A full or partial string value is supported as a search criterion.
- The string can contain any characters (letters, numbers, and special characters).

How to Perform This Search

This procedure shows how to search for projects by a full or partial name string.

Task ***To search for projects by name, do the following:***

1. Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.)
2. At the top of the **Projects** pane, select **Project Name** from search dropdown list on the left.
3. Enter the project name (or a partial string in the name) in the **Enter Project Name** field. The list of projects changes to reflect the search results, and a filtered count (for example, “(19 of 123)”) is provided in the **Projects** pane header to show the number of projects returned by the search.

   If no inventory items meet the specified criterion, the **Projects** pane shows “No Projects”.
4. Select a project from the filtered list to open its dashboard. See Showing Only Your Projects for details.

   Alternatively, open the project to view its inventory displayed on its **Project Inventory** tab. See Opening a Project for details.

Searching for Projects That Include Inventory Based on a Specific Component or Component Version

You can use the **Project Inventory** filter to search for those projects that include one or more inventory items based on a specific component or component version. The search returns a filtered list of projects; and, when a project in the list is opened, its inventory is also filtered by the search criterion.

This search method saves you time in locating specific inventory items that require attention across all projects. For example, you might also use this search method to pinpoint those projects containing inventory items affected by a recent component upgrade.

You can also use this search method to easily locate projects that contain inventory items impacted by a security vulnerability whose exact ID you do not know; you can search instead for projects with inventory based on a component and version known to be affected by the vulnerability. (This search method is an alternative to using the **Security Vulnerability** filter, which requires the exact vulnerability ID as the search criterion. See Searching for Projects That Include Inventory Impacted by a Specific Security Vulnerability.)

Rules When Performing This Search

When you search for projects that include inventory based on a specific component or component version, the following rules apply.

- A full or partial string value is supported as search criterion.
- All characters in the search string must be consecutive.
- The string value is case-insensitive and can contain letters, numbers, hyphens, and special characters. Spaces are also allowed.
- Only inventory items on the **Project Inventory** tab are supported in the search.

How to Perform This Search

Use this procedure to locate those projects that include inventory based on a specific component or component version.

Task ***To search for projects that include inventory based on a specific component or component version, do the following:***

1. Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.)
2. At the top of the **Projects** pane, select **Project Inventory** from search dropdown list on the left.
3. In the **Enter Search Criteria** field, specify the complete name or a partial string for one of the following:
    - The name of the component as it appears on the **Component Details** tab (for example, `struts2-core`) for an inventory item
    - The name of the component and version as it appears in the **Inventory Name** value (for example, `Apache Struts 2.3.14.3`)

   The list of projects changes to reflect the search results, and a filtered count (for example, “(19 of 123)”) is also provided in the **Projects** pane header to show the number of projects returned by the search.

4. Open one of the projects to see a filtered list of inventory items that contain the component or component and version. (See Opening a Project for details.)

Searching for Projects That Include Inventory Associated with a Specific License

You can use the **Project Inventory** filter to search for those projects that include one or more inventory items associated with a specific license. The search returns a filtered list of projects; and, when a project in the list is opened, its inventory is also filtered by the search criterion.

This search method saves you time in locating inventory items with license-related issues, such as those items that are associated with a high-risk license, across all projects.

Rules for Performing This Search

When you search for projects that include inventory associated with a specific license, the following rules apply:

- A full or partial string value is supported as search criterion.
- All characters in the search string must be consecutive.
- The string value is case-insensitive and can contain letters, numbers, hyphens, and special characters. Spaces are also allowed.
- Only inventory items on the **Project Inventory** tab are supported in the search.

How to Perform This Search

Use this procedure to locate those projects that include inventory associated with a specific license.

Task ***To search for projects that include inventory associated with specific license, do the following:***

1. Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.)
2. At the top of the **Projects** pane, select **Project Inventory** from search dropdown list on the left.
3. In the **Enter Search Criteria** field, specify the complete name or a partial string for one of the following:
    - The name of the **Selected License** as it appears on the **Component Details** tab (for example, `GNU General Public License v2.0`) for an inventory item
    - The license SPDX short identifier (for example, `GPL-2.0+`)
    - The license name as it appears in the **Inventory Name** value (for example, `GNU General Public License` or `GPL-2.0+`)

    The list of projects changes to reflect the search results, and a filtered count (for example, “(19 of 123)”) is also provided in the header on the **Projects** pane to show the number of projects returned by the search.
4. Open one of the projects to see a filtered list of inventory items that contain the license. (See Opening a Project for details.)

Searching for Projects That Include Inventory Impacted by a Specific Security Vulnerability

You might find it sometimes necessary to see how a specific security vulnerability impacts your organization. You can do this quickly by searching for all projects that include one or more inventory items impacted by the security vulnerability or advisory. The search returns a filtered list of projects; and, when a project in the list is opened, its inventory is also filtered by the search criterion.

Perform the search in one of the following ways:

- **If you know the exact ID of the security vulnerability or advisory**—Use the **Security Vulnerability** search filter with the *exact* security vulnerability ID as the search criterion, as described in this section.
- **If you do not know the ID of the security vulnerability or advisory**—Use the **Project Inventory** search filter to provide the name of the vulnerable component as the search criterion. See Searching for Projects That Include Inventory Based on a Specific Component or Component Version for details.

Note ▪ *A vulnerability or advisory might not have an ID, for example, in the case of a zero-day vulnerability for which an ID has not been published.*

Rules When Performing This Search

When you use the **Security Vulnerability** search filter to search for those projects that include inventory impacted by a specific security vulnerability, the following rules apply:

- Only one vulnerability ID can be specified as a search criterion.
- Only exact matches of the full vulnerability ID string are supported. Partial strings are not supported.
- The string you enter does not support spaces.
- Only published inventory items are searched.
- The search ignores inventory associated with a component version for which the vulnerability has been suppressed.
- The search does not validate the vulnerability ID you enter. If you enter an invalid ID, no results are returned in the **Projects** pane.

How to Perform This Search

Use this procedure to locate projects that include inventory impacted by the *exact* security vulnerability ID you specify.

Task ***To search for projects with inventory impacted by a specific security vulnerability, do the following:***

1. Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.)
2. At the top of the **Projects** pane, select **Security Vulnerability** from search dropdown list on the left.
3. In the **Enter Vulnerability ID** field, specify the complete ID of the vulnerability (for example, `CVE-2018-11776` for an NVD vulnerability).

4. Press Enter. The list of projects changes to reflect the search results, and a filtered count (for example, "(19 of 123)") is also provided in the header on the **Projects** pane to show the number of projects returned by the search.

   If no inventory items meet the specified criterion, the **Projects** pane shows "No Projects".

5. Open one of the projects to see a filtered list of inventory items that are impacted by the security vulnerability. (See Opening a Project for details.)

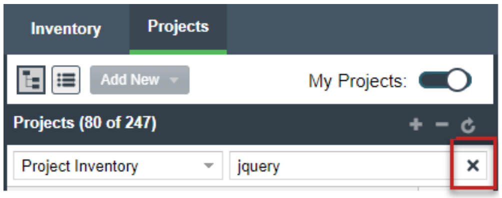
Restoring the Full Project Tree or List

Use this step to remove the current filter in effect in **Projects** pane to restore all projects in the project tree or list.

Task ***To restore the full list of projects, do the following:***

1. Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.)

2. Click the ✖ icon in the criterion field to remove the current **Projects** pane filter:



   The project tree or list is restored to show all projects.

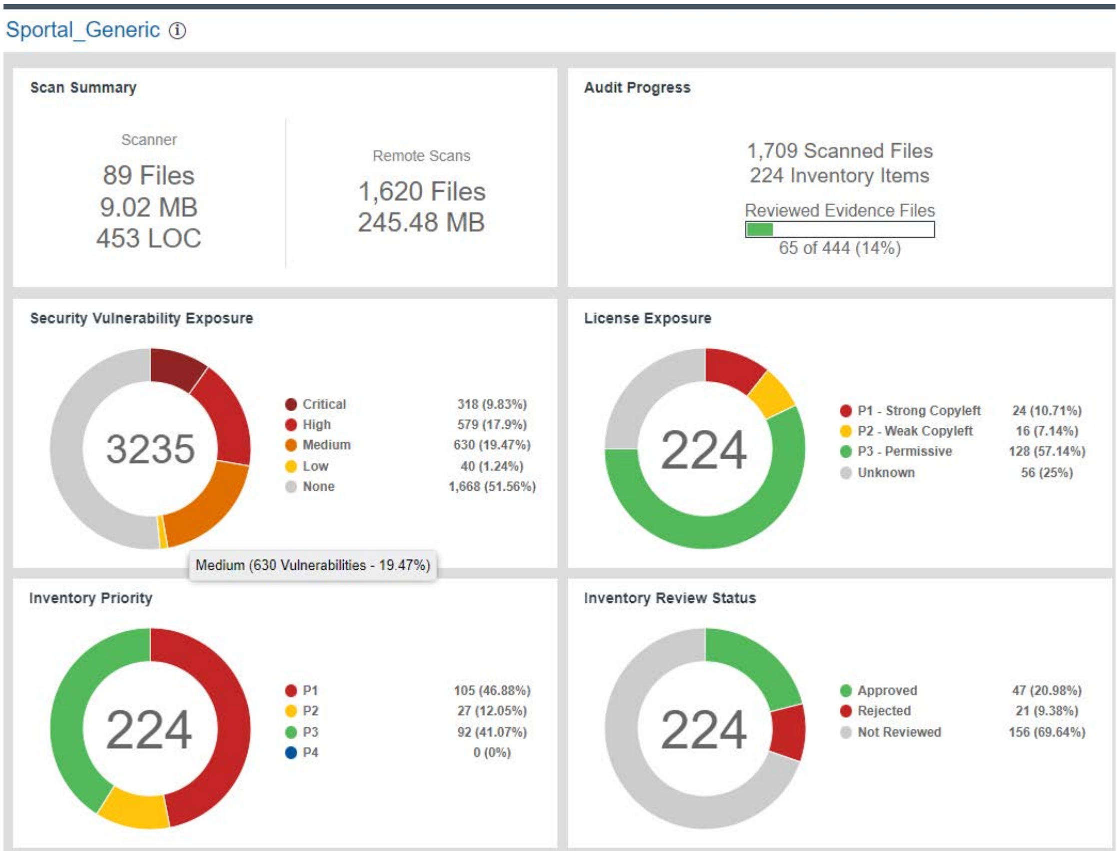
Using the Project Dashboard

When you select a project in the **Projects** pane, the project's dashboard is displayed, providing you with an interactive view of your project, including security-vulnerability and license exposure, codebase and inventory review statistics, and other information. The following topics describe this dashboard.

- Project Dashboard
- Navigating the Project Dashboard Features

Project Dashboard

The following image shows the dashboard for an example project.



Sportal_Generic ⓘ

Scan Summary

Scanner
89 Files
9.02 MB
453 LOC

Remote Scans
1,620 Files
245.48 MB

Audit Progress

1,709 Scanned Files
224 Inventory Items
Reviewed Evidence Files
65 of 444 (14%)

Security Vulnerability Exposure

3235

● Critical	318 (9.83%)
● High	579 (17.9%)
● Medium	630 (19.47%)
● Low	40 (1.24%)
● None	1,668 (51.56%)

Medium (630 Vulnerabilities - 19.47%)

License Exposure

224

● P1 - Strong Copyleft	24 (10.71%)
● P2 - Weak Copyleft	16 (7.14%)
● P3 - Permissive	128 (57.14%)
● Unknown	56 (25%)

Inventory Priority

224

● P1	105 (46.88%)
● P2	27 (12.05%)
● P3	92 (41.07%)
● P4	0 (0%)

Inventory Review Status

224

● Approved	47 (20.98%)
● Rejected	21 (9.38%)
● Not Reviewed	156 (69.64%)

Navigating the Project Dashboard Features

This procedure walks you through the dashboard features and explains how to use their interactive capabilities.

Task ***To use the project dashboard, do the following:***

1. Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.)
2. In the **Projects** pane on the left, click a project in the list of projects. The dashboard for the selected project is displayed in the right panel. (Alternatively, hover over the project entry in the list, and click the **Load Project Dashboard** icon 📊 in the entry to display the dashboard.)

   The project dashboard contains the following charts to provide an overview of the project's most recent scan and the resulting inventory:

   - **Scan Summary**—A summary of your most recent **Scanner** scan (that is, server scan) and most recent **Remote Scans**. If multiple scan-agent plugins are used for remote scanning, the **Remote Scans** summary shows combined totals from the most recent scans of all the agents.

     If only a server scan has occurred, the tile shows only **Scanner** totals. Likewise, if only remote scans have occurred, the tile shows only **Remote Scans** totals.

   - **Audit Progress**—For a **Scanner** (server) scan, a snapshot of the audit progress that users have made on those files containing OSS or third-party evidence. No audit progress is shown for remote scans.

The tile also shows the total number of scanned files and resulting inventory items for the project.

- **Security Vulnerability Exposure**—An interactive color-coded chart depicting percentages of security vulnerabilities by severity across all inventory in the project. You can hover over a color segment in the chart to see an actual number and percentage. To the right of the chart, a color legend also shows the total number and percentage of vulnerabilities of a given severity. The number in the center of the chart is the total number of security vulnerabilities found across all inventory.

  The colors in this chart indicating severity can vary depending on the CVSS version that Code Insight is using.

  Additionally, the counts in this chart do not include vulnerabilities are currently suppressed. See Working with Security Vulnerabilities for details.)

- **License Exposure**—An interactive color-coded chart depicting percentages of project inventory grouped by the *priority of the license* associated with the inventory. (License priority ranks licenses by their degree of impact on your organization's proprietary policies.) You can hover over a color segment in the chart to see an actual number and percentage. To the right of the chart, a color legend also shows the total number and percentage of inventory items associated with a given license priority. The number in the center of the chart is the total number of inventory items identified for the current project. For more information about license priority, see License Priority.

- **Inventory Priority**—An interactive color-coded chart depicting percentages of inventory grouped by *inventory priority* in the current project. (Inventory priority identifies which inventory items need more attention than others in the inventory review process; it is derived from an algorithm based in part on an item's associated license priority and security vulnerabilities.) You can hover over a color segment in the chart to see an actual number and percentage. To the right of the chart, a color legend also shows the total number and percentage of inventory items that are assigned the given inventory priority. For more information, see Inventory Priority.

- **Inventory Review Status**—An interactive color-coded chart that provides a visual of the how many inventory items have been approved, rejected, or not reviewed in the project. You can hover over a color segment in the chart to see an actual number and percentage. To the right of the chart, a color legend also shows the total number and percentage of inventory items that are assigned the given review status.

3. (Optional) Click a legend item or color segment in a chart to open the project to only those inventory items associated with the segment or legend item. See Opening a Project for details.

   Alternatively, you can open the project to view all its inventory (see Opening a Project), or select another project from the list of projects in the **Projects** pane to view the dashboard for that project.

Filtering Inventory for a Project from the Project Dashboard

After you create a project, and upload and scan a codebase, you can quickly filter the project's inventory to view potential problems and take steps to eliminate issues, such as high exposure items (shown in red), from your project inventory.

Task ***To quickly filter inventory on the project dashboard, do the following:***

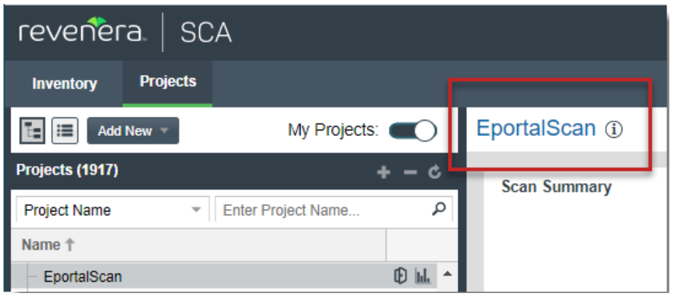
1. Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.)
2. In the **Projects** pane on the left, click a project in the list of projects. The dashboard for the selected project is displayed in the right panel.
3. In the project dashboard, navigate to the desired chart, and click a color-coded segment in the chart (or click a legend item next to the chart). The project is opened to its **Project Inventory** tab, displaying only those project inventory items associated with the information represented by the chart segment or legend item.
4. Click on a project inventory item listed to see more detail in the right pane of the **Project Inventory** tab. See Reviewing Published Inventory for a Project for details about this tab.

Opening a Project

Use this basic procedure to open a project.

Task ***To open a project, do the following:***

1. Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.)
2. In the **Projects** pane on the left, click a project in the projects display. The dashboard for the selected project is displayed in the right panel.
3. To open the project, either click the **Open Project** icon next to the project in the projects display, or click the project's name link in the upper left corner of the project dashboard (shown below).



The project is opened on either of the following tabs for the project:

- The **Project Inventory** tab if the project contains published inventory items. For more information about the **Project Inventory** tab, see Reviewing Published Inventory for a Project.
- The project's **Summary** tab if the project does not contain published inventory items. For more information about the **Summary** tab, see Managing Projects. For information about publishing inventory, see Publishing or Recalling Inventory Manually from the Analysis Workbench.

Note that the **Analysis Workbench** tab is also available for users with the proper permissions (although you have to navigate to open it). The Analysis Workbench enables a user to perform a deep analysis of the scan results. For more information, see The Analysis Workbench Layout.

Managing Items in the Projects Display

The following procedures describe how to manage the format of the project list:

- Access the Display of Projects
- Select the Projects Display Format
- Manage Items in the Project Tree Format
- Manage Items in the Plain List Format

Access the Display of Projects

This procedure describes how to access and to manage the display of available Code Insight projects.

Task ***To access the projects display, do the following:***

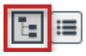
Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.) The display of projects is in the **Projects** pane on the left.

Select the Projects Display Format

The following describes how to toggle between project tree or plain list format for the project list. By default, the projects are in tree format.

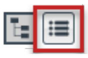
Task ***To select a format for the projects display, do either:***

To display the projects in tree format, click this icon at the top of the **Projects** pane:



By default, the projects are displayed in the tree format.

To display the projects in a plain list format, click this icon:



Manage Items in the Project Tree Format

The following describes ways to manage items (projects and folders) in the project tree format.

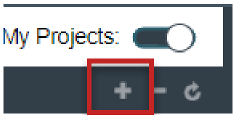
- Expanding or Collapsing All Folders
- Managing Items From the Right-Click Menu

Expanding or Collapsing All Folders

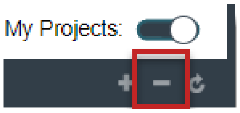
You can easily expand or collapse all folders in the project tree. By default, all folders are collapsed.

Task ***To expand or collapse all folders in the project tree, do the following:***

To expand all folders in the tree, click the **Expand All Folders** icon at the top of the **Projects** pane:



To collapse all folders in the tree, click the **Collapse All Folders** icon:



Managing Items From the Right-Click Menu

The right-click menu options described in this section are enabled or disabled depending on the permissions granted the current user.

Task ***To manage the items in the project tree from the right-click menu, do the following:***

Perform any of these tasks on an item in the project tree:

- **Rename a project** —(Project Administrator only) Double-click the project name and overwrite the current name with the new name.
- **Move a project to a different folder**—(Project Administrator only) Drag and drop the project to the desired folder.
- **Delete a project**—(Project Administrator only) Right-click the project and select **Delete Project**. See Deleting a Project for further details.
- **Create a project**—(User with Create Project permission) To create a project under a specific folder, right-click that folder or any project directly under that folder, and select **Create New Project | At This Level**. To create a project at the root level of the project tree, right-click anywhere in the tree, and select **Create New Project | At Root Level**.

  See Creating a Project for further details.
- **Create a folder**—(User with Create Project permission) To create a folder under a specific folder, right-click that folder or any project directly under that folder, and select **Create New Folder | At This Level**. To create a folder at the root level of the project tree, right-click anywhere in the tree, and select **Create New Folder | At Root Level**.
- **Delete a folder**—(User with Create Project permission) Right-click the folder and select **Delete Folder** (or click ✕ to the right of the folder). Any sub-folders under this folder are also deleted. Any projects under the deleted folder are moved to the parent folder or to the root of the project tree.

Manage Items in the Plain List Format

The following describes ways to manage items (projects and folders) in the plain list format.

Task ***To manage a project list in plain list format, do the following:***

Perform any of these tasks on an item in the project list:

- **Rename a project** —(Project Administrator only) Double-click the project name and overwrite the current name with the new name.
- **Delete a project**—(Project Administrator only) Right-click the project and select **Delete Project**. See Deleting a Project for further details.
- **Create a project**—(User with Create Project permission) Right-click any project in the list, and select **Create a New Project**. See Creating a Project for further details.

Managing Projects

The following topics describe how to use features on the **Summary** tab to manage the currently opened project. (Refer to the Code Insight User Roles and Permissions section for the various user roles required manage projects.)

- Opening the Project Summary Tab
- Success Messages When Working with Projects
- Assigning and Removing Project Users
- Editing the Project Definition and General Settings
- Setting Policies to Publish Inventory Automatically
- Updating Scan Settings for a Project
- Updating Inventory Review and Remediation Settings for a Project
- Connecting the Project to Remote Data Sources
- Identifying Child Projects for a Project
- Completing Custom Fields for the Project
- Assigning the Project to an SBOM Insights Bucket
- Changing the Project Contact
- Updating Third-Party Notices Across Inventory for a Project
- Rescanning Your Codebase (Server Scans Only)
- Exporting Project Data
- Importing Project Data
- Exporting Project Inventory to SBOM Insights
- Branching a Project

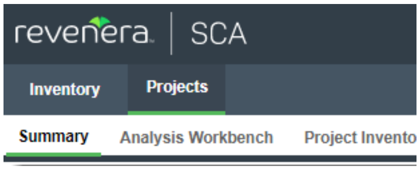
- Copying a Project
- Creating a Private Project
- Forcing an Automatic Review of All Inventory in a Project
- Deleting a Project

Opening the Project Summary Tab

The **Summary** tab for a given project displays important information about the project and provides access to the functionality used to manage the project.

Task ***To open the Summary tab for a given project, do the following:***

1. Open a project in the **Projects** view. (For instructions, see Opening a Project.)
2. Click the **Summary** button at the top of the window to open the **Summary** tab for the project.



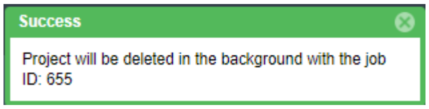
For a description of the fields and functionality available on the **Summary** tab, see Summary Tab.

Success Messages When Working with Projects

When you initiate a certain project operation (for example, a project deletion, notices update, or import among others), a message box is displayed in the upper right corner of the Code Insight user interface to inform you of one of the following:

- The operation has successfully completed.
- The operation has been successfully initiated as a background job in the **Jobs** queue. The message includes the job ID so that you can track the job (as described in Monitoring the Code Insight Jobs Queue).

The message persists for a couple of seconds, but you can click the ⊗ button in the box to close the message sooner.



Project will be deleted in the background with the job ID: 655

For more information about the Jobs queue, see Monitoring the Code Insight Jobs Queue.

Assigning and Removing Project Users

The Project Administrator assigns roles to users, enabling them analyze the codebase and manage and publish inventory, review published inventory, or view private projects. The Project Administrator can also create other Project Administrators.

The following are the available roles that users can have in a project:

- **Project Administrators** manage project users, manage project settings, upload and scan codebases, and rename, branch, and delete projects. They also manage Manage Source Control Management (SCM) and Application Lifecycle (ALM) instances.
- **Analysts** manage the codebase and inventory using the Analysis Workbench. They can upload and scan codebases, review files and add files to inventory, create new inventory, edit existing inventory, and publish and recall inventory. Analysts can also create and edit project inventory on the Project Inventory tab.
- **Reviewers** use the Project Inventory tab to approve and reject inventory, recall inventory, set inventory priority, and edit third-party notices and audit/guidance notes for the inventory.
- **Observers** can view inventory in a private project. They have read-only access to project inventory and can run reports. Development managers and executives are usually assigned this role. The Observer role is available for private projects only.

Note* ▪ *Private projects are hidden from all users except the Project Contact and those users assigned as Project Administrators, Analysts, Reviewers, or Observers of the project. For additional information about private projects, see Creating a Private Project.

For a reference to the various user roles and their permissions, refer to the Code Insight User Roles and Permissions section.

The following procedure describes how to assign users to project roles and remove users from these roles.

Task ***To assign users to or remove them from project roles, do the following:***

1. As the Project Administrator, navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Edit Project Users**. The **Edit project users** page appears.

   Note that all users assigned to a given role are shown on this page, whether the user was manually assigned the role or had inherited the role (for example, through project migration, designation as Project Contact, or Project Defaults set up by the System Administrator).
3. Do either of the following:
   - To assign users to a given role, drag and drop one or more user names from the **Select Users** list to the desired “role” pane (**Project Administrators**, **Analysts**, **Reviewers**, or **Observers**). Repeat this step a necessary. (A user can be assigned to multiple roles.)
   - To remove a user from a role, click ✖ next to the user’s name in the appropriate “role” pane. You can remove any user from a role, even those who inherited the role.

***Note* ▪ The *Observers* pane is visible for only private projects.**

4. Click **Close** when you have finished managing the project users.

Editing the Project Definition and General Settings

The Project Administrator can edit the project's definition and general settings.

Task ***To update project settings, do the following:***

1. As the Project Administrator, navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** window opens.
3. Select the **General** tab.
4. Update the fields as needed. Refer Edit Project: General Tab to for field descriptions.
5. Click **Save** to save the changes.

Updating Scan Settings for a Project

You can update the scan configuration by switching the project to a different scan profile, update which sub-folders to scan, and change settings for automatically publishing inventory during the scan (see Setting Policies to Publish Inventory Automatically).

See also the Edit Project: General Tab to configure the project setting that determines whether the scan retains inventory that has no files associations.

Task ***To update scan settings for the project, do the following:***

1. As the Project Administrator, navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** window opens.
3. Select the **Scan Settings** tab.
4. Update the fields as needed. Refer Edit Project: Scan Settings Tab for field descriptions.
5. Click **Save** to save the changes.

Setting Policies to Publish Inventory Automatically

Code Insight provides the ability to automatically publish inventory without the need for an analyst to be involved. This feature supports a fully automated end-to-end process where there is no human analyst involvement. (For example, the auto-publish feature works in conjunction with workflow policies that automatically review inventory items as they are published, as described in Managing Policies to Automatically Review Inventory.) If there is a human analyst involved, the auto-publish feature can be turned off, allowing the analyst to publish the inventory manually after analysis.

The following configuration defines the auto-publish feature for scans and rescans performed on the current project. When the auto-publish feature is enabled for the project, additional options are made available to do the following:

- Set the minimum inventory confidence level for publishing inventory.
- Determine whether to automatically mark files associated with an auto-published inventory as "reviewed".
- Determine whether to publish inventories with undetermined licenses (that is, their selected **License** value is **I don't know**).

***Note* ▪ Aside from this configuration for project scans, the auto-publish feature is also automatically applied when you create an inventory item from the *Project Inventory* tab.**

Task ***To set the auto-publish feature, do the following:***

1. As the Project Administrator, navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** window opens.
3. Select the **Scan Settings** tab.
4. Enable or disable **Automatically publish system-created inventory items**. When you enable this option, additional auto-publish options are made available for configuration. See Edit Project: Scan Settings Tab for a description these options.
5. When you have set the auto-publish feature, click **Save**. The **Summary** tab is opened.

Updating Inventory Review and Remediation Settings for a Project

You can overwrite default settings that configure the automation of the review, remediation, and status notification processes for published inventory in your project. These settings, which work in conjunction with the set of policies in the project's policy profile, are used to set up the following in your project:

- The policy profile to associate with the project. The policies in the selected profile work in conjunction with the review, remediation, and notification configuration defined on this tab.
- Automatic creation of manual review tasks for inventory items not reviewed by policy during publication performed as part of a scan. The tasks are automatically assigned to the default legal or security contact that you specify.

- Automatic creation of remediation tasks and associated external work items for inventory that is rejected either automatically by policy or during manual publication by an analyst. The tasks are automatically assigned to the default engineering contact that you specify.
- Automatic rejection of published inventory impacted by new vulnerabilities detected in the latest scan or Electronic Update.
- The automatic generation of email notifications only (instead of assigned tasks), which are sent to the Project Contact as alerts concerning the rejected or non-reviewed published inventory items.

Task ***To update settings that automate review, remediation, and status notification processes for published inventory, do the following:***

1. As the Project Administrator, navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, **Edit Project**. The **Edit Project** window opens.
3. Select the **Review and Remediation Settings** tab.
4. Update the fields as needed. Refer Edit Project: Review and Remediation Settings Tab to for field descriptions.
5. Click **Save** to save the changes.

Connecting the Project to Remote Data Sources

If your system is configured to connect to a remote data source, you will have access to update the following:

- Configuring Synchronization of Remote Data Sources to the Scan Server
- Associating the Project with an Application Life Cycle System to Create Work Items

Configuring Synchronization of Remote Data Sources to the Scan Server

Use the **Version Control Settings** tab on the **Edit Project** window to synchronize one or more Source Code Management (SCM) repositories to the Scan Server for your project so you can scan and audit code without manually moving that data to the server. For information about connecting to remote data sources, see Configuring Source Code Management.

Associating the Project with an Application Life Cycle System to Create Work Items

Code Insight integrates application lifecycle management (ALM) systems (also known as issue-tracking systems), enabling Code Insight users to create and manage external work items associated with inventory directly from Code Insight. With this integration, inventory requiring further review and remediation can be tracked externally as part of the user’s existing issue-tracking system.

For example, a Code Insight scan might uncover security vulnerabilities or copyleft licenses requiring further review by the Security and Legal teams. With an ALM integration, these reviews and any resulting remedial work can be quickly converted into work items in the ALM system directly from Code Insight. Users can also track the state of these work items—and open the items—from Code Insight.

Currently, Code Insight supports integration with only Jira as an ALM system. This chapter focuses the Project Administrator's task of configuring a Code Insight project for Jira integration so that the project's users can create and track issues on the Jira system from Code Insight.

The following topics provide the background and process for configuring projects to integrate with Jira:

- About the Jira Connector and Instances
- Associating a Code Insight Project with a Jira ALM instance
- Disassociating an Jira ALM Instance From a Project

About the Jira Connector and Instances

Integration with the Jira system is enabled through the Jira connector available with Code Insight. This connector supports pre-populated data (in the form of one or more *instances*) used to connect Code Insight with the Jira system and create Jira issues in that system. The Jira connector also controls the frequency of data synchronization between Code Insight and the server, enabling users to view the current state of these external Jira issues from within Code Insight. The Code Insight System Administrator has the responsibility of defining one or more of Jira ALM instances in Code Insight.

Once Jira ALM instances are available, the Code Insight Project Administrator can associate a Code Insight project with a specific Jira ALM instance so that project users can create track external Jira issues from Code Insight.

Associating a Code Insight Project with a Jira ALM instance

Use the following instructions to associate a Code Insight project with a Jira ALM instance.

Task ***To associate a Code Insight project with a Jira ALM instance, do the following:***

1. As the Project Administrator, navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** window opens.
3. Select the **ALM Settings** tab.
4. From the **ALM Instance** dropdown list, select the Jira ALM instance to associate to this project.

   The current settings for the Jira ALM instance are displayed.

   If no instances (or no appropriate instances) are available in the dropdown list, contact the System Administrator about the configuration of a Jira ALM instance that meets your needs.
5. Complete the fields on the **ALM Settings** tab as needed. Refer to Fields Defining an External Jira Issue for field descriptions and important information about the fields. Consider the following:

   - The **Default Project Key** field is the only field (other than the **ALM Instance** field) that requires a value when associating the project with the instance. If a default value is provided, ensure that it is correct.

- The remaining fields are optional when associating the project with the instance. A value that you enter for any of these remaining fields serves as a default for that field when a project user actually defines an external Jira issue for project inventory. Keep in mind that, although the default value for a field might be the one most commonly used (or that you desire to be used) for external issues, the user creating the issue can override the default as needed.
- If you enter a default value, ensure that it is valid. Validation of these field values takes place during the creation of a Jira issue. At that time, if information entered for these fields is invalid (for example, the **Default Assignee** value does not exist in the Jira system), the information will still be saved, but the user will not be able to create the issue on Jira system.

6. When you have completed the settings, click **Save** to associate the Jira ALM instance to the project.

Disassociating an Jira ALM Instance From a Project

The Project Administrator can disassociate an Jira ALM instance from a project at any time. If the association is removed, any existing Jira issues for the Code Insight project will remain in Jira, but the **Create Work Item** option on the **Task Details** dialog is disabled.

Task ***To disassociate an ALM instance from a project, do the following:***

1. As the Project Administrator, navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** window opens.
3. Select the **ALM Settings** tab.
4. In the **ALM Instance** dropdown list, change the selection to **None**.
5. Click **Save**.

Identifying Child Projects for a Project

Code Insight enables you to create and manage project hierarchies as a means to keep track of projects related each other. A project hierarchy is created by simply identifying one or more projects as *child projects* of another project (called the *parent project*). Once the hierarchy is created, links are established in the Code Insight Web UI between the parent project and the associated child projects so that you can easily move between projects to assess scan results and review inventory.

A project hierarchy is useful when your product application contains one or more modules, each with a codebase for which you want to set up a separate project to track and assess its open-source or third-party software. By setting up a project hierarchy, you can easily switch between the main project for your application (the parent project) and the projects for the modules (the child projects) to complete the work needed to build a composite Bill of Materials.

Note that a child project, in turn, can be a parent project to other projects. Likewise, a given parent project can be identified as a child project to other projects. Since hierarchies are created as needed, projects might have no association with a hierarchy.

Once the hierarchy for a given project is established either as a parent or a child, you can do the following:

- From the **Summary** tab for the project, view and link to any of its child and parent projects (see Summary Tab).

- From the global **Inventory** view, examine the inventory of its child projects as well as link to any these projects (see Inventory View).

Continue with these next topics for descriptions on how manage project hierarchies:

- Identifying Child Projects for a Given Project
- Disassociating a Child Project from a Parent Project

Identifying Child Projects for a Given Project

Use this procedure to associate one or more projects as child projects of a given project.

***Note** ▪ Ensure that each project that you want to identify as a child project has already been created.*

Task ***To identify one or more projects as child projects of another project, do the following:***

1. As Project Administrator, navigate to the **Summary** tab of the project for which you are identifying one or more child projects. (For navigation instructions, see Opening the Project Summary Tab.)
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** window opens.
3. Select the **Project Hierarchy** tab.
4. On the **Project Hierarchy** tab, click **Add Child Project**.
5. From the **Add Child Project** dialog that is displayed, select the project you that want to identify as a child project of the current project. (For a description of this dialog and the fields available on the **Project Hierarchy** tab, see Edit Project: Project Hierarchy Tab.)

   Once you select the project, you are returned to the **Project Hierarchy** tab, which now lists the new child project.
6. Repeat steps 4 and 5 to add other child projects to the current project.
7. Click **Save**.

Disassociating a Child Project from a Parent Project

Use these steps to disassociate a child project from its parent project. Once you disassociate the child project, it is removed from the parent project’s hierarchy.

For a description of the fields available on the **Project Hierarchy** tab used to perform this procedure, see Edit Project: Project Hierarchy Tab.

***Note** ▪ Disassociating a child project simply means that project is no longer identified as a child of the current project. This procedure does not delete or in any way change the project that you disassociate.*

Task ***To disassociate a child project from a parent project, do the following:***

1. As Project Administrator, navigate to the **Summary** tab of the project for which you are disassociating one or more child projects. (For navigation instructions, see Opening the Project Summary Tab.)
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** window opens.
3. Select the **Project Hierarchy** tab.
4. In the list of child projects on the tab, click **X** in the **Actions** column for the child project you want to disassociate from the parent project. A message box appears, prompting you to confirm the disassociation.

   Once you confirm to disassociate the project, child project is removed from the hierarchy. The links associated with this parent-child relationship are also removed from the **Summary** tabs for the parent project and the project that was disassociated. The links are also removed from **Inventory** view.
5. Repeat the previous step for each child project you want to disassociate from the current project.
6. Click **Save**.

Completing Custom Fields for the Project

The **Custom Fields** tab on the **Edit Project** window lists the fields that were defined specifically for Code Insight projects at your site. These fields provide users with helpful information that supplements the information provided by the standard Code Insight fields for projects. Provide values for these custom fields to describe the current project.

Note ▪ *If no custom fields for projects have been configured for your site, the following message is displayed on the tab: “There are no custom fields configured.” However, if custom fields have been defined for your site but none are currently not available for display, this tab is blank (that is, shows no message or fields).*

Task ***To complete the custom fields that describe the current project, do the following:***

1. As the Project Administrator, navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** window opens.
3. Select the **Custom Fields** tab.
4. For any or all of the fields on this tab, enter the requested information as it pertains to the current project. For more information about the completing the different types of fields that you might encounter, see Edit Project: Custom Fields Tab.

   For information about completing the **SBOM Bucket Name** field (if it is available), see Assigning the Project to an SBOM Insights Bucket.
5. Click **Save**.

Assigning the Project to an SBOM Insights Bucket

SBOM Insights (a Revenera SCA product) gives organizations the ability to manage security and legal risk by maintaining a complete, accurate SBOM (Software Bill of Materials) in the cloud. SBOM Insights aggregates this SBOM over multiple sources and provides full visibility of its contents to security and legal teams, as well as to supply chain partners.

If Code Insight has been configured to perform SBOM Insights exports, Project Analysts can export inventory data from a given Code Insight project to SBOM Insights. When the export process is finished, SBOM Insights automatically imports the exported data to a bucket, where the data is managed and aggregated with SBOMs from other sources. (For complete information about SBOM Insights, refer to the SBOM Insights user documentation.)

To enable a project to export inventory data to SBOM Insights, the Code Insight Project Administrator must assign the project to the specific SBOM Insights bucket in which the inventory exported from the project is imported.

Refer to the following sections for information about performing this task:

- Overview of the Export Configuration and Process
- Assigning the Project to an SBOM Bucket

Overview of the Export Configuration and Process

To provide context for the Project Administrator’s role in the process of exporting Code Insight inventory data to SBOM Insights, refer to the following table. It provides an overview of the configuration tasks and the process involved in the export.

Table 2-18 ▪ Configuration and Process Involved in Exporting Project Inventory to SBOM Insights

Phase	Performed By	Description	For More Information
1	Code Insight System Administrator	Configures Code Insight to enable SBOM exports.	Refer to “Configuring Code Insight for Exports to SBOM Insights” in the *Code Insight Installation & Configuration Guide*.
2	Code Insight Project Administrator	Assigns the Code Insight project to a specific SBOM Insights bucket.	See the current section.
3	Code Insight Project Analyst	Initiates the process that exports the project’s inventory to SBOM Insights and imports it to the specified bucket.	See Exporting Project Inventory to SBOM Insights.
4	SBOM Insights	Automatically imports the exported inventory to the assigned bucket as a set of “SBOM parts”.	Click here to open to the section in the SBOM Insights help describing SBOM parts and their import into SBOM Insights.

Table 2-18 ▪ Configuration and Process Involved in Exporting Project Inventory to SBOM Insights (cont.)

Phase	Performed By	Description	For More Information
5	Any Code Insight user	Accesses the Code Insight **Jobs** queue to track the progress of the export.	See Monitoring the Code Insight Jobs Queue.

Obtaining the Bucket Name

To assign the project to specific bucket, you must know the name of the bucket in SBOM Insights. If you have access to SBOM Insights and depending on your user role, you can do the following to determine the bucket name:

- With an **SBOM Viewer** role in SBOM Insights, you can view the current buckets in SBOM Insights and determine the name of the bucket.
- With an **SBOM Manager** role in SBOM Insights, you can both view the current buckets and create a bucket if necessary.

If you do not have access to SBOM Insights, work with an **SBOM Manager** user, who can determine the name of the appropriate bucket or create the bucket for you.

For more information about buckets and their creation in SBOM Insights, click here to open the appropriate section in the SBOM Insights user documentation.

Assigning the Project to an SBOM Bucket

Use the following procedure to assign a project to an SBOM Insights bucket.

Task ***To assign a project to an SBOM Insights bucket, do the following:***

1. As the Project Administrator, navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** window opens.
3. Select the **Custom Fields** tab.
4. In the **SBOM Bucket Name** field, enter the name of an existing bucket in SBOM Insights to which SBOM Insight will import the inventory data exported by Code Insight.
5. Click **Save**.

Once this bucket is assigned to the project, Analysts of the project can export the project’s inventory data to SBOM Insights.

Changing the Project Contact

Code Insight provides the ability to change the Project Contact for a given project. The Project Contact, initially the project creator, is the default contact for all task-workflow notifications generated during the inventory review process. That is, if a Legal, Security, or Development contact has not been explicitly assigned to the project through system Project Defaults or at the project-settings level, that contact defaults to the Project Contact. Additionally, the Project Contact is the default contact for any “miscellaneous” tasks created during an inventory review.

The current Project Contact, a Project Administrator, or a System Administrator can transfer the Project Contact role to different user. That user automatically inherits the roles the previous Project Contact user held.

***Note ▪** Changing a Project Contact is a silent transaction. No email notifications will be sent as part of this operation.*

Task ***To change the Project Contact, do the following:***

1. Log into Code Insight as the current Project Contact, a Project Administrator, or System Administrator.
2. Navigate to the **Summary** tab (see Opening the Project Summary Tab).
3. Select **Change Project Contact** from the **Manage Project** menu, or click the **Change Project Contact** button, whichever is available. The **Select New Project Contact** dialog appears.

   ***Note ▪** If you have not logged with the appropriate permissions, neither the menu option nor the button will be visible.*

4. Select a name in the list and click **Apply**. The **Summary** tab shows the selected name displayed in the **Project Contact** field.

Updating Third-Party Notices Across Inventory for a Project

Project Analysts can have Code Insight automatically update each inventory item in a project with the item’s appropriate third-party notices content obtained directly from the Revenera Data Library. The Analyst can choose to update the notices text for every inventory item in the project or for only those inventory items with an empty **Notices Text** field. When this feature is run, existing notices text for an inventory item is overwritten.

When initiated, the notices update is scheduled in the Code Insight **Jobs** queue.

***Note ▪** Code Insight also provides the option to update the **Notices Text** field manually for individual inventory items. For more information, see Finalizing the Notices Text for the Notices Report.*

Task ***To initiate a notices update across all inventory in a project, do the following:***

1. As a Project Analyst, navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Update Notices**.

   Note ▪ *The **Update Notices** menu option is temporarily disabled whenever a scan, rescan, Project Copy, Project Branch, or another **Update Notices** job is in progress or scheduled for the current project. Once these jobs complete (check the **Jobs** queue), the **Update Notices** menu option is re-enabled so that you can initiate the notices update process.*

3. From the **Update Notices** dialog, select either option:
   - **Update only if empty**—Only those inventory items in the project with an empty **Notices Text** field are updated with notices text retrieved for the component version from the Revenera Data Library. Inventory items with existing content in their **Notices Text** field are skipped during the update.
   - **Overwrite all**—All inventory items in the project are updated with the appropriate notices text retrieved from the Revenera Data Library. Any existing content in the **Notices Text** field is overwritten.
4. Click **OK** to initiate the update process. The success message “Update Notices job triggered with ID: *x*” (where *x* is the job ID) is displayed in the upper right of the screen.
5. To track the status of the **Update Notices** job, open the **Jobs** queue and locate the job ID. See Monitoring the Code Insight Jobs Queue for complete instructions on accessing and monitoring the **Jobs** queue.

   Note the following about **Update Notices** jobs in the **Jobs** queue:
   - When **Update Notices** jobs for other projects are triggered concurrently with your job, the first **Update Notices** job triggered is in the **Active** state in the **Jobs** queue. The remaining **Update Notices** jobs are placed in a **Scheduled** state and are run based on the scheduled order.
   - If a **PDL Update** (Electronic Update) or **Library Refresh** job is currently is progress, all **Update Notices** jobs after the start of the **PDL Update** or **Library Refresh** job are placed in a **Scheduled** state. Once the job is finished, the **Update Notices** jobs are run based on the scheduled order.

Rescanning Your Codebase (Server Scans Only)

During a server scan, Code Insight uses a combination of Automated Analysis and Advanced Analysis techniques to identify open-source and third-party content in your codebase (see About Code Insight Scans). Automated Analysis is always performed during a scan. Advanced Analysis is performed only if the Compliance Library (CL) has been installed in your Code Insight system and the scan settings for your project have enabled this type of analysis.

When you run the initial scan on your codebase, a *full* scan (that is, a scan on all codebase files) is automatically performed for both analysis techniques. For any subsequent codebase rescans that you initiate, you can manage the scan as follows:

- Force a full rescan (which can take considerable time since all codebase files will be rescanned).

- Allow default scan behavior for all regular rescans (that is, ones that are not forced full rescans). Default rescan behavior typically scans only those files that have changed since the last scan. However, certain Code Insight events that have occurred since the last scan can result in a full rescan.
- Configure the scan profile associated with the project so that all rescans (except forced full rescans) scan only changed files and skip unchanged files, despite any events that might have occurred.

The following topics provide information you should know about the rescan process and includes instructions on initiating a rescan:

- Default Rescan Behavior
- Configuring Rescans to Always Skip Unchanged Files
- Effects of Scan-Setting Changes on Rescans
- Handling of Edited Inventory During Rescans
- Initiating a Codebase Rescan
- Forcing a Full Codebase Rescan

Default Rescan Behavior

When a user initiates a regular rescan (that is, does not a *force* a full rescan), typically only codebase files that have changed since the last scan are rescanned. However, by default, certain Code Insight events that might have occurred since the last scan will determine whether the rescan performs a scan on all files (full rescan) or on only those codebase files that have changed since the last scan.

The following table lists these events and the type of default scan performed by each analysis technique—Automated Analysis and Advanced Analysis (if this technique is configured)—during the rescan.

Note that a System Administrator can configure the scan profile associated with a project to override the default rescan behavior dictated by these listed events, so that each rescan skips unchanged files and scans only those files that have changed, despite any events that might have occurred. For more information, see Configuring Rescans to Always Skip Unchanged Files.

Table 2-19 ▪ Default Rescan Behavior for Events

Event	Automated Analysis	Advanced Analysis	Notes
Change to codebase files	Only changed files scanned	Only changed files scanned	Changes in codebase files are determined by the MD5 hash digest of the files.

Table 2-19 ▪ Default Rescan Behavior for Events (cont.)

Event	Automated Analysis	Advanced Analysis	Notes
Change to Automated Analysis rule set	Full rescan	See Notes	Changes in Automated Analysis rules result in Automated Analysis performing a full rescan to reapply the changes to all files. (The rule changes are automatically pushed to your Code Insight server through an internal process and the weekly Electronic Update.) Additionally, Advanced Analysis performs a full rescan only if rule changes have occurred *and* either the CL version has changed or an NG-bridge update has occurred. (See the *Code Insight Installation and Configuration Guide* for information about NG-bridge updates.) ***Note ▪*** *If an override of default rescan behavior is in effect (see Configuring Rescans to Always Skip Unchanged Files), the rules are applied to only changed files; unchanged files are skipped.* ***Note ▪*** *Custom detection rules are applied only during the initial codebase scan and during a forced full rescan. They are not applied during a regular rescan initiated by the user.*
Code Insight version change	See Notes	See Notes	Automated Analysis performs a full rescan only if the new Code Insight version includes changes to the Automated Analysis framework. Advanced Analysis performs a full rescan only if the Code Insight version has changed *and* either the CL version has changed or an NG-bridge update has occurred. (See the *Code Insight Installation and Configuration Guide* for information about NG-bridge updates.) ***Note ▪*** *If an override of default rescan behavior is in effect (see Configuring Rescans to Always Skip Unchanged Files), the rescan skips unchanged files and scans only changed files, even if conditions are present to perform a full rescan for one or both analysis techniques.*

Table 2-19 ▪ Default Rescan Behavior for Events (cont.)

Event	Automated Analysis	Advanced Analysis	Notes
Scan profile setting change	Full rescan	See Notes	A full rescan for Automated Analysis is required if one or more specific scan profile settings have changed. See Effects of Scan-Setting Changes on Rescans for more information. Advanced Analysis also performs a full rescan only if one or more of the specific profile settings have changed *and* either the CL version has changed or an NG-bridge update has occurred. Keep in mind that changes to settings related to source-code matching result in an *expensive* full scan for Advanced Analysis. (See the *Code Insight Installation and Configuration Guide* for information about NG-bridge updates.) Scan profiles are configured by the Code Insight System Administrator as described in the *Code Insight Installation and Configuration Guide*. ***Note ▪*** *If an override of default rescan behavior is in effect (see Configuring Rescans to Always Skip Unchanged Files), the rescan skips unchanged files and scans only changed files, even if scan profile settings have changed.*

Configuring Rescans to Always Skip Unchanged Files

As described in Default Rescan Behavior, by default certain Code Insight events that might have occurred since the last scan can determine whether the rescan performs a scan on all files (full rescan) or on only those codebase files that have changed since the last scan.

However, the System Administrator can configure the scan profile associated with a project to override this default behavior. The configuration allows rescans to always skip unchanged files and scan only changed files, even if events that typically call for a full rescan have occurred. It also delineates *which* unchanged files are skipped: all unchanged files, only unchanged files that have been reviewed, only unchanged files that are associated with inventory, or only unchanged files that are both reviewed and associated with inventory.

See your System Administrator about the rescan configuration options.

Note ▪ *The current rescan configuration options are ignored if the user initiates a forced full rescan. All files—both changed and unchanged—are completely scanned.*

Effects of Scan-Setting Changes on Rescans

One type of change event that, by default, does result in a full rescan by either Automated Analysis or Advanced Analysis (or both) is an update to settings in the scan profile associated with the rescan. Depending on which settings have changed, the full rescan could be more expensive (requiring more time and resources) than other full rescans.

Note the following:

- If you have applied a new scan profile to your project, only those profile settings that are different from the settings in the previously associated profile will impact the rescan.
- If an override of the default rescan behavior is in effect (see Configuring Rescans to Always Skip Unchanged Files), no full rescan is performed even if any of the scan profile settings listed below have changed. The rescan skips unchanged files and scans only those files that have changed.
- If the following table shows that a change to a specific setting results in a full rescan for Advanced Analysis, note that the full rescan is performed only if, in addition to the setting change, either the CL version has changed or an NG-bridge update has occurred. Otherwise, the setting change results in a scan of only changed files.

The following table provides a list of the scan profile settings and the type of full rescan to expect should any of the settings be updated prior to a codebase rescan.

Table 2-20 ▪ Types of Full Rescan to Expect Should Scan Profile Settings Change

Scan Profile Settings	Automated Analysis	Advanced Analysis
A change to any of these settings: ● **Perform Package/License Discovery in Archive** ● **Dependency Support** ● **Automatically Add Related Files to Inventory**	Full rescan	—
A change to any of these settings: ● **Source Code Matches** Related fields: ● **Include System Identified Files** ● **Include Files with Exact Matches** ● **Minimum Source Code Matches**	—	Full rescan (expensive)
A change to any of these settings: ● **Exact Matches** ● **Search Terms** ● **Scan Inclusions**	—	Full rescan (expensive but less expensive than that performed when **Source Code Matches** or related fields change)

Handling of Edited Inventory During Rescans

Code Insight enables you to make changes to inventory both in the **Analysis Workbench** and on the **Project Inventory** tab. You create inventory items as well as edit both user-created and system-generated inventory. Edits to existing inventory can include changes to the following elements in an inventory item:

- The component version string or the associated license
- Codebase-file associations (only in the **Analysis Workbench**)
- Inventory properties, Notices text, and notes

However, normal Code Insight rescan behavior can result in actions that impact your inventory changes. For example, an updated Automated Analysis rule set might associate codebase files to an inventory item different from the one to which you have *manually* associated these files. Logically, the rescan should remove the associations you defined and re-apply them to the inventory item identified in the rule set. However, losing the manual changes might not be desirable.

The following topics describe how the rescan process handles edited inventory:

- Rescan Rules to Preserve Inventory Data
- Rescan Scenario: Handling of Files Manually Disassociated from Inventory Before Rescan

Rescan Rules to Preserve Inventory Data

In general, a rescan (full or on only changed files) can add or disassociate files and overwrite properties for any existing system-generated inventory that has not been manually updated. However, the rescan *does* retain the existing status and priority for such inventory items, as well as any existing notes or Notices text (although the scan can append new notes or Notices text).

For inventory that you have manually edited or created, the rescan applies the following rules:

- All the user-created inventory, its file associations, and edits are considered not system-updatable and therefore are preserved.
- Any manual change to a system-generated inventory item (including updates to the associated component) results in the inventory item being classified as user-created and therefore not system-updatable (see the previous rule.) However, the rescan can add additional files to the inventory item if the component, version, and license match.
- If one or more files were manually disassociated from a system-generated inventory item before the rescan, rescan logic assumes that these files were erroneously associated with the component initially. Therefore, the rescan does not attempt to re-associate these files to the inventory item; nor does it associate the files with another inventory item that uses the same component name (with a different version or license). The following example scenario illustrates this rule.

Rescan Scenario: Handling of Files Manually Disassociated from Inventory Before Rescan

The following scenario demonstrates how the rescan process handles files that you manually disassociated from a system-generated inventory item before the rescan.

Scenario

In this scenario, the initial scan on your codebase generated the inventory item **log4j 2.6** and associated the files **file1.jar** and **file2.jar** with the item.

However, after analyzing the inventory, you realize that **file2.jar** should be associated instead with **log4j 2.11**, an inventory item that does not exist in your current inventory. To remedy this, you perform the following steps:

1. Create an inventory item named **log4j 2.11**.
2. Disassociate the **file2.jar** from **log4j 2.6**.
3. Associate **file2.jar** with the inventory item **log4j 2.11** that you just you created.

On rescan, your edits remain intact:

- The file **file1.jar** remains associated with the inventory item **log4j 2.6**.
- The inventory item **log4j 2.11** that you created is preserved along with its association with the file **file2.jar**.

The rescan also results in the creation of a new system-generated inventory item, **log4j 2.10**. However, the rescan does not associate the file **file2.jar** with the new inventory item.

Initiating a Codebase Rescan

Use the following procedure to rescan your codebase.

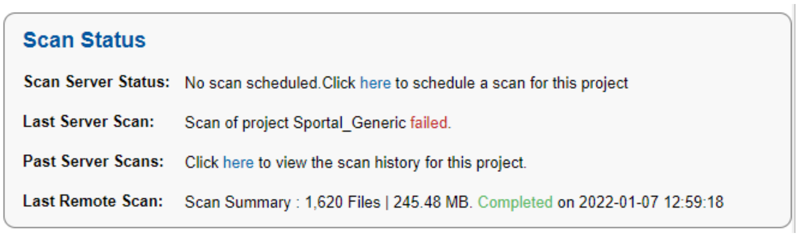
Refer to the Code Insight User Roles and Permissions section for role requirements to scan a codebase.

Task ***To start the rescan, do the following:***

1. Navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. Perform either action:
    - Click the **Start Scan** button.
    - Click the “here” link in **Scan Server Status** to schedule a scan.
3. Monitor the scan’s progress by clicking the “here” link in **Past Server Scans** to obtain information about scheduled, active, and past scans on the project. You can also monitor the scan in the **Jobs** queue (see Monitoring the Code Insight Jobs Queue). Note the following:
    - If a scan is running on another project, your scan will automatically start based on queue order. Additionally, if the Scan Server is temporarily inactive, the scan will automatically start based on queue order once the server is running again.
    - The project currently open can have only one scan in queue or in progress at a time. If you attempt to schedule a scan when your project already has a scan queued or running, the **Start Scan** button will be disabled until the scan completes. The **Start Scan** button is also disabled when a **Project Copy**, **Project Branch**, or **Apply Policy - Project** job for the project is scheduled or running. For more reasons for **Start Scan** button disablement and the actions you can take, see Actions to Take When the Start Scan Button is Disabled.

- If a report generation is currently in queue or in progress for the project, the scan is not triggered. Instead, a pop-up error message is displayed, explaining that you must wait until the report generation has completed before repeating this step to attempt to trigger the scan again.

Information about the server scan's progress is shown in the **Scan Server Status** section on the **Summary** tab.



Scan Status

Scan Server Status: No scan scheduled.Click here to schedule a scan for this project

Last Server Scan: Scan of project Sportal_Generic failed.

Past Server Scans: Click here to view the scan history for this project.

Last Remote Scan: Scan Summary : 1,620 Files | 245.48 MB. Completed on 2022-01-07 12:59:18

When the scan completes, **Last Server Scan** will display one of the following messages:

- **Completed**—The scan succeeded with no warnings during scan or analysis. This message appears on the screen in green.
- **Completed with warnings**—The scan succeeded but the analysis produced warnings. For more information, check the **scanEngineDetail** log for the Scan Server.
- **Failed**—The scan failed. This message appears on the screen in red. For more information, see Scan Failure Reasons and Troubleshooting Measures.

For an overall understanding of the scan results, see Overview of Scan Results.

Forcing a Full Codebase Rescan

A forced full-codebase rescan enables you to scan your entire codebase at any time even if no change has occurred in your codebase, in your scan settings, or with the Code Insight or Compliance Library (CL) version. Such a rescan might be required, for example, to view the latest changes to inventory or to apply any new custom detection rules. See the following topics for more information:

- Forcing a Full Rescan
- Custom-rule Application During a Forced Full Rescan

Keep in mind that a full rescan can take considerable time.

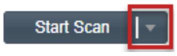
For general information about how any rescan handles existing system-generated inventory and manually created or updated inventory, see Handling of Edited Inventory During Rescans. For specific information about how the forced full rescan applies custom rules to existing system-generated inventory, see Custom-rule Application During a Forced Full Rescan.

Forcing a Full Rescan

Use the following procedure to initiate a full codebase rescan.

Task ***To force a full project rescan, do the following:***

1. As Project Administrator or Analyst, navigate to the **Summary** tab (see Opening the Project Summary Tab).
2. Click the drop-down arrow next to the **Start Scan** button.



3. Select **Full Rescan**.

   A confirmation message box is displayed asking you to confirm that you want to continue with the rescan.

4. Select **Yes**.

   Information about the scan's progress and its completion status is shown in the **Scan Status** section. For details, see the last step in Initiating a Codebase Rescan.

Custom-rule Application During a Forced Full Rescan

Custom rules are applied only during the initial codebase scan and during a forced full rescan. During the forced full rescan, if a previously scanned file now matches a new custom rule, the existing system-generated inventory item for that file is overwritten with the custom-rule data. The following two scenarios describe this over-write process.

Note ▪ *Custom rules affect only system-generated inventory items that have not been manually updated. The rules have no impact on manually-created (custom) inventory items and on system-generated inventory items that users have updated.*

Scenario 1: The custom-rule data identifies the same component version and license as the existing inventory item

In this case, the scan applies the custom rule by updating the existing inventory item as follows:

- Appends new detection notes to reflect the custom rule.
- Updates the **Created by** field value for the inventory item to **High Confidence Custom Auto-WriteUp Rule** in the **Analysis Workbench**.

Note that, in this scenario, the scan retains the **Audit Notes**, **As-Found License Text**, or **Notices Text** content defined in the existing inventory item. It does not append custom-rule data to these fields.

Scenario 2: The custom-rule data identifies a component version and license different from the existing inventory item

In this case, the scan applies the custom rule as follows:

- Creates a new inventory item based on the custom-rule data.
- Retains the existing inventory item, but disassociates its files and adds them to the new inventory item.
- Applies the status and priority of the existing inventory to the new inventory item.
- Appends any **Audit Notes**, **As-Found License Text**, or **Notices Text** content defined in the custom rule.

Exporting Project Data

Code Insight allows you to export your project data to a JSON data file for other uses—but especially for importing data into other Code Insight projects. You can run a project data export by selecting the **Export Project Data** option from the **Manage Project** menu on a project's **Summary** tab.

For complete information about the export feature (including how to export project data using the public REST API), see Exporting and Importing Project Data.

Importing Project Data

Code Insight allows you to import data from one Code Insight project into another project. The data to be imported must be in a properly formatted and archived JSON file, such the archive resulting from a project data export (see in Exporting Project Data).

You can set up and execute a project data import by selecting the **Import Project Data** option from the **Manage Project** menu on the project's **Summary** tab. For complete information about the import feature (including how to import project data using the public REST API), see Exporting and Importing Project Data.

Exporting Project Inventory to SBOM Insights

SBOM Insights (a Revenera SCA product) gives organizations the ability to manage security and legal risk by maintaining a complete, accurate SBOM (Software Bill of Materials) in the cloud. SBOM Insights aggregates this SBOM over multiple sources and provides full visibility of its contents to security and legal teams, as well as to supply chain partners.

If Code Insight has been configured to perform SBOM Insights exports, Project Analysts can export inventory from a given Code Insight project to SBOM Insights. When the export process is finished, SBOM Insights automatically imports the exported data as "SBOM parts" to a specific SBOM bucket. (This bucket is assigned to the current project by the Code Insight Project Manager, as described in Assigning the Project to an SBOM Insights Bucket.) From this bucket, the parts are managed and aggregated with parts from other buckets (sources) to create a complete SBOM.

For complete information about SBOM Insights, click here to access the SBOM Insights user documentation.

***Important ▪** The option to export inventory to SBOM Insights is available only if the Code Insight System Administrator has configured Code Insight for this type of export. (If the option is available, it is displayed on the **Manage Project** menu on the project's **Summary** tab.)*

Task ***To export project inventory to SBOM Insights, do the following:***

1. As a Project Analyst, navigate to the **Summary** tab for the project whose inventory you want to export to SBOM Insights (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Export to SBOM Insights**.

   ***Note ▪** The **Export to SBOM Insights** menu option is disabled whenever a scan, rescan, Project Copy, or another SBOM Insights export job is in progress or scheduled for the current project. Once these jobs complete (check the **Jobs** queue), the **Export to SBOM Insights** menu option is re-enabled so that you can initiate the export process.*

3. When prompted, select **Yes** to proceed with the export process.

    - If the export is successfully initiated, the message "Export to SBOM Insights bucket job is initiated with ID: *x*" (where x is the job ID) is displayed in the upper right of the screen.
    - If no SBOM Insights bucket has been specified for this project, an error message is displayed. Contact the Project Administrator for assistance.

4. To track the status of the export job, open the **Jobs** queue and locate the job ID. See Monitoring the Code Insight Jobs Queue for complete instructions on accessing and monitoring the **Jobs** queue.

    Note the following about SBOM Insights export jobs in the queue:

    - The job status changes to **Complete** only after the inventory data has been exported *and* SBOM Insights has imported this data to the specified bucket.
    - If an invalid bucket has been specified for this project, the export job ends with a **Failed** status. Contact the Project Administrator for assistance.
    - If an Electronic Update or Library Refresh is currently is progress, all subsequent **Export to SBOM Insights** jobs are placed in a **Scheduled** state. Once the update is finished, the export jobs are run based on the scheduled order.
    - When other projects trigger **Export to SBOM Insights** jobs concurrently with your job, the first export job triggered is in the **Active** state. The remaining export jobs are placed in a **Scheduled** state and are run based on the scheduled order.

Branching a Project

Code Insight provides the **Branch Project** wizard to automate the process of branching one Code Insight project to another, enabling the branched project to preserve any file-audit data, inventory, and inventory-review data that was created in the project from which you are branching.

The following sections describe how to use the **Branch Project** wizard to set up and start the branching process and what happens during the branching operation.

- Project-Branching Terminology
- Overview of the Branching Operation
- Setting Up and Starting the Project-Branching Process
- Other Considerations About the Project-Branching Operation

Project-Branching Terminology

The following terminology is used in the descriptions that follow:

- **Source project**—The Code Insight project whose data you are branching to another project.
- **Branched project**—The new project to which you are branching data from the source project.

Overview of the Branching Operation

The following provides a basic overview of what happens during the project-branching operation.

Table 2-21 ▪ Phases of the Project-Branching Operation

Branching Phase		Description
Phase 1	**Launch of Branch Project wizard**	The user opens the **Branch Project** wizard to set up the project-branching operation. Details about the entire setup process is described in Setting Up and Starting the Project-Branching Process.
Phase 2	**Creation of branched project during setup**	As part of the setup, the branched project is created once the project properties on the **Project Information** page of the wizard have been validated (upon clicking **Next**), as described in Step 1: Creating the Branched Project.
Phase 3	**Codebase uploads and SCM synchronization configuration during setup**	As part of the setup, one or more codebases can be uploaded directly from the **Upload Codebase** page in the wizard, as described in Step 2: Uploading a Codebase (Optional). Additionally, as part of setup, the user can configure one or more Source Control Management (SCM) instances from the **Version Control Settings** page, enabling the branch process to synchronize the project with remote codebase repositories in your site's SCM applications. Unlike codebase uploads, synchronization takes place once the automated part of the branching process begins (Phase 5). Codebase uploads and synchronization are optional, as the user might want to simply perform an inventory copy (Phase 7).
Phase 4	**Initiation of project-branching operation**	After all appropriate setup information is provided in the wizard, the user initiates the branching operation by clicking **Finish** on the **Summary** page of the wizard, as described in Step 5: Initiating the Branching Operation.
Phase 5	**Synchronization with remote codebases through SCM instances**	During the branching operation, the branched project is synchronized with one or more remote Source Code Management (SCM) repositories to obtain codebase files. This synchronization occurs only if SCM instances were configured during setup on the **Version Control Settings** page of the wizard (as described in Step 3: Configuring Synchronization with a Source Code Management Instance (Optional)).
Phase 6	**Scan of branched project codebase**	The branching operation then scans the codebase of the branched project. The codebase includes any of the uploaded codebase files as well as any codebase files obtained through synchronization with SCM instances. Note that the size of the codebase can impact the length of time needed for the scan.

Table 2-21 ▪ Phases of the Project-Branching Operation (cont.)

Branching Phase		Description
Phase 7	**Copy of project information**	The branching operation copies (that is, imports) file-audit data, inventory, and inventory-review information from the source project to the branched project. If no codebases have been uploaded or obtained through synchronization with SCM repositories, only inventory and inventory-review information is imported; no information about files associated with the inventory is included in the import.
Phase 8	**Operation completion**	Once the branching operation successfully completes, users can open the branched project and begin auditing files and reviewing inventory.

Setting Up and Starting the Project-Branching Process

The following procedures highlight important information about the various steps that the **Branch Project** wizard guides you through in setting up a project-branching operation.

- Opening the Branch Project Wizard
- Step 1: Creating the Branched Project
- Step 2: Uploading a Codebase (Optional)
- Step 3: Configuring Synchronization with a Source Code Management Instance (Optional)
- Step 4: Configuring a Project Copy
- Step 5: Initiating the Branching Operation

You can cancel the setup process at any time, as described in Canceling the Branching Setup Process. Also review Other Considerations About the Project-Branching Operation for special notes pertaining to the project-branching process.

Opening the Branch Project Wizard

Begin the setup for the project-branching by opening the **Branch Project** wizard from the source project.

Task ***To open the Branch Project wizard, do the following:***

1. Navigate to the **Summary** tab of the source project (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Branch Project**.

   The **Branch Project** wizard opens to the **Introduction** page.
3. Click **Next** to navigate to the **Project Settings** page.

***Note** ▪ The **Manage Project | Branch Project** option is disabled if the source project has not yet been successfully scanned or if a codebase upload or a **Project Scan**, **Apply Policy - Project**, or report generation job is currently in progress on the project.*

Step 1: Creating the Branched Project

The **Project Settings** page in the **Branch Project** wizard enables you to define the properties for the branched project and then creates the project once you click **Next** to move to the next wizard page. By default, property values are pre-populated with values from the source project. However, you can edit these properties as needed.

Task ***To create the branched project, do the following:***

1. On the **Project Settings** page in the **Branch Project** wizard, identify the properties for the new branched project. Initially, the properties from the source project pre-populate this page, but you can edit these properties as needed. See Branch Project: Project Information for a description of each property.

   The option you choose for **Source Code Options** determines how the branching process obtains the codebase files for the branched project. Initially, the **Upload Codebase** option is selected by default. The **Merge from Source Control** option is also selected if the source project obtains codebase files from synchronization with a Source Control Management instance. However, you can edit these options as needed for the branched project. If you select neither option, only inventory and inventory-review information is copied to the branched project; no information about the files associated with inventory is branched.

2. Click **Next** to create the project and proceed to the next appropriate page in the wizard.

   - If selected **Upload Codebase** in the **Source Code Options** section, the **Update Codebase** page opens. See Step 2: Uploading a Codebase (Optional).
   - If you selected only **Sync from Source Control** in the **Source Code Options** section, the **Version Control Settings** page opens. See Step 3: Configuring Synchronization with a Source Code Management Instance (Optional).
   - If you selected neither option, the **Project Copy Settings** page opens. See Step 4: Configuring a Project Copy.

   Once the project is created, you cannot edit the project **Name** should you return to this page to update project properties. Additionally, once a codebase is uploaded for the branched project, the **Scan Server** and **Upload Codebase** options on the **Project Information** page will be disabled.

Step 2: Uploading a Codebase (Optional)

The **Upload Codebase** page in the **Branch Project** wizard identifies and uploads one or more codebase files for the branched project.

This page is enabled only if you selected the **Upload Codebase** option from the previous **Project Information** page.

The uploaded codebases can be used in conjunction with codebases obtained through synchronization with your site’s SCM applications to provide the complete set of codebase files for the branched project. See Step 3: Configuring Synchronization with a Source Code Management Instance (Optional).

If you decide not to upload a codebase, clicking **Next** moves you to the next appropriate wizard page. (When this page is enabled but no codebase is uploaded, you can always return to this page to perform a codebase upload during setup if you want.)

Task ***To upload a codebase for the branched project, do the following:***

1. On the **Upload Codebase** page in the **Branch Project** wizard, select the archive containing the codebase to upload.
2. Define the properties for the upload. See Branch Project: Upload Codebase for a description of the properties.
3. Click **Upload Project Codebase** to verify the properties and proceed with uploading the codebase. A message displays when the upload has completed successfully.
4. Repeat the above steps for each codebase you want to upload.
5. When the codebases have been uploaded, click **Next** to proceed to the next appropriate page in the wizard:
    - If you selected **Sync from Source Control** on the **Project Information** page, the **Version Control Settings** page in the wizard opens. See Step 3: Configuring Synchronization with a Source Code Management Instance (Optional).
    - If you did not select **Sync from Source Control**, the **Project Copy Settings** page opens. See Step 4: Configuring a Project Copy.

Step 3: Configuring Synchronization with a Source Code Management Instance (Optional)

The **Version Control Settings** page in the **Branch Project** wizard configures one or more Source Control Management (SCM) instances, enabling the branching operation to synchronize the branched project with remote codebase repositories in your site's SCM applications. This synchronization takes place once the automated part of the branching process begins (see Step 5: Initiating the Branching Operation). For more information about how to set up for SCM synchronization and how the synchronization process works, refer to Configuring Source Code Management.

This page is enabled only if the **Sync from Source Control** option is selected on the previous **Project Information** page.

By default, any SCM instances used by the source project are automatically copied to this page, each instance defined on a separate tab. However, you can edit or remove any of these instances or add new ones as needed for the branched project. Alternatively, you can choose not to include any SCM instances on the **Version Control Settings** page, but can always return to this page to add instances later during setup if you want.

If you also uploaded codebases from the **Upload Codebase** page, the codebase files obtained through SCM synchronization, as defined on this page, will be added to the uploaded codebase files to provide the complete codebase for the branch product.

Task ***To configure synchronization with SCM instances, do the following:***

1. On the **Version Control Settings** page in the **Branch Project** wizard, update or delete any currently defined SCM instances or click **Add Instance** to create a new instance as needed. See Branch Project: Version Control Settings for more information about managing the SCM instances and the properties used to define each instance.
2. For any given SCM instance, click the **Test Connection** to ensure that branching operation can successfully connect to the remote repository specified in the instance.
3. When SCM instance configuration is complete, click **Next** to perform a final connection test on all the SCM instances defined and, and if the connections are successful, proceed to the **Project Copy Settings** page in the wizard. See Step 4: Configuring a Project Copy.

Step 4: Configuring a Project Copy

The **Project Copy Settings** page in the **Branch Project** wizard defines the parameters used by the branching process to import file-audit data, inventory, and inventory-review information from the source project to the branched project. By default, this page shows the properties used by the source project. However, you can edit these properties as needed for the branched project.

Note *▪ During the import, the branching process ignores the setting* ***On rescan or import, delete inventory without any associated files*** *for the branched project and always creates inventory. (This setting, found on the* ***Manage Project | Edit Project | General*** *tab on the project's* **Summary** tab*, can be accessed once the branching process is completed.)*

If neither **Upload Codebase** nor **Sync from Control Version** was selected on the **Project Information** page, this import process copies only inventory and inventory-review information from the source project to the branched project. With this scenario, no file information will be associated with the inventory copied to the branched project.

Task ***To define project copy settings for the branching operation, do the following:***

1. On the **Project Copy Settings** page in the **Branch Project** wizard, edit the properties that the branching operation will use to import file-audit and inventory information from the source project to the branched project. See Branch Project: Project Copy Settings for more information about these properties.
2. Click **Next** to verify the properties and, if no errors exist, move to the **Summary** page in the wizard. See Step 5: Initiating the Branching Operation.

Step 5: Initiating the Branching Operation

The **Summary** page in the **Branch Project** wizard provides an overview of the parameters that you defined for the project-branching process. From this page, you can start the branching process. Alternatively, you can navigate back to other pages in the wizard to make changes before starting the branching process, or you cancel the entire branching setup.

Task ***To initiate the branching operation, do the following:***

1. On the **Summary** page in the **Branch Project** wizard, review the list of properties defined for the branching operation.
2. If you need to make any changes to the current configuration for the branching operation, click **Back** to navigate backwards through the wizard pages. Alternatively, you can click any of the enabled tabs for wizard pages to move directly to a page.
3. To start the branching operation, click **Finish** on the **Summary** page. The project dashboard for the branched project is displayed, showing the current status of the branching process.

When the branching operation is finished, you can open the branched project and navigate to its **Project Inventory** tab or **Analysis Workbench** to proceed with auditing codebase files or reviewing inventory.

For an overview of the branching-operation phases, see Overview of the Branching Operation.

Canceling the Branching Setup Process

During the project-branching setup process, you can press the **Cancel** button on the **Branch Project** wizard window (or close the window) at any time to cancel the setup. A pop-up message will ask you to confirm the cancellation. If you select **Yes**, the branched project and its uploaded codebases will be removed from the Code Insight system.

Other Considerations About the Project-Branching Operation

The following are additional notes about the branching operation:

- Users can run parallel branching operations on the same source project.
- If a user initiates the branching process in one browser tab, opens the same Code Insight instance in another browser tab, and then navigates to the branched project in the second tab, the branching status information for the branched project should be the same as in the first browser tab.
- The project-branching session might time out (for example, due to a delayed scan phase because the scan is queued). Should the session time out, the user must use the `DELETE projects/{projectId}` REST API to remove the branched project from Code Insight. Once the project is deleted, the user can then rerun the **Branch Project** wizard to recreate the branched project and initiate the branching operation.
- Should an SCM synchronization or the scan fail during the automated part of the branching operation, the branch process is terminated. The user can navigate to the **Summary** tab of the branched project to view any captured scan details.
- During the branching process, the branch codebase is scanned before the import phase begins. Therefore, any file association for a source inventory item most likely also exists in the branch inventory item by the time the import starts. By design, when the import detects the same file association for both the source and branch inventory item, it adds no new file association to the branch inventory item. However, when the import processes this association, a “duplicate entry” exception similar to the following might be logged:

  ```
  Duplicate entry ‘entry_id’ for key 'pse_inventory_group_files.UNIQ_FILE_GROUP
  ```

This exception is benign and has no impact on the regular file-processing behavior—that is, the existing file association is retained; no new file association is added.

Copying a Project

The Project Copy feature copies the information from an existing Code Insight project—including its project settings, user information, source-code files and folders, scan evidence, inventory details, alerts, and certain scan history details—to a new project. Project Copy offers an alternative to project-branching and has the following advantages over the branching process:

- **No scan needed**—The branching process triggers a full scan of the branched project (which, in most cases, is not needed because the source project already contains base-line scan results that are copied to the branched project). In contrast, the Project Copy process simply copies all scan results from the source project to the target project without running a scan on the target.
- **Previously removed false-positive inventory not reinstated**—The scan on a branched project can pull in false-positive inventory that might have been cleaned up in the source project. On the other hand, the Project Copy process copies inventory information “as is” from the source project to the target project. If information in the source project had been cleaned up, it remains cleaned up in the target because no scan is triggered on the copied data.
- **No need to confirm project settings prior to the copy**—As part of the branch-project process, you must confirm individual project settings for the branched project before running the branch process. (In most cases, no changes are needed.) During the Project Copy, source project settings are simply copied to the target project without the extra step to confirm settings. If any changes are needed, you can apply them to the target project after the copy.

The following sections describe how you initiate a Project Copy and what happens during the copy process:

- Running the Project Copy
- Information Copied to the Target Project
- Errors During a Project Copy

Running the Project Copy

The following information describes how to initiate a Project Copy process:

- Required Conditions for Accessing the Project Copy Feature
- Actions Blocked During the Copy Process
- Initiating the Copy Process
- Project Copy Completion

Required Conditions for Accessing the Project Copy Feature

A Project Copy requires the following conditions. If any of these conditions are not met, you are blocked from opening the Project Copy feature to begin setup.

- You must be a Project Administrator of the source project.
- The source project cannot be a legacy “Inventory Only” project.

- The source project must have been previously (and successfully) scanned.
- The Scan Server must be running so that it can communicate with the Core Server as needed.
- A scan on the source project cannot be in progress. (If you select **Copy Project** option from the **Manage Project** menu while a scan is running, you receive an error message.)

Actions Blocked During the Copy Process

You blocked from doing the following during the Project Copy:

- Performing a full or incremental scan on the source project.
- Uploading a codebase to the source project.
- Accessing the newly created target project. Its dashboard remains blank (except for progress information about the Project Copy) until copy process is complete.

Initiating the Copy Process

Use the following procedure to initiate a Project Copy process to copy the information from a scanned project (source) to a new project (target). As part of the copy process, Project Copy will create the target project with the same project settings as the source project and with the name you provide.

Task ***To initiate a Project Copy process, do the following:***

1. Navigate to the **Summary** tab of the source project (see Opening the Project Summary Tab).
2. From the **Manage Project** menu, select **Copy Project.** If any of the conditions listed in Required Conditions for Accessing the Project Copy Feature are not met, an error message is displayed describing the issue. You cannot proceed with the copy until all conditions are met.

   Note ▪ *The **Copy Project** option is disabled if an **Update Notices**, **Export Project Data**, **Apply Policy - Project**, or **Import Project Data** job is currently scheduled or running.*

3. In the **New Project Name** field on the **Project Copy** window, enter the name you want to give the target project that the copy process will create to receive the copied information. The project name must meet these criteria:

   - It must be a unique project name within your Code Insight system.
   - The initial character in the name cannot be the space character.

   Note ▪ *Code Insight determines whether the target project name is unique during the actual copy process (that is, while the **Project Copy** job in the **Jobs** queue is in an **Active** state). If the name already exists, the job fails with an error message. See step 5 for information about monitoring the **Jobs** queue.*

4. Click **OK**. A success message (displayed in the upper right of the screen) both indicates that the Project Copy was added to the **Jobs** queue and provides the job ID.

   You can monitor the Project Copy progress (see the next step) or proceed to perform other operations on the source project.

5. Monitor the progress of the Project Copy from one or both of the following locations. (For information about the types of information copied to the target project, see Information Copied to the Target Project.)

    - **Jobs queue**—Open the **Jobs** queue and use the job ID to locate the **Project Copy** job and track its status. (Use the instructions in Monitoring the Code Insight Jobs Queue to access and monitor the queue.) The **Jobs** queue enables you to see the status and position of the copy job in relation to other jobs running in the Code Insight system. The copy job can run concurrently with most jobs, including scans on other projects, with the following exceptions:

        - Multiple Project Copy jobs for the same source project can be queued but cannot run concurrently. The first copy job triggered for the project will be the first placed in **Active** state. The remaining copy jobs triggered for the project continue in a **Scheduled** state and are run based on the appropriate queue order.

        - If a **Project Copy** job is already scheduled when a Library Refresh or Electronic Update is added to the queue, the Library Refresh or Electronic Update will run first (once any already active jobs finish). The **Project Copy** job remains in a **Scheduled** state and will run according to queue order once the refresh or update is complete. (A Library Refresh and Electronic Update have priority over all other scheduled jobs.)

    - **Target project dashboard**—Navigate to the dashboard of the target project (which is created once the job has an **Active** status in the **Jobs** queue), and check the description of the copy's progress displayed on the dashboard. You might need to refresh the dashboard to obtain the latest information.

    ***Note** ▪ While you can view progress information on the dashboard of the target project, you cannot open the project until the copy is complete.*

Project Copy Completion

Once the copy process is finished, the **Completed** status is displayed in the **Jobs** queue. (If the process has failed, see Errors During a Project Copy for more information about full and partial failures and possible remediation.)

Additionally, the dashboard for the target project is refreshed with the information and statistics of the scanned data (as copied from the source project). You can then access the target project to perform regular project and inventory operations.

Information Copied to the Target Project

The following table describes the information that Project Copy copies (or does not copy) from the source project to the target project.

Table 2-22 ▪ Information Copied to the Target Project During a Project Copy

Project Information	How Handled During Project Copy
Project settings	The following describes the project settings that are copied/not copied from the source project to create the target project. The following settings descriptions are organized by the tab labels used in the **Edit Project** window in the UI.

Table 2-22 ▪ Information Copied to the Target Project During a Project Copy (cont.)

Project Information	How Handled During Project Copy	
	General	All settings on the **General** tab are copied to the target project. The project is created in the same folder (or in the root) as the source project.
	Scan Settings	All scan settings on the **Scan Settings** tab are copied to the target project, including the selected codebase path for the scan.
	Version Control Settings	All SCM instances defined for the source project on the **Version Control Settings** tab are automatically copied to the target project. Note that the copy process does not test the instance's connection to the remote repository nor synchronize with the repository, as is done when SCM instances are actually created in a project.
	Review and Remediation Settings	All automated and manual review options and remediations options (including the legal, security, and developer contact information) on the **Review and Remediation Settings** tab are copied to the target project.
	ALM Settings	All Application Life Cycle (ALM) instances and their properties defined on the **ALM Settings** tab are copied to the target project.
	Project Hierarchy	Only child-project definitions and links in the source project are copied to the target project. No parent project information is copied.
	Custom Fields	Custom fields and their values in the source project are copied to the target project. A custom field with no value in the source project has no value in the target project.
Project users	The following describes how user information in the source project is handled in the target project: ● Users assigned to the Project Administrator, Analyst, Reviewer, and Observer (for a private project) roles in the source project will be copied to the same roles in the target project. ● The Project Contact in the source project is not copied to the target project. Instead, the person who initiated the Project Copy is designated as the Project Contact in target project. ● Default project role assignments at the global level are not copied to the target project. Only the user assignments that currently exist for roles in the source project are copied (which might include default users from the global level if these defaults were never changed in the source project).	

Table 2-22 ▪ Information Copied to the Target Project During a Project Copy (cont.)

Project Information	How Handled During Project Copy
Project provenance	The **Provenance** field on the **Summary** tab in the target project is updated to show the hyperlinked name of the source project used in the Project Copy: **Copied from** <sourceProjectName> **(Id:** <sourceProjectID>**)** When you click the link, you navigate to the **Summary** tab of the source project. (If the source project has been deleted, a message is displayed, stating that the project no longer exists. Once clicked, the link for the deleted project is permanently disabled.)
Source code	Source code is copied from the source project to the target project as follows: ● Whether uploaded or the result of using SCM synchronization, the physical source code in the source project's scan path folder is copied to the scan path folder in the target project. That is, an exact copy of the source files in the source project will be present in the target project once the copy is complete. ● If a source-code file in the source project contains symbolic links, the links are copied with the file to the target project. ● The properties of a source code folder in the source project are preserved in the target project. ● If a source-code folder in the source project contains shortcuts, the shortcuts are copied with the folder to the target project. ● If a source-code folder is hidden in the source project's scan path, the hidden folder is copied along with the folder's other directories and subdirectories to the target project.
Files	The following file-system information resulting from the most recent scan (Scan Server or a remote scan agent or both) is copied from the source project to the target project: ● Folders and files scanned by the Scan Server or remote scan-agents ● Scan roots (for the Scan Server or remote scan agents) ● The alias for a file system scanned by a remote scan agent ● Metadata for physical files obtained through an upload or SCM synchronization ● File-system hierarchy (so that file paths are maintained under the root, inside folders and sub-folders, and so on)

Table 2-22 ▪ Information Copied to the Target Project During a Project Copy (cont.)

Project Information	How Handled During Project Copy
Scan history	The only scan history copied to the target project is information about the last successful server and/or remote scan on the source project. However, this information is accessible from only the Code Insight database, not the UI. Consequently, no details about the most recent scan (by the Scan Server and/or scan agents) are initially displayed in the **Scan Status** section of the **Project Summary** tab in the target project; and, when you click the link to show past server scans, the resulting **Scan History** window is empty. Note, however, that the details and history for subsequent scans on the target project will be stored in the database and be viewable from the **Summary** tab. ***Note*** *▪ Any subsequent scan run on the target project is considered a rescan (incremental or forced full).*
Evidence	All evidence discovered in the source project's files during the project's most recent scan (by a Scan Server or a remote scan agent) is copied to the target project. This evidence includes licenses, copyrights, search terms, email addresses, URLs, and source-code matches. Note the following about the evidence copied to the target project: ● Scanned files are visible in the **Code Files** pane in the Analysis Workbench. ● Partial evidence in file content in the **Analysis Workbench** is highlighted as expected. ● Audit and review information for files is copied. ● File filters based on evidence will operate as expected during file searches.

Table 2-22 ▪ Information Copied to the Target Project During a Project Copy (cont.)

Project Information	How Handled During Project Copy
Inventory and file association	All inventory items are copied, including each item's name, description, ID, published status, review status, alerts, workflow URL and request details, and custom field values. This information is displayed on the target project's **Project Inventory** tab and the **Analysis Workbench**. The following also takes place during the copy process: ● The mappings of all inventory dependencies in the source project are maintained in the target project. ● Information about the inventory owner is copied. (An *inventory owner* is the entity that discovered or identified the inventory item—such as Code Insight's automated analyzer or detection-rule facility during a scan, or the user who created the custom inventory item.) ● An inventory item's file associations in the source project are copied to the target project. (This includes associations with remotely scanned files and folders.) ● The **Created** and **Updated** dates for an inventory item in the target project are automatically updated to reflect the date of the Project Copy. The inventory's published date is also updated with this date in the Code Insight database (but no corresponding field is provided in the UI). ● The **Provenance** value for an inventory item in the target project is automatically updated to identify (and provide a link to) the item's immediate predecessor, which, in this case, is the inventory item in the source project. (The linked **Provenance** value for an inventory item enables you to trace the origin of the item through its chain of predecessors.) Note that the following inventory details are *not* handled by Project Copy: ● The open and closed tasks assigned to an inventory item are not copied to the target project. ● After the Project Copy, the **Inventory History** window for an inventory item in the target project will list the Project Copy event only. No history for the item from the source project is copied.

Errors During a Project Copy

The following sections describe the possible errors encountered during Project Copy process:

- Errors Resulting in Failure of the Entire Project Copy
- Errors Resulting in a Partial Copy
- Post-Failure Steps

Errors Resulting in Failure of the Entire Project Copy

If the Project Copy encounters an error while performing any of the following operations, the entire copy process is stopped with a **Failed** status in the **Jobs** queue and on the target project's dashboard:

- Initiation of the copy process
- Validation that the target project name is unique in the system
- Copy of project settings
- Copy of information about the last successful scan
- Copy of physical source code

Errors Resulting in a Partial Copy

If the sub-operation that copies any of the following entities fails, the entire Project Copy process fails (with a **Failed** status); but any data copied up to the point of error is retained.

- File data
- Inventory
- Inventory alerts
- Inventory workflow
- Inventory custom fields
- Inventory file associations

Post-Failure Steps

When a Project Copy fails, the target project's dashboard indicates which copy operation failed but does not display scan statistics as it normally would after a successful copy. You can still explore the project's **Summary** tab, **Project Inventory** tab, and **Analysis Workbench** to view what was copied.

To attempt the Project Copy again, navigate to the target project's **Summary** tab and delete the project. The source project remains available, enabling you to remediate the copy issue and run the Project Copy again.

Creating a Private Project

When a project is created, the default visibility for the project is **Public**, which means that any Code Insight user has read-only access to the project. To what degree a user can interact with the project depends on whether the user has a project role and what the role is. (See Creating a Project for instructions on creating a public project.)

Security-conscious project creators can control access to their projects within the enterprise by setting a project's visibility to **Private**. This feature gives project creator the ability to hide sensitive information from general view and select specific users who can view the project. Private projects are hidden from all users except the Project Contact and those users assigned as Project Administrators, Analysts, Reviewers, or Observers of the project. Additionally, project and vulnerability ID searches will not return private projects unless the user performing the search has the permissions to see a given private project.

When a private project is created, the creator automatically becomes the Project Contact and is assigned to the Project Administrator, Analyst, Reviewer, and Observer roles. These roles enable the creator to initially manage the project and its users, analyze the project codebase, and review project inventory. However, creators can remove themselves from any of these roles to let others handle project responsibilities.

Note ▪ *Users who have System Administrator privileges but are not part of a Private project can see the project in the list of projects in the* ***Projects*** *view, access the* ***Summary*** *tab for the project, and change the project contact.*

For information roles, see Assigning and Removing Project Users.

Task ***To create a private project, do the following:***

1. Navigate to the **Projects** view. (See Opening the Projects View if additional instructions are needed.)
2. In the **Projects** pane on the left, click **Add New**. The **Add Project** dialog appears with default values appearing in all the fields but **Name**.
3. In the **Name** field, enter a name for the new private project.
4. From the **Project Visibility** dropdown list, select **Private**.
5. Complete the other fields as described in About Code Insight Projects.
6. Click **Save** to save the new private project.

   This project is visible in the list of projects to only the Project Contact and any Project Administrator, Analyst, Reviewer, or Observer of the project. Additionally, the project and vulnerability ID searches will not return private projects unless the user performing the search has the permissions to see these projects.
7. (Optional) Assign roles to users who will interact with the private project. For more information, see Assigning and Removing Project Users.

Forcing an Automatic Review of All Inventory in a Project

Code Insight provides an Apply Policy feature that enables you to force an automatic review of all published inventory in a given project based on the project’s associated review policy profile. In this way, if the policy profile has been modified, users do not have to wait for a scan or manually unpublish and re-publish inventory items individually to apply the changed policy across project inventory items to automatically approve or reject them.

During the automatic review, the project’s review policy is applied to all project inventory. If the inventory item meets at least one of the criteria, the item is assigned an **Approved** or **Rejected** status, overwriting the current status. The review status of those inventory items that meet no policy criteria remain as is. (Note that the user can configure the review *not* to overwrite any current status that was manually set for inventory.)

The following topics provide more information about forcing the automatic review:

- Required Permissions for Forcing an Automatic Review
- Initiating the Automatic Review of Project Inventory

For more information about review policy profiles, see Managing Policies to Automatically Review Inventory.

Required Permissions for Forcing an Automatic Review

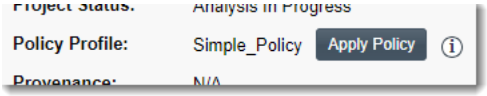
To perform this feature, a user must have **Reviewer** permissions on the project. For any other user of the project (including the project's analysts, observers, contacts, and administrators as well as non-users of the project), the **Apply Policy** button is disabled. (A tooltip for the disabled button explains the same.)

Initiating the Automatic Review of Project Inventory

Use the following procedure to perform an automatic review of all inventory in a given project.

Task ***To apply the current review policy for a project to all published inventory in the project, do the following:***

1. Navigate to the **Summary** tab of the source project (see Opening the Project Summary Tab).
2. In the **Project Details** pane, locate the **Policy Profile** field, which shows the review policy profile associated with the project.

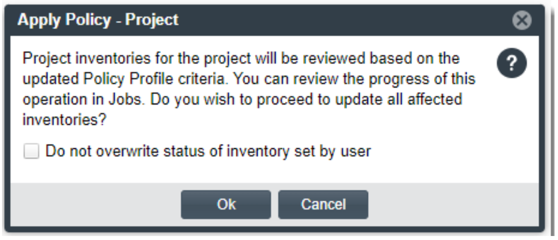


Project Status: Analysis in Progress
Policy Profile: Simple_Policy
Provenance: N/A

3. (Optional) To review the latest criteria in the policy profile, click the **Information** icon to the right of the field. (The contents of the profile are read-only.)
4. To force an automatic review of inventory in the project, click **Apply Policy**.

Note ▪ *This button is disabled if you do not have* ***Reviewer*** *permissions on the project or if a scan, rescan, Project Copy, or Project Branch is scheduled or in progress.*

The **Apply Policy - Project** confirmation pop-up window is displayed, explaining the operation and providing the option *not* to overwrite a current review status that was manually set for an inventory item within the project.



Apply Policy - Project

Project inventories for the project will be reviewed based on the updated Policy Profile criteria. You can review the progress of this operation in Jobs. Do you wish to proceed to update all affected inventories?

□ Do not overwrite status of inventory set by user

5. Do one of the following:
   - Leave the option unselected if you want the automatic review to overwrite the current status of all inventory that meets at least one of the policy criteria. (Default)
   - Select the option if you want the automatic review to *not* to overwrite any current status that was manually set for an inventory item. For all other inventory, their status is overwritten if they meet at least one of the policy criteria.
6. Click **OK** to continue with the operation (or **Cancel** to discontinue and close the pop-up).

If you selected **OK**, a success message (displayed in the upper right of the screen) indicates that the **Apply Policy - Project** job was triggered and provides the job ID.

Additionally, the **Apply Policy** and **Start Scan** buttons—as well as the **Branch Project** and **Copy Project** options on the **Manage Project** menu—are disabled on the **Summary** tab until the current automatic review is complete.

7. To monitor the progress of the Apply Policy operation, open the **Jobs** queue and use the job ID to locate the **Apply Policy - Project** job and track its status. (Use the instructions in Monitoring the Code Insight Jobs Queue to access and monitor the queue.) Consider the following about the queue process for this job:

    - This job will execute immediately as long as no other jobs are in an **Active** (currently running) or **Scheduled** state. See the next bullet for the exception.
    - If a scan for another project is currently active when the **Apply Policy - Project** job is queued (and no other jobs are currently scheduled or active), the **Apply Policy - Project** job is placed in an **Active** state and runs concurrently with the scan.
    - If Core Server jobs (including **Apply Policy - Project** jobs for other projects) are currently active or scheduled when the **Apply Policy - Project** job is added to the queue, the job is placed in a **Scheduled** status and automatically runs once these other jobs are complete.

      Note *▪ Core Server jobs include all job types except scans and rescans.*

    - Once the **Apply Policy - Project** job is scheduled or in progress, other Core Server jobs added to the queue are placed in **Scheduled** status and will run according to queue order after the **Apply Policy - Project** job completes.
    - If an **Apply Policy - Project** job is already scheduled when a Library Refresh or Electronic Update is added to the queue, the Library Refresh or Electronic Update will run first (once any already active jobs finish). The **Apply Policy - Project** job remains in a **Scheduled** state and will run according to queue order once the refresh or update is complete. (A Library Refresh and Electronic Update have priority over all other scheduled jobs.)
    - Should a server shutdown occur when an **Apply Policy - Project** job is running, that job fails. Jobs in **Scheduled** status (including other **Apply Policy - Project** jobs) at the time of shutdown do not fail. Once the server restarts, the scheduled jobs will run according to queue order.

After the Job Completes

Any change in the review status for a given inventory item that occurred during the automatic review is tracked in the item’s **Inventory History**.

Currently, no tasks (including remediation tasks) are automatically created for inventory items once the automated review completes.

Deleting a Project

Project Administrators can use this procedure to delete any of their projects. When a project is deleted, the following components are deleted:

- **From the Code Insight database**—The project record and all scan results, inventory, alerts, tasks, and user audit work associated with the project.
- **From the Scan Server**—The project's codebase files.

Task ***To delete a project, follow these steps:***

1. As Project Administrator, perform either step:
    - Navigate to the **Summary** tab (see Opening the Project Summary Tab), and from the **Manage Project** menu, select **Delete Project**.
    - Navigate to the **Projects** view (see Opening the Projects View), right-click the project in the project display, and select **Delete Project**.
2. When prompted, select **Yes** to proceed with the deletion. The project deletion process is added to the **Jobs** queue with the message in the upper right of the current screen: "Project will be deleted in the background with the job ID: <jobId>".

    Note ▪ *If the Scan Server associated with the project is temporarily inactive or is disabled, a pop-up is displayed to inform you that the server is down. The deletion process does not proceed.*

3. To view the status of the project deletion, open the **Jobs** queue and locate the job ID. For instructions, see Monitoring the Code Insight Jobs Queue.

Impact of Other Jobs on Scheduling Project Deletions

The following describes how the scheduling of a project deletion can be impacted by other Code Insight jobs.

Multiple Project Deletions Triggered

When multiple project deletions have been triggered simultaneously, the first deletion triggered is in the **Active** state in the **Jobs** queue. The remaining deletions are placed in the **Scheduled** state and are run based on the scheduled order.

Electronic Update or Library Refresh Running

If a deletion job is already scheduled when a Library Refresh or Electronic Update is added to the queue, the Library Refresh or Electronic Update will run first (once any already active jobs finish). The deletion job remains in a **Scheduled** state and will run according to queue order once the refresh or update is complete. (A Library Refresh and Electronic Update have priority over all other scheduled jobs.)

Scan Running on the Same Project

A project deletion task cannot run on a project currently being scanned.

- When you attempt to trigger the deletion from the project's **Summary** page, the **Manage Project** > **Delete Project** option is disabled.
- If you trigger the deletion from the project's right-click menu in the **Projects** view, the deletion job is added to the **Jobs** queue. However, its final status shows as **Failed**, and the **Error** column for the job states "You cannot delete this project or scan folder while there are tasks scheduled or active for the project".

Once the scan or rescan is complete, you can re-trigger the deletion task.

Managing Policies to Automatically Review Inventory

This section describes the purpose of review policies in Code Insight and how to manage them. The topics are covered in this section:

- Understanding Policy Profiles
- How Policy Profiles Work in the Automated Inventory-Review Process
- Opening the Policy Page
- Adding or Editing a Policy Profile
- Viewing an Existing Policy
- Copying a Policy Profile
- Associating a Policy Profile with a Project
- Managing Policies to Automatically Review Inventory
- Forcing an Automatic Review of Inventory Across All Projects

Understanding Policy Profiles

Policy profiles are used by Code Insight to automate the inventory review process—that is, automatically mark published inventory items as approved or rejected—without the need for a manual review. (Inventory that are neither approved or rejected by policy will require a manual review.) Policy profiles can be defined up-front or revised during the manual inventory review process. The Code Insight Administrator grants the **Manage Policy** role to users who have rights to manage policy profiles. Typically, these would be legal or security experts.

Code Insight provides a default policy profile (called *Default License Policy Profile*) that can be used as is, modified, or copied to fit your need. This policy profile contains typical settings for a team who is distributing software. You can also create policies from scratch.

How Policy Profiles Work in the Automated Inventory-Review Process

A policy profile is a set of policies whose criteria is based on OSS or third-party component versions, licenses, or security vulnerability score and severities. A given policy profile can be associated with one or more projects to enable automatic reviews of inventory items within any of these projects. (These reviews are triggered by a number of different events described later in this section.)

During a review, the policy criteria are evaluated against a given published inventory item to automatically approve or reject the inventory item. Any conflicting criteria are resolved in favor of an automated rejection of the inventory item. In other words, the rejection per a single criterion will result in an overall rejection of an inventory item despite the number of approvals per other criteria.

When published, if an inventory meets no criteria in the policy, the system can leave the inventory item in a **Not Reviewed** state, thus requiring the inventory to be manually reviewed.

The following sections provide more information the application of the review policy:

- Events Triggering an Automatic Review of Inventory
- User Actions Triggering an Automatic Review of Inventory
- Further Automation of the Inventory Review Process

Events Triggering an Automatic Review of Inventory

In general, whenever an inventory item is published either manually or during a scan or rescan, an automated review by policy takes place. Additionally, any inventory updated during a scan, rescan, Electronic Update, or Library Refresh is automatically reviewed.

User Actions Triggering an Automatic Review of Inventory

The following user actions also trigger an automatic review of inventory.

- Saving any updates to the component, version, license, or usage information in an existing published inventory item. See Editing Inventory from the Analysis Workbench and Editing Inventory from the Project Inventory Tab.
- Creating an inventory item from the **Project Inventory** tab. (The item is automatically published, thus triggering an automatic review.) See Creating Inventory from the Project Inventory Tab.
- Initiating an **Apply Policy** job (as needed) to automatically review the inventory in a given project against the current review policy with which the project is associated. See Forcing an Automatic Review of All Inventory in a Project.
- Initiating an **Apply Policy** job (as needed) to automatically review the inventory across all projects associated with a given policy profile in the Code Insight instance. See Forcing an Automatic Review of Inventory Across All Projects.

Because these actions trigger can an automatic review, users do not have to manually unpublish and re-publish individual inventory items for an immediate application of the latest policy. Nor do users have to wait for the next scan (or such event) for an automated review process across inventory.

Further Automation of the Inventory Review Process

Users can further automate the review workflow by configuring project-level parameters that determine the actions Code Insight takes once inventory is rejected or given a **Not Reviewed** status during an automatic review by policy. For example, a remediation or review task can be automatically created for such inventory. See Updating Inventory Review and Remediation Settings for a Project for more information.

Opening the Policy Page

Use the following procedure to open the **Policy** page, which provides access to the functionality needed to manage policy profiles. The procedure assumes that you have logged into Code Insight.

Task ***To open the Projects view, do the following:***

Open the **Policy** page using either of these methods:

- From the Code Insight dashboard (which is displayed when you start Code Insight), click **view policy**. See Opening Code Insight for details on accessing the dashboard.
- Click the ☰ icon in the upper right corner of the Code Insight web page to open the Code Insight main menu. Select **POLICY** from the menu.

The **Policy** page is displayed, showing a list of all available policy profiles. For more information about this page, see Policy Page.

Note ▪ *If the Code Insight System Administrator has changed Code Insight's CVSS version configuration (for example, from CVSS v2.0 to 3.x), policies defined for these profiles might have automatically changed based on the new version's scoring system. For more information, see Impact on Policies When Code Insight's CVSS Configuration Changes for details.*

Adding or Editing a Policy Profile

The following procedure describes how to add a new policy profile or edit an existing one. Only users who have Policy Manager permissions can create or edit a policy.

Task ***To add a new policy profile or edit an existing one, do the following:***

1. Open the **Policy** page (see Opening the Policy Page).
2. To edit an existing policy profile, select it from the list, and click the **Edit** icon ✎.

   or

   To add a new policy profile, click **Add Policy**.

   The **Policy Details** window for the policy profile is displayed.

3. Refer to the associated help (or to Policy Details Window) for details about the fields used to define the policy profile.

4. Click **Save** to save the updates or to add the new policy profile.

Viewing an Existing Policy

The following procedure describes how to open a read-only version of a policy profile defined on your Code Insight instance. This version provides a quick way to scan the policy profile details without having to scroll through all the controls and space required when updating a profile in the editable version.

This function is available to all users, whether or not they have Policy Manager permissions.

Note **▪** *In addition to the procedure described here, users can also open a read-only version of the policy profile associated with a given project by clicking the* ***View Policy Details*** *link on the project's* ***Summary*** *tab.*

Task ***To view an existing policy profile, do the following:***

1. Open the **Policy** page (see Opening the Policy Page).

2. From the policy list, select the policy profile you want to view, and click **View** icon .

   A read-only version of the **Policy Details** window for the profile is displayed.

3. Refer to the associated help (or to Policy Details Window) for details about the fields used to define the policy profile.

4. When you have finished reviewing the policy profile, click **Close**.

Copying a Policy Profile

The following procedure describes how to create a copy of an existing policy profile. This can be useful for creating a template that be used to create other policy profiles or for backing up an existing profile.

Only users with Policy Manager permissions can perform this function.

Task ***To copy an existing policy, do the following:***

1. Open the **Policy** page (see Opening the Policy Page).

2. From the policy list, select the policy profile to copy, and click **Copy** icon .

   The **Policy Details** window opens, showing a new policy profile called **Copy of** *`policy name`* and having the same policies has the original profile. You can then edit the new profile to change its name or update its policies. See Adding or Editing a Policy Profile.

Associating a Policy Profile with a Project

The Project Administrator can associate a project with a policy profile by editing the project (see Updating Inventory Review and Remediation Settings for a Project). If no policy is explicitly selected for a project, the *Default License Policy Profile* is used.

Forcing an Automatic Review of Inventory Across All Projects

Code Insight provides an Apply Policy feature that enables you to force an automatic review of published inventory in all projects currently associated with a given policy profile. In this way, if the policy profile has been modified, users do not have to wait for a scan on each project or manually unpublish and re-publish individual inventory items within each project to apply the changed policy across inventory items to automatically approve or reject them.

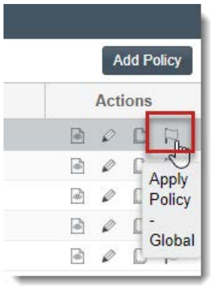
During the automatic review, the status of those inventory items that meet any of the current policy criteria will be overwritten based on the criteria. (These items include those with a status manually set by a user.) The status of those inventory items that meet no policy criteria will remain as is.

Important *▪ Currently this feature is available for only Code Insight instances that use the MySQL database. It is not available those instances that use the SQL Sever database. Additionally, a special server deployment configuration is recommended for running this feature. Refer to the “Supported Deployment Configurations” section in the Code Insight Installation & Configuration Guide.*

To perform this feature, a user must have the system permission **Manage Policy**.

Task ***To apply a given new or modified policy profile to inventory in all projects associated with the profile, do the following:***

1. Open the **Policy** page (see Opening the Policy Page).
2. From the policy list, select the policy profile that you want to apply to inventory across all projects currently associated with it.
3. Click **Apply Policy - Global** icon for the policy. (This icon is not available if you do not have **Manage Policy** permission.)



The **Apply Policy on All Projects** confirmation pop-up window is displayed.

4. On the pop-up, click **OK** to proceed with the Apply Policy operation or **Cancel** to discontinue it. See the note below.

    Important ▪ *Once review begins, all inventory that meet the policy criteria are reviewed by the selected policy and their current review status is overwritten. This includes inventory whose current review status was set manually by a user. Currently no option exists to skip such inventory items during the review process. If you do not want the operation to overwrite statuses that were manually set, click* ***Cancel****.*

    If you selected **OK**, a success message (displayed in the upper right of the screen) indicates that the Apply Policy job was added to the **Jobs** queue and provides the job ID.

5. To monitor the progress of the Apply Policy, open the **Jobs** queue and use the job ID to locate the **Apply Policy - Global** job and track its status. (Use the instructions in Monitoring the Code Insight Jobs Queue to access and monitor the queue.) Consider the following about the queue process for this job:

    - This job will execute immediately as long as no other jobs in the queue are in an **Active** state (that is, currently running).
    - If jobs are currently active when the **Apply Policy - Global** job is added to the queue, the job is queued with a **Scheduled** status and automatically run after the currently active jobs complete.
    - Other jobs added to the jobs queue once the **Apply Policy - Global** job is scheduled or in progress are placed in **Scheduled** status and will run according to queue order after the **Apply Policy - Global** job completes.
    - If a **Apply Policy - Global** job is already scheduled when a Library Refresh or Electronic Update is added to the queue, the Library Refresh or Electronic Update will run first (once any already active jobs finish). The **Apply Policy - Global** job remains in a **Scheduled** state and will run according to queue order once the refresh or update is complete. (A Library Refresh and Electronic Update have priority over all other scheduled jobs.)

After the Job Completes

Once the **Apply Policy - Global** job completes, any change in the review status of a given inventory item is recorded in that item's **Inventory History**.

Currently, no tasks (including remediation tasks) are automatically created for inventory items once the automated review completes.

Tracking the Progress of an Electronic Update

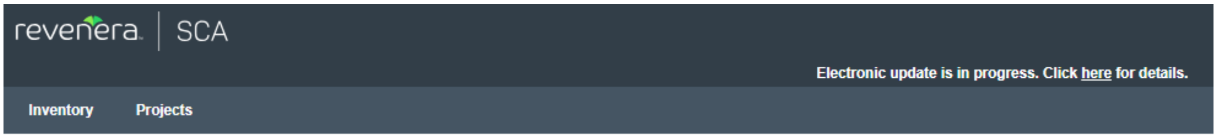
Electronic Updates keep your local Code Insight database up to date with the latest component, version, license, and vulnerability information available in the Code Insight data library. You can schedule these updates to execute automatically at a regular frequency or manually through the Administration interface. More information about Electronic Updates is found in the “Setting Up Electronic Updates” section in the *Code Insight Installation & Configuration Guide*.

Refer to the following sections for more information about how users are notified of a currently running Electronic Update and how they can monitor its progress.

- Electronic Update Notification
- Monitoring the Progress of Electronic Update Phases

Electronic Update Notification

Whenever an Electronic Update is currently running, a notification message is displayed in the heading of the Code Insight user interface, indicating that an update is in progress.



Electronic update is in progress. Click here for details.

Users can track the progress of each phase of the Electronic Update by clicking the “here” link within the notification message, as described in Monitoring the Progress of Electronic Update Phases. Additionally, users can monitor the general state of the Electronic Update (**Scheduled**, **Active**, and so forth) in the Code Insight **Jobs** queue, described briefly in About Electronic Update Jobs.

Note ▪ *The notification message is displayed in the user interface only when an Electronic Update is in the* ***Active*** *state in the* ***Jobs*** *queue. It is not displayed when the update is in the* ***Scheduled*** *state. See About Electronic Update Jobs for more information.*

Once the Electronic Update completes, the notification message is automatically removed. (Users cannot manually remove the message.)

About Electronic Update Jobs

When an Electronic Update is triggered either manually or automatically at its scheduled time, a **PDL Update** job is added to the Code Insight **Jobs** queue. This job will execute immediately as long as no other jobs in the queue are in an **Active** state (that is, currently running). If other jobs are active, the update will be queued with a **Scheduled** status and automatically run after these currently active jobs complete. Additionally, once the Electronic Update is added to the **Jobs** queue (as **Active** or **Scheduled**), any new or already scheduled jobs remain in the **Scheduled** state until the update is complete. The scheduled jobs will then run in appropriate queue order.

Exceptions to the Jobs Queue Process

The following are exceptions to the above **Jobs** queue process when an Electronic Update is added to the queue:

- One or more active scans can still be running at the point when the Electronic Update is placed in an **Active** state. The active update waits for these scans to complete before it actually executes.
- If a **Library Refresh** job is already scheduled when the Electronic Update is added to the **Jobs** queue, the **Library Refresh** job takes priority and runs first while the update is placed in a scheduled state until the refresh completes. The Electronic Update then takes priority.

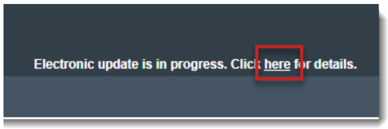
For more information the Electronic Update as job and about the **Jobs** queue in general, see Monitoring the Code Insight Jobs Queue.

Monitoring the Progress of Electronic Update Phases

The following instructions describe how to monitor the progress of each phase in a currently running Electronic Update. This task is performed by clicking the “here” link in the Electronic Update notification message displayed on each Code Insight page when an update is in progress. (For more information about the message, see Electronic Update Notification.)

Task ***To monitor the progress of an Electronic Update, do the following:***

1. Within the Electronic Update notification message, click the underlined **here** link.

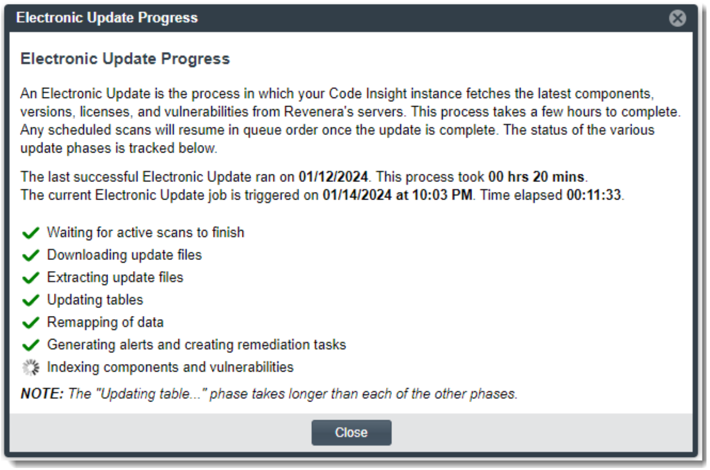


Electronic update is in progress. Click here for details.

The **Electronic Update Progress** pop-up window is displayed, listing the phases of the Electronic Update in sequential order. As each phase completes, it is marked with a green check.

In addition to the progress of the update, the window shows the following:

- The start date of the Electronic Update and its current elapsed time.
- The date on which the last successful Electronic Update was run and its total execution time. If no successful update has run previously, this information is not displayed.



Electronic Update Progress

Electronic Update Progress

An Electronic Update is the process in which your Code Insight instance fetches the latest components, versions, licenses, and vulnerabilities from Revenera's servers. This process takes a few hours to complete. Any scheduled scans will resume in queue order once the update is complete. The status of the various update phases is tracked below.

The last successful Electronic Update ran on **01/12/2024**. This process took **00 hrs 20 mins**.
The current Electronic Update job is triggered on **01/14/2024 at 10:03 PM**. Time elapsed **00:11:33**.

✓ Waiting for active scans to finish
✓ Downloading update files
✓ Extracting update files
✓ Updating tables
✓ Remapping of data
✓ Generating alerts and creating remediation tasks
Indexing components and vulnerabilities

NOTE: *The "Updating table..." phase takes longer than each of the other phases.*

This example window shows the progress of a *server* Electronic Update. The progress for a *local* update does not include the **Downloading**... and **Extracting**... phases. For a description of the various phases and the difference between a server and local update, see Description of the Electronic Update Phases.

2. Click **Close** at anytime to close the pop-up window.

Description of the Electronic Update Phases

The following table provides a description of each Electronic Update phase that is visibly tracked on the **Electronic Update Progress** pop-up window. (This pop-up window is accessed from the Electronic Update notification message that is displayed on each Code Insight page when an update is in progress, as described in Monitoring the Progress of Electronic Update Phases.)

The phases indicated in the table as "Server update only" do not apply to a local update. All other phases apply to both server and local updates. For a information about the differences between these two types of updates, see Differences in Process Between Server and Local Electronic Updates.

Table 2-23 ▪ Description of the Electronic Update Phases

Phase	The Electronic Update is...
Waiting for active scans to finish	Waiting for any active scans to complete before starting. ***Note ▪*** *Normally other jobs are not allowed to run when an Electronic Update is* ***Active****. The one exception is that one or more scans can still be running at the point when the Electronic Update is placed in an* ***Active*** *state. Therefore, this phase is included. For more details, see About Electronic Update Jobs.*
Downloading update files	(Server update only) Downloading the files containing the latest component and CVSS data from Revenera to the Code Insight server.
Extracting update files	(Server update only) Extracting data files from the downloaded `update.zip` file.
Updating table <TABLE_NAME> (x/27)	Updating the twenty-seven Code Insight database tables with the latest component, license, and CVSS data. The name of the table currently being updated is listed, along with its position in the table sequence (for example, table number 11 out of 27).
Remapping of data	Remapping of any custom data that was created before the update but that is now recognized and reinstated by the update. (Data is considered custom if it did not originally exist in the Code Insight database but was manually created by users through Code Insight.) License details are also updated during this phase.
Generating alerts and creating remediation tasks	Processes security vulnerability information and does the following: ● Generates alerts in the user interface for the inventory items associated with the new vulnerabilities. ● Issues email notifications to the project owners, identifying the specific inventory associated with each new vulnerability. ● Creates remediation tasks inventory rejected during this phase, as dictated by the project's policy profile and remediation options (see Updating Inventory Review and Remediation Settings for a Project).

Table 2-23 ▪ Description of the Electronic Update Phases (cont.)

Phase	The Electronic Update is...
Indexing components and vulnerabilities	Indexes all the current entities in the updated Code Insight database.

Differences in Process Between Server and Local Electronic Updates

Code Insight enables you to configure the Electronic Update to run as either a server or local update, as described in detail in the *Code Insight Installation & Configuration Guide*. The basic difference between the two methods is the means by which the Code Insight server obtains the files required to run the update. As a result, the phases monitored in the **Electronic Update Progress** pop-window for these two type of updates are slightly different.

- A **server** Electronic Update performs all phases of the update automatically, including the download of the update files from Revenera. Therefore, all phases available on the **Electronic Update Progress** pop-up window are tracked.
- For a **local** (offline) update, a user must manually download the Electronic Update files from Revenera to a local machine accessible to the Code Insight server. The Electronic Update then locates the downloaded files, uploads them to the Code Insight server, and extracts them. These operations must be completed before the local Electronic Update is eligible for **Active** status. As a result, the initial **Downloading upload files** and **Extracting upload files** phases are *not* monitored in the **Electronic Update Progress** pop-up window.

Managing Authorization Tokens

Code Insight uses a JSON Web Token (JWT) to authorize user access to the Code Insight public REST interface. You might be required to explicitly enter an authorization token for certain functionality that uses this REST interface directly (that is, not through Code Insight web UI), such as the following:

- Project import and export processes (see Exporting and Importing Project Data)
- The execution of remote scan agents (see Performing Remote Scans)

Code Insight enables you to generate and manage one or more of these authorization tokens.

An authorization token is for use by the Code Insight user account that creates it. Thus, an authorization token that your user account generates will give you REST access to only the Code Insight functionality for which your account has permissions. Additionally, you can view and manage only those authorization tokens for the user account under which you are logged in.

Authorization tokens are created and managed from **Preferences** page, as described in the following procedures:

- Accessing the Preferences Page
- Generating an Authorization Token
- Copying the Authorization Token to the Clipboard
- Editing the Token Name
- Deleting an Authorization Token

Accessing the Preferences Page

Use these steps to open the **Preferences** page.

Task ***To open the Preferences page, use these steps:***

1. Click the following icon in the upper right corner of the Code Insight web page to open the Code Insight main menu:

   ☰

2. Select **PREFERENCES** from the menu to open the **Preferences** page.

Generating an Authorization Token

Use the following procedure to generate an authorization token.

Task ***To generate an authorization token, do the following:***

1. Access the **Preferences** page (see Accessing the Preferences Page).
2. From the **AUTH Tokens** pane, click **Add Token**.
3. Enter a name for the new token and specify an expiration date (or choose **Never Expires**).
4. Click **Save**.

Copying the Authorization Token to the Clipboard

Use the following procedure to copy an authorization token to the clipboard so that you can paste it in your REST API interface.

Task ***To copy an authorization token to the Clipboard, do the following:***

1. Access the **Preferences** page (see Accessing the Preferences Page).
2. From the **AUTH Tokens** pane, locate the token you want to copy, and click the **Copy to clipboard** 📋 icon in the **Actions** column.
3. Click **Select Token Text** to select the entire token value, and then press Ctrl + C to copy it.
4. Paste token in the appropriate location for use by the REST interface.

Editing the Token Name

You can edit only the name of an authorization token, not its expiration date or value.

Task ***To edit the token name, do the following:***

1. Access the **Preferences** page (see Accessing the Preferences Page).
2. From the **AUTH Tokens** pane, locate the token you want to edit, and click the Edit ✎ icon.
3. Update the token name as needed.
4. (Optional) To copy the token value to the Clipboard for pasting into the REST interface, click **Select Token Text** to select the entire token value, and then press Ctrl + C to copy it.

Deleting an Authorization Token

Use the following procedure to delete an authorization token.

Task ***To delete an authorization token, do the following:***

1. Access the **Preferences** page (see Accessing the Preferences Page).
2. From the **AUTH Tokens** pane, locate the token you want to edit, and click the Edit ✖ icon.

Downloading Code Insight Log Files

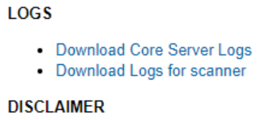
Code Insight allows Code Insight System Administrators to download Code Insight log files that have been generated for the Core Server and each Scan Server. The downloads are in `.zip` format, enabling you to easily distribute log files as needed for analysis or troubleshooting purposes.

Task ***To download log files, do the following:***

1. Log into Code Insight as a Code Insight System Administrator.
2. Click the following icon in the upper right corner of the Code Insight web page to open the Code Insight main menu:

   ☰

3. Select **HELP** from the menu to open the **Help** page.
4. Navigate to the **Logs** section, and click the link for the type of logs you want to download.



LOGS

- Download Core Server Logs
- Download Logs for scanner

DISCLAIMER

The following logs are available:

- **Download Core Server Logs**—The logs generated by the Code Insight Core Server. These include `core.log` and `core.update.log`.
- **Download Logs for {scannerName}**—The logs for the Scan Server whose name is specified in the link label. (A separate download link is generated for each Scan Server configured in your Code Insight system.) Scan Server logs include Tomcat logs, as well as `codeaware.log`, `codeware.update.log`, `scanEngineDetail.log`, and possible archived versions of these logs.

3

Performing Advanced Searches

The following sections describe Code Insight's advanced inventory searching capabilities:

- Advanced Inventory Searches
- Dependencies in Advanced Searches

Advanced Inventory Searches

You can perform simple searches on the **Project Inventory** tab or in the **Analysis Workbench** to find inventory items whose names match text strings (or, in the **Analysis Workbench**, to display only published or non-published inventory). However, Code Insight also provides an Advanced Inventory Search feature that enables you to refine an inventory list to focus on specific inventory. (For details about this feature, see Advanced Inventory Search Dialog, Searching Published Inventory, or Performing Inventory Searches in the Analysis Workbench.)

The Advanced Inventory Search feature offers many combinations of search criteria. The following table, which is arranged by persona (job function or department), presents a number of advanced searches and their typical results.

Table 3-1 ▪ Sample Advanced Searches

Persona	Search Type	Finds...	Example
Any	By inventory, component, or license keyword	Inventory of interest based on full or partial inventory, component or license name. Useful when you want a quick search for a specific component or license across all of your inventory items.	Inventory Name = *zlib 1.2.8 (zlib/ libpng License)* Inventory Name = *zlib* License Name = *EPL*

Table 3-1 ▪ Sample Advanced Searches (cont.)

Persona	Search Type	Finds...	Example
Any	By criticality (priority)	Most critical inventory that requires security or legal review based on presence of high-severity vulnerabilities or P1 licenses. Useful when you want to prioritize your inventory review by most important findings.	**Option 1:** Inventory Priority = *P1* **Option 2:** Security Vulnerability Severity = *High* or *License Priority = P1 - Viral/Strong Copyleft*
Any	By manually created inventory "finds" not associated with a component-version-license instance	Inventory items whose type is **License Only** or **Work in Progress** (or **Component**). Helpful to locate those manually created inventory items that still need further investigation and a solid component-version-license association before they can be published or removed.	Inventory Type = *Work in Progress*
Any	By review status	Inventory whose status is **Approved**, **Rejected**, or **Not Reviewed** (requires further review. Useful to identify items that are yet to be reviewed. Also when you are further qualifying other search criteria with an additional expression based on review status.	Inventory Review Status = *Approved*
Any	By dependencies	Only dependency inventory items (both first-level and transitive dependencies), only top-level inventory items (excluding all dependency inventory items), or all inventory items. Useful for focusing on or filtering out dependency inventory items.	Dependency Options = *All Inventory Items* Dependency Options = *Only Top-Level Inventory Items* Only Dependency Inventory Items = *Only Dependency Inventory Items*
Any	By dependency scope	Those dependencies required at runtime (**Runtime**) or those dependencies not required at runtime (**Non-Runtime**) or dependencies of any scope, including **N/A**. (**All**).	Dependency Scope = *Non-Runtime*

Table 3-1 ▪ Sample Advanced Searches (cont.)

Persona	Search Type	Finds...	Example
Any	By inventory age	Inventory created within the specified time range. Useful to filter to recent inventory items, which is especially valuable when a user logs into Code Insight at a regular interval (daily, weekly, etc.).	Inventory Age = *Last 7 Days*
Any	By notification	Published inventory items that have new security vulnerability alerts or that have been rejected due to new non-compliant security vulnerabilities. You can select one or both options. Useful for filtering to published inventory items that have important new security information or that have been rejected due to new security issues that are non-compliant with policy.	Inventory with Open Alerts = *checked* Inventory Rejected Due to New Non-Compliant Security Vulnerabilities = *checked*
Any	By task status	Inventory tasks by their Open or Closed status. Useful for determining the work required before the inventory review process can be completed. Also useful for locating inventory whose closed tasks might need to be reopened for extra work.	Task Status = *Open* Task Status = *Closed*
Any	By task type	Inventory tasks by their type. Useful for filtering to inventory that requires a manual legal or security review (**Manual Inventory Review**), source-code changes to make it compliant or secure (**Remediate Inventory**), or another type of effort (**Miscellaneous**).	Task Type = *Manual Inventory Review* Task Type = *Remediate Inventory* Task Type = *Miscellaneous* Task Type = *Any*

Table 3-1 ▪ Sample Advanced Searches (cont.)

Persona	Search Type	Finds...	Example
Any	By inventory task age	Inventory tasks created within the specified date range. Useful for keeping track of new work to be performed on inventory and old work still needs to addressed.	Inventory Tasks Age = *Last 7 days* Inventory Tasks Age = *Custom Date Range From: 09/05/2018 To: 10/31/ 2018*
Any	By inventory task owner	Inventory tasks owned by a specific user. Useful for determining the workload of a specific user.	Inventory Tasks Owner = *Any* Inventory Tasks Owner = *Only mine* (current user) Inventory Tasks Owner = *<Username>* (selected user)
Analyst, Reviewer	By value in custom inventory fields	Inventory whose custom inventory field values contain a specific string.	<customInventoryFieldLabel> Contains <Search Text>
Analyst, Reviewer	By Confidence Level	Inventory generated with a specific level of confidence (High, Medium, or Low). The level is based on the measure of the strength of the discovery technique used to generate the item. (See Inventory Confidence.) Useful for determining whether the item should be triaged or reviewed to validate or further refine the finding.	Inventory Confidence Level = *High* (or *Medium* or *Low*)
Security Analyst	By vulnerability ID	Inventory with a specific vulnerability (NVD CVE or Secunia Advisory). Useful when you are looking for inventory exposing you to a specific security issue, typically a newsworthy event.	Security Vulnerability ID = *SA71946*

Table 3-1 ▪ Sample Advanced Searches (cont.)

Persona	Search Type	Finds...	Example
Security Analyst	By security risk exposure	Inventory containing security vulnerabilities of a specified severity. Useful to filter to inventory items that require immediate attention based on your corporate security policy. For example, we must address all high-severity security issues in the current release.	Security Vulnerability Severity = *High*
Security Analyst	By security vulnerability age	Inventory with new security vulnerabilities since a specified date. Useful to see which inventory items have new security vulnerabilities reported against them based on the specified date range.	Security Vulnerability Age = *Last day*
Security Analyst	By security risk exposure and vulnerability age	Inventory with new security vulnerabilities of a specified severity since a specified date. Useful to see which inventory items have new security vulnerabilities reported against them based on the specified date range and a certain severity.	Security Vulnerability Age = *Last day* and Security Vulnerability Severity = *High*
Security Analyst	By inventory alert	Published inventory items that have new security vulnerability alerts or that have been rejected due to new non-compliant security vulnerabilities. You can select one or both options. Useful for filtering to inventory items that have important new security information or that have been rejected due to new security issues that are non-compliant with policy.	Inventory with Open Alerts = *checked* Inventory Rejected Due to New Non-Compliant Security Vulnerabilities = *checked*

Table 3-1 ▪ Sample Advanced Searches (cont.)

Persona	Search Type	Finds...	Example
Security Analyst	By new vulnerabilities (requires re-review)	Inventory that has gained a new security vulnerability since a specified date. Useful to determine which inventory items require another look from a security analyst due to new associated vulnerabilities.	Review Status = *Approved* and *Security Vulnerability Age = Last 7 days*
Legal	By license risk exposure	Most critical inventory that requires legal review (contains a P1 license - Viral/Strong Copyleft). Useful to prioritize legal work based on license classification.	License Priority = *P1 - Viral/Strong Copyleft*
Analyst	By inventory that requires a re-review based on missing license	Approved inventory with a missing license. Useful to catch scenarios where items were approved without an associated license. This should be a rare event.	Inventory Review Status = *Approved* and License Priority = *No License Found*
Eng. Mgr./ Final Reviewer	Stop shipment!	Approved inventory that may require a stop shipment due to high severity vulnerability or P1 license. Useful to identify cases that would break the build. These are items that were approved at the time of review, but since then have a different license or high-severity vulnerability.	Inventory Review Status = *Approved* or License Priority = *P1 - Viral/Strong Copyleft* or Security Vulnerability Severity = *High*

Dependencies in Advanced Searches

Code Insight is able to scan archived and multi-layer codebases. When inventory items from these codebases are published, dependencies can be published as well. However, when performing searches on published inventory, the amount of data returned can be immense. So it is important to consider whether to include or exclude them in your inventory searches.

4

Exporting and Importing Project Data

The following sections describe the project export and import functionality in Code Insight:

About Exporting and Importing

Code Insight provides export and import functionality for project data. The export-import process can be performed on the same server or across servers. Project data can be quickly imported into an empty project to create inventory or imported into a scanned project to create inventory with file associations. The imported inventory in both cases is “live”—that is, ready to be reviewed and edited.

Export and import functionality is useful in any of these following scenarios:

- **Backup of project and audit data**—Use export to create a full backup of a Code Insight project. The backup data file includes project and scan details, all inventory (with inventory details, field values, file associations, and inventory review status), the review status of files, and any custom data. The project data can be restored to a new project for an archived view or for ongoing scanning and auditing.
- **Copying or branching a project**—Use the export-import process to create an exact copy of a project for future scanning and audit work. To do this, export the data from the source project, scan the target project (pointed to the same codebase), and import the data into the target project. The target project can be used for continued scanning and analysis work while the source project remains unchanged.
- **Versioning a project**—Use the export-import process to apply analysis work performed on one product version to the next product version. For example, you can apply the analysis work performed on project “foo-v1” to project “foo-v2”. To do this, export the data from “foo- v1”, scan “foo-v2”, and import the data into “foo-v2”.

- **Audit work reuse**—As in the versioning example above, you can use the export-import process to apply analysis work performed on one project to another project containing a subset of similar files. Export data from “project1”, scan “project2” (pointed to “project2” codebase), and import the data into “project2”. Likewise, this process can be used to apply analysis work from several different projects to the current project.
- **Sharing of live audit results between teams**—Use the export-import process to share live audit results between teams or with Revenera Professional Services. For example, you can export data from “project 1” on “instance 1” and import the data into “project 2” on “instance 2”. Results are imported either into an empty project for a live view of inventory or into a scanned project for a live view of inventory with file associations and access to the codebase file tree.

  Note ▪ *An empty project is one to which no codebase has been uploaded or synchronized or that has not yet been scanned.*

- **Migrating audited projects from Code Insight v6 to v7**—Use the import process to create live project inventory in Code Insight v7 from an exported v6 project or workspace data. After exporting data from a v6 project, run the Audit Data Migration Tool (available for download in the Product and License Center) to map inventory fields from Code Insight v6 to v7 and to convert the exported data to the proper JSON data format required for import into v7. Then perform an import into a Code Insight v7 project. Import into an empty project (see note for previous bullet) for a live view of inventory only or into a scanned project for a live view of inventory with file associations.
- **Creating inventory from an external system**—Use import to create project inventory in Code Insight from a data file containing legacy or external data. This type of import requires the conversion of the legacy data to the required JSON format prior to importing into the new project.

Export/Import Processes with Legacy Projects

Previous to Code Insight 2020 R3, projects were one of two types—*standard* or *inventory-only*. A standard project managed the results of a server scan—that is, a scan performed by the Scan Server on a codebase that was uploaded or synchronized to the Scan Server. The inventory-only project managed the results of a remote scan, performed by a scan agent on a remote codebase. (The agent sent the scan results to the project on the Code Insight server.) The scan results in both project types included an inventory of open-source and third-party software. Additionally, scan results in the standard project included information about the codebase files, enabling users to perform file operations. The inventory-only project, however, included no information about the remote codebase files.

Beginning with Code Insight 2020 R3, all scanning—server and remote—is accomplished in a single “unified” project, simply called “project” in this documentation. In addition to an inventory of open-source and third-party software, the new project type can include information about codebase files from both server and remote scans, enabling file operations on all project files.

Migration to the New Project Type Introduced in 2020 R3

During an upgrade from a pre-2020 R3 Code Insight release to the current release, standard projects are automatically migrated to the new project type.

Inventory-only projects, now called *legacy projects*, continue to be supported with limitations. For example, these projects support only 2020 R2 or earlier scan-agent plugins and allow imports only from other legacy projects. Note that legacy projects will be deprecated in the future.

If you want to migrate an legacy project to the new project type, see the following Knowledge Base article in the Revenera Community:

https://community.flexera.com/t5/FlexNet-Code-Insight-Customer/Code-Insight-2020-R3-Changes-to-Projects/ta-p/160059

New Export JSON Elements

Starting in Code Insight 2020 R3, the JSON output for an export process run contains the following elements to support remote codebase-file information that can be exported from a Code Insight project created in 2020 R3 or later:

- **remoteAlias**— The unique name for a remote scan agent used by the project
- **remoteScanFolders**—A list containing the scan root for each alias
- **remoteFilePaths** (for inventories)—The list of all remote file paths associated with inventory
- **remoteReviewedPaths**—The list of all reviewed remote files

Prerequisites When Using the REST Interface

The following are prerequisites when using the Code Insight REST interface to execute an export or import:

- REST Client or Command-Line Tool Supporting cURL
- Authorization Token
- Project ID

These prerequisites are not needed if you are performing an export or import using the Code Insight Web UI.

REST Client or Command-Line Tool Supporting cURL

A REST client or command-line interface with cURL support is required to execute the cURL commands that call the REST API to export or import project data.

To download cURL, go to https://curl.haxx.se/download.html. Once cURL is installed, ensure that its path is added to your PATH environment variable so that you can use it with batch or PowerShell scripts and call it from the command prompt of any working directory.

Authorization Token

The export and import REST interface requires a valid JSON Web Token (JWT) for the owner of the project from which data is to be exported or to which data is to be imported, depending on the function being performed. For instructions on obtaining the JWT, see Managing Authorization Tokens. The token is not required when using the Code Insight Web UI to export or import project data.

Project ID

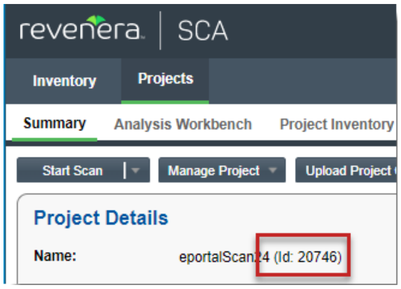
The export and import REST interface requires the ID of the project from which you are exporting data or to which you are importing data, depending on the function being performed. The following procedures describe methods for locating the project ID.

Locating the Project ID in the Code Insight Web UI

The following are some ways to locate the project ID in the Code Insight Web UI.

Task ***To obtain the project ID through the Web UI, do the following:***

Locate the **Name** value on the **Project Summary f**or the project. The ID is displayed in parentheses next to the name:



Name: eportalScan24 (Id: 20746)

Retrieving the Product ID Using the REST Interface

Retrieve the project ID by issuing a cURL command that calls the **Get Project Id** REST API.

Important ▪ *If want to copy and paste the cURL command directly from these instructions, copy it to a text editor first to remove formatting and any line breaks or extra spaces.*

Task ***To obtain the project ID by calling the Get Project Id REST API, do the following:***

Execute the following cURL command to invoke the **Get Project Id** REST API. Replace the highlighted variables with your server host ID (hostname or IP address plus the port), project name, and authorization token.

```
curl -X GET "HOST:PORT/codeinsight/api/project/id?projectName=PROJECT_NAME" -H "accept:
application/json" -H "Authorization: Bearer JWT_TOKEN"
```

The following is an example:

```
curl -X GET "http://localhost:8888/codeinsight/api/project/id?projectName=AllTypes" -H "accept:
application/json" -H "Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsInVzZXJJZCI6MSwiaWF0IjoxNTY2ODU5NTg2fQ.qV2j8ZLgNGNJsT8OdPR
wvE0-0y1x7w-0zr5h7Jz2d9uqY8tvACsV68posEUO9tD-YXlgXznX-IGnrnopDU7G3w"
```

Note ▪ *If the project name contains a space or special character, replace the character with its encoded version. For example, for the project* `project foo`*, you would provide the name* `project%20foo`*, where the space is replaced with the encoded character* `%20`*.*

The response contains the project ID (in this example, 164):

```
{"Content: ":164}
```

Exporting Project Data

The following sections provide the details about exporting project data in Code Insight:

- About an Export
- Types of Data Exported
- Prerequisites for Exporting Data
- Exporting Project Data Using the Web UI
- Exporting Project Data Using the REST Interface

Important ▪ *The instructions in this section assume that you are exporting project data using the Code Insight version for which this documentation was published. If you are exporting project data using another version of Code Insight, refer to the documentation for that version for export instructions.*

About an Export

The Code Insight project-data export feature is available through these interfaces:

- Code Insight Web UI, as described in Exporting Project Data Using the Web UI.
- Code Insight REST interface, as described in Exporting Project Data Using the REST Interface.

During the export process, project data is exported to a JSON data file and then compressed in a `.zip` archive. The archive file can be stored for backup purposes or imported into a project (described in Importing Project Data).

When performed through the Web UI, the export runs as a background process that does not interfere with analysis work, scanning, or other operations.

Restarts of the Code Insight server (or the machine running the server) do not have any impact on saved exported data.

Types of Data Exported

The export always processes project data in full—that is, there is no way to limit the exported data. Thus, the data exported includes the following:

- **Export information**—The project contact, the version of Code Insight in which the export was run, the Compliance Library and Electronic Update versions currently used by the project, the date and time of the export, and more.
- **Project details and scan settings**—The name, description, scan profile, policy profile, and scanroot path for the project.
- **Inventory**—All data for the project's inventory, including inventory fields, publication and review statuses, associated codebase files, and associated repository items.
- **Reviewed files**—The absolute file path and MD5 for each project codebase file marked as reviewed.
- **Custom data**—Custom inventory and any custom components, versions, and licenses to which this inventory is mapped.
- **Custom fields**—Custom field values for inventory and projects.

Note ▪ *Some data exported is for informational purposes only and is not necessarily processed during an import.*

Prerequisites for Exporting Data

To export data, ensure that the following prerequisites are met:

- A Code Insight v7 instance currently running.
- An existing, non-empty project (containing at least one inventory item) on that instance.
- Items listed in Prerequisites When Using the REST Interface (if performing the export using REST API).

Exporting Project Data Using the Web UI

Refer to the following sections for the information needed to export project data using the Code Insight Web UI.

- Using the Web UI to Perform the Export
- Impact of Other Jobs in Progress or Scheduled for the Current Project
- Downloading the Archive Containing the Current Exported Data for the Project

Using the Web UI to Perform the Export

When project data is exported through the Web UI, the export operation is added to the **Jobs** queue and runs in the background. During the export process, the exported data is written to a JSON data file, which is then compressed in a `.zip` archive and saved to an `exports` folder in the Code Insight installation.

Note ▪ *When a data export job for a given project is initiated, the options to perform a scan, rescan, Project Copy, or a Project Branch on this same project are disabled until the export is finished.*

Task ***To export project data, do the following:***

1. As Project Administrator or Analyst, navigate to the project's **Summary** tab (see Opening the Project Summary Tab).
2. (Optional) If you want a backup of the existing `.zip` file containing your project's most recently exported data, follow the instructions in Downloading the Archive Containing the Current Exported Data for the Project. This process downloads the `.zip` file from the `exports` folder in your Code Insight installation to your browser's download site.

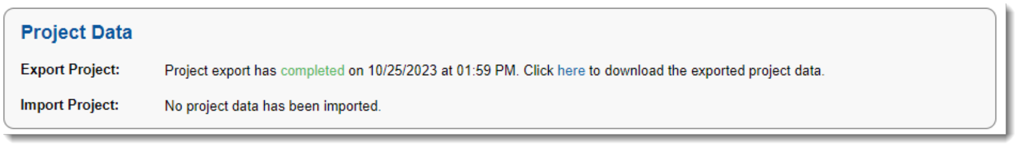
   Note ▪ *The export you are about to perform overwrites the existing archive in the `exports` folder.*

3. From the **Manage Project** menu, select **Export Project Data**. Either of the following occurs:
   - If a `.zip` file of data exported from this project already exists in the `exports/<projectId>` folder of your Code Insight installation, a popup is displayed. Continue with Step 4.
   - If no file exists, the export proceeds. The message "Project Data Export is initiated with Job ID: x" is temporarily displayed in the upper right of the screen. Skip to Step 5.

   Note ▪ *Due to certain events, the **Export Project Data** menu option might be temporarily disabled. For more information, see Impact of Other Jobs in Progress or Scheduled for the Current Project.*

4. In the popup, select whether you want to overwrite the existing exported data:
   - **Yes**—The export process will overwrite the current archive containing the previously exported data with an archive containing the results of the new export.

     When you select this option, the export is immediately initiated and the message "Project Data Export is initiated with Job ID: x" is temporarily displayed in the upper right of the screen. The export job is added to the **Jobs** queue and will run in the background in queue order. Continue with Step 5.
   - **No**—The export process will not be performed at this time. The pop-up closes and you remain on the **Summary** tab.
5. To track the status of the export, do either:
   - Check the **Export Project** field in the **Project Data** section on the **Summary** tab. (For a description of the various statuses available in this field, see the field descriptions in Summary Tab.)



Project Data

Export Project: Project export has completed on 10/25/2023 at 01:59 PM. Click here to download the exported project data.

Import Project: No project data has been imported.

- Open the **Jobs** queue and use the job ID to locate the **Project Export** job and track its status. (Use the instructions in Monitoring the Code Insight Jobs Queue to access and monitor the queue.) The **Jobs** queue enables you to see the status and position of the export job in relation to other jobs running in the Code Insight system.

  Note ▪ *When other projects trigger* ***Project Export*** *jobs concurrently with your job, the first export job added to the queue will be the first export job placed in* ***Active*** *state. The remaining export jobs continue in a* ***Scheduled*** *state and are run based on the scheduled order.*

During the export process, the project data is exported to a JSON data file, which is then compressed in a `.zip` archive and saved to the `exports/<projectId>` folder in your Code Insight installation. Both the `.zip` file and the JSON file inside the `.zip` file have the same name, `<projectId>-export-<date>_<time>`. This new `.zip` file overwrites the existing `.zip` file containing the project's previously exported data.

When the export process is finished, the "completed" status is displayed both in the **Jobs** queue and for the **Export Project** field (in the **Project Data** section on the **Summary** tab). The **Export Project** field also provides an option to download the `.zip` file containing the exported data (see Downloading the Archive Containing the Current Exported Data for the Project).

Job Conflicts When Attempting to Run an Export

The following sections describe the possible job conflicts that can occur when you attempt to run an export on a given project. The sections also describe how Code Insight manages these conflicts.

- Impact of Other Jobs in Progress or Scheduled for the Current Project
- Impact of System-Wide Jobs or Jobs for Other Projects in Progress or Scheduled

Impact of Other Jobs in Progress or Scheduled for the Current Project

When certain other jobs for the current project are in progress or scheduled to run, the **Export Project Data** menu option is temporarily disabled. Consider the following operations that can impact the availability of this option:

- The **Export Project Data** menu option is disabled whenever a scan, rescan, or another export is in progress or scheduled for the *current project*.
- When a Project Branch is in progress or scheduled for the current project, the **Export Project Data** menu option remains enabled for the current project, but it is disabled for the target project.

Once the conflicting job completes, the **Export Project Data** menu option is re-enabled so that you can initiate an export process. (You can check the **Jobs** queue to view the status of the conflicting job. See Monitoring the Code Insight Jobs Queue for help accessing and monitoring the queue.)

Impact of System-Wide Jobs or Jobs for Other Projects in Progress or Scheduled

The following describes the impact that certain system-wide jobs or jobs for other projects have on the current project:

- If an export job for another project is in progress or is scheduled, an export initiated for the current project is placed in a **Scheduled** state in the **Jobs** queue. It will run in scheduled order after the export already in progress (and any previously scheduled exports) complete.
- If an Electronic Update or Library Refresh is progress or is scheduled, an export initiated for the current project is placed in a **Scheduled** state in the **Jobs** queue and will run in scheduled order once the Electronic Update or Library Refresh is complete. (The Electronic Update or Library Refresh has priority over all other scheduled jobs.)

You can check the **Jobs** queue to view the status of conflicting jobs. See Monitoring the Code Insight Jobs Queue for help accessing and monitoring the queue.

Downloading the Archive Containing the Current Exported Data for the Project

When a data export is run for a project, the exported data is saved to a `<projectId>-export-<date>_<time>.zip` file located in the `exports/<projectId>` folder of your Code Insight installation. Each subsequent data export performed for the project overwrites the previous archive.

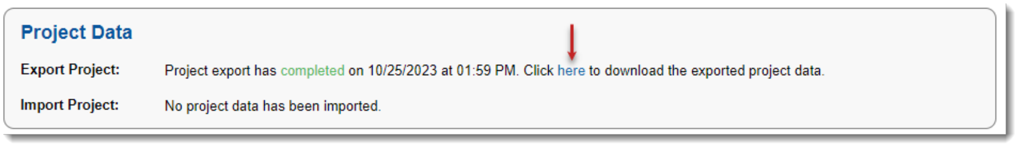
Code Insight provides the option to download a copy of a project's current export archive to the default download location for your browser. You might want to perform this download for one or more reasons, including:

- Provide easy access to the archive when selecting it for a project import. (Once the archive file is downloaded, you can copy it to any location.)
- Secure a backup copy of the most recently exported data.

Use the following procedure to download this archive from the Code Insight installation folder.

Task ***To download the archive containing the a project's most recently exported data, do the following:***

1. As Project Administrator or Analyst, navigate to the project's **Summary** tab (see Opening the Project Summary Tab).
2. Locate the **Export Project** field in the **Project Data** section.
3. Click the "here" link in the status description.



Project Data

Export Project: Project export has completed on 10/25/2023 at 01:59 PM. Click here to download the exported project data.

Import Project: No project data has been imported.

The file is downloaded to your browser's default location.

__Note__ ▪ If no export was previously run on the project, the __Export Project__ field shows “No project data has been exported”, and consequently no link is available. This same message is also displayed for projects in a Code Insight 2023 R4 or later instance that has been upgraded, but the `exports` folder has not been copied to the upgraded instance.

Exporting Project Data Using the REST Interface

Use the following information to export project data by issuing a cURL command that calls the **Export Project Data** REST API. During the export process, the project data is written to a JSON data file, which is then redirected to a `.zip` archive.

__Important__ ▪ When you use the __Export Project Data__ REST API, the export does not run as a background job as it does when executed through the UI.

__Note__ ▪ If want to copy and paste the cURL command directly from these instructions, copy it to a text editor first to remove formatting and any line breaks or extra spaces.

Task ***To export project data by calling the exportProjectData REST API, do the following:***

1. Ensure that the requirements listed in Prerequisites for Exporting Data and Prerequisites When Using the REST Interface are met.

2. To initiate the export process, execute the following cURL command to invoke the **Export Project Data** REST API using the GET method.

```
curl -X GET "HOSTNAME:PORT/codeinsight/api/project/exportProjectData?projectId=PROJECT_ID" -H
"accept: application/json" -H "Authorization: Bearer JWT_TOKEN" > PROJECT_DATA_FILE.zip
```

In the command syntax, replace the highlighted variables with your server host name (machine name or IP address) along with the port, the ID of the project you are exporting, and your authorization token.

Also, replace `PROJECT_DATA_FILE` with the name that you want to use for the `.zip` archive to which the data file containing the exported data will be redirected. (The data file will use the same name, but with a `.json` extension.) If you provide the name of the `.zip` file only, the file is downloaded by default to the directory from which the export command was executed. Alternatively, you can download the file to another location by including the full path to this other location in the file name.

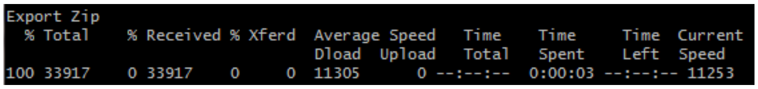
__Note__ ▪ If an archive with the same name already exists in the download location, the new archive replaces the existing one.

Example Command Calling the “Export Project Data” REST API

The following is an example cURL command calling the **Export Project Data** REST API:

```
curl -X GET "http://localhost:8888/codeinsight/api/project/exportProjectData?projectId=164" -H
"accept: application/json" -H "Authorization: Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsInVzZXJJZCI6MSwiaWF0IjoxNTY2ODU5NTg2fQ.qV2j8ZLgNGNJsT8OdPR
wvE0-0y1x7w-0zr5h7Jz2d9uqY8tvACsV68posEUO9tD-YXlgXznX-IGnrnopDU7G3w" > ProjectKDR.zip
```

As the command executes, the status of the export process is displayed in the command prompt window:



```
Export Zip
 % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                Dload  Upload   Total   Spent    Left  Speed
100 33917    0 33917    0     0  11305      0 --:--:--  0:00:03 --:--:-- 11253
```

When the export completes, the `.zip` archive containing the exported data is downloaded to the directory from which the export command was executed. For instance, if the export command was executed from the `C:/fnci/project_export` directory and the output redirect value is `ProjectKDR-export-02-20-2021_10-42.zip`, the following is the archive location:

```
C:/fnci/project_export/ProjectKDR-export-02-20-2021_10-42.zip
```

Note ▪ *If the output redirect value in the example had included the full path to another location, the `.zip` file would have been downloaded to this other location.*

The name of the data file contained in the archive is `ProjectKDR-export-02-20-2021_10-42.json`.

Importing Project Data

The following sections provide the details about importing project data:

- About an Import
- Prerequisites for Importing Data
- Import Behavior and Configuration
- Importing Project Data Using the Web UI
- Importing Project Data Using the REST API
- Verifying the Import Results

Important ▪ *The instructions in this section assume that you are importing project data using the Code Insight version for which this documentation was published. If you are importing project data using another version of Code Insight, refer to the documentation for that version for import instructions.*

About an Import

The Code Insight project-data import feature is available through these interfaces:

- Code Insight Web UI, as described in Importing Project Data Using the Web UI
- Code Insight REST interface, as described in Importing Project Data Using the REST API

The following sections provide overview information about the import process.

Input Used in the Import Process

The input for a project data import is an archived JSON data file containing project data. During the import process, this JSON data file (called the *import data file* in this documentation) is extracted from the archive and imported to a specific Code Insight project, called the *target project*.

You can create this input archive in a couple of ways:

- Use the Code Insight project-data export feature (see Exporting Project Data) to export data from a Code Insight v7 project.
- Export data from a Code Insight v6 project (see the Code Insight 6.14.x documentation for instructions). Then use the Audit Migration Tool to convert the exported data into the format required for importing in a v7 project.
- Manually create the import data file using data exported from an external system and formatting it as JSON code. Then compress the data file as a `.zip` archive.

In all cases, the input for the import process must be a data file whose contents are in the expected JSON format and that has been archived as a `.zip` file.

Target Projects

The project data import is performed on a project—called the *target project*—that has been created in Code Insight. This target can contain the results of a server or remote scan (or both types of scans) performed on the target codebase prior to the import. (See Scan Types for a description of server and remote scans.) The import process creates inventory with file associations and provides access to codebase files through the **Analysis Workbench** in the target project. The inventory is live—that is, ready to be edited and reviewed.

Imported Data

The data imported can include project and scan details, inventory (with inventory details, field values, file associations, and inventory review status), the review status of files, and any custom data.

About Importing Data to Current and pre-2020 R3 Projects

The following are the allowed and disallowed import scenarios involving current projects and projects created before Code Insight 2020 R3. (Also see Legacy Projects.)

- You can import data exported from a project created in 2020 R3 or later into another 2020 R3 or later project.
- You can import data exported from a project created in 2020 R3 or later to a standard project (created in 2020 R2 or earlier) that has been migrated to 2020 R3 or later.

- You cannot import data exported from a project created in 2020 R3 or later to a legacy inventory-only project that has been migrated to 2020 R3 or later.

Prerequisites for Importing Data

To import data, ensure that the following prerequisites are met:

- A Code Insight v7 instance currently running.
- An existing Code Insight project to which data will be imported (that is, a target project).

  ***Note** ▪ Ensure that a scan is not running on this project during the import.*

- A `.zip` archive containing the JSON data file with the project data to be imported (see Input Used in the Import Process).
- A completed Electronic Update for the Code Insight system to which the target project belongs.
- Items listed in Prerequisites When Using the REST Interface (if performing the import using REST API).

Import Behavior and Configuration

The following sections provide information you should know about the Code Insight import behavior and ways to configure this behavior.

- About File Processing During an Import
- Default Criteria for Handling File and Inventory Comparisons During the Import
- Available Import Options to Configure Import Behavior
- Other Import Considerations

About File Processing During an Import

Depending on the import configuration, the import process might need to match files paths in the import data file with file paths in the target project codebase to do the following:

- Create new file associations in inventory in the target project.
- Mark reviewed files in the import data file as reviewed in the target project.
- Determine empty files.

The file path and the file MD5 value are the key criteria used to locate target codebase files that match files in the import data file. When the MD5 value is used as a criterion, the MD5 for a file in the import data file must have an exact MD5 match in the target codebase. However, when the file path is used as a criterion, the file-matching process can apply various rules.

File-Path Processing

When the file path is used as a criterion for matching, the import process internally subtracts the root path from the absolute path of codebase files in the import data file and in the target project. The result is the *complete file* path for a given file, as illustrated in these examples:

- **Absolute file path**—`/home/fnci/scanRoot/1/ePortal-1.3/src/gettext.c`
- **Root path**—`/home/fnci/scanRoot/1/`
- **Complete File path**—`/ePortal-1.3/src/gettext.c`

Then, based on the file-path criterion selected by the user, the import locates matching files by searching complete file paths, partial paths, or simply file names. The following examples illustrate a complete path in comparison with a partial path or file name:

- **Complete path**—`/ePortal-1.3/copy1/src/gettext.c`
- **Partial path**—`/copy1/src/gettext.c` **or** `/src/gettext.c`)
- **File name**—`gettext.c`

When users select the partial path as a criterion for matching files, they must also provide a desired directory depth that defines the partial path.

Default Criteria for Handling File and Inventory Comparisons During the Import

The following information describes the *default* criteria used for handling file and inventory comparisons during an import. You can modify this criteria as needed by reconfiguring import options, as described in the next section, Available Import Options to Configure Import Behavior.

- Only those files whose complete files paths match in both the import data file and the scanned target codebase can be associated with inventory in the target project. See Option for Creating New File Associations in Target Inventory for more information.
- Only those files whose MD5s and complete paths match in both the import data file and the scanned target codebase are processed when marking files as reviewed in the target project. See Option for Marking Target Codebase Files as Reviewed for more information.
- Only those inventory items in the import data file whose associated files match file paths in the target codebase are created in the target project (if these inventory items do not already exist in the target project). See Option to Create Empty Inventory for more information. (The import REST interface uses this behavior as the default; the Web UI might use different default behavior based on project settings.)
- When an inventory item in the import data file is an exact match to an inventory item in the target project, any notes content defined for the inventory item in the import data file overwrites the notes content defined in the target inventory item. See Options for Handling Inventory Notes and Custom Fields for more information.

The default import logic is configurable, as described in the next section.

Available Import Options to Configure Import Behavior

The following options can be specified to override the default import behavior described in the previous section.

- Option for Creating New File Associations in Target Inventory
- Option for Marking Target Codebase Files as Reviewed
- Option to Create Empty Inventory
- Options for Handling Inventory Notes and Custom Fields
- Options for Handling Inventory Usage Values

Additionally, the section Specifying File-Matching Criteria in the Import REST Interface provides more details about the file-matching criteria used when creating new file associations in target inventory or when marking target codebase files as reviewed.

The information in this section directly applies to the procedures on how to execute an import using the Web UI or the REST interface described later. However, because the configuration of import behavior is a thoughtful process and configurations must be set up before initiating an import, the configuration options in detail are described here.

Option for Creating New File Associations in Target Inventory

The import Web UI and REST interface provide options to set the file-matching criteria needed by the import to create new file associations in target project inventory when that inventory is identical to inventory in the data import file. (For a description of *identical* inventory items, see Identical Inventory.) The import process will create a new file association in the identical target inventory item only if the defined file-matching criteria is met. If a file association in the import data file already exists in the target inventory item, no new file association is added to the target inventory item.

By default, a file associated with an inventory item in the import data file can be added to an identical inventory item in the target project codebase only if the file's complete path matches in both the import data file and in the target project codebase. For example, a file with a complete path of `/ePortal-1.3/src/gettext.c`, listed in the import data file as belonging to "Inventory Item 1", will be considered for association with this same inventory item in the target project only if the target project codebase contains a file with the same complete path, `/ePortal-1.3/src/gettext.c`.

However, you can set different criteria for adding associated files to target inventory, such as requiring that only partial paths or MD5 values match or requiring that both MD5 values and paths match. For more details, see the following topics:

- "Add Files to Inventory" Option in the Web UI
- "addFilesToInventory" Attribute in the REST Interface

Note *▪ The same file-matching criteria defined for creating new file associations in target inventory is also used in determining empty inventory. See Option to Create Empty Inventory.*

“Add Files to Inventory” Option in the Web UI

The field **Add Files to Inventory** on the **Import Project Data** dialog is used to set the file-matching criteria for creating new file associations in target inventory. For a description of the criteria options available with this field, see Import Project Data Dialog. For complete instructions on using the Web UI to import project data, see Importing Project Data Using the Web UI.

“addFilesToInventory” Attribute in the REST Interface

The `addFilesToInventory` attribute is used in the **Import Project Data** REST API to set the file-matching criteria for creating new file associations in inventory. The following sections provide attribute details:

- About the “addFilesToInventory” Attribute
- Example “addFilesToInventory” Syntax in Import cURL Command

For instructions on executing **Import Project Data** API, see Importing Project Data Using the REST API.

About the “addFilesToInventory” Attribute

The `addFilesToInventory` attribute must be set to `true` to enable the import process to add files to inventory in the target project. Along with this attribute, a second attribute, `inventoryFileMatchingCritieria`, must be included to set the file-matching criteria used to determine whether a given file can be added to target inventory. A third attribute is required to set the directory depth if you specify partial-path criteria. For more information about setting file-matching criteria, see Specifying File-Matching Criteria in the Import REST Interface.

If the user explicitly sets the `addFilesToInventory` attribute to `false` (or omits this attribute entirely), the import does not associate any additional files to inventory in the target project.

Example “addFilesToInventory” Syntax in Import cURL Command

The following shows an example of the `addFilesToInventory` attribute in a cURL command that calls the REST `import` endpoint:

```
curl -H "Authorization:Bearer  %jwt%" -F importFile=@"FileToImport.zip" -F projectImportModel={
"createEmptyInventory” : true, “overwriteInventoryNotes” : true, “addFilesToInventory” : true,
“inventoryFileMatchingCriteria” : “MD5_AND_PARTIAL_FILEPATH”, “inventoryDirectoryDepth” : 2,
“markFilesAsReviewed” : true, “reviewFileMatchingCriteria” : “PARTIAL_FILEPATH”,
“reviewDirectoryDepth” : 2, "resetInventoryUsage": false" };type=application/json  http://
hostname:8888/codeinsight/api/projects/{projectId}/import
```

For complete instructions about using the cURL command to execute an import, see Importing Project Data Using the REST API.

For details about the implementation of the **Import Project Data** REST API, access the Code Insight Swagger documentation. To do so, click the ☰ icon in the upper right corner of the Code Insight Web UI, select **Help**, and click the **REST API Guide** link.

Option for Marking Target Codebase Files as Reviewed

The import Web UI and REST interface provide options to set the criteria needed to mark file target codebase files as reviewed.

Note *▪ For this option, the import compares only those files in the import data file that are marked as reviewed with files in the target codebase.*

By default, the import can mark an unreviewed file in the target codebase as reviewed only if a file in the import data file has the same MD5 and complete file path as the target file. For example, suppose a file marked as “unreviewed” in the target project codebase has a complete path of `/ePortal-1.3/src/gettext.c`. The import can mark this file as reviewed only if the import data file contains a file whose complete path is `/ePortal-1.3/src/gettext.c` and whose MD5 is the same as the MD5 of the file in the target project codebase.

However, you can set different import criteria for marking files in the target project codebase as reviewed, such as requiring that only partial paths or only MD5 values match or requiring that both MD5 values and partial paths match. For more details, see the following topics:

- “Add Files to Inventory” Option in the Web UI
- markFilesAsReviewed Attribute in the REST Interface

“Mark Files as Reviewed” Option in the Web UI

The field **Mark Files as Reviewed** on the **Import Project Data** dialog is used to set the file-matching criteria for marking target codebase files as reviewed during the import process. For a description of the criteria options available with this field, see Import Project Data Dialog. For complete instructions on using the Web UI to import project data, see Importing Project Data Using the Web UI.

markFilesAsReviewed Attribute in the REST Interface

The `markFilesAsReviewed` attribute is used in the **Import Project Data** REST API to set the file-matching criteria for marking target codebase files as reviewed during the import process. The following sections provide attribute details:

- About the “markFilesAsReviewed” Attribute
- Example “markFilesAsReviewed” Syntax in Import cURL Command

For instructions on executing **Import Project Data** API, see Importing Project Data Using the REST API.

About the “markFilesAsReviewed” Attribute

The `markFilesAsReviewed` attribute must be set to `true` to enable the import process to mark reviewed files in the import data file as reviewed in the target project. Along with this attribute, a second attribute, `reviewFileMatchingCritieria`, must be included to set the file-matching criteria used to determine whether a given unreviewed file in the target codebase can be flagged as “Reviewed”. A third attribute is required to set the directory depth if you specify partial-path criteria. For more information about setting file-matching criteria, see Specifying File-Matching Criteria in the Import REST Interface.

If the user explicitly sets the `markFilesAsReviewed` attribute to `false` (or omits this attribute entirely), the import does not change the “Reviewed” or “Not Reviewed” status of files in the target project codebase. The statuses remain as they were before the import process was performed.

Example “markFilesAsReviewed” Syntax in Import cURL Command

The following shows an example of the explicit use of the `markFilesAsReviewed` attribute in a cURL command that calls the REST `import` endpoint:

```
curl -H "Authorization:Bearer  %jwt%" -F importFile=@"FileToImport.zip" -F projectImportModel={
"createEmptyInventory” : true, “overwriteInventoryNotes” : true, “addFilesToInventory” : true,
“inventoryFileMatchingCriteria” : “MD5_AND_PARTIAL_FILEPATH”, “inventoryDirectoryDepth” : 2,
“markFilesAsReviewed” : true, “reviewFileMatchingCriteria” : “PARTIAL_FILEPATH”,
“reviewDirectoryDepth” : 2, "resetInventoryUsage": false" };type=application/json  http://
hostname:8888/codeinsight/api/projects/{projectId}/import
```

For complete instructions about using the cURL command to execute an import, see Importing Project Data Using the REST API.

For more details about the implementation of the **Import Project Data** REST API, access the Code Insight Swagger documentation. To do so, click the ☰ icon in the upper right corner of the Code Insight Web UI, select **Help**, and click the **REST API Guide** link.

Option to Create Empty Inventory

The import Web UI and REST interface provide an option to specify whether “empty” system-generated inventory items are still processed in the target project during the import. Empty inventory items either have no file associations in the data file to be imported or *do* have associated files in the import data file but no matching paths or MD5s for these files in the target project codebase (also see Complete vs Partial Paths in the Import Path-Matching Process).

When this option is enabled, all inventory items in the import data file—with or without matching associated files in the target codebase—are created in the target project during the import.

When the option is disabled, the import process does not create empty inventory items in the target project. It creates only inventory items whose associated files are found in the target codebase. To define what constitutes “matching files” between the import data file and the target project codebase, this option depends on the criteria set for the Option for Creating New File Associations in Target Inventory.

Note ▪ *If you are importing from a scanned project into an inventory-only project, which has no codebase, ensure this option is enabled so that inventory is generated in the new project.*

For more details about this option, see the following topics:

- Option in the Web UI
- “createEmptyInventory” Attribute in the REST Interface

Option in the Web UI

The option to create empty inventory is available in the Web UI as the project setting, **On the data import or rescan, delete inventory with no associated files**, located on the Edit Project: General Tab. Ensure that this field is properly set for the import you are about to perform. (You can always reset this value for the project once the import is complete.) See Editing the Project Definition and General Settings for details.

The default for this setting is defined at a global-project level by the Code Insight System Administrator. (The initial global setting disables the creation of empty inventory, but obviously this can be changed at the administrator's discretion to affect all projects.)

"createEmptyInventory" Attribute in the REST Interface

The option to create empty inventory is available as the `createEmptyInventory` attribute when invoking the **Import Project Data** REST API. The following sections provide attribute details:

- About the "createEmptyInventory" Attribute
- Example "createEmptyInventory" Syntax in Import cURL Command

About the "createEmptyInventory" Attribute

To enable the creation of empty inventory items in the target project, explicitly include the `createEmptyInventory` attribute and set it to **`true`** in the cURL command that calls the `import` REST endpoint. All inventory items in the import data file—with or without matching associated files in the target codebase—are created in the target project during the import.

To disable the creation of empty inventory items in the target project, explicitly include the `createEmptyInventory` attribute and set it to **`false`** in the cURL command. Only those inventory items with associated files in the import data file that match files in the target codebase are created in the target project.

If you omit the `createEmpyInventory` attribute entirely from cURL command, the import process uses the value of the `deleteEmptyInventory` attribute defined for the project (the same setting as the **On the data import or rescan, delete inventory with no associated files** value on the Edit Project: General Tab.) To override this project setting for the current import process only, explicitly include the `createEmptyInventory` attribute with the appropriate setting when invoking the import endpoint.

Example "createEmptyInventory" Syntax in Import cURL Command

The following example shows the attribute explicitly included in the cURL command that calls the `import` REST endpoint:

```
curl -H "Authorization:Bearer  %jwt%" -F importFile=@"FileToImport.zip" -F projectImportModel={
"createEmptyInventory” : true, “overwriteInventoryNotes” : true, “addFilesToInventory” : true,
“inventoryFileMatchingCriteria” : “MD5_AND_PARTIAL_FILEPATH”, “inventoryDirectoryDepth” : 2,
“markFilesAsReviewed” : true, “reviewFileMatchingCriteria” : “PARTIAL_FILEPATH”,
“reviewDirectoryDepth” : 2, "resetInventoryUsage": false" };type=application/json  http://
hostname:8888/codeinsight/api/projects/{projectId}/import
```

For complete instructions about using the cURL command to execute an import, see Importing Project Data Using the REST API.

For more details about the implementation of the **Import Project Data** REST API, access the Code Insight Swagger documentation. To do so, click the ☰ icon in the upper right corner of the Code Insight Web UI, select **Help**, and click the **REST API Guide** link.

Options for Handling Inventory Notes and Custom Fields

The import Web UI and REST interface enable you to specify whether the values for “notes” fields and custom fields defined for an inventory item in the import data file should overwrite or be appended to the values for these same fields defined for the identical inventory item in the target project. (For a description of *identical* inventory items, see Identical Inventory.)

The “notes” fields include the following:

- **Notices Text**
- **Audit Notes**
- **Usage Guidance**
- **Remediation Notes**

Note ▪ *For a custom field to be imported for an inventory item, the label of the custom field in the import data file must match the label of a custom field for the target inventory item.*

For more details about this option to overwrite target inventory notes, see the following topics:

- “Inventory Notes Handling” Options in the Web UI
- “overwriteInventoryNotes” Attribute in the REST Interface

“Inventory Notes Handling” Options in the Web UI

Select the appropriate **Inventory Notes Handling** option on the **Import Project** dialog to configure whether the values for “notes” fields and custom fields are overwritten in target inventory during the import. For a description of this option, see Import Project Data Dialog. For complete instructions on using the Web UI to import project data, see Importing Project Data Using the Web UI.

“overwriteInventoryNotes” Attribute in the REST Interface

The option to overwrite target inventory “notes” and custom fields is available as the `overwriteInventoryNotes` attribute in the **Import Project Data** REST API. The following sections provide attribute details:

- About the “overwriteInventoryNotes” Attribute
- Example “overwriteInventoryNotes” Syntax in the Import cURL Command

About the “overwriteInventoryNotes” Attribute

To overwrite the value of each “notes” field or custom field in the target inventory item with the value of the corresponding field for the inventory item in the import data file, explicitly include the `overwriteInventoryNotes` attribute and set it to `true` in the cURL command that calls the `import` REST endpoint. Note that, if the data for a given field is blank in the import data file, no overwrite occurs; any existing value for the field in the target inventory is retained.

To append the value of each “notes” or custom field in the import data file to the value of the same field in the target inventory item, explicitly include the `overwriteInventoryNotes` attribute and set it to **`false`** (or omit the attribute entirely). The appended value is separated from the existing value with a line break and the following heading:

```
Copied during import from <ProjectName>:<InventoryName> (TimeStamp)
```

However, If the value for any of these fields is the same for the inventory item in both the import data file and the target inventory, no value is appended.

Example “overwriteInventoryNotes” Syntax in the Import cURL Command

The following example shows the attribute explicitly included in the cURL command that calls the `import` REST endpoint:

```
curl -H "Authorization:Bearer  %jwt%" -F importFile=@"FileToImport.zip" -F projectImportModel={
"createEmptyInventory” : true, “overwriteInventoryNotes” : false, “addFilesToInventory” : true,
“inventoryFileMatchingCriteria” : “MD5_AND_PARTIAL_FILEPATH”, “inventoryDirectoryDepth” : 2,
“markFilesAsReviewed” : true, “reviewFileMatchingCriteria” : “PARTIAL_FILEPATH”,
“reviewDirectoryDepth” : 2, "resetInventoryUsage": false" };type=application/json  http://
hostname:8888/codeinsight/api/projects/{projectId}/import
```

For complete instructions about using the cURL command to execute an import, see Importing Project Data Using the REST API.

For more details about the implementation of the **Import Project Data** REST API, access the Code Insight Swagger documentation. To do so, click the ☰ icon in the upper right corner of the Code Insight Web UI, select **Help**, and click the **REST API Guide** link.

Options for Handling Inventory Usage Values

The import Web UI and REST interface enable you to specify whether the import process should copy the current inventory Usage values to the target project or reset all values for imported inventory to the system default value, `Unknown`. For details, see the following topics:

- “Inventory Usage Handling” Options in the Web UI
- “resetInventoryUsage” Attribute in the REST Interface

The default import behavior is to reset Usage values for imported inventory to the default value, `Unknown`.

“Inventory Usage Handling” Options in the Web UI

Select the appropriate **Inventory Usage Handling** option (**Reset usage field values to system default** or **Copy existing usage field values**) on the **Import Project** dialog to configure how inventory usage values should be handled. For a description of these two options, see Import Project Data Dialog. For complete instructions on using the Web UI to import project data, see Importing Project Data Using the Web UI. The default is to reset all values to the default **Unknown**.

“resetInventoryUsage” Attribute in the REST Interface

The attribute `resetInventoryUsage` in the **Import Project Data** REST API is used to determine how the import process should handle Usage values for imported inventory. The following sections provide attribute details:

- About the “resetInventoryUsage” Attribute
- Example “resetInventoryUsage” Syntax in the Import cURL Command

About the “resetInventoryUsage” Attribute

If you set the `resetInventoryUsage` attribute to **`true`** (or omit the attribute entirely) in the cURL command that calls the `import` REST endpoint, the import process resets all Usage values for imported inventory to the system default value, `Unknown`. (This is the default behavior.)

Conversely, to configure the import process to copy existing Usage values to the imported inventory, you must explicitly include the `resetInventoryUsage` attribute in the cURL command and set it to **`false`**.

Example “resetInventoryUsage” Syntax in the Import cURL Command

The following example shows the attribute explicitly included in the cURL command that calls the `import` REST endpoint:

```
curl -H "Authorization:Bearer  %jwt%" -F importFile=@"FileToImport.zip" -F projectImportModel={
"createEmptyInventory” : true, “overwriteInventoryNotes” : false, “addFilesToInventory” : true,
“inventoryFileMatchingCriteria” : “MD5_AND_PARTIAL_FILEPATH”, “inventoryDirectoryDepth” : 2,
“markFilesAsReviewed” : true, “reviewFileMatchingCriteria” : “PARTIAL_FILEPATH”,
“reviewDirectoryDepth” : 2, "resetInventoryUsage": false" };type=application/json  http://
hostname:8888/codeinsight/api/projects/{projectId}/import
```

For complete instructions about using the cURL command to execute an import, see Importing Project Data Using the REST API.

For more details about the implementation of the **Import Project Data** REST API, access the Code Insight Swagger documentation. To do so, click the ☰ icon in the upper right corner of the Code Insight Web UI, select **Help**, and click the **REST API Guide** link.

Specifying File-Matching Criteria in the Import REST Interface

If the `addFilesToInventory` attribute (see Option for Creating New File Associations in Target Inventory) or the `markFilesAsReviewed` attribute (see Option for Marking Target Codebase Files as Reviewed) is set to **`true`**, you must include an additional attribute that defines the file-matching criteria needed to compare files in the import data file with the target codebase files:

- For `addFilesToInventory`, include the `inventoryFileMatchingCritieria` attribute.
- For `markFilesAsReviewed`, include the `reviewFileMatchingCritieria` attribute.

Note ▪ *This `...FileMatchingCriteria` attribute is required whenever `addFilesToInventory` or `markFilesAsReviewed` is set to **`true`**. If this attribute is omitted, an error occurs when you attempt to execute the import command.*

If you are using the **Import Project Data** REST API to execute the import process, refer the information in this section for details about the criteria.

If you are performing the import through the Code Insight Web UI, refer to the Import Project Data Dialog for criteria descriptions.

Available File-Matching Criteria in the Import REST Interface

The following describes the available criteria for the `inventoryFileMatchingCritieria` attribute or `reviewFileMatchingCritieria` attribute used to locate file matches between the import data file and the target project codebase.

Table 4-1 ▪ Possible Values for File-Matching Criteria

Attribute Value	Description
MD5_AND_COMPLETE_ FILEPATH	A file's MD5 value and complete path (including the file name) in the import data file must match the MD5 and complete path of a file in the target project codebase.
MD5_AND_PARTIAL_FILEPATH	A file's MD5 value and partial path (including the file name) in the import data file must match the MD5 and partial path of a file in the target project codebase. For this criterion, you must include an additional attribute to define the depth of the partial path. See Specifying Directory Depth for Partial-Path Criteria in the REST Interface for more information.
MD5_AND_FILENAME	The MD5 and name of a file in the import data file must match the MD5 and name of a file in the target project codebase.
MD5	A file's MD5 value in the import data file must match an MD5 in the target project codebase.
COMPLETE_FILEPATH	The complete path of a file (including the file name) in the import data file must match a complete file path in the target project codebase.
PARTIAL_PATH	A file's partial path (including of the file name) in the import data file must match the partial path of a file in the target project codebase. For this criterion, you must include an additional attribute to define the depth of the partial path. See Specifying Directory Depth for Partial-Path Criteria in the REST Interface for more information.
FILENAME	The name of the file in the import data file must match a file name in the target project codebase. (No path is compared in the file-matching process.)

Specifying Directory Depth for Partial-Path Criteria in the REST Interface

If you specify `MD5_AND_PARTIAL_PATH` or `PARTIAL_PATH` as the value for the `inventoryMatchingCriteria` or the `reviewFileMatchingCriteria` attribute, you must include another attribute that defines the directory depth by which to match the partial paths:

- For the `inventoryMatchingCriteria`, include the `inventoryDirectoryDepth` attribute.

- For `reviewFileMatchingCriteria`, include the `reviewDirectoryDepth` attribute.

Note* ▪ *This* `...DirectoryDepth` *attribute is required whenever a partial-path criterion is specified for file matching. If this attribute is omitted, an error occurs when you attempt to execute the import command.

Provide a value 1 through 20 to designate the number of directories above the file name that must be the same when matching file paths in the import data file with file paths in the target project codebase.

For example, suppose a file has a complete path `/ePortal-1.3/copy1/src/gettext.c`. If a partial path criterion is set with a directory depth of 2 (that is, 2 directories above the file name), the partial path `copy1/src/gettext.c` in the import data file must match the same path in the target project codebase to meet the criterion.

The partial-path depth enables the import to match files when the codebase location is different between the target project codebase and the project codebase whose scanned results are included in the import data file. See also Complete vs Partial Paths in the Import Path-Matching Process.

Other Import Considerations

Consider the import behavior when it processes the following:

- Complete vs Partial Paths in the Import Path-Matching Process
- Identical Inventory
- Unreviewed Files
- Custom Fields for Inventory and Projects

Complete vs Partial Paths in the Import Path-Matching Process

When the import process considers whether to add a file to inventory or mark it as reviewed, the file-matching criteria can mandate that the path of the file match between the import data file and the target project. For example, if the file-matching criteria requires that the *complete* paths of files match, the file `/ePortal-1.3/src/gettext.c`—the only file belonging to “InventoryItem 1.0 (License1)” in the import data file—is considered to be a different file from `/ePortal-2.0/src/gettext.c` in the target project. As a result, `ePortal-2.0/src/gettext.c` cannot be associated with “InventoryItem 1.0 (License1)”; and, if the `createEmptyInventory` value is **`false`**, “InventoryItem 1.0 (License1)” is not created in the target project since it has no associated file. Additionally, `ePortal-2.0/src/gettext.c` will not be marked as reviewed in the target project.

In order for a file in the target project to be treated as identical to a file in the data file, the paths for the two files must match in both locations. To accomplish this, you can use “partial path” and its directory depth as the file-matching criteria. In the case above, you would ensure that the file paths match by selecting “partial path” and setting its directory depth to 1 (one directory above the file name). In this way, the import is matching only the partial path /src/gettect.c for both files. Alternatively, you can manually manipulate paths in the import data file to match those in the target project, but you are strongly recommended to apply the “partial path” and directory depth criteria instead.

Identical Inventory

An inventory item in the source project is considered identical to an inventory item in the target project if both items are associated with the same unique combination of component-version-license (CVL). By default, identical inventory is reconciled during the import by overriding all fields in the target inventory item with information from the import data file. In cases where the source inventory item has empty fields, the data in the target inventory item will be left as is (that is, will not be removed).

However, Code Insight does provide the option to *append* the contents of notes fields in the import data file to target inventory. See Options for Handling Inventory Notes and Custom Fields for details.

Unreviewed Files

A codebase file that was flagged as “unreviewed” in the source Code Insight project (that is, the project from which the import data file was created) does not retain its unreviewed status in the target project if the target codebase scan has marked the corresponding target file as reviewed. This occurs because the import data file stores information for only those codebase files that have been marked as reviewed in the source project. Hence, no information about the unreviewed file exists in the import data to overwrite information for this same file in the target project.

If, during a standard import, you want the target project codebase to retain the reviewed and unreviewed status of codebase files in the source project, manually flag all of the codebase files in the target project as “unreviewed” before importing the data file. In this way, any unreviewed files in the source project remain unreviewed in the target project; files marked as reviewed in the source project are included in the import data file and will be marked as reviewed in the target project if they meet the import file-matching criteria.

Custom Fields for Inventory and Projects

Custom field values are imported for the project or inventory items as long as the following conditions are met:

- The name and field type of a custom field for the project in the import data file must match the name and field type of a custom field defined in the target project. Additionally, if the field type is “dropdown”, the value of the field in the import file must match one of the “dropdown” values defined for the field in the target project.

  The value imported for a custom project field overwrites the value of the corresponding field in the target project. If the imported field has no value, the target field value is null.

- The name of a custom field for an inventory item in the import data file must match the name of a field defined for the identical inventory item in the target project. Additionally, depending on how you configure the import process, the value in the import data file can overwrite or be appended to the value of the field for the target inventory item. See “Inventory Notes Handling” Options in the Web UI.

Importing Project Data Using the Web UI

Use the following instructions to import project data using the Code Insight Web UI.

Task ***To import project data, do the following:***

1. Ensure that all requirements in Prerequisites for Importing Data are met.
2. As Project Administrator or Analyst, navigate to the **Summary** tab (see Opening the Project Summary Tab).

3. From the **Manage Project** menu, select **Import Project Data**. (For information about the occasions when this option is temporarily disabled, see Import Menu Option Disabled When Certain Other Jobs Are Scheduled or Running.)
4. Complete the fields as described in Import Project Data Dialog, and click **OK**. Once an internal temporary copy of the export data file is created for the import job, the dialog closes and the message “Project Data Import is initiated with Job ID: x” is temporarily displayed in the upper right of the screen.

   The import job is now added to the **Jobs** queue and will run in the background in queue order.
5. To track the status of the export, do either:
   - Check the **Import Project** field in the **Project Data** section on the **Summary** tab. (For a description of the various statuses for this field, see the field descriptions in Summary Tab.)
   - Open the **Jobs** queue and use the job ID to locate the **Project Import** job and track its status. (Use the instructions in Monitoring the Code Insight Jobs Queue to access and monitor the queue.) The **Jobs** queue enables you to see the status and position of the import job in relation to other jobs running in the Code Insight system.

     Note ▪ *When other projects trigger **Project Import** jobs concurrently with your job, the first import job added to the queue will be the first import job placed in **Active** state. The remaining import jobs continue in a **Scheduled** state and are run based on the scheduled order.*

   Once the export process has completed, the “completed” status is displayed in the **Jobs** queue and on the **Import Project** field (in the **Project Data** section on the **Summary** tab).
6. Verify that the import results are what you expected. See Verifying the Import Results.

Import Menu Option Disabled When Certain Other Jobs Are Scheduled or Running

The **Import Project Data** menu option is temporarily disabled whenever an Electronic Update is in progress or scheduled for the Code Insight instance. Additionally, the menu option is temporarily disabled whenever one of these jobs is in progress or scheduled for the *current project*: a scan, rescan, SBOM Insights export, Project Copy, Project Branch, report generation, or another import job.

Once the conflicting job completes, the **Import Project Data** menu option is re-enabled so that you can initiate the import process. (You can check the **Jobs** queue to view the status of the conflicting job. See Monitoring the Code Insight Jobs Queue for help accessing and monitoring the queue.)

Importing Project Data Using the REST API

Use the following instructions to import project data by explicitly executing a cURL command that calls the **Import Project Data** REST API. (Alternatively, you can invoke this API using an API client such as Postman. See Importing Project Data Using the Postman API Client.)

Note ▪ *When you use the **Import Project Data** REST API, the import does not run as a background job as it does when executed through the UI.*

Task ***To run an import, do the following:***

1. Ensure that all prerequisites in Prerequisites for Importing Data and Prerequisites When Using the REST Interface are met.
2. Set up a cURL command to invoke the **Import Project Data** REST API (`import` endpoint). You choose one of these methods in which to execute the command:
    - Explicitly Provide the Import Attributes in the cURL Command
    - Point the cURL Command to a File Containing the Import Attributes
3. Execute the command.

    When the import is complete, a status message with **OK** will appear in the command prompt window. If the import is not successful, a status code and error message is displayed.
4. Verify that the import results are what you expected. See Verifying the Import Results.

Explicitly Provide the Import Attributes in the cURL Command

One option for setting up the cURL command that invokes the **Import Project Data** REST API (`import` endpoint) is to explicitly include the import attributes in the command. For the complete instructions on running the import using this command, return to Importing Project Data Using the REST API.

Note ▪ *If want to copy and paste the cURL command directly from these instructions, copy it to a text editor first to remove formatting and any line breaks or extra spaces.*

The following shows the cURL command syntax with the available import attributes:

```
curl -H "Authorization:Bearer  JWT_TOKEN" -F importFile=@"FILE_TO_IMPORT.zip" -F
projectImportModel={ \"createEmptyInventory\” : true/false, \“overwriteInventoryNotes\” : true/
false, \“addFilesToInventory\” : true/false, \“inventoryFileMatchingCriteria\” :
\“FileMatchingCriteria\”, \“inventoryDirectoryDepth\” : 1-20, \“markFilesAsReviewed\” : true/false,
\“reviewFileMatchingCriteria\” : \“FileMatchingCriteria\”, \“reviewDirectoryDepth\” : 1-20,
\“resetInventoryUsage\” : true/false };type=application/json  http://HOSTNAME:PORT/codeinsight/api/
projects/PROJECT_ID/import
```

This next code excerpt is an example of the cURL command. Refer to Available Import Options to Configure Import Behavior for details about the available import attributes.

```
curl -H "Authorization:Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsInVzZXJJZCI6MSwiaWF0IjoxNTY2ODU5NTg2fQ.qV2j8ZLgNGNJsT8OdPR
wvE0-0y1x7w-0zr5h7Jz2d9uqY8tvACsV68posEUO9tD-YXlgXznX-IGnrnopDU7G3w" -F importFile=@"1184-export-4-
02-20-2021-40.zip" -F projectImportModel={ "createEmptyInventory” : true, “overwriteInventoryNotes”
: false, “addFilesToInventory” : true, “inventoryFileMatchingCriteria” : “MD5_AND_PARTIAL_FILEPATH”,
“inventoryDirectoryDepth” : 2, “markFilesAsReviewed” : true, “reviewFileMatchingCriteria” :
“PARTIAL_FILEPATH”, “reviewDirectoryDepth” : 2, "resetInventoryUsage": false" };type=application/
json  http://localhost:8888/codeinsight/api/projects/217/import
```

Note the following about the example command. Users must explicitly provide their own values in the command based on their environment.

- JWT_TOKEN has been replaced with an example user authorization token.
- HOSTNAME:PORT has been replaced with an example machine name and port—in this case, **localhost:8888**. (The IP address for the machine can be used instead of the machine name.)
- FILE_TO_IMPORT has been replaced with the name of an import data file (that is, the file name of the zip file to be imported) in the example. In this case, the import data file is **1184-export-02-20-2021_09-40**.zip. The entire file name must be enclosed in quotes.
- PROJECT_ID has been replaced with the ID of the project to which data is being imported (in this case, **217**). See Project ID for help on obtaining the project ID.
- The value for the inventoryFileMatchingCriteria and reviewFileMatching attributes must be enclosed in quotes.

Point the cURL Command to a File Containing the Import Attributes

Another option for setting up the cURL command that invokes the **Import Project Data** REST API (import endpoint) is to save the import attributes to a .json file and point to that file in the cURL command. For the complete instructions on running the import using this command, return to Importing Project Data Using the REST API.

Task ***To set up a cURL command that points to a file containing the import attributes, do the following:***

1. Create and save a .json file containing the import attributes. Refer to Available Import Options to Configure Import Behavior for details about the available import attributes.

   The following shows sample file contents:

   ```
   {
         "createEmptyInventory": false,
         "overwriteInventoryNotes": true,
         "addFilesToInventory": true,
         "inventoryFileMatchingCriteria": "COMPLETE_FILEPATH",
         "inventoryDirectoryDepth" : 2
         "markFilesAsReviewed": true,
         "reviewFileMatchingCriteria": "MD5_AND_COMPLETE_FILEPATH",
         "reviewDirectoryDepth": 2,
         "resetInventoryUsage": false,
   }
   ```

   Note that the value for the inventoryFileMatchingCriteria attribute (not shown) and the reviewFileMatching attribute must be enclosed in quotes.

   For purposes of example, this file is saved as Import217Settings.json, but you can provide any name with the json extension. Alternatively, you can save the json-formatted contents as a simple text file. However, when you point to the file in the cURL command, provide the file name only, not extension. For example, for Import217Settings.txt, provide only **Import217Settings** as the file name in the cURL command; do not include the .txt extension.

2. Set up the cURL command, pointing to the file containing the import attributes.

Note ▪ *If want to copy and paste the cURL command directly from these instructions, copy it to a text editor first to remove formatting and any line breaks or extra spaces.*

The following shows the cURL command syntax:

```
curl -H "Authorization:Bearer  JWT_TOKEN" -F importFile=@"FILE_TO_IMPORT.zip" -F
projectImportModel=@"IMPORT_SETTINGS.json;type=application/json" http://HOSTNAME:PORT/
codeinsight/api/projects/PROJECT_ID/import
```

Here is an example of the cURL command:

```
curl -H "Authorization:Bearer
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJhZG1pbiIsInVzZXJJZCI6MSwiaWF0IjoxNTY2ODU5NTg2fQ.qV2j8ZLgNGNJsT8
OdPRwvE0-0y1x7w-0zr5h7Jz2d9uqY8tvACsV68posEUO9tD-YXlgXznX-IGnrnopDU7G3w" -F importFile=@"1184-
export-02-20-2021_09-40.zip" -F projectImportModel=@"Import217Settings.json;type=application/
json" http://localhost:8888/codeinsight/api/projects/217/import
```

Consider the following about the example command. (Keep in mind that users must explicitly provide their own values in the command based on their environment.)

- `JWT_TOKEN` has been replaced with an example user authorization token.
- `HOSTNAME:PORT` has been replaced with an example machine name and port—in this case, `localhost:8888`. (The IP address for the machine can be used instead of the machine name.)
- `FILE_TO_IMPORT` has been replaced with the name of an import data file (that is, the file name of the zip file to be imported) in the example. In this case, the import data file is `1184-export-02-20-2021_09-40.zip`. The entire file name must be enclosed in quotes.
- `ImportSettings` has been replaced with the name of the example file containing the import attributes. In this case, the file is `Import217Settings.json`. The entire file name and type must be enclosed in quotes.
- `PROJECT_ID` has been replaced with the ID of the project to which data is being imported (in this case, `217`). See Project ID for help on obtaining the project ID.

Importing Project Data Using the Postman API Client

Instead of explicitly executing a cURL command that calls the **Import Project Data** REST API, you can invoke the API using an API client such as Postman. For instructions on running an import using Postman, see the following Knowledge Base article in the Revenera Software Community:

https://community.flexera.com/t5/FlexNet-Code-Insight-Customer/Using-Postman-to-Execute-a-Project-Data-Import-in-FlexNet-Code-Insight/ta-p/145532

Verifying the Import Results

Use this procedure to verify that the import process completed as expected.

Task ***To verify that the import results are as expected, do the following:***

1. Open the target project in Code Insight and navigate to the **Project Inventory.**
2. Confirm that the total number of inventory items includes the newly imported items. (Keep in mind that, by default, only inventory with matching associated files in the target codebase are imported.)
3. Confirm that the inventory items contain accurate inventory details and file path associations.
4. If the import results are not what you expect, adjust the import configuration (see Import Behavior and Configuration), and run the import again.

Possible Benign Error When Processing Existing File Associations

During the import process, if a file association in the import data file already exists in the target inventory item, no new file association is added to the target inventory item. However, when the import processes this association, a “duplicate entry” exception similar to the following might be logged:

```
Duplicate entry ‘entry_id’ for key 'pse_inventory_group_files.UNIQ_FILE_GROUP
```

This exception is benign and has no impact on the regular file-processing behavior—that is, the existing file association is retained; no new file association is added.

5

Automated Analysis

The following topics describe the Automated Analysis of features in Code Insight:

- What is Automated Analysis?
- Supported Development Ecosystems
- Supported Archive Formats
- Additional Rule-based Detection Capabilities
- Handling of “Work in Progress” Inventory

What is Automated Analysis?

Code Insight provides Automated Analysis capability to automatically inventory various package formats without the need for manual analysis. New automated detection rules are delivered to Code Insight as part of the Electronic Update process and through internal processes.

Automated Analysis is used in both scanning scenarios outlined below:

- Local scanning where the codebase is uploaded to the Scan Server or synchronized to the server from a Source Control Management system like Git or Perforce.
- Remote scanning, where a scan-agent plugin performs a scan remotely on built artifacts or source code on an Engineering build server and sends results back to Code Insight. This applies to full scans performed internally by Automated Analysis, not forced

Supported Development Ecosystems

Code Insight provides native support for operating in many development ecosystems (each encompassing a language, package type, and public registry). See the following topics for more information:

- Supported Ecosystems
- Notes About Ecosystem Support

- More About Code Insight Support for Dependencies

Supported Ecosystems

The table below provides the following information about each ecosystem that Code Insight supports in the Automated Analysis process:

- **Language/File Type**—The code language or file type supported by the ecosystem.
- **Package**—The name of a package type in the ecosystem.
- **Registry**—The URL for the public registry or repository that hosts the package type.
- **Manifest File**—The file for which the Code Insight scan searches to locate a package of this type.
- **Top-level Inv.**—The indicator ✓ for "yes" or a dash (—) for "no", showing whether the Code Insight scan supports the detection of third-party software in the package (displayed as top-level inventory).
- **Direct Dep., Trans. Dep.**—The indicator ✓ for "yes" or a dash (—) for "no", showing whether the Code Insight scan detects the direct (first-level) dependencies and transitive dependencies (that is, dependencies of dependencies) of the component's top-level inventory.
- **Notes**—Link to notes (if available) pertaining to Code Insight's support of the specific ecosystem.

Table 5-1 ▪ Supported Ecosystems

Language/ File Type	Package	Registry	Manifest File	Top-level Inv.	Direct Dep.	Trans. Dep.	Notes
BitBake, BitBake recipe	Yocto	N/A	`.bb`	✓	N/A	N/A	See Yocto Ecosystems.
C, C++	Debian	https://tracker.debian.org https://www.debian.org	`.deb`	✓	N/A	N/A	See Debian Ecosystems.
C++, FORTRAN, Java, JavaScript, Lua, Python, R, Ruby, Scala	Conda	https://anaconda.org/	`index.json`	✓	✓	—	See Conda Ecosystems.
DLL/EXE	PE Header	N/A	`.dll, .exe`	✓	N/A	N/A	—

Table 5-1 • Supported Ecosystems (cont.)

Language/ File Type	Package	Registry	Manifest File	Top-level Inv.	Direct Dep.	Trans. Dep.	Notes
Go	glide	https://pkg.go.dev/	`glide.yaml`	✓	—	—	See Go Ecosystems.
	godep		`godeps.json`	✓	—	—	
	govendor		`vendor.json`	✓	—	—	
	module		`go.mod` `_go.mod`	✓	✓	✓	
			`go.sum`	—	✓	✓	
Java	Gradle	http://search.maven.org/	`build.gradle`	✓	✓	✓	See Gradle Ecosystems.
			`*.versions.toml`	N/A	✓	✓	
	Maven		`pom.xml`	✓	✓	✓	See Maven Ecosystems.
			`.jar`	✓	✓	✓	
			`.pom`	✓	✓	✓	
JavaScript	Bower	https://registry.bower.io/packages/	`bower.json`	✓	✓	—	—
			`.bower.json`	✓	✓	—	—
			`package.json`	✓	✓	—	—
Kotlin/ (.kts)	Gradle	http://search.maven.org	`build.gradle.kts`	✓	✓	✓	See Kotlin Support.
			`*.versions.toml`	N/A	✓	✓	
.NET	NuGet	https://api.nuget.org/v3-flatcontainer/	`.csproj`	✓	✓	✓	The registry URL provided is a parent URL and will not work as is. Provide the fully formed URL to access the appropriate component. Also see .NET Ecosystems.
			`.nupkg`	✓	✓	✓	
			`.nuspec`	✓	✓	✓	

Table 5-1 ▪ Supported Ecosystems (cont.)

Language/ File Type	Package	Registry	Manifest File	Top-level Inv.	Direct Dep.	Trans. Dep.	Notes

NodeJS	NPM	https://registry.npmjs.org/	`package.json`
			`package-lock.json`
			`OR`
			`npm-shrinkwrap.json`
				✓	✓	✓	See NPM Ecosystems.

	Yarn	https://registry.npmjs.org/	`package.json` `yarn.lock`	✓	✓	—	See Yarn Ecosystems.
PHP	Composer	https://packagist.org/	`composer.json`	✓	✓	—	—
			`composer.lock`	✓	✓	—	—
Python	PyPI	https://pypi.org/	`PKG-INFO`	✓	—	—	See PyPI Ecosystems.
			`requirements.txt`	N/A	✓	—	
			`setup.py`	✓	✓	—	
			`.whl`	✓	✓	—	
			`.egg`	✓	✓	—	
RPM	RPM Header	N/A	`.rpm`	✓	N/A	N/A	See RPM Ecosystems.
			`.spec`	✓	N/A	N/A	
Ruby	Gem	https://rubygems.org/api/v1	`.gem`	✓	✓	—	The registry URL provided is a parent URL and will not work as is. Provide the fully formed URL to access the appropriate component. Also see Ruby Ecosystems.
			`Gemfile`	✓	✓	—	
			`.gemspec`	✓	✓	—	
Rust	Cargo	https://crates.io/	`Cargo.toml`	✓	—	—	See Cargo Ecosystems.
			`.crate`	✓	—	—	

Table 5-1 ▪ Supported Ecosystems (cont.)

Language/ File Type	Package	Registry	Manifest File	Top-level Inv.	Direct Dep.	Trans. Dep.	Notes
Swift, Obj-C	CocoaPods	https://cocoapods.org/	`Podfile.lock`	✓	✓	✓	See Cocoapod Ecosystems.
			`.podspec`	✓	✓	✓	
Various	Git Repo	https://github.com	`config`	✓	—	—	See Git Ecosystems.

Notes About Ecosystem Support

The following sections provide additional information (such as limitations, requirements, and clarifications) to consider for the various ecosystems supported in the Code Insight Automated Analysis process:

- Cargo Ecosystems
- Cocoapod Ecosystems
- Conda Ecosystems
- Git Ecosystems
- Go Ecosystems
- Gradle Ecosystems
- Kotlin Support
- Maven Ecosystems
- .NET Ecosystems
- NPM Ecosystems
- PyPI Ecosystems
- RPM Ecosystems
- Ruby Ecosystems
- Yarn Ecosystems
- Yocto Ecosystems

Cargo Ecosystems

Note the following about Cargo ecosystems:

- Code Insight reports top-level inventory from pre-build and post-build artifacts such as `Cargo.toml` and `.crate` files, respectively.
- If a probable component is discovered but not matched against the crates forge in the Code Insight Data Library, Code Insight creates a non-published Work In Progress inventory item for the component.

Cocoapod Ecosystems

Note the following about Cocoapod ecosystems:

- For the detection of direct dependencies, having the `.podspec` file in the scan directory is desirable. In cases where the `.podspec` file is not present, the system does the following:
    - Uses the `podfile.lock` file to identify direct dependencies.
    - Uses the directory name to identify top-level inventory.
- For the detection of transitive dependencies, having both the `.podspec` and the `podfile.lock` file is desirable. In cases where `podfile.lock` is missing, the system uses the following external APIs to retrieve transitive-dependency information.
    - **https://cdn.jsdelivr.net/cocoa/Specs**
    - **https://cdn.cocoapods.org/**

Conda Ecosystems

First-level dependencies are supported for `index.json`, but the semver resolution of version is not yet supported.

Debian Ecosystems

Scans discover and report top-level inventory in Debian (`.deb`) packages as follows:

- If an inventory item discovered in a Debian package has a match in the Debian forge, the item is created as a published top-level inventory item with a High confidence level.
- Any item that does not have a match in the forge is created as an unpublished top-level inventory item with a “Work in Progress” designation and a Low confidence level.

Git Ecosystems

Code Insight scans the configuration file (`config` or `gitconfig`) inside a `.git` folder in a project codebase to identify OSS and third-party components and evidence and then uses this information to create inventory items.

***Note** ▪ To support the detection of components in a Git repository, the configuration file in a `.git` folder will always be included in scans even if this folder has been added to the **Scan Exclusions** list in the scan profile.*

Go Ecosystems

Note the following for Go ecosystems:

- Currently, Code Insight supports the discovery of inventory in scans of only pre-built artifact source code.
- Go module inventory is reported from the Go forge. If an item is not found in the Go forge, Code Insight searches for it in the GitHub forge.
- All Go module inventory will be associated with the `go.mod` or `_go.mod` file.
- The name of Go module inventory includes the module name, version, and license name (if applicable) in the format *componentName <version> (<license>)*. The following is an example:

    match v1.1.0 (MIT)

- If the codebase is uploaded from the release section of the VCS repository, Code Insight must use the version in the name of the project's parent folder as the version in the top-level inventory name. Any changes to the version in the parent folder name can result in the wrong version being reported in the inventory.
- For a pseudo version of a direct or transitive dependency, Code Insight shows only the main version part. For example, if the pseudo version is v0.0.0-20191202100458-e7afc7fbc510, Code Insight displays the dependency version as **v0.0.0** (created as a custom version).
- Top-level inventory that is generated with a Work in Progress type during Automated Analysis is not automatically published in the Analysis Workbench (even if the auto-publish feature is enabled for the project). You must manually publish such inventory items.

Direct Dependencies in Go Mod Pre-Built Artifacts

Code Insight supports the detection of direct dependencies in Go modules (`go.mod` and `_go.mod` files).

Transitive Dependencies in Go Mod Pre-Built Artifacts

All transitive dependencies found in the Go module `go.sum` are mapped to top-level inventory.

Gradle Ecosystems

Code Insight reports top-level inventory whenever a `build.gradle` file is located in the scan path. The top-level inventory name is either the `rootProject.name` attribute in `settings.gradle` or the name of the directory in which `build.gradle` resides.

***Note ▪** The `settings.gradle` file should be available in the same directory as `build.gradle`.*

Dependencies are reported from `build.gradle` if they are referenced with a `dependencies` tag within the file.

Gradle Version Catalog

A Gradle version catalog is a list of dependencies, represented as dependency coordinates, from which a user can select when declaring dependencies in a build script.

If the version catalog exists in a `*.versions.toml` file and the dependencies in the catalog are also listed in the `versionCatalogs` tag (available in the `settings.gradle` and `settings.gradle.kts` files), Code Insight will support these same dependencies referenced in any other `build.gradle` or `build.gradle.kts` files.

As recommended by Gradle, Code Insight supports, by default, the `libs.versions.toml` file if it exists under the project root or `gradle` folder.

Additionally, if the scan's profile is configured to detect **Only First Level Dependencies** or **All Transitive Dependencies**, top-level inventory items are created for all items in the `libraries` section of the `*.versions.toml` file, with the exception of those items also referenced in `build.gradle` or `build.gradle.kts`. These top-level inventories are created with their **Part of Product** usage property set to **No**.

Dependency Scope Reported

A dependency scope of either **Runtime** or **Non-Runtime** can be reported for dependencies found in the `build.gradle` or `build.gradle.kts` file. For more information, see Dependency Scopes.

Kotlin Support

Code Insight currently provides Beta support for Kotlin kts files. Code Insight reports top-level inventory from a .kts file and any dependencies with a dependencies tag within the file.

The top-level inventory name is either the rootProject.name attribute in the settings.gradle.kts file or the name of the directory in which build.gradle.kts resides.

Note* ▪ *The settings.gradle.kts file should be available in the same directory as build.gradle.kts.

The Gradle version catalog is also supported for build.gradle.kts files. See Gradle Version Catalog.

Maven Ecosystems

A dependency scope of either **Runtime** or **Non-Runtime** can be reported for dependencies found in .jar, pom.xml, and .pom files. For more information, see Dependency Scopes.

.NET Ecosystems

When .NET projects are created using .NET Core *and* no information about top-level items is available within the .csproj file, Code Insight does the following:

- Reports top-level inventory by using the .csproj filename as the inventory name
- Locates and reports the inventory's associated direct and transitive dependencies from this same file

This process requires an Internet connection to access the NuGet forge. For offline connections, top-level items and any associated dependencies are reported only if the top-level items are directly available in the .csproj file.

NPM Ecosystems

Note the following for NPM ecosystems:

- Code Insight provides scan support for package.json alone or for package.json with either package-lock.json or npm-shrinkwrap.json (if either exists).
- The package-lock.json or npm-shrinkwrap.json file is scanned only if it co-exists with package.json. The package.json file contains the component and dependency data. The package-lock.json or npm-shrinkwrap.json file is used to identify the exact dependency versions for components that Code Insight should pull from package.json.
- If both package-lock.json or npm-shrinkwrap.json are present with package.json, Code Insight scans npm-shrinkwrap (along with package.json) and ignores package-lock.json.

PyPI Ecosystems

Code Insight supports the discovery of top-level inventory and direct dependencies for both pre-build and post-build artifacts of a Python project. Pre-build artifacts include source packages, such as tar.gz, .zip, and other such files. Post-build artifacts are binary packages such as .whl and .egg files.

More About Direct Dependencies in Pre-Build Artifacts

Direct dependencies for the pre-build artifacts are retrieved from the `requirements.txt` file if it exists. (In the absence of `requirements.txt`, direct dependencies are reported from the `install_requires` section in the `setup.py` file.)

When direct dependencies are retrieved from `requirements.txt`, the top-level inventory item to which these dependencies are mapped is determined as follows:

- If `PKG-INFO` or `setup.py` resides in the same directory as `requirements.txt`, the top-level inventory item is determined by information in either `PKG-INFO` or `setup.py`.
- If `PKG-INFO` or `setup.py` does not reside in the same directory as `requirements.txt`, the top-level inventory item is determined in one of two ways:
    - If the Code Insight obtains the codebase through a `git sync` or `git clone` operation, the top-level inventory item to which direct dependencies are mapped is created from the configuration information found in the `.git` file.
    - If the codebase has been directly downloaded from a GitHub or PYPI repository and then uploaded to Code Insight for the scan, the top-level inventory item is created using the name of the directory under which `requirements.txt` resides. Direct dependencies identified in `requirements.txt` are then mapped to this inventory item.

      Note that, upon creation, such an inventory item is considered a “place holder” item because it is created from a directory name, which might or might not be a valid component name. The item is published during the automated analysis only if its name matches a valid component in the Code Insight Data Library, its forge is PyPI or GitHub, and it meets your site’s inventory publication policies. Otherwise, the item remains unpublished for further review.

      The inventory type for the item is determined as follows:

        - If the component name matches a component name in the Code Insight Data Library, the inventory type is **Component**.
        - If the component is not found in the Data Library but the inventory’s license matches a license in the Data Library, the inventory type is **License Only**.
        - If neither the component nor license has a match in the Data Library, the inventory type is **Work In Progress**.

More About Direct Dependencies in Post-Build Artifacts

Direct dependencies for the post-build artifacts are retrieved from the indicator files within these built artifacts.

- **.whl file**—The top-level inventory and direct dependencies are reported from the `METADATA` file, which resides within the `.whl` file. The `Requires-Dist` tag specified within the `METADATA` file resolves the direct dependencies.
- **.egg file**—The top-level inventory is reported from the `PKG-INFO` file, which resides within the `.egg` file. The direct dependencies are reported from the `requires.txt` file, which is bundled along with `PKG-INFO` file inside the `.egg` file.

RPM Ecosystems

Note the following about RPM Ecosystems:

- If automated analysis discovers the URL for a component in an RPM package or RPM `.spec` file but the component does not currently exist in the Code Insight database, the scan reports it as a *published* Work in Progress inventory item. (The discovered URL is also reported in the scan results.)
- If a `.spec` file has no Name tag value or if this value is a variable, no inventory is reported from this file.
- If a `.spec` file has no Version tag value or if this value is a variable, the scan reports the inventory from this file as Work in Progress inventory with no version. The inventory is published if the URL for the associated component is also reported in the scan results.
- If a `.spec` file contains a URL defined with a variable, the scan reports the inventory from this file as Work in Progress inventory with an unresolved URL.

Ruby Ecosystems

Note the following for Ruby ecosystems:

- For RubyGem projects, Code Insight shows all platform-related dependencies and those dependencies that are not part of a “test” or “dev” group as inventory. Any gems identified as “dev” or “test” are not considered for inventory.
- Only SemVer expressions in the *major.minor.patch* format are supported to resolve dependencies listed in the manifest file.

Yarn Ecosystems

Note the following for Yarn ecosystems:

- Code Insight provides scan support for `package.json` alone or for `package.json` with the `yarn.lock` file.
- The scan `yarn.lock` file is scanned only if it co-exists with `package.json`. (The `package.json` file contains the component and dependency data. The `yarn.lock` file is used to identify the exact dependency versions for components that Code Insight should pull from `package.json`.)

Yocto Ecosystems

Code Insight parses a `.bb` file only if it contains an `SRC_URI` property value that starts with **`git://`** or **`https://`**. If the `SRC_URI` property contains more than one URI, only the first supported URI is considered.

More About Code Insight Support for Dependencies

Code Insight supports scanning for top-level inventory items, direct dependencies, and transitive dependencies. The scan profile, managed by the Code Insight System Administrator, is used to configure of the desired depth of scan with respect to dependencies. See About Scan Profiles for information about scan profiles. The following sections provide more insights about dependency scanning:

- Dependency Scanning
- Dependency Scopes

Dependency Scanning

When configuring a scan or analyzing scan results, consider the following about scanning dependency scanning:

- Dependencies represent open-source packages that are referenced by the scanned codebase, but not necessarily present in the codebase.
- Dependency scanning is designed to be used when scanning pre-build artifacts, typically found in source-code bundles. Since this scenario relies on package-management configuration files, it is not 100% precise in the resolution of the declared dependencies. In many cases, dependencies will be resolved to the latest available version within the declared range. However, this version can differ from the actual package version pulled down as part of the build.
- Dependency scanning is not designed for scanning post-build artifacts when using the scan-agent plugins to scan on the build servers as part of the build process. In such scenarios, all dependencies have already been resolved by the build system and are present in the scanned codebase.

Dependency Scopes

A dependency has a scope of either runtime (that is, the dependency is required during application runtime) or non-runtime (it is not required during runtime). Depending on the value of the **Report Non-Runtime Dependencies** option in the scan profile, scan results can include just runtime dependencies *or* both runtime and non-runtime dependencies. (This option is available only when the scan profile is configured for first-level-dependency or transitive-dependency scans via the **Dependency Support** field.)

The scope of a reported dependency is shown in the **Dependency Scope** field listed in the inventory details on the **Project Inventory** tab and in the **Analysis Workbench**. The value of this field is either **Runtime** or **Non-Runtime**. This general scope designation is based on the ecosystem-specific scope with which the dependency is defined in the code.

See the following topics for more information:

- Current Code Insight Support for Dependency Scopes
- Gradle Dependency Scopes Supported by Code Insight
- Maven Dependency Scopes Supported by Code Insight
- NPM Dependency Scopes Supported by Code Insight

Current Code Insight Support for Dependency Scopes

Code Insight currently reports scopes for dependencies found only in the following manifest files for the given ecosystem. (Dependencies not found in these files show **N/A** for **Dependency Scope**.)

- `build.gradle` and `build.gradle.kts` files in Gradle Ecosystems
- `.jar`, `pom.xml`, and `.pom` files in Maven Ecosystems
- `package.json` file, `package-lock.json`, or `npm-shrinkwrap.json` in NPM Ecosystems

The next sections list the ecosystem-specific scopes with which dependencies found in these files can be defined and that the scan translates to the broader **Runtime** or **Non-Runtime** scope in the scan results. (The lists show only those scopes currently supported by Code Insight.)

Gradle Dependency Scopes Supported by Code Insight

The following list shows runtime and non-runtime Gradle-specific scopes with which dependencies found in a `build.gradle` or `build.gradle.kts` file can be defined. The Gradle scope for a given dependency is converted to the broader **Runtime** or **Non-Runtime** scope in the scan results.

Table 5-2 ▪ Gradle Dependency Scopes for Runtime and Non-Runtime

Runtime Scopes	Non-Runtime Scopes
`implementation`	`compileOnlyApi`
`api`	`compileOnly`
`runtime`	`testImplementation`
`runtimeOnly`	`testCompileOnly`
`runtimeElements`	`testRunTimeOnly`
`runtimeClasspath`	`compileOnlyApi`
`providedCompile`	`apiElements`
`providedRuntime`	`compileClasspath`
`default`	`testCompileClasspath`
`sourceSetRuntime`	`testRuntimeClasspath`
`sourceSetRuntimeOnly`	`annotationProcessor`
`sourceSetRuntimeClasspath`	`testCompile`
`apk`	`testRuntime`
`compile`	`sourceSetCompile`
`kapt`	`sourceSetImplementation`
`classPath`	`sourceSetCompileOnly`
	`sourceSetCompileClasspath`
	`sourceSetAnnotationProcessor`
	`provided`
	`testFixturesImplementation`
	`kaptAndroidTest`
	`kaptTest`
	`testAnnotationProcessor`
	`androidTestImplementation`
	`androidTestApi`
	`androidTestUtil`
	`EnforcedPlatform`
	`debugImplementation`
	`releaseImplementation`
	`androidTestImplementation`
	`androidTestCompile`
	`gradleApi`
	`gradleTestKit`

Maven Dependency Scopes Supported by Code Insight

The following list shows runtime and non-runtime Maven-specific scopes with which dependencies found in a `.jar`, `pom.xml`, or `.pom` file can be defined. The Maven scope for a given dependency is converted to the broader **Runtime** or **Non-Runtime** scope in the scan results.

Table 5-3 ▪ Maven Dependency Scopes for Runtime and Non-Runtime

Runtime Scopes	Non-Runtime Scopes

`compile`
`provided`
`runtime`
	`test`
	`system`
	`import`

Additional Notes About Maven Dependency Scopes

The following dependency behavior occurs during transitive scan (that is, scans whose scan profile is configured with the **All Transitive Dependencies** option).

- The scan reports dependencies from a dependency management tag with an `import` scope for a given `pom.xml` file. However, if these dependencies have dependencies from a dependency management tag in their respective `pom.xml` files, these next-level dependencies are not reported.
- Dependencies defined with the `test` scope for a given `pom.xml` file are reported. However, dependencies (also with a `test` scope) of these dependencies are not reported.

NPM Dependency Scopes Supported by Code Insight

The following list shows runtime and non-runtime NPM-specific scopes with which dependencies can be defined. The NPM scope for a given dependency is converted to the broader **Runtime** or **Non-Runtime** scope in the scan results.

Table 5-4 ▪ NPM Dependency Scopes for Runtime and Non-Runtime

Runtime Scopes	Non-Runtime Scopes

`dependencies`
`bundled`
`optional`
`peer dependencies`
	`devDependencies`

Additional Note About NPM Dependency Scopes

If only the `package.json` exists in the NPM ecosystem, transitive dependencies with a `devDependencies` scope are not reported during transitive scans (that is, scans whose profile is configured with the **All Transitive Dependencies** option). This behavior is in conformity with the NPM ecosystem behavior.

Supported Archive Formats

Note the following about the archives supported by Automated Analysis:

- Automated Analysis uses 7-Zip to read archive files. Refer to 7z Format on the external 7-Zip website to view the archive formats currently supported by this archiver product.
- Automated Analysis discovers inventories and evidences associated with `.apk` archives that are uploaded as part of the codebase (but are not part of an ecosystem, as listed in the Supported Ecosystems table). Note that Automated Analysis does not fetch any dependency information from these archives. The names of inventories generated from `.apk` archives are suffixed with “found inside <archive_name>.apk”.

Additional Rule-based Detection Capabilities

Automated detection used in the Automated Analysis process can also generate findings based on other rule-based techniques that include the following:

- Search term analysis
- File name analysis
- CDN analysis

Handling of “Work in Progress” Inventory

Any Work in Progress inventory item generated by a scan (whether performed by the Scan Server or by a remote scan plugin) will *not* be automatically published despite the project’s associated scan profile and its **Auto-Publish** rule based on the confidence level of inventory. (For more information about a project’s scan settings, see Updating Scan Settings for a Project and Edit Project: Scan Settings Tab.)

Note ▪ *The behavior described in this section does not always apply to Work in Progress inventory reported in the scan results for RPM packages and RPM `.spec` files. In certain cases, Work in Progress inventory items reported from these files are automatically published. See RPM Ecosystems for more information.*

Consider the following:

- Work in Progress inventory items that have been manually created or edited are not impacted by this rule during a rescan.
- This rule was introduced in Code Insight 2023 R2. For projects last scanned prior to Code Insight 2023 R2, this rule is applied accordingly to inventory generated during the next full rescan.

  Note ▪ *Forced full rescans (invoked by users) on the Scan Server and full rescans triggered internally by the different analysis techniques during regular rescans will regenerate previous system-generated inventory (ignoring any edits) and unpublish any previously published Work in Progress inventory according to this rule. See Rescanning Your Codebase (Server Scans Only) for more information.*

6

Performing Remote Scans

This section discusses Code Insight remote scanning. The following topics are covered in this section:

- About Remote Scans
- Code Insight Plugins
- Important: Plugin Upgrades in Code Insight

For complete information about performing remote scans, refer to the *Code Insight Plugins Guide*.

About Remote Scans

Code Insight has the ability to scan files on a remote system and manage the inventory items created from this remote location. This remote scan allows you to integrate automatic package-level scanning into your build process using a Code Insight scan-agent plugin. This integration includes automated package discovery (see Automated Analysis) and targeted components.

Refer to the following sections for more information:

- Creating a Project Without Uploading a Codebase
- Overview of How to Set Up for Remote Scanning
- How Remote Scans Work
- Viewing the Remote Scan Status
- Support for Processing Remote Scan Results in the Background

Creating a Project Without Uploading a Codebase

Some organizations might be interested in reviewing the inventory that results from a scan of their product's post-build artifacts on the build server. Other organizations might want to review the inventory resulting from a codebase scan but are reluctant to upload their product codebase (or synchronize a Source Control Management repository) to Code Insight. Instead, they want to keep their codebase in its existing development system due to security, consistency, or other concerns.

To address these requirements, Code Insight provides scan-agent plugins t

Using Scan-Agent Plugins

Code Insight offers scan-agent plugins that scan codebase files or built artifacts wherever they reside and send the results as inventory to the Code Insight Core Server for review and remediation by users. This process requires a Code Insight project on the Core Server for handling the returned results, but requires no codebase upload or synchronization to Code Insight.

Using Both a Scan Agent-Plugin and the Scan Server

Organizations might still want to upload a their product codebase to Code Insight to perform a server scan, but then use a scan plugin to remotely scan post-build artifacts directly on the build server. They can use the same Code Insight project to handle the results of both scans, enabling them to compare the resulting inventories, resolve discrepancies, and determine a final inventory list.

Overview of How to Set Up for Remote Scanning

The following is an overview of setting up for remote scanning.

Table 6-1 ▪ Overview of Setting Up for Remote Scanning

Phase	Description
1	Create a project in Code Insight. See About Code Insight Projects.
2	Create a valid JSON Web Token (JWT) for the user whose account will be used to connect to Code Insight. For instructions on generating the JWT, see Managing Authorization Tokens.
3	Install and configure the appropriate scan-agent plugin. (For information how to install and configure the plugin, see the *Code Insight Plugins Guide*.) As part of the configuration process, you will need to provide the name of the project that you created, the URL of the Code Insight core server, and the JWT.

When the scan-agent plugin is invoked (for example, during a build in Jenkins), the remote codebase will be scanned and any identified inventory items will be created in the existing project on the Code Insight server for further review and remediation.

How Remote Scans Work

Once a Code Insight scan-agent plugin is installed and the scan is configured as part of your build process, the scan agent, when run, collects and sends the scan results back to a project in Code Insight. The results provide information about the scanned files (including any license evidence found) and published inventory awaiting review, management, and remediation through Code Insight user interface.

As with published inventory generated by the Code Insight scan server, published inventory generated by a scan-agent plugin can be automatically reviewed by license or security policies as part of the scan and, for inventory not reviewed by policy, can be reviewed manually by legal or security experts. Security alerts with corresponding email notifications will be generated for any inventory item with new security vulnerabilities.

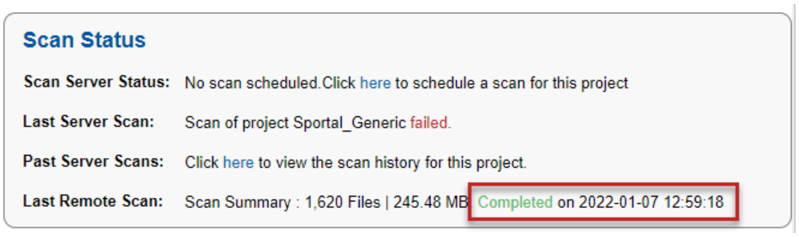
Considerations

Consider the following:

- For files scanned by a Code Insight scan-agent plugin on a remote system, currently only license evidence found in these files is currently reported in Code Insight.
- Code Insight does not generate email notifications for remote scan events.

Viewing the Remote Scan Status

To view the status of the import of scan results into your Code Insight project from the most recent remote scan run for your project, navigate to the project's **Summary** page. In the **Scan Status** section, locate the status and timestamp of the latest import. (See Summary Tab for a description of all the remote-scan details.)



Scan Status

Scan Server Status: No scan scheduled.Click here to schedule a scan for this project

Last Server Scan: Scan of project Sportal_Generic failed.

Past Server Scans: Click here to view the scan history for this project.

Last Remote Scan: Scan Summary : 1,620 Files | 245.48 MB Completed on 2022-01-07 12:59:18

Support for Processing Remote Scan Results in the Background

For scans performed by the generic or Docker Images scan-agent plugin only, the phase in which the scan results are processed in Code Insight for a given project is run as a background job that users can track in the **Jobs** queue. During such scans, the plugin sends the scan results in JSON format to Code Insight as a `.txt` file to be stored temporarily in Code Insight. Once the results are successfully sent, the job ID for processing the results in Code Insight is created and the job is added to the job queue. (The ID is also returned to the plugin to indicate that the results were successfully sent.) Users can track the progress of the results-processing as a **Remote Scan** job in the **Jobs** queue. (Progress is shown as **Active**, **Scheduled**, **Completed**, or **Failed**.) When Code Insight completes the processing, the temporary file is removed.

For more information about the **Jobs** queue, see Monitoring the Code Insight Jobs Queue.

For all other plugin scans, as soon as Code Insight receives the results, they are processed in the foreground.

About Running a Other Jobs Along with a Job That Processes Remote Scan Results

Note the following about running a job for remote-scan processing along with other jobs:

- Only one **Remote Scan** job can run on a project at a given time. All other **Remote Scan** jobs for the project are placed in a **Scheduled** state and run in scheduled order.
- When a **Remote Scan** job for a specific project is in progress or waiting to run (with an **Active**, **New**, **Waiting on Update**, or **Scheduled** status), a job for a project import, SBOM Insights export, report generation, or project deletion for the same project is placed in a **Scheduled** state and run in scheduled order. Likewise, when any of these other jobs are currently in progress or waiting to run, a new **Remote Scan** job for the same project is placed in a **Scheduled** state and run in scheduled order.
- When a **Remote Scan** job for a specific project is in progress or waiting to run (with an **Active**, **New**, **Waiting on Update**, or **Scheduled** status), a branching process cannot be performed on the same project. (The **Branch Project** option on the **Manage Project** menu for the project is disabled.) The user must wait for the **Remote Scan** job to complete before attempting to run the branching process again.

  Similarly, if a branching process is currently in progress or waiting to run for a specific project and a remote scan is attempted for the same project, the target project displays the message “Cannot import data for remote scan since project branching is in progress for the same project <projectid>”.
- When a **Remote Scan** job for a specific project is in progress or waiting to run (with an **Active**, **New**, **Waiting on Update**, or **Scheduled** status), a copy of this project cannot be performed. If a user attempts to run a project copy, a pop-up is displayed, explaining this situation. The user must wait for the **Remote Scan** job to complete before attempting to run the copy again.

  Similarly, if a project copy is currently in progress or scheduled and a remote scan is attempted for the same project, both the source and the target project display the message “Cannot import data for remote scan since project copy is in progress for the same project <projectid>”.
- If an Electronic Update is currently scheduled or in progress, all subsequent **Remote Scan** jobs are placed in a **Scheduled** state. Once the update is finished, these jobs are run based on the scheduled order.
- If a **Library Refresh** job is currently scheduled or in progress and a **Remote Scan** job is added to the queue, the job fails. You must wait until the Library Refresh is complete before having the scan-agent plugin run the scan again.

Code Insight Plugins

Code Insight offers the following scan-agent plugins for remote scanning.(Refer to the *Code Insight Plugins Guide* for a list of requirements for each scan-agent plugin.)

Table 6-2 ▪ Overview of the Standard Plugins

Build Environment	Code Insight Plugin	Performs automated scanning of...
IDEs	Eclipse	An Eclipse workspace in the Eclipse IDE environment.
	Visual Studio	A Visual Studio solution.

Table 6-2 ▪ Overview of the Standard Plugins (cont.)

Build Environment	Code Insight Plugin	Performs automated scanning of...
CI Tools	Azure DevOps	An Azure DevOps workspace as part of the build process.
	Bamboo	A Bamboo workspace as part of the build process (on Local Agents only)
	GitLab	GitLab projects as part of the build process.
	Jenkins	A Jenkins workspace as part of the build process. A separate plugin is available (called the Scan Schedule Plugin) that enables you to simply schedule the scan of a codebase residing on the Code Insight Scan Server via the Jenkins scheduler.
	TeamCity	TeamCity projects as part of the build process.
Package Manager and Build Tools	Ant	Apache Ant as part of the build process.
	Gradle	Gradle projects as part of the build process.
	Maven	Maven projects as part of the build process.
Container Platforms	Docker Images	Docker images on a Docker server.

Additionally, a generic scan-agent plugin is available with Code Insight that enables you to scan arbitrary file systems of your choice. It also easily integrates with certain Engineering systems, such as TeamCity and GitLab, to perform scans as part of a build process and can serve as an example for developing your own scan-agent plugin (as described in the *Code Insight Plugins Guide*).

Important: Plugin Upgrades in Code Insight

In Code Insight 2020 R2 and earlier, scan-agent plugins required an inventory-only project on Code Insight to which to send the results of the remote scans. Starting in Code Insight 2020 R3, the scan-agent plugins were upgraded to send the results of remote scans to a project type (also introduced in 2020 R3) that can manage results from both remote scans and scans performed by the Scan Server.

Currently, Code Insight continues to support existing inventory-only projects, enabling users to scan these projects using only the older plugins installed from previous Code Insight releases. However, inventory-only projects will be deprecated in a future release. For more information about the plugins upgrade, see the *Code Insight Plugins Guide*. Also see Legacy Projects.

7

Configuring Source Code Management

Code Insight provides a connector that allows you to use Source Code Management systems (SCM) as a source for codebase data. This section discusses the following topic:

- Managing Source Code Management (SCM) Instances
- Configuring an SCM Git Instance
- Configuring an SCM Perforce Instance
- Configuring an SCM Subversion Instance
- Configuring an SCM TFS Instance

Managing Source Code Management (SCM) Instances

Code Insight provides a Source Code Management (SCM) facility that synchronizes codebases on remote repository servers to the Scan Server. These codebases can then be scanned, enabling users to audit and review the results in Code Insight. Synchronization with remote codebases is enabled through Code Insight SCM connectors and through SCM instances configured for these connectors at the project level. The following sections provide information about using the SCM facility.

- Prerequisites
- About SCM Connectors, Instances, and Synchronizations
- Adding an SCM Instance to the Code Insight Project
- Testing an SCM Instance Connection
- Synchronizing an SCM Instance
- Editing an SCM Instance
- Deleting an SCM Instance

Prerequisites

Before performing the procedures in this section, ensure that an SCM command-line client is properly installed on the Code Insight Scan Server and that connectivity between the SCM client and the remote SCM server is properly configured. Refer to the “Integrating with Source Code Management” chapter in the *Code Insight Installation and Configuration Guide* for details.

If Code Insight is running as a service, the user context under which the service runs must have appropriate permissions to run the SCM client.

About SCM Connectors, Instances, and Synchronizations

The following describes the Code Insight SCM entities, operations, and the relationships between them.

SCM Connectors

The Code Insight SCM facility includes a set of connectors that enable codebase repositories on remote servers to synchronize to the Scan Server. Currently, SCM connectors are available for Git, Perforce, Subversion, and TFS repository types.

SCM Instances

Access to a given remote repository is enabled through an SCM instance defined at the project level by the Project Administrator. Each instance is configured for the appropriate SCM connector and identifies both the URL for a specific repository and the credentials needed to access that repository.

A given instance can identify the URL for a single repository only. (The exception is the SCM Git instance, which allows URLs for multiple repositories.) However, a project can have multiple SCM instances across one or more connector types.

SCM Synchronizations

Once the SCM instances for a project are defined and their connections tested, the Project Administrator should perform an initial synchronization to ensure that the codebases represented by these instances are properly synchronized to the Scan Server. Additionally, each time a scan is initiated on the project, another synchronization is automatically performed to ensure that the latest codebase(s) are in place before starting the scan. (The Project Administrator can continue to initiate synchronizations manually as needed for the project.)

The repository for a given instance is synchronized to following location on the Scan Server:

<scanroot>/<projectID>/<scm_instance_ID>

where

- <scanroot> is the Scan Server root directory.
- <projectID> is the ID of project to which you have added SCM instances.
- <scm_instance_ID> is the internal ID for the SCM instance, such **git.0**, **git.1**, and so forth.

Adding an SCM Instance to the Code Insight Project

The following procedure describes how to add an SCM instance in a Code Insight project.

Task ***To add or edit an SCM instance, do the following, do the following:***

1. Navigate to the **Summary** tab for the project for which you want to synchronize remote codebase files to the Scan Server.
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** page opens.
3. Select the **Version Control Settings** tab.
4. Select the desired SCM connector from the **Application** dropdown list. For continued instructions, see the appropriate section:
    - Configuring an SCM Git Instance
    - Configuring an SCM Perforce Instance
    - Configuring an SCM Subversion Instance
    - Configuring an SCM TFS Instance

Testing an SCM Instance Connection

If you add an SCM instance or edit any of the fields that define an existing instance, you should test the connection to the remote repository to ensure the repository is responsive.

Task ***To test your connection, do the following:***

1. Navigate to the **Summary** tab for the project for which you synchronizing remote codebase files to the Scan Server.
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** page opens.
3. Select the **Version Control Settings** tab.
4. Select the **Instance** tab for the connection you want to test.
5. Click **Test Connection** to confirm that Code Insight can connect to the repository. If the connection is successful, Code Insight displays a success message in the upper right corner.

    If the connection is not successful, an error popup is displayed, describing the error. You can edit the instance properties as needed and click **Test Connection** again.

Synchronizing an SCM Instance

Whenever you add an SCM instance or edit any of the fields that define an existing instance, first test its connection to the remote repository (see Testing an SCM Instance Connection). If the connection is successful, you should manually initiate a synchronization to ensure that the codebases across all SCM instances for the project are properly synchronized to the Scan Server.

Use the following procedure to manually initiate a synchronization.

Note *▪ Even though the scan process automatically performs a synchronization before actually scanning, a synchronization test initiated manually after any changes to a project's SCM instances can help to avoid problems with a subsequent scan and its results.*

Task ***To synchronize SCM instances to the Scan Server, do the following:***

1. Navigate to the **Summary** tab for the project for which you are synchronizing remote codebase files to the Scan Server.
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** page opens.
3. Select the **Version Control Settings** tab.
4. Click **Sync Now** to synchronize the codebase repositories from all the project's SCM instances to the following location on the Scan Server:

   <scanroot>/<projectID>/<scm_instance_ID>

   where <scanroot> is the Scan Server root directory, <projectID> is the ID of project for which you have added or edited SCM instances, and <scm_instance_ID> is the internal ID for the SCM instance, such **git.0**, **git.1**, and so forth.
5. If the synchronization successfully completes, a message in the upper right corner of the screen confirms this success.

   If the synchronization encounters an error, an error pop-up is displayed, explaining the problem. You can address the issue and retry the synchronization.

Additional Information About Synchronizations

Note this additional information about synchronizations:

- If multiple SCM instances have been added, clicking **Sync Now** will synchronize all instances in your project to the Scan Server. If the synchronization fails for one instance, the overall synchronization fails.
- If the Scan Server assigned to the project to which you are synchronizing codebase files is disabled, the **Sync Now** button is also disabled. Consider reassigning the project to an enabled Scan Server. (If necessary, see your Code Insight System Administrator for information about which servers are enabled).
- The numbered labels for the SCM instance tabs on the **Version Control Settings** tab are not the same as the SCM instance IDs, which are generated internally and used to identify the instance subdirectory on the Scan Server. (You can retrieve these internal IDs by calling the GET **scminstances** REST API for a project.) When using the REST interface to manage SCM instances, you must use the internal instance IDs.
- Unlike the other SCM instance types, an SCM Git instance allows URLs for multiple repositories. For more information about the synchronization of multiple URLs, see Processing of Git Repository URLs During a Synchronization.

Editing an SCM Instance

Use these steps to edit an existing SCM instance.

Task ***To edit an SCM instance, do the following:***

1. Navigate to the **Summary** tab for the project for which you are synchronizing remote codebase files to the Scan Server.
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** page opens.
3. Select the **Version Control Settings** tab.
4. Select the **Instance** tab for the instance you want to edit.
5. Update the fields as needed. Refer to the appropriate section for a description the instance fields:
   - Fields Used to Configure a SCM Git Instance
   - Fields Used to Configure an SCM Perforce Instance
   - Fields Used to Configure an SCM Subversion Instance
   - Fields Used to Configure an SCM TFS Instance
6. When you have completed the updates, click **Test Connection** to determine whether the instance can successfully connect to the remote repository. See Testing an SCM Instance Connection for details.
7. Click **Sync Now** to ensure that the codebase repository successfully synchronizes to the Scan Server. See Synchronizing an SCM Instance for details.

Deleting an SCM Instance

Use these steps to delete an SCM instance if it is no longer needed.

Task ***To delete an SCM instance, do the following:***

1. Navigate to the **Summary** tab for the project to which you intend to synchronize (or are synchronizing) codebase files.
2. From the **Manage Project** menu, select **Edit Project**. The **Edit Project** page opens.
3. Select the **Version Control Settings** tab.
4. Select the **Instance** tab for the instance you want to delete.
5. Click **Delete Instance**.

   A message is displayed, warning you that by deleting the instance, the codebase files can be permanently removed during the next scan on the project.
6. Click **Yes** to proceed (or **No** to discontinue the instance deletion).

Configuring an SCM Git Instance

Code Insight provides an SCM connector that enables codebases hosted on a Git server to synchronize to the Scan Server for scanning. The synchronization is enabled through an SCM Git instance, which both identifies the Git repositories that will synchronize to Scan Server and provides the credentials needed to access these repositories during the synchronization.

Refer to the following topics for more information:

- Adding an SCM Git Instance to the Code Insight Project
- Fields Used to Configure a SCM Git Instance
- Processing of Git Repository URLs During a Synchronization
- Migration of SCM Git Instances from a Code Insight Version Prior to 2023 R2

Adding an SCM Git Instance to the Code Insight Project

The following procedure describes how to add an SCM Git instance in a Code Insight project.

You can configure one or more Git instances, each instance identifying one or more repositories. All repositories identified in a given instance must use the same set of credentials.

Task ***To configure an SCM Git instance, do the following:***

1. Use the instructions in Adding an SCM Instance to the Code Insight Project to navigate to the **Version Control Settings** tab.
2. On the tab, select **Git** from the **Application** dropdown list.
3. Click **Add Instance**.
4. Define the SCM Git instance, referring to Fields Used to Configure a SCM Git Instance for a description of the instance settings. Alternatively, use the inline help provided for each field on the tab. As mentioned previously, you can enter multiple Git repository URLs in the instance.
5. Once you have completed the fields, click **Test Connection** to determine whether the instance can connect to all the repository URLs listed. If all connections are successful, a success message is displayed in the top-right corner of the screen.

   If one or more connections in the instance are unsuccessful, an error popup is displayed, showing the list of repository connections that failed. You can edit the fields as needed and test the connections again.
6. Once you have successfully set up the connections in the SCM Git instance, click **Add Instance** to create another SCM Git instance. Alternatively, select a different repository type from the **Application** dropdown list to create SCM instances of that type.
7. As a best practice, after you configured and tested the SCM instances for the project, click **Sync Now** to ensure that the codebases across all SCM instances for the project are properly synchronized to the Scan Server. See both Synchronizing an SCM Instance for general synchronization information and Processing of Git Repository URLs During a Synchronization for information specific to synchronization of Git instances.
8. Click **Save** to save the changes across all SCM instances to the Code Insight database.

Fields Used to Configure a SCM Git Instance

The following settings are used to configure a SCM Git instance.

Table 7-1 ▪ Setting Used to Configure an SCM Git Instance

SCM Git Instance Setting	Description
Git Repository URL(s)	Enter (or manage) one or more the repository URLs. You can enter a maximum of 50 URLs. For more information about adding and managing URLs in this field, see Adding and Managing Repository URLs in the “Git Repository URL(s)” Field.
Git Username	Provide the user name for authenticated access to *all* repositories listed in **Git Repository URL(s)** for this instance. Leave this and the **Git Password** field blank for an “anonymous” or SSH access. The blank values force the system to look for an SSH key pair on the server instead of attempting to verify a user name and password. (The *Code Insight Installation & Configuration Guide* provides for instructions on configuring Git over SSH.) Consult the Code Insight System Manager to ensure you are using the correct user credentials.
Git Password	Enter the password associated with the user name provided. Leave this and the **Git Username** field blank for an “anonymous” or SSH access. See the **Git Username** description for more information.

Adding and Managing Repository URLs in the “Git Repository URL(s)” Field

The following describes how to manage repository URLs in the **Git Repository URL(s)** field.

- Adding URLs
- Editing a URL
- Deleting a URL

Adding URLs

Use following instructions to add URLs to the **Git Repository URL(s)** field.

Task ***To add a URL:***

1. Click the add icon + above the **Git Repository URL(s)** field, and enter the URL in the text box that is displayed. The URL should be in one of these basic formats:

    - `http(s)://<host.xz>/<path>/to/repo.git`

- `<user>@<host>:<path>/repo.git`

Optionally, add a specific branch name, commit ID, or tag name to the end of the URL to specify a specific view of the repository. Use the appropriate delimiter to separate any of these elements from the main URL:

- For a specific branch, use ~~ before the branch name.
- For a specific tag, use ^^ before the tag name. (A tag marks a specific point in time in the history of the repository, such a version release or a bug fix.)
- For a specific commit, use >> before the commit ID.

Note ▪ *Because these delimited elements are mutually exclusive, you can add only one to a given URL. While you can save a URL that has multiple delimited elements, the test and synchronization processes will flag the URL as invalid.*

2. (Optional) Repeat the previous step to add another URL. You can enter a maximum of 50 URLs.

Examples

The following are example URLs that use the various formats described:

- `git@git.eng.flexera.com:org1/repo1.git`
- `git@git.eng.flexera.com:org1/repo2.git>>Commit123`
- `git@git.engi.flexera.com:org1/repo3.git~~dev-branch`
- `git@git.eng.flexera.com:org1/repo4.git^^bug-fix-4`
- `https://github.com/scaqaadmin/testgit_fnci.git^^v2018r1`
- `https://github.com/scaqaadmin/testgit_fnci>>217b44c0eb2b47bb43e22772ef109711ee5cfb3a`

Editing a URL

Use this procedure to edit a URL in the **Git Repository URL(s)** field.

Task ***To edit a URL:***

Click within the URL and edit it as needed within its text box.

Deleting a URL

Use this procedure to delete a URL from the **Git Repository URL(s)** field.

Task ***To delete an URL:***

1. Click within the URL to select it.
2. Click the remove icon -.

Processing of Git Repository URLs During a Synchronization

The following describes how SCM synchronization handles repository URLs identified in an SCM Git instance.

Codebase Structure on the Scan Server

All repositories for a single instance (Git, Perforce, Subversion, or TFS) are synchronized to the following location on the Scan Server:

<scanroot>/<projectID>/<scm_instance_ID>

where <scanroot> is the Scan Server root directory, <projectID> is the ID of project for which you have added or edited SCM instances, and <scm_instance_ID> is the internal ID for the SCM instance, such **git.0**, **git.1**, and so forth.

For a Git instance, each *different* repository identified in the instance is synchronized to a separate folder under the <scm_instance_ID> folder on the Scan Server. The folder name is based on the repository name in the URL. For example, suppose the following URLs are provided in the same instance:

```
https://github.com/scaqaadmin/scagit_prime1.git~~main
https://github.com/scaqaadmin/testgit_fnci>>217b44c0eb2b47bb43e22772ef109711ee5cfb3a
```

The two repositories are synchronized to the following respective locations on the Scan Server, where **C:\CodeBase** is the Scan Server root directory, **3388** is the project ID, and **git.0** is the SCM instance ID:

- `c:\CodeBase\3888\git.0\scagit_prime1`
- `c:\CodeBase\3888\git.0\testgit_fnci`

Synchronization of Multiple URLs for the Same Repository

The synchronization process maintains only one folder per Git repository on the Scan Server. If multiple URLs in an instance identify the same repository (but, for example, point to different branches, tags, or commit IDs within the repository), the synchronization retains the codebase for only the *last* URL listed for the repository in the **Git Repository URL(s)** field.

For example, suppose the following URLs are listed in this order in the **Git Repository URL(s)** field for an SCM Git instance. Each of these URLs identify the same Git repository, `testgit_fnci`, but point to different views of the repository:

```
https://github.com/scaqaadmin/testgit_fnci~~main
https://github.com/scaqaadmin/testgit_fnci>>217088
https://github.com/scaqaadmin/testgit_fnci~~nuget
```

The synchronization processes these URLs sequentially, overwriting each preceding codebase in the `testgit_fnci` folder on the Scan Server until the codebase for the last URL listed for `testgit_fnci` (in this example, `https://github.com/scaqaadmin/testgit_fnci~~nuget`) is synchronized. This is the codebase that is retained in the `testgit_fnci` folder on the Scan Server. (Any files common among each of the overwritten codebases are retained in the final synchronized codebase.)

Migration of SCM Git Instances from a Code Insight Version Prior to 2023 R2

Since Code Insight 2023 R2, SCM has supported the identification of multiple repositories in the **Git Repository URL(s)** field for a single SCM Git instance on a project's **Version Control Settings** tab. The previous **Branch/Tag/Commit** field on the tab was removed, requiring that any branch, tag, or commit delimiter be appended as part of the actual repository URL entered in the **Git Repository URL(s)** field.

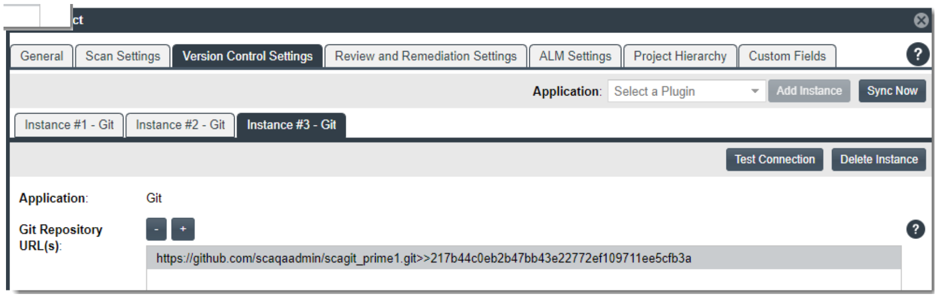
Due to this change, the migration of pre-2023 R2 SCM Git instances to the current version are handled as described in the following scenarios. Note that every migrated instance will contain only one URL in its **Git Repository URL(s)** field.

Scenario 1: A pre-2023 R2 instance containing a Git URL and no value in the "Branch/Tag/Commit" field

The **Git Repository URL(s)** field in the migrated instance shows the same URL used in the instance in the previous version.

Scenario 2: A pre-2023 R2 instance containing a Git URL and only a commit ID in the "Branch/Tag/Commit" field

The **Git Repository URL(s)** field in the migrated instance lists this URL with the commit ID separated by the appropriate delimiter, as in this example:



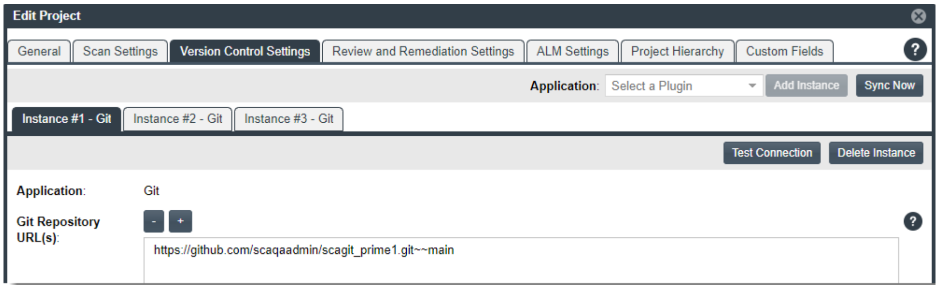
Application: Select a Plugin

Application: Git

Git Repository URL(s): https://github.com/scaqaadmin/scagit_prime1.git>>217b44c0eb2b47bb43e22772ef109711ee5cfb3a

Scenario 3: A pre-2023 R2 instance containing a Git URL and only a branch name in the "Branch/Tag/Commit" field

The **Git Repository URL(s)** field in the migrated instance lists this URL with the branch name separated by the appropriate delimiter, as in this example:



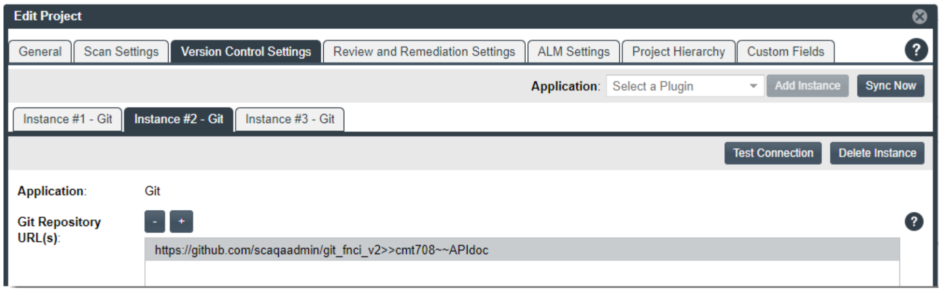
Application: Select a Plugin

Application: Git

Git Repository URL(s): https://github.com/scaqaadmin/scagit_prime1.git~~main

Scenario 4: A pre-2023 R2 instance containing a Git URL and both a commit ID and a branch name in the "Branch/Tag/Commit" field

The **Git Repository URL(s)** field in the migrated instance lists a single URL containing both the commit ID and the branch name, separated by the appropriate delimiters, as in this example:



Application: Git

Git Repository URL(s): https://github.com/scaqaadmin/git_fnci_v2>>cmt708~~APIdoc

The single URL in the migrated instance contains more than one delimited element (in this example, both a commit ID and a branch name). Because these elements are mutually exclusive, Code Insight allows only one per URL. (While you can save an instance containing a URL with multiple delimited elements, the test or synchronization process will flag the URL as invalid.) Therefore, edit the instance to show separate URLs, each pointing to the same repository but with a single delimited element. Based on the example above, you would edit the **Git Repository URL(s)** field in the instance to show these two URLs:

```
https://github.com/scaqaadmin/git_fnci_v2>>cmt708
https://github.com/scaqaadmin/git_fnci_v2~~APIdoc
```

Alternatively, create a separate instance for each URL.

Also keep in mind that, during synchronization when multiple URLs point to the same repository, only the codebase for the last URL processed in sequential order is retained on the Scan Server. Any files common among the URLs are also retained.

Configuring an SCM Perforce Instance

Code Insight provides an SCM connector that enables a codebase hosted on a Perforce server to synchronize to the Scan Server for scanning. The synchronization is enabled through a SCM Perforce instance, which both identifies the Perforce repository that will synchronize to Scan Server and provides the credentials needed to access the repository during the synchronization.

Refer to the following topics for more information:

- Adding an SCM Perforce Instance to the Code Insight Project
- Fields Used to Configure an SCM Perforce Instance

Adding an SCM Perforce Instance to the Code Insight Project

The following procedure describes how to add an SCM Perforce instance to the Code Insight project.

Task ***To configure an SCM Perforce instance, do the following:***

1. Use the instructions in Adding an SCM Instance to the Code Insight Project to navigate to the **Version Control Settings** tab.
2. On the tab, select **Perforce** from the **Application** dropdown list.
3. Click **Add Instance**.
4. Define the SCM Perforce instance, referring to Fields Used to Configure an SCM Perforce Instance for a description of the instance settings. Alternatively, use the inline help provided for each field on the tab.
5. Once you have completed the fields, click **Test Connection** to determine whether the instance can connect to the repository URL listed. If the connection is successful, a success message is displayed in the top-right corner of the screen.

   If the connection is unsuccessful, an error popup is displayed, briefly explaining the error. You can edit the fields as needed and test the connection again.
6. Once you have successfully set up the connection in the SCM Perforce instance, click **Add Instance** to create another SCM Perforce instance. Alternatively, select a different repository type from the **Application** dropdown list to create SCM instances of that type.
7. As a best practice, after you configured and tested the SCM instances for the project, click **Sync Now** to ensure that the codebases across all SCM instances for the project are properly synchronized to the Scan Server. See Synchronizing an SCM Instance for more information.
8. Click **Save** to save the changes across all SCM instances to the Code Insight database.

Fields Used to Configure an SCM Perforce Instance

The following settings are used to configure an SCM Perforce instance on Code Insight.

Keep the following in mind as you set up the instance, especially when providing the **Username** and **Password** credentials:

- The repository identified by the instance must reside on a Perforce server that is configured with Security Level 1, 2, or 3. The Code Insight Perforce connector does not support instances created for a Perforce server configured with Security Level 0, in which users are created without passwords.
- The Code Insight Perforce connector supports LDAP authentication on Perforce. If Perforce is configured with LDAP, you must provide the appropriate LDAP credentials for the **Username** and **Password** fields to access the Perforce repository identified by the instance.

Table 7-2 ▪ Setting Used to Configure an SCM Perforce Instance

SCM Perforce Instance Setting	Description

URL (P4PORT)	Enter the URL of the Perforce instance with which to synchronize. Note the following example URL formats:
	For a TCP connection
	`tcp:<p4ServerHostID>:<p4Port>`
	For an SSL connection
	`ssl:<p4ServerHostID>:<p4Port>`
	`p4ServerHostID` and `p4Port` identify the hostID (hostname or IP address) and port of the Perforce server.

Username (P4USER)	Provide the user name that has access to the Perforce depot to which this instance is synchronizing. If Perforce is configured with LDAP authentication, provide the LDAP user name.
Password (P4PASSWD)	Provide the password associated with the user name (see the previous field). If Perforce is configured with LDAP authentication, provide the LDAP password associated with the LDAP user name. If you are using a P4 ticket provided by the Perforce administrator, this field is optional.
Branch Spec (P4CLIENT)	Provide the path (in the following format) to the Perforce branch to which this instance is synchronizing: `<depot>/<projectPath>`
Changelist No	(Optional) Provide a changelist number only if this instance is synchronizing to a particular changelist. Otherwise, this value defaults to the latest revision.
Label	(Optional) Provide a label for the perforce branch.

Configuring an SCM Subversion Instance

Code Insight provides an SCM connector that enables a codebase hosted on a Subversion server to synchronize to the Scan Server for scanning. The synchronization is enabled through an SCM Subversion instance, which both identifies the Subversion repository that will synchronize to Scan Server and provides the credentials needed to access the repository during the synchronization.

Refer to the following topics for more information:

- Adding an SCM Subversion Instance to the Code Insight Project
- Fields Used to Configure an SCM Subversion Instance

Adding an SCM Subversion Instance to the Code Insight Project

The following procedure describes how to add an SCM Subversion instance to the Code Insight project.

Task ***To configure an SCM Subversion instance, do the following:***

1. Use the instructions in Adding an SCM Instance to the Code Insight Project to navigate to the **Version Control Settings** tab.
2. On the tab, select **Subversion** from the **Application** dropdown list.
3. Click **Add Instance**.
4. Define the SCM Subversion instance, referring to Fields Used to Configure an SCM Subversion Instance for a description of the instance settings. Alternatively, use the inline help provided for each field on the tab.
5. Once you have completed the fields, click **Test Connection** to determine whether the instance can connect to the repository URL listed. If the connection is successful, a success message is displayed in the top-right corner of the screen.

   If the connection is unsuccessful, an error popup is displayed, briefly explaining the error. You can edit the fields as needed and test the connection again.
6. Once you have successfully set up the connection in the SCM Subversion instance, click **Add Instance** to create another SCM Subversion instance. Alternatively, select a different repository type from the **Application** dropdown list to create SCM instances of that type.
7. As a best practice, after you configured and tested the SCM instances for the project, click **Sync Now** to ensure that the codebases across all SCM instances for the project are properly synchronized to the Scan Server. See Synchronizing an SCM Instance for more information.
8. Click **Save** to save the changes across all SCM instances to the Code Insight database.

Fields Used to Configure an SCM Subversion Instance

The following settings are used to configure an SCM Subversion instance for the Code Insight project.

Table 7-3 ▪ Settings Used to Configure an SCM Subversion Instance

SCM Subversion Instance Setting	Description
Subversion URL	Enter the URL of the Subversion repository containing the revision that you want to synchronize to your project. Use the following format: <protocol>://<host>/<subversionRoot>/<repository>
Username	Provide the user name needed to access the repository. Leave this field blank to make an anonymous connection.

Table 7-3 ▪ Settings Used to Configure an SCM Subversion Instance (cont.)

SCM Subversion Instance Setting	Description
Password	Provide the password needed to access the repository. Leave this field blank to make an anonymous connection.
Revision	(Optional) Enter the Subversion revision that you want to synchronize to your project. Leave this field blank to default to the latest revision.

Configuring an SCM TFS Instance

Code Insight provides an SCM connector that enables a codebase hosted on a Team Foundation Server (TFS) server to synchronize to the Scan Server for scanning. The synchronization is enabled through an SCM TFS instance, which both identifies the TFS repository that will synchronize to Scan Server and provides the credentials needed to access the repository during the synchronization. Refer to the following for more information:

- Adding an SCM TFS Instance to the Code Insight Project
- Fields Used to Configure an SCM TFS Instance

Adding an SCM TFS Instance to the Code Insight Project

The following procedure describes how to add an SCM TFS instance to the Code Insight project.

Task ***To configure an SCM TFS instance, do the following:***

1. Use the instructions in Adding an SCM Instance to the Code Insight Project to navigate to the **Version Control Settings** tab.
2. On the tab, select **TFS** from the **Application** dropdown list.
3. Click **Add Instance**.
4. Define the SCM TFS instance, referring to Fields Used to Configure an SCM TFS Instance for a description of the instance settings. Alternatively, use the inline help provided for each field on the tab.
5. Once you have completed the fields, click **Test Connection** to determine whether the instance can connect to the repository URL listed. If the connection is successful, a success message is displayed in the top-right corner of the screen.

   If the connection is unsuccessful, an error popup is displayed, briefly explaining the error. You can edit the fields as needed and test the connection again.
6. Once you have successfully set up the connection in the SCM TFS instance, click **Add Instance** to create another SCM TFS instance. Alternatively, select a different repository type from the **Application** dropdown list to create SCM instances of that type.

7. As a best practice, after you configured and tested the SCM instances for the project, click **Sync Now** to ensure that the codebases across all SCM instances for the project are properly synchronized to the Scan Server. See Synchronizing an SCM Instance for more information.
8. Click **Save** to save the changes across all SCM instances to the Code Insight database.

Fields Used to Configure an SCM TFS Instance

The following settings are used to configure an SCM TFS instance for the Code Insight project.

Table 7-4 ▪ Settings Used to Configure an SCM TFS Instance

SCM TFS Instance Setting	Description

TFS URL	Provide the URL of the TFS with which to synchronize. Note the following example URL formats.
	For the latest version of TFS:
	`<protocol>:<tfs_host>:<port>/<collection>/<project>`
	For earlier versions of TFS:
	`<protocol>:<tfs_host>:<port>/<collection>/<tfsroot>/<project>`

Username	Provide the user name that has access to the TFS collection to which this instance is synchronizing. If you are synchronizing with a VSTS project in TFS, enter the user name from the alternate authentication credentials enabled in VSTS. For details about enabling alternate credentials, refer to “Special Requirement for a VSTS Project in TFS” in the “Integrating with Source Code Management” chapter in the *Code Insight Installation and Configuration Guide*.
Password	Provide the password associated with the user name provided. If you are synchronizing with a VSTS project in TFS, enter the password from the alternate authentication credentials enabled in VSTS.
Changeset	(Optional) Provide the changeset number to which the SCM TFS instance is synchronizing. If no value is provided, this value defaults to the latest revision. If a changeset and label are both specified (see the **Label** description next), the label is ignored, and the instance synchronizes to the changeset.
Label	(Optional) Provide a specific label to which the SCM TFS instance is synchronizing. If a label and changeset (see the previous **Changeset** description) are both specified, the label is ignored, and the instance synchronizes to the changeset.

8

Pages and Panels

These sections serve as a reference to the following elements in the Code Insight Web UI:

- Add Project Dialog
- Add Token Dialog
- Add User Dialog
- Advanced File Search Add Dialog
- Advanced File Search Dialog
- Advanced Inventory Search Dialog
- ALM Tab
- Analysis Workbench
- Branch Project: Project Copy Settings
- Branch Project: Project Information
- Branch Project: Summary
- Branch Project: Upload Codebase
- Branch Project: Version Control Settings
- Branch Project Wizard
- Code Insight Dashboard
- Component Details Window
- Create (or Edit or View) Scan Profile Dialog
- Create Custom Detection Rule Dialog
- Custom Detection Rules Tab
- Edit (Default) Project Users Page

- Edit Custom Rule Dialog
- Edit Project: Custom Fields Tab
- Edit Project: General Tab
- Edit Project: Project Hierarchy Tab
- Edit Project: Review and Remediation Settings Tab
- Edit Project: Scan Settings Tab
- Edit Token Dialog
- Edit User Dialog
- Electronic Updates Tab
- Email Server Tab
- Evidence Details Tab in the Analysis Workbench
- File Search Results Pane
- Global Component & License Lookup Tab
- Import Project Data Dialog
- Inventory Details Tab in the Analysis Workbench
- Inventory History Window
- Inventory View
- Jobs Queue
- LDAP Tab
- License Details Window
- Lookup Component Window
- Policy Details Window
- Policy Page
- Preferences Page
- Project Defaults Tab
- Project Inventory Details Pane
- Project Inventory Tab
- Projects Pane and Associated Dashboard
- Reports Tab
- Scan History Dialog
- Scan Profiles Tab
- Scan Server Dialog
- Scan Servers Tab

- Security Vulnerabilities Window
- Select a New Project Contact Page
- Summary Tab
- Suppress Vulnerability Window
- Suppressed Versions of <component> for <vulnerability> Window
- Suppressed Vulnerabilities Tab
- System Settings Tab
- Unsuppress Vulnerability Window
- Users/Permissions Tab
- Versions for <component> Window

Add Project Dialog

The **Add Project** dialog displayed when you select to create a project from the **Project** pane (see Creating a Project). From this dialog, you define the basic properties for the new project using the following fields:

Table 8-1 ▪ Add Project Dialog

Column/Field	Description
Name	Enter a name for the new project.
Project Visibility	Select the default for visibility status—**Public** or **Private**—for the project. Any user in the system read-only access to a public project. To what degree a user can interact with the project depends on whether the user has a project role and what the role is—Project Administrator, Analyst, or Reviewer. However, private projects are hidden from all users except the Project Contact and those users assigned as Project Administrators, Analysts, Reviewers, or Observers of the project. Additionally, project and vulnerability ID searches will not return private projects unless the user performing the search has the permissions to see a given private project.
Scan Server	Select the local Scan Server for this project. Once this project is scanned, the Scan Server cannot be changed. (Select a Scan Server even if you are using a remote scan agent plugin to scan the project.)

Add Token Dialog

The **Add Token** dialog appears when you click the **Add Token** button on the **Preferences** page. It lets you create an authorization token (that is, a JSON Web Token known as a JWT) to be used to authenticate calls to Code Insight REST APIs. The token is associated with the user account under which you are logged in or whose password you have changed in the **Change Password** fields on the **Preferences** page. The dialog has the following fields:

Table 8-2 ▪ Add Token Dialog

Column/Field	Description
Name	Enter a name for the token you are creating.
Token Validity	Select one of the validity periods: ● **Never Expires**: The authorization token never expires. ● **Expires On**: The authorization token is valid until the date you pick on the Validity Calendar.
Validity Calendar	If you check the **Expires On** option, the validity calendar becomes active. Type an expiration date (for example, 10/10/10), or click the calendar icon and select a date.
Save	Click this button to save the token.
Cancel	Click this button to exit the **Add Token** dialog without saving the token.

See Also
Preferences Page
Edit Token Dialog

Add User Dialog

The **Add User** dialog on the **Administration** page allows you to add new users to the Code Insight system. The dialog contains the following columns and fields:

Table 8-3 ▪ Add User Dialog

Column/Field	Description
Login	Enter the login of the new user.
First Name	Enter the first name of new user.
Last Name	Enter the last name of new user.
Email	Displays the email address of the user associated with the login. To change the user’s email address, type over the existing email address.

Table 8-3 ▪ Add User Dialog (cont.)

Column/Field	Description
Password	The current password is not displayed in this field. A user with Code Insight System Administrator permissions can type a new password in this field.
Password Confirm	The current password is not displayed in this field. A user with Code Insight System Administrator permissions can type a new password in this field.
Question	The prompt that the user must answer to retrieve a forgotten password.
Answer	The answer to the question in the previous field.
Submit	Select **Submit** to have the system save your user edits. A prompt appears to notify you that your edits have been saved.
Cancel	Select Cancel to return to the Administration Users tab without saving your changes.

See Also
Users/Permissions Tab

Advanced File Search Add Dialog

The **Advanced File Search Add** dialog allow you to select a standard search, create your own search, or delete an existing search. The dialog has the following fields:

Table 8-4 ▪ Advanced File Search Add Dialog

Column/Field	Description
Name	The name of the search. For example, *Files not in inventory*.
Description	A short description of the search. For example, *Files not associated with inventory items*.

Table 8-4 ▪ Advanced File Search Add Dialog (cont.)

Column/Field		Description
Criteria	Use these fields to build the new search.	
	Add Criteria	From the dropdown list, select search criteria. To add more criteria, click **Add Criteria** and select another item from the dropdown list. When you select search criteria from the dropdown list, a boolean operator appears in the center dropdown, and a new dropdown appears from which you must select a criteria value by which to search the selected field. ***Note*** *▪ Files scanned by a remote scan agent can be searched by only the following criterion:* ***File Size****,* ***File Path****,* ***File Digest****,* ***Review Status****,* ***Inventory Status, Evidence status****,* ***Has license matches****,* ***Does not have license matches****, and* ***License****.*
	Add Criteria Group	Click to add a group of criteria.
Save	Click to save the new search.	
Save and Search	Click to execute the new search without saving the search for future use.	
Search without saving	Click to execute the new search without saving the search for future use.	
Cancel	Click to close the **Search Files** dialog without searching.	

Advanced File Search Dialog

The **Advanced File Search** dialog allows you to select a standard search, create your own search, or delete an existing search. The dialog has the following fields:

Table 8-5 ▪ Advanced File Search Dialog

Column/Field	Description
Add New	Click this button to access the **Advanced File Search Add** dialog.
Name	The name of the search. For example, *Files not in inventory*.
Description	A short description of the search. For example, *Files not associated with inventory items*.
✖	Click to delete a search.
Search	Click to execute the selected search.

Table 8-5 ▪ Advanced File Search Dialog (cont.)

Column/Field	Description
Close	Click to close the Search Files dialog without searching.

Advanced Inventory Search Dialog

The **Advanced Inventory Search** dialog is opened when you click the **Advanced Search** button at one of the following locations:

- **Inventory Items** pane on the Project Inventory Tab
- **Inventory Items** pane in the Analysis Workbench
- Inventory View

Considerations When Using Advanced Inventory Searches in the Analysis Workbench

Note the following when using the Advanced Inventory Search feature in the **Analysis Workbench**.

- If the **Inventory Items** list is filtered by published or not-published items (before or after using an Advanced Inventory search), the resulting inventory list is based on the published/not-published filter *and* the **Advanced Inventory Search** criteria.
- Search results from **Advanced Inventory Search** criteria and the results of inventory searches based on associated codebase files are mutually exclusive and will overwrite each other in the **Inventory Items** pane. (For more information about inventory searches based on an inventory's associated codebase files, see Showing Inventory Associated with Files Selected in the Codebase List.)

Field Descriptions

The **Advanced Search** dialog provides the following options that enable you to search project inventory in a variety of ways.

Table 8-6 ▪ Advanced Inventory Search Dialog

Section	Field	Description
Inventory Items	The following options enable you to filter inventory by inventory attributes.	
	Inventory Name	Enter the whole or partial inventory name by which to filter the inventory display. For example, if you enter apache in this field, Code Insight will find all inventory items that contain the *apache* string in their names. The name filter you enter here is automatically copied to the name filter field in the **Inventory Items** pane on the **Project Inventory** tab or in the **Analysis Workbench**. Likewise, if you have entered a name filter on the **Inventory Items** pane, it is automatically copied to this field on the **Advanced Inventory Search** dialog. This behavior enables you to keep the name filter persistent. You can always change or remove this filter as necessary at these locations.
	Inventory Review Status	Select one or more of the following checkboxes to filter the inventory display based on the review status of inventory items: ● **Approved**—Show only inventory that has been reviewed and approved, either manually by a reviewer or automatically during the auto-publish process. ● **Rejected**—Show only inventory that has been reviewed and rejected, either manually by a reviewer or automatically during the auto-publish process. ● **Not Reviewed**—Show only inventory that has not yet been reviewed. For more information about the review status, see Review Status of Inventory. When you select multiple options for this field, the search always applies “or” logic between the selections within the field.
	Inventory Priority	Select one or more of checkboxes (**P1**, **P2**, **P3**, or **P4**) to search the inventory by inventory priority. For more information about inventory priority, see Inventory Priority. When you select multiple options for this field, the search always applies “or” logic between the selections within the field.

Table 8-6 ▪ Advanced Inventory Search Dialog (cont.)

Section	Field	Description
	Inventory Type	Select one or more of these options to search inventory by its type: ● **Component**—Inventory based on registered component instances. (A registered component instance represents a unique component-version-license combination found in the Code Insight Data Library or database.) ● **License Only**—”Placeholder” inventory that a user might still need to verify as a valid inventory and whose evidence includes one or more groups of codebase files of unknown origin governed by a specific license. Such inventory is usually identified by the temporary name “Files under <License_name>”. ● **Work in Progress**—”Placeholder” inventory that a user might still need to verify as valid inventory and whose evidence includes third-party code or artifacts. When you select multiple options for this field, the search always applies “or” logic between the selections within the field.
	Created By	Select one of these options to search inventory by its creator type: ● **Any**—Show all inventory no matter the value of an inventory item’s **Created By** field. ● **Users**—Show only inventory that has been manually created by users. (This type of inventory shows the user’s first and last name in the inventory item’s **Created By** field.) ● **System**—Show only inventory generated automatically by Code Insight. (This type of inventory shows **System** in the inventory item’s **Created By** field.)
	Dependency Options	Select one of the following options to filter the inventory based on dependency level: ● **All Inventory Items**—Show all inventory (that is, all top-level inventory items, along with their first-level and transitive dependencies). ● **Only Top-Level Inventory Items**—Show all top-level inventory items only. No first-level or transitive dependencies are displayed. ● **Only Dependency Inventory Items**—Show only first-level and transitive dependencies. No top-level inventory is displayed.

Table 8-6 ▪ Advanced Inventory Search Dialog (cont.)

Section	Field	Description
	Dependency Scope	Select the dependency scope by which to filter inventory. The scope indicates whether or not the dependency is required at runtime. For more details about dependency scopes, see Dependency Scopes in the **Automated Analysis** section. ● **All**—Show inventory of any dependency scope: **Runtime**, **Non-Runtime**, or **N/A**. Inventory items with an **N/A** dependency scope are classified as neither runtime nor non-runtime dependencies. The **N/A** scope includes top-level inventory, those dependencies for which Code Insight does not currently support the reporting of scope, and migrated inventory for which a scan has not been run. ● **Runtime**—Show only dependencies that have a runtime scope (that is, are required at runtime). This scope selection is not applicable when the scan profile is configured to report no dependencies. ● **Non-Runtime**—Show only those dependencies that have a non-runtime scope (that is, are not required at runtime). This scope selection is not applicable under either of these conditions: ● The scan profile is configured to report no dependencies. ● The scan profile is configured to report only runtime dependencies. ***Note ▪*** *Your access to inventory of a specific scope in a project can change if a certain reconfiguration has previously occurred—for example, a change to the scan profile or a re-upload of updated runtime and non-runtime dependencies—and a rescan or full rescan has subsequently taken place.*

Table 8-6 ▪ Advanced Inventory Search Dialog (cont.)

Section	Field	Description
	Inventory Age	Select one of the following to filter the inventory display by the time frame in which the inventory items were published: ● **Last 1 day**—Show inventory published in the last day. For example, if today is Feb 6th, search from Feb 5th 12 AM. ● **Last 7 days**—Show inventory published in the last week. For example, if today is Feb 6th, search from Jan 30th 12 AM. ● **Last month**—Show inventory published in the last month. For example, if today is Feb 6th, search from Jan 7th 12 AM (30 days). ● **Custom Date Range**—Show inventory published within the specified time frame. Select a beginning (**From**) and ending (**To**) date from the pop-up calendar. ● **Any**—Show all published inventory.
	Inventory Notifications	Select one or more of the following checkboxes to filter the inventory display based on security vulnerability alerts: ● **Inventory with Open Alerts**—Show only inventory items that have open vulnerability alerts (that is, alerts for vulnerabilities that were discovered post-publication and have not been closed). ● **Inventory Rejected Due to New Non-Compliant Security Vulnerabilities**—Show inventory items that have been rejected due to new security alerts that are non-compliant with policy. When you select multiple options for this field, the search always applies “or” logic between the selections within the field.
	Inventory Confidence Level	Select one or more Confidence levels—**High**, **Medium**, or **Low**—by which to filter system-generated inventory items in the inventory display. The Confidence level is the measure of the strength of the discovery technique used by Code Insight to generate an inventory item. For a description of the Confidence levels and how they are used, see Inventory Confidence. When you select multiple options for this field, the search always applies “or” logic between the selections within the field.

Table 8-6 ▪ Advanced Inventory Search Dialog (cont.)

Section	Field	Description
	Inventory Usage	Usage describes how the OSS or third-party software (represented by a given inventory item) is used in your product. Select one or more values for one or more **Inventory Usage** criteria to filter inventory its usage. Note the following about the processing of this criteria: ● The criterion for each usage property defaults to **Any**, meaning the inventory can have any value in the search. The search does not filter inventory by a criterion with the **Any** value since no specific values are selected. ● You can select one or more values for a given **Inventory Usage** criterion. An inventory item must match a selected value for this criterion to be considered in the search. For example, if you select **Internal** and **Hosted** for the **Distribution Type** criterion, an inventory item must match either **Internal** or **Hosted** to be considered in the search. ● If you define multiple **Inventory Usage** criteria, the search uses “and” logic to process inventory against this criteria. That is, to be considered in the search, an inventory item must match a selected value for each **Inventory Usage** criterion. For example, if you select **Internal** and **Hosted** for the **Distribution Type** criterion and select **Dynamically Linked** for the **Linking** criterion, only inventory defined with **Internal** *or* **Hosted** for its **Distribution Type** *and* **Dynamically Linked** for its **Linking** property will be considered in the search.
		Distribution Type — The option indicating how the OSS or third-party component associated with an inventory item is distributed. The distribution type can affect license priority and obligations. ● **Internal**—The component is distributed internally only (for example, as an internal test framework included in the codebase but not distributed publicly with the software package). ● **External**—The component is a separate entity from your software package. It might be shipped as a separate component along with the software package or deployed through some method, such as a private cloud at the customer site. ● **Hosted**—The component is hosted in your company’s data center (for example, as a SAAS application) ● **Unknown**—The distribution type is unknown.

Table 8-6 ▪ Advanced Inventory Search Dialog (cont.)

Section	Field	Description	
		Part of Project	The option indicating whether the OSS or third-party component is part of the core product or an infrastructure piece such as a build or test tool. This can affect whether third-party notices are required for this item. The available values are be **Yes**, **No**, and **Unknown**.
		Linking	The option identifying how your software package links to the OSS or third-party component libraries. This method can affect license priority and obligations. ● **Not linked**—The software package uses no links to the component libraries. ● **Statically linked**—The component libraries are included in the software materials and thus linked statically. ● **Dynamically linked**—The component libraries are brought in at runtime. ● **Unknown**—The type of linking is unknown.
		Modified	The option indicating whether code from the OSS or third-party package has been modified for use by your organization. The available values are **Yes**, **No**, and **Unknown**.
		Encryption	The option indicating whether the component provides the encryption capabilities used in the product. Encryption can affect export controls. The available values are **Yes**, **No**, and **Unknown**.
Inventory Tasks	The following options filter inventory to show only those inventory items that have tasks. Refine the search using one or more task attributes—for example, task status, type, age, or owner.		
	Task Status	Select one of the following to filter the inventory display by the current status of the tasks associated with inventory: ● **Open Tasks**—Show inventory associated with at least open task. ● **Closed Tasks**—Show inventory associated with at least one closed task. ● **All Tasks**—Show all inventory associated with tasks, open or closed.	

Table 8-6 ▪ Advanced Inventory Search Dialog (cont.)

Section	Field	Description
	Tasks Type	Select one of the following to filter the inventory display by the type of task associated with inventory: ● **Manual inventory review**—Show inventory associated with a least one task requesting that a manual legal or security review be performed. (This review is needed to flag the inventory as accepted or rejected.) ● **Remediate Inventory**—Show inventory (currently or previously rejected) associated with at least one task requesting that software development take some action to make rejected inventory acceptable. ● **Miscellaneous**—Show inventory associated with at least one task requesting that additional attention of some sort be given to the inventory. ● **Any**—Show all inventory associated with tasks of any type.
	Inventory Tasks Age	Select one of the following to filter the inventory display by the time frame in which tasks associated with inventory items have been created: ● **Last day**—Show inventory associated with at least one task created within the last day. For example, if today is Feb 6th, search from Feb 5th 12 AM. ● **Last 7 days**—Show inventory associated with at least one task created within the last week. If today is Feb 6th, search from Jan 30th 12 AM. ● **Last month**—Show inventory associated with at least one task created within the last month. If today is Feb 6th, search from Jan 7th 12 AM (30 days). ● **Custom Date Range**—Show inventory associated with at least one task created in the specified time frame. Select a beginning (**From**) and ending (**To**) date from the pop-up calendar. ● **Any**—Show all inventory associated with tasks, no matter when the tasks were created.

Table 8-6 ▪ Advanced Inventory Search Dialog (cont.)

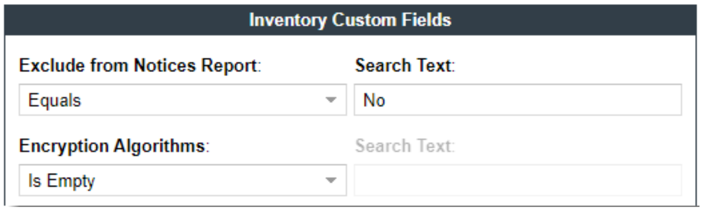
Section	Field	Description
	Inventory Task Owner	Select one of the following to filter the inventory display by the user who is assigned to tasks associated with inventory items: ● **Only Mine**—Show inventory associated with at least one task assigned to you (the current user). ● **Specific User**—Show inventory associated with at least one task assigned to the specified user. (A **Select user** pop-up enables you to select the user.) ● **Any**—Show all inventory associated with tasks, no matter to whom the tasks are assigned.
Inventory Custom Fields		The section is displayed only if one or more custom inventory fields have been defined for your site. If such fields have been defined, each field is listed, enabling you to set up a criterion for a given field that filters inventory by the field's value (or its lack of value). For each field whose value you want to use as a criterion for filtering inventory, do the following: **1.** Under the field name, select the search operation (**Contains**, **Equals**, or **Is Empty**) in the field on the left. A search based on the **Is Empty** criterion considers the field as *empty* if it has no value or only empty lines (or if it is designated as null in the REST interface). (An empty line is created by pressing the Return key but typing no characters.) The search ignores any field populated with one or more numbers, characters, or symbols. **2.** In the **Search Text** field on the right, enter the partial or full field value by which to search inventory. (This field is disabled if **Is Empty** is selected.)  Inventory Custom Fields Exclude from Notices Report: Equals Search Text: No Encryption Algorithms: Is Empty Search Text: If you have set up multiple custom fields as criteria, the **And** or **Or** operator pertaining across all fields in the advanced search (as selected for **Apply *x* Criteria** in the dialog) is applicable across the custom-field criteria. ● To appear in search results when **Or** is selected for the advanced search, an inventory item must contain at least one of the custom-field criteria you defined. ● To be a candidate in the search results when **And** is selected, an inventory item must meet *all* the custom-field criteria you defined.

Table 8-6 ▪ Advanced Inventory Search Dialog (cont.)

Section	Field	Description
Security Vulnerabilities	The following options enable you to filter inventory by the attributes of the security vulnerabilities associated with inventory items. If you accessed this dialog from the Inventory View, setting any of the following security-vulnerability criteria might increase the inventory search time significantly. ***Note*** ▪ *When you search by the ID or severity of a suppressed vulnerability, the results do not include inventory items associated with component versions for which the vulnerability was suppressed.*	
	Security Vulnerability ID	Enter the complete valid ID for the security vulnerability by which to filter the inventory display to show only those inventory items associated with the specified vulnerability.
	Security Vulnerability Severity	Select one or more vulnerability severity levels by which to filter the inventory display to show only those inventory items associated with at least one vulnerability that has one of the selected severities. The severity-level options differ depending on the CVSS version used by Code Insight. If CVSS v3.x (3.0 and 3.1) is used, the following severity options are available: ● Critical (CVSS score 9.0 - 10.0) ● High (CVSS score 7.0 - 8.9) ● Medium (CVSS score 4.0 - 6.9) ● Low (CVSS score 0.1 - 3.9) ● None (CVSS score = 0) If CVSS v2.0 is used, these severity options are available: ● High (CVSS score 7.0 - 10.0) ● Medium (CVSS score 4.0 - 6.9) ● Low (CVSS score 0.1 - 3.9) ● Unknown (N/A) For more information about vulnerability severities, see Security Vulnerabilities Associated with Inventory. When you select multiple options for this field, the search always applies “or” logic between the selections within the field.

Table 8-6 ▪ Advanced Inventory Search Dialog (cont.)

Section	Field	Description
	Security Vulnerability Age	Select one of the following options to filter the inventory display by the time frame in which security vulnerabilities associated with inventory items were detected. ***Note ▪*** *The detection date is either the inventory creation date (if a vulnerability was reported when the inventory was created) or the date that a new vulnerability applicable to this inventory was delivered by the update service.* ● **Last day**—Show inventory associated with at least one vulnerability detected within the last day. For example, if today is Feb 6th, search from Feb 5th 12 AM. ● **Last 7 days**—Show inventory associated with at least one vulnerability detected within the last week. For example, if today is Feb 6th, search from Jan 30th 12 AM. ● **Last 30 days**—Show inventory associated with at least one vulnerability detected within the last month. For example, if today is Feb 6th, search from Jan 7th 12 AM. ● **Custom Date Range**—Show inventory associated with at least one vulnerability detected within a specific time frame. Select a beginning (**From**) and ending (**To**) date from the pop-up calendar. ● **Any**—Show all inventory associated with security vulnerabilities, no matter when the vulnerabilities were detected.
Licenses and Versions	The following options enable you to filter inventory by attributes of the selected license for inventory items. If you accessed this dialog from the Inventory View, setting any of the following license criteria might increase the inventory search time significantly.	
	License Name	Enter the full or partial license name by which to filter the inventory display. For example, if you enter bsd in this field, Code Insight will find all inventory items whose **Selected License** value has the *bsd* string in its name.

Table 8-6 ▪ Advanced Inventory Search Dialog (cont.)

Section	Field	Description
	License Priority	Select one or more license priorities by which to filter the inventory display. The display will show only those inventory items whose **Selected License** has one of the priorities you select: ● **P1**— Viral/Strong Copyleft ● **P2**—Weak Copyleft/Commercial/Uncommon ● **P3**—Permissive/Public Domain ● No License Found For more information about license priority, see Auditing Scan Results in the Analysis Workbench. When you select multiple options for this field, the search always applies “or” logic between the selections within the field.
	Version	Select **No Associated Version** to filter to those licenses with no version associated with them.
Actions	The following are actions you can take to define criteria logic and apply the filters.	
	Apply *And* |*Or* Criteria	Select the boolean operator to apply to the search criteria: ● **Or**—To be included in the search results, an inventory item must contain at least one of the criteria you selected on this dialog. ● **And**—To be included in the search results, an inventory item must meet all the criteria across the advanced search, as selected in this dialog. (This is the default operator.)
	Apply	Click this button to apply the selected search criteria and return to the **Inventory Items** list (on the **Project Inventory** tab or in the **Analysis Workbench**) or to the **Inventory** view to see the results.
	Clear Form	Click this button to return the search criteria configuration to its default state.
	Close	Click this button to close this dialog and return to the **Inventory Items** list or the **Inventory** view without applying your search criteria.

ALM Tab

Code Insight integrates with application lifecycle management (ALM) systems, enabling Code Insight users to create and manage external work items from Code Insight. An *external work* item helps users keep track of inventory review or remediation work that needs to be performed outside of Code Insight.

The **ALM** tab on the **Administration** page is used by a Code Insight System Administrator to configure ALM instances. An *ALM instance* contains the information needed by Code Insight to access an ALM system and set up a work item. Once a Code Insight project is associated with an ALM instance, users can create work items on the ALM system and track work progress—all from Code Insight.

Currently, because Code Insight supports integration only with Jira as an ALM system, the descriptions in this section focus on the setup up a Jira ALM instance used to create issues on a Jira server from Code Insight.

The following sections describe the fields and variables used to define a Jira ALM Instance:

- Fields Used to Configure a Jira ALM Instance
- Use of Code Insight Variables in Text

For a complete description of how to configure and manage Jira ALM instances, refer to the "Integrating with Application Lifecycle Management" chapter of the *Code Insight Installation & Configuration Guid*e.

Fields Used to Configure a Jira ALM Instance

The following table describes the fields that the System Administrator uses to create or update a Jira ALM instance for integrating with the Jira system.

Table 8-7 ▪ ALM Tab

Group	Field	Description
Connector-level fields	The following fields configure an ALM connector (in this case, the Jira connector). The "Integrating with Application Lifecycle Management" chapter of the *Code Insight Installation & Configuration Guid*e provides more information about the relationship between the Jira connector and instances.	
	Application	To add instances to the Jira connector, select **Jira** from the dropdown. Your selection is displayed in the non-editable **Application** field in the body of the instance definition. The field is required.
	Add Instance	Click this button to open a new **Instance #*n* - Jira** tab (within the **ALM** tab) to create a new Jira ALM instance.

Table 8-7 ▪ ALM Tab (cont.)

Group	Field	Description
	Existing Issues Sync Frequency	Click the ✎ to the right of this field to select the synchronization frequency that will apply to *all* the instances configured on the **ALM** tab. The synchronization process keeps Code Insight up to date with the status of the Jira issues created from Code Insight through the Jira ALM instances. Synchronization options include: • **Never**—Never run a synchronization. (Default) • **Hourly**—Run a synchronization every *x* number of hours. For example, enter **2** to run a synchronization every 2 hours. • **Daily**—Run a synchronization daily at *x* clock time. For example, select **1:15 AM** from the associated list to run a synchronization at 1:15 am every day. (You can also enter a custom time.) • **Weekly**—Run a synchronization on *x* weekday at *x* clock time every week. For example, select **Monday** and **11:00 PM** from the associated lists to run a synchronization every Monday at 11 pm. (You can also enter a custom clock time.) Click ✓ to accept the updated synchronization frequency or ✖ to restore the previous frequency.
Functions available for the currently open instance tab	The following buttons apply to the currently opened Jira ALM instance tab.	
	Test Connection	Click this button to validate that the supplied **ALM Instance Name**, **JIRA Server URL**, **Authentication Type**, **JIRA Username**, and **JIRA Password/API Token** for the current Jira ALM instance enables Code Insight to successfully connect to the Jira Server. If the connection fails, check that these fields contain valid information. See a description of these fields later in this table.
	Delete Instance	Click this button to delete the current Jira ALM instance. If one or more projects are currently associated this instance, an error message is displayed, stating that you cannot delete the project due to project “references”. You must disassociate all projects from the instance before you can delete it.

Table 8-7 ▪ ALM Tab (cont.)

Group	Field	Description
Fields required to connect to the Jira system	The fields described in this section provide the information necessary for this Jira ALM instance to connect to the Jira Server. These are the only fields required to set up the Jira ALM instance. (All remaining fields on this tab are optional when creating the instance.) Once you have supplied values for these fields, you can use the **Test Connection** to determine whether your configuration successfully connects to the Jira Server. If not, ensure that you have entered a valid value each field. The connection is also tested when you attempt to save the instance. ***Important*** *▪ For a description of the prerequisites needed to help ensure a successful connection, see the “Prerequisites for Configuring the Jira Connector” section in the “Code Insight Installation & Configuration Guide”.*	
	ALM Instance Name	(Required) Enter a name for the Jira ALM instance. The name must be unique among all other Jira ALM instances defined in your Code Insight installation.
	JIRA Server URL	(Required) Enter the URL of the Jira server to connect to. Provide the URL in the format `http(s):<serverName_or_ipAddress>`.
	Authentication Type	(Required) Select the type of Jira deployment at your site: ● **Jira (On Cloud): Basic HTTP**—Jira is deployed on the Cloud Jira Server. You must provide a Jira user name and API token in the **Jira Username** and **Jira Password/API Token** fields, respectively. ● **On Premise: Basic Auth**—Your site uses an on-premise Jira Server and Data Center that requires a Jira user name and password in the **Jira Username** and **Jira Password/API Token** fields, respectively, as credentials. ● **On Premise Jira: Bearer Token**—Your site uses an on-premise Jira Server and Data Center that requires a personal access token (PAT) in the **Jira Password/API Token** field as credentials. (No **Jira Username** value is required.)

Table 8-7 ▪ ALM Tab (cont.)

Group	Field	Description
	Use the following fields to provide the credentials needed for the selected **Authentication Type**. For a successful connection, ensure that the specified user is a valid user on the Jira Server. Additionally, make sure that the user has full access to the URL instance specified for the Jira Server. This is particularly important if Captcha or Single Sign-On is enabled on the server. ***Note ▪*** *After a successful connection, the user is automatically designated as the reporter of the each Jira issue created from this instance.*	
	JIRA Username	(Required for **Jira (On Cloud): Basic HTTP** and **On Premise: Basic Auth**) Enter the user name for the Jira user.
	JIRA Password/API Token	(Required) Enter the appropriate value based on the **Authentication Type** selection: ● **For “Jira (On Cloud): Basic HTTP”**—Enter the API token associated with the user name. ● **For “On Premise: Basic Auth”**—Enter the password associated with the user name. ● **For “On Premise: Bearer Token”**—Enter the user’s personal access token (PAT).
Jira project for the instance	The following field identifies the Jira project (on the Jira server) with which any Jira issue created from this instance will be associated.	
	Default Project Key	Provide a default value for the key that identifies the Jira project with which the Jira issues (created from this instance) will be associated. This value can be edited when a project is associated with the instance. This field is optional when configuring the instance (but is required when a Code Insight project is associated with this instance, as described in Associating the Project with an Application Life Cycle System to Create Work Items).

Table 8-7 ▪ ALM Tab (cont.)

Group	Field	Description
Fields used to define a Jira issue	The fields described in this section are used to define the Jira issue. Note the following: ● These fields are optional when setting up an instance. ● A value entered for any of these fields serves as the field’s default. However, it can be overwritten by a Project Manager when associating a project with this instance or by a project user when creating a Jira issue. (For more information, see Associating the Project with an Application Life Cycle System to Create Work Items and Creating and Viewing External Work Items for a Project Inventory Task.) ● If you enter a default value, ensure that it is valid. Validation of these field values takes place during the creation of a Jira issue. At that time, if information entered for these fields is invalid (for example, the **Assignee** value does not exist in the Jira system), the information will still be saved, but the user will not be able to create the issue on the Jira server. ***Important ▪*** *A Jira issue on the Jira server must include the* ***Issue Type****,* ***Priority****,* ***Assignee****,* ***Summary Text****, and* ***Description Text*** *fields because these are the fields used to define the Jira issue in Code Insight. Any other field used to define a Jira issue on the Jira server must be configured as optional (that is, as not requiring a value). Deviation from this field configuration on the Jira server can cause the creation of Jira issues from Code Insight to fail. For complete information, see the “Prerequisites for Configuring the Jira Connector” section in the “Code Insight Installation & Configuration Guide”.*	
	Default Issue Type	Enter the type of issue created on the Jira server—**Bug** or **Task**.

Table 8-7 ▪ ALM Tab (cont.)

Group	Field	Description
	Default Priority	Select the priority level for the Jira issue: **1**—Highest **2**—High **3**—Medium **4**—Low **5**—Lowest ***Important ▪*** *The priority scheme for issues on the Jira server must match this priority scheme. If a different priority scheme is used on the Jira server, the creation of the Jira issue from Code Insight can fail. For complete information, see the "Prerequisites for Configuring the Jira Connector" section in the "Code Insight Installation & Configuration Guide".*
	Default Assignee	Enter the email for the user on the Jira server to whom you want to assign any Jira issues created from this instance.
	Default Summary Text	Enter the text that will display as the summary for the issue on the Jira server. This field supports the use of Code Insight variables, as described in Use of Code Insight Variables in Text.
	Default Description Text	Enter the text that will display as the description for the issue on the Jira server. This field supports the use of Code Insight variables, as described in Use of Code Insight Variables in Text.

Use of Code Insight Variables in Text

The **Default Summary Text** and **Default Description Text** fields support Code Insight variables that automatically pass information about the current Code Insight project and inventory item to the content in these fields.

Supported Variables

The following table lists the variables available for use in the text entered in the fields:

Table 8-8 ▪ Supported Code Insight Variables For Use in Work-Item Summary and Description Text

$PROJECT_NAME	Name of the Code Insight project containing the issue
$INVENTORY_ITEM_NAME	Name of the inventory item containing the issue
$COMPONENT_NAME	Name of the component associated with the inventory item
$VERSION_NAME	Version of the component associated with the inventory item
$LICENSE_NAME	Name of the selected license for the inventory item
$NUMBER_VULNERABILITIES	Total number of security vulnerabilities associated with the inventory item
$NUMBER_FILES	Total number of files associated with the inventory item
$INVENTORY_URL	Link to the inventory item

When the Jira issue is created, the included variables are replaced by their respective values.

Example Use of Variables

The following is example text you might enter in the **Default Summary Text** field. The text includes some of the available variables:

```
The $INVENTORY_ITEM_NAME inventory item in the project $PROJECT_NAME contains
$NUMBER_VULNERABILITIES vulnerabilities that require review. Go to $INVENTORY_URL to see the
vulnerable inventory item.
```

If your Code Insight project name is *MySampleProject* and the name of the inventory item name for which you create a work item is *Apache Commons BeanUtils*, the work item and Jira issue will display the following summary:

```
The Apache Commons BeanUtils 1.7.0 (Apache 2.0) inventory item in the project MySampleProject
contains 18 vulnerabilities that require review. Go to https://my.sample.server:8888/codeinsight to
see the vulnerable inventory item
```

Analysis Workbench

The **Analysis Workbench** is a facility that lets you examine the evidence in a project's scanned codebase in your project and interact with the inventory resulting from the scan.

The **Analysis Workbench** has the following fields.

__Note__ ▪ For files scanned by a Code Insight scan-agent plugin on a remote system, only license evidence is currently reported in Code Insight. The __Analysis Workbench__ indicates which remotely scanned files contain license evidence (a green icon is displayed next to the files under a remote scan-agent node in the __Codebase Files__ and __File Search Results__ pane) and lets you view this evidence on the __Evidence Details__ pane and a file’s __Evidence Summary__ pane.

__Note__ ▪ Some panes do not contain data until you choose a file in another pane.

Table 8-9 ▪ Analysis Workbench

Column/Field	Description
Legend	A color-coded and hyperlinked guide to the files and inventory in your scanned codebase: • **New Evidence** 📄: Click this link to filter the search results to display only files that are new since the last scan. If only a single scan took place, all files with evidence are displayed in the **Files Search Results** pane. • **Reviewed** ✓: Click this link to display files in the **File Search Results** pane that have been reviewed. • **Exact** 🟥: Click this link to display files in the **File Search Results** pane that are exact matches. • **Copyrights** 🟦: Click this link to display files in the **File Search Results** pane that contain copyright text. • **Email/URLS** 🟪: Click this link to display files in the **File Search Results** pane that contain email addresses and URLs. • **Licenses** 🟩: Click this link to display files in the **File Search Results** pane that contain licenses. • **Search Terms** 🟦: Click this link to display files in the **File Search Results** pane that match default search terms. • **Source** 🟨: Click this link to display files in the **File Search Results** pane that match

Table 8-9 ▪ Analysis Workbench (cont.)

Column/Field	Description
Codebase Files pane	A tree listing the files in the project codebases. The tree can include one or more nodes, each node identifying a specific Scan Server or remote scan agent and listing the files scanned by that scanner. A given file in the list can show the following information: ● If the file contains evidence, one or more color-coded indicators for the type of evidence. See the previous **Legend** description for the indicator meanings. ● A check mark if the file has been reviewed. To search for files and folders whose name contains a specific string, provide the string (at least three characters) in the **Enter search string...** field and click 🔍. The list is expanded as much as necessary to highlight the matching files. Click > or < to navigate to each result highlighted in the list. To explore more information about a file and its evidence, click its hyperlinked file name. Alternatively, right-click the file name for more options (see Managing the Codebase Files for more information).
File Search Results pane	A tree listing the files resulting from a search (other than the name search used in the **Codebase Files** pane). The tree is organized by the nodes and directories containing the files. Drill down into the nodes and directories to view the files. A given file in the list can show the following information: ● If the file contains evidence, one or more color-coded indicators for the type of evidence. See the previous **Legend** description for the indicator meanings. ● A check mark if the file has been reviewed. To explore more information about a file and its evidence, click its hyperlinked file name. Alternatively, right-click the file name for more options (see Managing the Codebase Files for more information).
File Details tab	Click a codebase file in the **Codebase Files** or **File Search Results** pane on the left to open the **Files Details** tab (in the middle pane). This tab includes a expandable header that lists metadata about the selected codebase file, as well as the three sub-tabs—**Evidence**, **Exact Matches**, and **Partial Matches**—available to examine the file's open-source or third-party evidence. For more information, see Examining and Managing Open-Source Evidence for a Given File.

Table 8-9 ▪ Analysis Workbench (cont.)

Column/Field	Description	
Inventory Items (*x*) pane	The list of inventory items for the project. You can filter this list by published or not-published inventory or by inventory name.	
	Current View	The project inventory that is being displayed—inventory items filtered by codebase files to which they are associated or by inventory attributes.
	Published (*x*)/Not Published (*x*)	Select the option to filter the inventory to all published or all not-published items. The *x* value is the number of each type. If you filter by published or not-published items and then filter by **Advanced Search** criteria (or vice versa), the resulting inventory list is based on the published/not-published filter AND the **Advanced Search** criteria.
	Clear Filter	Clears any inventory-list filters that have been applied (name, published, or not-published filters and **Advanced Search** filters).
	Advanced Search	Click this button to perform an advanced search on the inventory (for example, by inventory review status, priority, confidence level or associated licenses, security vulnerabilities, tasks, and more). The Advanced Inventory Search Dialog is displayed.

Table 8-9 ▪ Analysis Workbench (cont.)

Column/Field		Description
	inventory name filter	Enter a string by which to search and filter the inventory by name. As you type each character in the string, the list is automatically filtered according to the entered characters. This current name filter is automatically copied to the Advanced Inventory Search Dialog if you perform an Advanced Inventory Search (by clicking **Advanced Search**). Likewise, if you enter a name filter on the **Advanced Inventory Search** dialog, it is copied back to this field. This behavior enables you to keep the name filter persistent. However, you can always change or remove the filter in either location.
	Add New	Click to create a new inventory item. The **New Inventory Item** tab is opened. See Creating Inventory From the Inventory Items List for more information.
	Name	The column listing the name for each inventory item in the **Inventory Items** list. Click the column header to sort the list alphabetically in ascending or descending order by inventory name. To show or edit information about an inventory item, click the hyperlinked inventory name to open its **Inventory Details** tab (in the middle pane). See Inventory Details Tab in the Analysis Workbench for details.
	#Files	The column listing the number of codebase files associated with each inventory item in the **Inventory Items** list. Click the column header to sort the list in ascending or descending order by the number of files associated with the items.
	Publish	To publish an inventory item in the **Inventory Items** list, right-click the item and select **Publish Inventory**.
	Recall	To recall a published inventory item in **Inventory Items** list if it does not fit the criteria for published inventory, right-click the item and select **Recall Inventory**. The status of the item is changed to “Not Published”.
	Delete	To remove an inventory item from the **Inventory Items** list, right-click the item and select **Delete**.
Inventory Details tab	To show or edit information about an inventory item in the **Inventory Items** list, click the hyperlinked inventory item to open its **Inventory Details** tab (in the middle pane). See Inventory Details Tab in the Analysis Workbench for details.	

Table 8-9 ▪ Analysis Workbench (cont.)

Column/Field	Description
Evidence Details	Click **Evidence Details** (in the middle pane header) to open the **Evidence Details** tab in the middle tab. From here, you can view a summary of OSS and third-party evidence found across the codebase during the last scan. You can also filter the evidence based on files selected in the **Codebase Files** pane; or filter the files in the **Codebase Files** pane by selected evidence.

See Also
Opening the Analysis Workbench
The Analysis Workbench Layout
Reviewing Published Inventory for a Project

Branch Project: Project Copy Settings

The **Project Copy Settings** page in the **Branch Project** wizard defines the parameters used by the branching process to import file-audit data, inventory, and inventory-review information from the source project to the branched project.

If neither **Upload Codebase** nor **Sync from Control Version** was selected on the **Project Information** page, this import process copies only inventory and inventory-review information from the source project to the branched project. In this scenario, no file information will be associated with the inventory copied to the branched project.

For a description of the procedures related to the **Project Copy Settings** page, see the following:

- Branching a Project
- Step 4: Configuring a Project Copy

The following describes the properties and actions available on the **Project Copy Settings** page:

Table 8-10 ▪ Branch Project: Project Copy Settings Page

Field	Description
Add Files to Inventory	Refer to Import Project Data Dialog for a description of this field and its related **File matching criteria** fields. The branched project is the same as the “target project” in this description. If the branching process is performing an inventory-only copy to the branched project (that is, if neither **Upload Codebase** nor **Sync from Control Version** was selected on the **Project Information** page), this criterion is ignored during the branching process.

Table 8-10 ▪ Branch Project: Project Copy Settings Page (cont.)

Field	Description
Mark Files as Reviewed	Refer to Import Project Data Dialog for a description of this field and its related **File matching criteria** fields. The branched project is the same as the “target project” in this description. If the branching process is performing an inventory-only copy to the branched project (that is, if both **Upload Codebase** and the **Sync from Control Version** are not selected on the **Project Information** page), this criterion is ignored during the branching process.
Inventory Notes Handling	Refer to Import Project Data Dialog for a description of this field. The branched project is the same as the “target project” in this referenced description.
Inventory Usage Handling	Select one of these options to define how the branch process should handle the Usage attributes for inventory items: ● **Copy existing usage field values**—Copy the existing Usage values for the inventory items from the source project to the branched project. (Default) ● **Reset usage field values to system default**—Do not copy existing Usage values for inventory items from the source project to the branched project. Instead, in the branched project, reset all Usage fields for these inventory items to the system default value: **Unknown**. For a description of the inventory usage fields, refer to Usage tab in the **Project Inventory Details Pane** topic.
Next	Click this button to validate the information on this page. If errors are found, you must correct them before moving to the next page. If no errors exist, the **Summary** page in the wizard open.
Back	Click this button to move to the previous wizard page. Note that if you navigate back to the **Project Information** page, you cannot edit the **Name** field for project. Additionally, if you uploaded a codebase from the **Upload Codebase** page, the **Scan Server** and **Upload Codebase** options on the **Project Information** page are disabled.
Cancel	Click this button to cancel the project-branching setup. The branched project and any uploaded codebases for the project are deleted from Code Insight.

Branch Project: Project Information

The **Project Information** page in the **Branch Project** wizard identifies the essential information needed to create the branched project. Certain fields are pre-populated with values from the current project but can be edited as needed for the new project. After you complete the fields and click **Next**, the branched project is created if the information you provided is valid.

For a description of the procedures related to the **Project Information** page, see the following:

- Branching a Project
- Step 1: Creating the Branched Project

The following describes the properties and actions available on the **Project Information** page:

Table 8-11 ▪ Branch Project: Project Information Page

Field	Description
Name	Enter a name for the branched project. The name must be unique in your Code Insight system.
Description	If necessary, edit the description for the branched project. This field is initially populated with the description of the source project.
Policy Profile	If necessary, select a different policy profile for the branched project to automate its inventory review process. This field is initially populated with the policy profile used by the source project. A given policy profile uses a combination of policies to automatically mark published inventory items as approved or rejected without the need of a manual review. (Inventory items that are neither approved or rejected by policy are marked as **Not Reviewed** and will require a manual review.) For more information about policy profiles, see Policy Details Window and Managing Policies to Automatically Review Inventory.
Scan Server	If necessary, select a different Scan Server that will scan the codebase for the branched project. This field is initially populated with the Scan Server used by the source project.
Scan Profile	If necessary, select a different scan profile that defines the settings applied whenever the branched project is scanned. This field is initially populated with the scan profile used by the source project.

Table 8-11 ▪ Branch Project: Project Information Page (cont.)

Field	Description
Project Visibility	If necessary, change the visibility attribute—**Public** or **Private**—of the branched project. ● **Public**—A project that provides read-only access to any user in the system. To what degree a user can interact with the project depends on whether the user has a project role and what the role is—Project Administrator, Analyst, or Reviewer. ● **Private**—A project that is hidden from all users except the Project Contact and those users assigned as Project Administrators, Analysts, Reviewers, or Observers of the project. Additionally, project and vulnerability ID searches will not return private projects unless the user performing the search has the permissions to see a given private project. This field is initially populated with the attribute of the source project.
Project Risk	If necessary, change the vulnerability risk value (**Low**, **Medium**, or **High**) of the branched project. This field is initially populated with the risk value of the source project.
Project Status	If necessary, change the status of the branched project. (The meaning of these statuses might be adjusted for your site.) ● **Not Started**—Indicates that the project scan results are not yet available for manual analysis. ● **Analysis in Progress**—Indicates that the project scan results are available for manual analysis, or a manual analysis is underway. ● **Analysis Completed**—Indicates that manual analysis is finished; and the published inventory is ready for legal, security, or software-engineering review. ● **Project Complete**—Indicates that the project analysis and review processes are complete, and notices content for all OSS and third-party software is finalized. This field is initially populated with the status of the source project.

Table 8-11 ▪ Branch Project: Project Information Page (cont.)

Field	Description
Source Code Options	Select one or both options (or neither option) defining the method for obtaining source code for branched project: ● **Upload Codebase**—The source codebase is uploaded from an archive accessible from current instance. By default, this option is always selected. (You can upload multiple codebases.) ● **Sync from Source Control**—The source code base is obtained through a synchronization process with one or more instances of your site's Source Control Management (SCM) system. For more information about the synchronization process with SCMs, see Configuring Source Code Management. This option is selected by default only if SCM instances were configured in the source project. Alternatively, to copy only inventory and inventory-review information from the source project to the branched project, do not select either of these options. No file information will be associated with the inventory copied to the branched project.
Copy Project Users	Select the option that determines whether project roles of the source project are copied to the branched project: ● **Yes**—All current project-role assignments—including Project Administrators, Analysts, Reviewers, Observers, and the Legal, Developer and Security contacts—are copied from the source project to the branched project once the branching process completes. (This is the default for the branching process.) The user who performs the branching process is assigned the Project Contact role. ● **No**—The Project Administrator user who is creating the branched project is assigned to each of the project roles in the branched project. Additionally, those users assigned to roles by the System Administrator as global defaults for all new projects are added to the branched project (see "Setting Project Defaults" in the *Code Insight Installation & Configuration Guide*).

Table 8-11 ▪ Branch Project: Project Information Page (cont.)

Field	Description
Retain Child Project Links	Select the option that determines whether the entire child hierarchy of the source project is copied to the branched project. This hierarchy includes the child projects directly associated with the source project, any child projects of those projects, and so on. (See Identifying Child Projects for a Project for more information about project hierarchy.) When this hierarchy is copied, links to the child projects are also copied. This option does not copy the parent hierarchy that might be associated with the source project. ● **Yes**—The entire child hierarchy of the source project is copied to the branched project, along with links to the projects in the hierarchy. ● **No**—The child hierarchy of the source project is *not* copied to the branched project.
Next	Click this button to create the project and move to the next wizard page: ● If selected **Upload Codebase** in the **Source Code Options** section, the Branch Project: Upload Codebase page opens. ● If you selected only **Sync from Source Control** in the **Source Code Options** section, the Branch Project: Version Control Settings page opens. ● If you selected neither option, the Branch Project: Project Copy Settings page opens. However, if errors are found on this page, you must correct them before the project can be created and you can move to the next page.
Cancel	Click this button to cancel the project-branching setup.

Branch Project: Summary

The **Summary** page in the **Branch Project** wizard provides an overview of the parameters that you defined for the project-branching process. From this page, you can start the branching process. Alternatively, you can navigate back to other pages in the wizard to make changes before starting the branching process, or you cancel the entire branching setup.

Once the branching process starts, any SCM synchronization is performed first. Then the branching process scans the branch-project codebase and finally performs an import to copy file-audit data, inventory, and inventory-review information from the source project to the branched project. For an overview of the branching-operation phases, see Overview of the Branching Operation.

For a description of the procedures related to the **Summary** page, see the following:

- Branching a Project
- Step 5: Initiating the Branching Operation

The following describes the actions available on the **Summary** page:

Table 8-12 ▪ Branch Project: Summary Page

Field	Description
Finish	Click this button to initiate the project-branching process. The dashboard for the new branched project is displayed, showing the current status of the branching process. Once the branching is finished, you can navigate to the **Project Inventory** tab and the **Analysis Workbench** for the branched project to proceed with the file-audit and inventory-review processes.
Back	Click this button to move to the previous wizard page. Alternatively, you can click any of the enabled tabs for wizard pages to move directly to a page. This backward navigation allows you to make changes to the parameters you set for the project-branching process. Note that if you navigate back to the **Project Information** page, you cannot edit the **Name** field for project. Additionally, if you uploaded a codebase from the **Upload Codebase** page, the **Scan Server** and **Upload Codebase** options on the **Project Information** page are disabled.
Cancel	Click this button to cancel the project-branching setup. The branched project and any uploaded codebases for the project are deleted from Code Insight.

Branch Project: Upload Codebase

The **Upload Codebase** page in the **Branch Project** wizard identifies and uploads one or more codebase files for the branched project. For each codebase you want to upload, repeat the process of selecting the codebase archive, specifying the upload options, and then uploading the codebase.

This page is enabled only if you selected the **Upload Codebase** option from the previous **Project Information** page.

If you also selected the **Sync with Source Control** option on the **Project Information** page, the codebases obtained through synchronization with your Source Control Management system (described in Branch Project: Version Control Settings) during the branching operation will be added to the codebases uploaded from this page. For a description of the procedures related to the **Upload Codebase** page, see the following:

- Branching a Project
- Step 2: Uploading a Codebase (Optional)

The following describes the properties and actions available on the **Upload Codebase** page:

Table 8-13 ▪ Branch Project: Upload Codebase Page

Field	Description
Select Archive File	Click this button to select the archive file containing the codebase you want to upload. The selected archive is displayed in the text box next to this button. Only a `.zip`, `.tar`, `.tar`, `.gz`, `.7z`, `.iso`, or `.ova` archive is accepted. The maximum archive file size is 10 GB.
Delete existing project codebase files	Select this option to delete all codebase files that you might have already uploaded from this page and replace them with files from the codebase currently selected for upload. If you leave this option unchecked, files from the selected codebase are added to the already uploaded files.

Table 8-13 ▪ Branch Project: Upload Codebase Page (cont.)

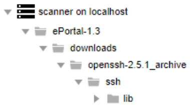
Field	Description
Archive Expansion Options	Configure the behavior of archive expansion during the upload: ● **Uploaded file only**—Extract the files from the uploaded archive. Any extracted archives are not expanded. ● **Uploaded file and first-level archives only**—Extract the files from the uploaded archive and expand all first-level archives in the codebase. Note that the expanded archive itself is retained along with its extracted contents in the parent folder. ● **Uploaded file and all contained archives**—Extract the files from the uploaded archive and expand archives at all levels (that is, archives with archives within archives and so forth) in the codebase. Note that each expanded archive is retained along with its extracted contents in the parent folder. Configure the of the upload process once archives are expanded. These settings are optional and are enabled only if the **Uploaded file and first-level archives only** or **Uploaded file and all contained archives** option has been selected. ● **Delete archive files after expansion**—Remove those archives that have been expanded during an upload. (The archive is removed from the uploaded codebase after the upload is finished.) If you leave this option unselected, the archive is retained as an additional file directly under its parent folder. ● **Append value to expanded archive directory name**—(Optional) Define a string to append to the name of any folder automatically created during the upload to store an archive's contents. After the codebase is scanned, this appended string helps you to identify those folders in the codebase tree whose contents were extracted from archives, especially if the original archives were removed from the codebase during the upload (see the previous option). For example, suppose the appended value is `_archive`, and the upload process extracts an archive called `7z.zip`. After the upload process expands the archive, the name of the folder containing the archive contents becomes `7z_archive`, as shown in this example. Note that the example also shows that the 7z.zip archive has been removed due to the selection of **Delete archives after expansion**.  scanner on localhost ePortal-1.3 downloads openssh-2.5.1_archive ssh lib This appended value has a maximum of 20 characters and does not support certain special characters. (Hover over the ⓘ icon for a list of unsupported characters.) For more information about archive expansion during a codebase upload, see More About Archive Expansion Behavior During Codebase Uploads.

Table 8-13 ▪ Branch Project: Upload Codebase Page (cont.)

Field	Description
Upload Project Codebase	Click this button to upload the selected codebase. Once the upload completes, you can upload another codebase by selecting its archive, specifying the appropriate expansion option for the upload, and clicking this button again. You can upload as many codebases as needed.
Next	Click this button to move to the next available wizard page. If you selected **Sync from Source Control** on the **Project Information** page, the **Version Control Settings** page in the wizard opens. If you did not select this option, the **Project Copy Settings** page opens.
Back	Click this button to navigate back to the **Project Information** page. Note that you cannot edit the **Name** field for project. Additionally, if you uploaded a codebase, the **Scan Server** and **Upload Codebase** options on the **Project Information** page are disabled.
Cancel	Click this button to cancel the project-branching setup. The branched project and any uploaded codebases for the project are deleted from Code Insight.

Branch Project: Version Control Settings

The **Version Control Settings** page in the **Branch Project** wizard configures one or more Source Control Management (SCM) instances, enabling the branching process to synchronize the branched project with remote codebase repositories in your site's SCM applications. The synchronization takes place once the branching process begins. For more information about how to set up for SCM synchronization and how the synchronization process works, refer to Configuring Source Code Management.

By default, any SCM instances used by the source project are automatically copied to this page, each instance defined on a separate tab. However, you can edit or remove any of these instances or add new ones for the branched project. Alternatively, you can choose not to include any SCM instances on the **Version Control Settings** page, but can always return to this page to add instances later during setup if you want.

This page is enabled only if you selected the **Sync from Source Control** option from the previous **Project Information** page.

If you also uploaded codebases from the **Upload Codebase** page, the codebase files obtained through SCM synchronization, as defined on this page, will be added to the uploaded codebase files to provide the complete codebase for the branch product.

For a description of the procedures related to the **Version Control Settings** page, see the following:

- Branching a Project
- Step 3: Configuring Synchronization with a Source Code Management Instance (Optional)

The following describes the properties and actions available on the **Version Control Settings** page:

Table 8-14 ▪ Branch Project: Version Control Settings

<table>
<tr><th>Field</th><th colspan="2">Description</th></tr>
<tr><td><strong>Add Instance</strong></td><td colspan="2">Click this button to create a new instance to connect to and synchronize with a remote SCM repository.</td></tr>
<tr><td><strong>Application</strong></td><td colspan="2">Select the type of SCM application to which the remote repository belongs: <strong>Git</strong>, <strong>Perforce</strong>, or <strong>TFC</strong> (Team Foundation Server).</td></tr>
<tr><td rowspan="4"><strong>Instance tab</strong></td><td colspan="2">If the source project contains SCM instances, these are copied to the branched project, each on a separate tab. Additionally, a new tab is created for each new SCM instance you decide to set up. The following provides information about the properties required for each instance and the actions you can perform on the instance.</td></tr>
<tr><td><strong>Instance properties</strong></td><td>Provide or edit the properties for the new or existing SCM connection instance. Click the appropriate link below for a description of the properties used to configure an instance based on the SCM application type.
● <strong>Git instance</strong>—Refer to Configuring an SCM Git Instance.
● <strong>Perforce instance</strong>—Refer to Configuring an SCM Perforce Instance.
● <strong>TFS instance</strong>—Refer Configuring an SCM TFS Instance.</td></tr>
<tr><td><strong>Test Connection</strong></td><td>To ensure that Code Insight is able to connect to the remote SCM repository specified by the instance, click this button. After a moment, Code Insight displays a success message if the connection is made. If the connection is not successful, ensure that your entries on the <strong>Instance</strong> tab are correct, and click <strong>Test Connection</strong> again.</td></tr>
<tr><td><strong>Delete Connection</strong></td><td>Click this button to permanently remove this instance from the branched project (and thus from the Code Insight system). Keep in mind that, if this instance was copied from the source project and you delete it here, the branch might no longer contain the same codebase files as the source project.</td></tr>
<tr><td><strong>Next</strong></td><td colspan="2">Click this button to perform a final “test connection” on each connection instance incrementally.
If all connections are successful, the <strong>Project Copy Settings</strong> page in the wizard is opened. If a connection to a given repository fails, an error message is displayed, specifying which connection instance has failed. You must correct all connection issues before proceeding with the project-branching setup.</td></tr>
<tr><td><strong>Back</strong></td><td colspan="2">Click this button to move to the previous wizard page.
Note that if you navigate back to the <strong>Project Information</strong> page, you cannot edit the <strong>Name</strong> field for project. Additionally, if you uploaded a codebase from the <strong>Upload Codebase</strong> page, the <strong>Scan Server</strong> and <strong>Upload Codebase</strong> options on the <strong>Project Information</strong> page are disabled.</td></tr>
</table>

Table 8-14 ▪ Branch Project: Version Control Settings (cont.)

Field	Description
Cancel	Click this button to cancel the project-branching setup. The branched project and any uploaded codebases for the project are deleted from Code Insight.

Branch Project Wizard

The **Branch Project** wizard automates the process of branching from one Code Insight project to another, enabling the target branched project to preserve any file-audit data, inventory, and inventory-review information that was created in the source project (the project from which you are branching). The wizard is accessed from the **Manage Project | Branch Project** option on the **Summary** page of the source project. For complete information about using the wizard, refer to Branching a Project.

The **Branch Project** wizard opens to the **Introduction** page. From this page, you click **Next** to begin the steps necessary to set up your project-branching process. The wizard navigates you through the following pages to complete the setup and initiate the branching process.

For a complete description of the procedures related to the **Branch Project** wizard and the project-branching process in general, see Branching a Project.

Table 8-15 ▪ Branch Project Wizard Pages

Wizard Page	Description
Branch Project: Project Information	The **Project Information** page in the **Branch Project** wizard enables you to define the properties needed to create the branched project. Once you navigate from this page, the project is created.
Branch Project: Upload Codebase	The **Upload Codebase** page enables you to upload one or more codebases to the branched project. If necessary, you can also synchronize with one or more Source Control Management instances (see Branch Project: Version Control Settings) to obtain the complete codebase for the project. Uploading a codebase is optional.
Branch Project: Version Control Settings	The **Version Control Settings** page sets up the properties needed to synchronize the branched project with one or more remote codebase repositories from your site’s Source Control Management applications. The connection to each SCM repository is defined on this page as an SCM instance, with each instance on a separate tab. The synchronization is performed once the project-branching process begins. If you uploaded codebase files from the **Upload Codebase** page (see Branch Project: Upload Codebase), the codebase files obtained through SCM synchronization, as defined on this page, will be added to the uploaded codebase files to provide the complete codebase for the branch product. Synchronizing with Source Code Management instances is optional.

Table 8-15 ▪ Branch Project Wizard Pages (cont.)

Wizard Page	Description
Branch Project: Project Copy Settings	The **Project Copy Settings** page identifies the parameters used by the branching process to import inventory as well as file-audit and inventory-review information from the source project to the branched project. If you have selected to not include codebase files in the branching process (that is, you have neither uploaded codebases nor enabled synchronization with remote codebases through SCM instances), only inventory and inventory-review information is imported to the branched project from the source project. No file-related information is imported.
Branch Project: Summary	The **Summary** page provides an overview of the parameters that you defined for the project-branching process. From this page, you can start the branching process or navigate back to other pages in the wizard to make changes before starting the branching process. You can also cancel the entire branching setup.

Code Insight Dashboard

The Code Insight dashboard is displayed when you access Code Insight. The dashboard contains the following information:

Table 8-16 ▪ Code Insight Dashboard

Column/Field	Description
analyzed	The number of lines of code that have been analyzed since Code Insight was installed.
scanned	The number of scanned files that have been scanned since Code Insight was installed. This total reflects all files scanned during successful project scans on the Scan Server. It does not reflect files scanned by remote scan plugins. To examine file totals for all scans (successful and failed) per the Scan Server *and* remote scan plugins for a given project, view the project dashboard or the Analysis Workbench (see Using the Project Dashboard or The Analysis Workbench Layout, respectively).
identified	The number of open-source or third-party components that were identified in your codebase.

Table 8-16 ▪ Code Insight Dashboard (cont.)

Column/Field	Description
View Inventory	Select this option to open the **Inventory** view, which provides a compilation of inventory across all current Code Insight projects. The list can be filtered at a basic level to show inventory for all projects, only your projects, or for a specific project. Filtering can be further refined by vulnerability severity, review status, and many other criteria. For more information, see Viewing Inventory Across All Projects.
Go to Project	Select this option to open the **Projects** view, which provides access to all current projects in Code Insight. For more information, see Accessing Projects in Code Insight.
View Policy	Select this option to open the **Policy** page, where you have access to all policies that automate the review process of project inventory when it is published. For more information, see Managing Policies to Automatically Review Inventory. Access to this page requires **Manage Policy** permissions.
Administration	Select this option to open the **Administration** page, where you have access to Code Insight administrative functionality. See the *Code Insight Installation and Configuration Guide* for a description of administrative tasks. Access to this page requires Code Insight System Administrator permissions.

***Note ▪** If this is the first time Code Insight has been accessed or if no codebase has been analyzed, the **analyzed**, **scanned**, and **identified** fields will be empty.*

See Also
Projects Pane and Associated Dashboard
Policy Page
Users/Permissions Tab
Electronic Updates Tab
Email Server Tab
LDAP Tab
ALM Tab
Scan Servers Tab
Scan Profiles Tab

Component Details Window

The following window is displayed when you click ⓘ next to an OSS or third-party component listed in the following locations:

- The **Component Name** column on the **Component** tab on the **Global Component & Lookup** page.
- The **Component** field on the **Inventory Details** tab in the **Analysis Workbench**.

The window shows publicly available information about the component, including the following properties.

Table 8-17 ▪ Component Details Window

Column/Field	Description
Component	The name of the OSS or third-party component and its internal ID, as identified in the Code Insight Data Library. If you are on the **Inventory Details** tab in the **Analysis Workbench**, you can associate the current inventory item with a different component (see Editing Inventory from the Analysis Workbench).
Version	The component version and its internal ID, as identified in the Code Insight Data Library. If you are on the **Inventory Details** tab in the **Analysis Workbench**, you can associate the inventory item with a different version of the component (see Editing Inventory from the Analysis Workbench).
Forge	The external repository associated with the component. You can click the forge link to open the forge website.
Possible Licenses	License candidates associated with this component. Click the ⓘ icon next to a given license to view information about the license on the License Details Window.
Custom Component	The **Yes** or **No** value indicating whether the component is custom (created by a user) or provided as part of the Code Insight Data Library
Vulnerabilities	A bar graph showing the count of known vulnerabilities by severity color for the component. Click the graph to view the list of vulnerabilities and their details. For details about the graph and vulnerabilities in general, see Security Vulnerabilities Associated with Inventory. If no vulnerabilities have been found for the inventory item, the value **No** is displayed in place of the graph.
Encryption	The **Yes**, **No**, or **N/A** value indicating whether the component provides the encryption capabilities used in your product or whether these capabilities are not applicable. Encryption can affect export controls.

Table 8-17 ▪ Component Details Window

Column/Field	Description
CPE	The list of CPE names—from the National Vulnerability Database—that are mapped to the component. CPE (Common Platform Enumeration) is a structured naming scheme that includes the component's vendor and product names in the following format: **cpe://<part>:<vendor>:<product**> where <part> is either **a** (applications), **h** (hardware platforms), or **o** (operating systems). ***Note ▪*** *The data provided represents only the part, vendor, and product; the version information is truncated from the CPE string.*

Create (or Edit or View) Scan Profile Dialog

The following table describes the fields that define a standard or custom scan profile on the **Create Scan Profile**, **Edit Scan Profile**, and **View Scan Profile** dialogs. Code Insight System Administrators access these dialogs from the **Scan Profiles** tab on the **Administration** page.

The **Create Scan Profile** enables you to add a custom profile, the **Edit Scan Profile** lets you update a selected standard or custom profile, and the **View Scan Profile** shows a read-only view of the current settings for a selected scan profile.

For more information about using these dialogs, see “Managing Scan Profiles” in the “Configuring Code Insight” chapter in the *Code Insight Installation & Configuration Guide*.

About Standard Scan Profiles

The following are the standard (pre-defined) scan profiles that ship with Code Insight. You can modify these profiles (with the exception of the Standard Scan Profile), assign them to projects, or use them as templates for creating your own scan profiles.

- **Basic Scan Profile (without CL)**—Defines a scan that uses Automated Analysis to detect evidence of open-source software (OSS) and third-party code in your codebase and generate an inventory of the findings. This scan does not perform exact-file or source-code matching and therefore does not use the Compliance Library (CL).
- **Standard Scan Profile**—Defines a scan that includes the basic scan features but also performs exact-file matching (that is, identifies codebase files that have an exact MD5 match in the CL). This scan requires the CL. This is the scan profile used as a template when you create a new profile. It cannot be modified.
- **Comprehensive Scan Profile**—Defines a scan that includes the basic scan features but also performs exact-file and source-code matching. (Source-code matches are strings in the codebase files that have an exact match to content in files in the CL). This scan requires the CL.

The table below shows the default value for a given setting in each of the standard scan profiles.

Scan Profile Settings

The following table defines that scan settings used to define a scan profile, custom or standard.

For your reference, the table contains additional columns (Basic, Standard, Comprehensive) to indicate the default value for a given setting in each of the standard scan profiles.

Table 8-18 ▪ Scan Profile Dialog

Field	Description	Basic	Standard	Compre-hensive
Name	Enter or edit the profile name.	**Basic Scan Profile**	**Standard Scan Profile**	**Compre-hensive Scan Profil**e
Perform Package/ License Discovery in Archives	Select this option to have the Scan Server recursively perform package discovery and license detection within all archive files encountered in the project codebase. By default, this option is selected.	Selected	Selected	Selected
Dependency Support	Determine the level of dependency scanning to be performed by the Scan Server. The available options include: ● **No Dependencies**—Only top-level inventory items are reported without any dependencies. (Default) ● **Only First Level Dependencies**—Only first-level (or direct) dependencies are reported along with top-level inventory items. ● **All Transitive Dependencies**—All first-level and transitive dependencies are reported along with top-level inventory items. The Scan Server calls out to the relevant package management repository to obtain transitive dependency information. For a description of Code Insight dependency support for supported ecosystems, see "Automated Analysis" in the *Code Insight User Guide*.	**No Depend-encies**	**No Depend-encies**	**No Depend-encies**

Table 8-18 ▪ Scan Profile Dialog (cont.)

Field	Description	Basic	Standard	Compre-hensive
Report Non-Runtime Dependencies	(Available if **Only First Level Dependencies** or **All Transitive Dependencies** is selected for **Dependency Support**) Specify whether the scan should report only runtime dependencies or both runtime and non-runtime dependencies. (Runtime dependencies are required during application runtime; non-runtime dependencies are not.) For more information, see Dependency Scopes. ● **Enabled**—Report both runtime and non-runtime dependencies. ● **Disabled**—Report only runtime dependencies.(Default)	N/A	N/A	N/A
Automatically Add Related Files to Inventory	Select this option to have the system associate additional files to existing inventory items based on the data available in automatic detection rules.	Selected	Selected	Selected
Rescan Options	By default, when a user initiates a regular rescan (that is, not a forced full rescan), only those files that have changed since the last scan are scanned. However, certain Code Insight events that have occurred since the previous scan can result in a rescan of all files (a full rescan). For a description of these events, see “Default Scan Behavior” in the *Code Insight User Guide*. These options are used to override this default rescan behavior so that, even if any of the events that would normally call for a full rescan have occurred, all rescans will skip unchanged files and scan changed files only.			
Do not rescan files that have not changed since previous scan	Select this option so that rescans always skip unchanged files and scan only those files that have changed since the last scan (even if events have occurred since the last scan that call for a full rescan).	Not selected	Not selected	Not selected
Apply this option to:	If the **Do not rescan files...** option is selected, further clarify *which* unchanged files to skip during the rescan: ● All unchanged files ● Only unchanged files marked as reviewed ● Only unchanged files associated with inventory ● Only unchanged files that are both marked as reviewed and associated with inventory	N/A	N/A	N/A

Table 8-18 ▪ Scan Profile Dialog (cont.)

Field	Description	Basic	Standard	Comprehensive
Exact Matches	Select this option to enable the detection and recording of scanned files that exactly match entire-file data in the Compliance Library (CL).	**Disabled**	**Enabled**	**Enabled**
Source Code Matches	Select this option to enable the detection and recording of any source-code snippets in the scanned files that match data in the Compliance Library (CL). If you enable this source-code matching, specify any of the following additional parameters for the matching process.	**Disabled**	**Disabled**	**Enabled**
	Include System-Identified Files Select this option if you want the Scan Server to perform source-code matching for files that have already been associated with one or more inventory items during automated analysis.	N/A	N/A	Selected
	Include Files with Exact Matches Select this option if you want the Scan Server to perform source-code matching for files that have already been identified as having exact-file matches in the CL.	N/A	N/A	Selected
	Minimum Source Code Matches Enter the minimum number of source-code matches that the scan needs to detect in a given codebase file before reporting the file as having such matches. (A *source-code match* is a snippet of code in a codebase file that matches an open-source code snippet found in the CL data.) Enter a new minimum value from 1 to 32767. (The default is 3.) For example, if this value is increased to `10`, ten code snippets in a given codebase file must match data in the CL before the scan reports the file as having source-code matches. In general, the higher this value, the fewer source-code matches an analyzer has to review.	N/A	N/A	**1**
Search Terms	Provide a list of search terms to be used in the scan. Use the **+** button to add a term and the **-** button to remove a term.	Standard terms listed	Standard terms listed	Standard terms listed

Table 8-18 ▪ Scan Profile Dialog (cont.)

Field	Description	Basic	Standard	Compre-hensive
Scan Exclusions	Provide a list of file extensions to be excluded from the scan. Use the **+** button to add an exclusion term and the **-** button to remove an exclusion. See “Creating Exclusion Patterns for Scan Profiles” in the *Code Insight Installation & Configuration Guide* for further instructions.	Standard exclusions listed	Standard exclusions listed	Standard exclusions listed

See Also
Scan Profiles Tab
About Code Insight Scans
Applying a Scan Profile to the Project
Edit Project: Scan Settings Tab

Create Custom Detection Rule Dialog

The **Create Custom Detection Rule** dialog enables you to create a custom detection rule. You can define these rules as needed to supplement the internal detection rules used by Automated Analysis to automatically create inventory during a scan. The custom detection rules are saved to the Code Insight Data Library for global use across projects. For more information about custom detection rules, see Managing Custom Detection Rules.

This dialog is accessed from two locations:

- From the **Inventory Details** tab in the **Analysis Workbench** for an inventory item of the “component” type—whether system-generated or manually created—to which codebase files have been manually associated (as described in Creating a Custom Detection Rule Within Context of an Inventory Item).
- From **Custom Detection Rules** tab accessed from the **Data Library** page on the Code Insight main menu (as described in Creating a Custom Detection Rule from Scratch).

The ability to edit certain fields depends on how you accessed the dialog. To help explain these differences, the following table designates the two access locations as “**Inventory Details** tab” and “**Custom Detection Rules** tab”.

The following describes the columns and actions you can perform from the **Custom Detection Rule** dialog. Unless specified as “Required” in this table, the fields are optional

Table 8-19 ▪ Create Custom Detection Rule Dialog

Category	Column/Field	Description
Inventory Name	Use this field if you want to enter a custom name for the inventory item created by this rule. This name overwrites the default *component version (license)* name that the rule normally assigns to the inventory item, based on the **Component** and **License** attributes specified. Note that, depending on how you access the **Create Custom Detection Rule** dialog, this field either initially is empty or explicitly contains the default inventory name. (If this field is empty or contains all blank spaces, the default name is assigned to an inventory item.) Either way, you have the option to leave the field as is or edit it to provide a custom inventory name. The maximum size for this value is 255 characters. ***Note ▪** When two rules are defined with these same **Component** and **License** attributes but with different inventory names, only the most recently created rule is applied during scans.*	

Table 8-19 ▪ Create Custom Detection Rule Dialog (cont.)

Category	Column/Field	Description
Component selection	The following fields describe the component on which the custom detection rule is based. ● If you have accessed this dialog from the **Inventory Details** tab for an inventory item in the **Analysis Workbench**, these fields are auto-populated with component information from the inventory item. The **Component** and **License** fields are not editable. ● If you have accessed this dialog from the **Custom Detection Rules** tab on the **Data Library** page, these fields are populated once you select the component. The fields *are* editable as described below.	
	Component	(Required) The name of the component on which this detection rule is based. ● If you accessed this dialog from the **Custom Detection Rules** tab, click **Lookup Component** to select the component and its version, license, and forge URL. The **License** and **URL** fields are populated accordingly. You cannot edit this field directly, but you can always select a different component. ● If you accessed this dialog from the **Inventory Details** tab, this field is not editable.
	License	(Required) The license associated with the component. ● If you accessed this dialog from the **Custom Detection Rules** tab, you cannot edit the field directly once it is populated from the component selection, but you can select a different license. To do so, click ✎ to switch to another license and, optionally, change the component version. Additionally, click ⓘ to view the details and text of the selected license as stored in the Code Insight Data Library. ● If you accessed this dialog from the **Inventory Details** tab, this field is not editable.
	Description	Enter a description of the component (or update the pre-populated description).
	URL	(Required) Enter the forge URL for the component (or update the pre-populated URL).

Table 8-19 ▪ Create Custom Detection Rule Dialog (cont.)

Category	Column/Field	Description
License, notices, and note content	The following fields are used to provide license or notice content and audit notes to be included in the inventory items created by this rule. These field are editable. If you accessed this dialog from the **Inventory Details** tab in the **Analysis Workbench**, these fields might be pre-populated with information from the manually created inventory. However, you can edit this information as needed.	
	As-Found License Text	Enter (or update) the license content that was discovered for the component during scans. This information is considered for use in the Notices report. If no **Notices Text** content is provided (see next field), the Notices report uses the content in this field as the license text for the third-party component. For more information, see Finalizing the Notices Text for the Notices Report.
	Notices Text	Enter (or update) the exact license content to include in the Notices report. This content is usually a modification of the text in **As-Found License Text**. (You can copy the **As-Found License Text** content to the **Notices Text** field and edit it.) If content exists in this field, the Notices report uses it as the license text for the third-party component and ignores any information in the **As-Found License Text** field. For more information, see Finalizing the Notices Text for the Notices Report.
	Audit Notes	Enter (or update) any notes or findings per analysis of the inventory item that might be helpful to the inventory reviewers.
File MD5 list	(This single field for specifying file criteria is available only if you have accessed the current dialog from the **Inventory Details** tab in the **Analysis Workbench** to create a rule within the context of an inventory item.) The **File MD5** list box is pre-populated with a set of file criteria used for detecting the third-party or OSS component for which the rule is being created. The criteria is based on the MD5 value of each file associated with the inventory item in which context you are creating the rule. To identify which of the displayed criteria the rule should apply, click the checkbox next to each desired criterion. (Be sure to clear the checkbox next to any criterion that you want to exclude from the rule.) At least one criterion must be selected. Consider that, if the custom detection rule is defined with multiple file criteria, the scan uses OR logic when processing the criteria against the target codebase. Consequently, only one file match between codebase and the rule is required to automatically create an inventory item. For a comprehensive list of rule-processing behavior, see Rule-Processing Considerations.	

Table 8-19 ▪ Create Custom Detection Rule Dialog (cont.)

Category	Column/Field	Description
Detection Criteria	(This field and its related **File MD5** or **File Path** field for specifying file criteria are available only if you have accessed the current dialog from the **Custom Detection Rules** tab on the **Data Library** page to create the rule from scratch.) Select the type of file criteria that you are specifying to detect the presence of the third-party or OSS component: ● **File MD5**—The file in each criterion is identified by its MD5 value. (Default) ● **File Path**—The file in each criterion is identified by its file path. The set of file criteria in the rule must be of the same criteria type. If you attempt to set up detection criteria for both types, keep in mind that you lose the criteria for the type that is currently not selected for **Detection Criteria** when you save the rule. A custom detection rule allows only a single set of criteria (**File MD5** or **File Path**) to exist at any one time. Also consider that, if the custom detection rule is defined with multiple file criteria, the scan uses OR logic when processing the criteria against the target codebase. Consequently, only one file match between codebase and the rule is required to automatically create an inventory item. For a comprehensive list of rule-processing behavior, see Rule-Processing Considerations. At least one criterion for the rule's specified criteria type is required.	
	File MD5 grid	(Available if **Detection Criteria** is **File MD5**) Add and manage the file criterion consisting of the file name and the MD5 value for each file used as an indicator of the existence of the component. At least one file criterion is required. ● **To add a file criterion**—Click **Add File** and enter the file's name and MD5 value in the **Name** and **MD5** fields, respectively, in the new row. ● **To edit a file criterion**—Click within the **Name** or **MD5** field in the row for the criterion and make the textual changes. ● **To remove a file criterion**—Click ✖ at the end of the row for the criterion.

Table 8-19 ▪ Create Custom Detection Rule Dialog (cont.)

Category	Column/Field	Description
	File Path text box	(Available if **Detection Criteria** is **File Path**) Add and manage the file criterion consisting of the file path for each file used as an indicator of the existence of the component. At least one file criterion is required. ● **To add a file path**—Click the Add icon + and enter the file's path in the new row. You can provide the file's absolute or relative path or enter a path pattern. A path *pattern* consists of the asterisk symbol * within the path, denoting any number of directories or files. For example, the following path pattern indicates that any file with the extension `.h` under the directory `root` will be considered detection criteria for the rule. `**/root/*.h` ● **To edit a file path**—Click within the path row and make the textual changes. ● **To remove a file path**—Click within the path row, and then click the Remove icon -.
Actions	The following are actions conclude the rule-creation session.	
	Save	Click **Save** to save the new custom detection rule to the Code Insight Data Library. You will be asked for confirmation to proceed with the creation.
	Cancel	Click **Cancel** to cancel the rule creation process. You will be asked for confirmation to proceed with the cancellation.

See Also
Managing Custom Detection Rules
Creating a Custom Detection Rule Within Context of an Inventory Item
Creating a Custom Detection Rule from Scratch
Finalizing the Notices Text for the Notices Report

Custom Detection Rules Tab

The **Custom Detection Rules** tab on the **Data Library** page lists the custom detection rules currently available for use in codebase scans in your Code Insight system. You can define custom rules as needed to supplement the internal detection rules used by Automated Analysis to automatically create inventory during a scan. These custom rules are saved to the Code Insight Data Library for global use across projects.

From this page, you can also select to edit a rule or remove a rule from your Code Insight system.

For more information about custom detection rules, see Managing Custom Detection Rules.

The following describes the columns and actions you can perform from the **Custom Detection Rules** tab.

Table 8-20 ▪ Custom Detection Rules Tab

Category	Column/Field	Description
Actions	**Enter Component Name search string**	To locate specific custom detection rules, enter a component-name string by which to filter the list of detection rules. The search results show only those rules whose component name contains the search string you provided. (This filter applies to only those custom rules visible in the UI; no call is made to the Data Library.)
	Create Custom Rule	Click to open the **Custom Detection Rule** dialog to create a new custom detection rule. See Create Custom Detection Rule Dialog for details.
Custom rule entry	The following columns provide details about each custom detection rule and give you access to actions you can take on the rule.	
	Component	The name of the component on which the custom detection rule is based.
	Version	The component version.
	License	The license found in the Code Insight Data Library and associated with the component.
	URL	The forge URL for the component.
	Actions for custom rule entry	Actions you can perform on the currently selected rule: • Click ✎ to edit the custom detection rule. The **Edit Custom Rule** dialog is opened. See Edit Custom Rule Dialog for details. • Click ✖ to delete the custom detection rule from your Code Insight system. The rule will no longer be applied during project scans.

See Also
Managing Custom Detection Rules
Create Custom Detection Rule Dialog
Edit Custom Rule Dialog

Edit (Default) Project Users Page

The **Edit Project Users** page, accessed from the **Manage Project** menu on the **Summary** tab for a specific project, is used to assign Code Insight users to various roles for the project.

The **Edit Default Project Users** page, available from the **Project Defaults** tab on the **Administration** page, is used to set project role assignments that default for any new project created (but which can then be edited at the project level on the **Edit Project Users** page).

For a description of the project roles and more information about the procedures used to manage them on the **Edit Project Users** page, see Assigning and Removing Project Users. (These same procedures basically apply to the **Edit Default Project Users** page.) Additionally, for a description of the permissions enabled for each project role, see the appendix Code Insight User Roles and Permissions.

The following describes the fields on the **Edit Project Users** and **Edit Default Project Users** page:

Table 8-21 ▪ Edit (Default) Project Users Page

Column/Field	Description
Select Users	The list of all users defined for your Code Insight system. From this list, you select the users to which you want to assign project roles.
Add User	Select one or more users in the **Select Users** pane, and then select the appropriate option—**Add to Analysts**, **Add to Reviewers**, or **Add to Observers**— from the **Add User** dropdown list to add the users to the desired “role” pane. This procedure is an alternative to dragging and dropping users to the appropriate “role” pane.
Enter Search Criteria	Enter a full or partial user name to search for a user in the system and click 🔍.
Project Administrators	The pane listing users who are currently assigned to the Project Administrator role for any project when it is created. To assign additional users to this role, drag and drop one or more users from the **Select Users** list to this pane. (Alternatively, select **Add to Project Administrators** from the **Add User** dropdown list to add the selected users to the pane.)
Analysts	The pane listing users who are currently assigned to the Analyst role for any project when it is created. To assign additional users to this role, drag and drop one or more users from the **Select Users** list to this pane. (Alternatively, select **Add to Analysts** from the **Add User** dropdown list to add the selected users to the pane.) To remove a user from this role, click ✖ next to the user name in the pane.
Reviewers	The pane listing users who are currently assigned to the Reviewer role for any project when it is created. To assign additional users to this role, drag and drop one or more users from the **Select Users** list to this pane. (Alternatively, select **Add to Reviewers** from the **Add User** dropdown list to add the selected users to the pane.) To remove a user from this role, click ✖ next to the user name in the pane.

Table 8-21 ▪ Edit (Default) Project Users Page (cont.)

Column/Field	Description
Observers	The pane listing users who are currently assigned to the Observer role. To assign additional users to this role, drag and drop one or more users from the **Select Users** list to this pane. (Alternatively, select **Add to Observers** from the **Add User** dropdown list to add the selected users to the pane.) To remove a user from this role, click ✖ next to the user name in the pane. ***Note ▪** On the **Edit Project Users** page, the **Observers** pane is visible only for private projects. On the **Edit Default Project Users** page, this pane is always visible, enabling you to assign observers that will default for any private project that might be created. For more information, see Creating a Private Project.*
Close	Click this button to save your changes.

See Also
Assigning and Removing Project Users

Edit Custom Rule Dialog

The **Edit Custom Rule** dialog enables you to edit an existing custom detection rule.

Custom detection rules as defined needed to supplement the internal detection rules used by Automated Analysis to automatically create inventory during a scan. These custom rules are saved to the Code Insight library for global use across projects. For more information about custom detection rules, see Managing Custom Detection Rules.

The following table describes the fields, buttons, and icons on the **Edit Custom Rule** dialog. You can edit any of the fields, using the methods described in the table. Unless specified as “Required” in this table, the fields are optional.

Table 8-22 ▪ Edit Custom Rule Dialog

Category	Column/Field	Description
Inventory Name	The name for the inventory items created by this rule. By default, the name assigned to the inventory items is *component version (license)*, based on the **Component** and **License** attributes specified. In some cases, the default name is automatically displayed in this field. When this field is empty (or contains all blank spaces), the default name is also assigned to the inventory items. Alternatively, enter a custom name in this field that overwrites the default name for the inventory items created by the rule. ***Note ▪** When two rules are defined with these same **Component** and **License** attributes but with different inventory names, only the most recently created rule is applied during scans.*	
Component selection	The following fields describe the component on which the custom detection rule is based. The fields are editable as described below.	
	Component	(Required) The name of the component on which this detection rule is based. You cannot edit this value directly, but you can switch to another component. To do so, click **Lookup Component** to select another component, along with its version, license, and forge URL.
	License	(Required) The license associated with the component. You cannot edit this value directly, but you can select a different license. Click ✏ to switch to another license and, optionally, change the component version. Additionally, you can click ⓘ to view the details and text of the selected license as stored in the Code Insight Data Library.
	Description	A description of the component.
	URL	(Required) The forge URL for the component.

Table 8-22 ▪ Edit Custom Rule Dialog (cont.)

Category	Column/Field	Description
License, notices, and note content	The following fields are used to provide license or notice content and any audit notes for the inventory item generated from this rule. These fields are editable.	
	As-Found License Text	The license content that was discovered for the inventory item during scans. This information is considered for use in the Notices report. If no **Notices Text** content is provided (see the next field), the Notices report uses the content in this field as the license text for the third-party component. For more information, see Finalizing the Notices Text for the Notices Report.
	Notices Text	The exact content to include in the Notices report. This is usually a modification of the text in **As-Found License Text** pane. (You can copy the **As-Found License Text** content to the **Notices Text** field and edit it.) If content exists in this field, the Notices report uses it as the license text for the third-party component and ignores any information in the **As-Found License Text** pane. For more information, see Finalizing the Notices Text for the Notices Report.
	Audit Notes	Any notes or findings per the analysis of the inventory item that might be helpful to the inventory reviewers.

Table 8-22 ▪ Edit Custom Rule Dialog (cont.)

Category	Column/Field	Description
File criteria	The following fields identify the set of file criteria used by the rule to detect the third-party or OSS component and create the associated inventory. The set of criteria is based on either the file path or MD5 value of the files.	
	Detection Criteria	(Required) The type of file criteria used to detect the presence of the third-party or OSS component: ● **File MD5**—The file for each criterion is identified by its MD5 value. ● **File Path**—The file for each criterion is identified by its file path. If you want to switch the current **Detection Criteria** type from **File MD5** to **File Path** or vice versa, know that once you enter the new set of criteria and save the rule, the criteria for the type currently not selected for **Detection Criteria** is automatically deleted. A custom detection rule allows only a single set of criteria to exist at any one time. Consider that, if the custom detection rule is defined with multiple file criteria, the scan uses OR logic when processing the criteria against the target codebase. Consequently, only one file match between codebase and the rule is required to automatically create an inventory item. For a comprehensive list of rule-processing behavior, see Rule-Processing Considerations. At least one criterion for the rule's specified criteria type is required.
	File MD5 grid	(Available if **Detection Criteria** is **File MD5**) The file criteria consisting of the file name and the MD5 value for each file used as an indicator of the existence of the component. At least one file criterion is required. To manage the file criterion in the grid: ● **Add a file criterion**— Click **Add File** and enter the file's name and MD5 value in the **Name** and **MD5** fields, respectively, in the new row. ● **Edit a file criterion**—Click within the **Name** or **MD5** field for the criterion and make the textual changes. ● **Remove a file criterion**—Click ✕ at the end of the criterion's row.

Table 8-22 ▪ Edit Custom Rule Dialog (cont.)

Category	Column/Field	Description
	File Path text box	(Available if **Detection Criteria** is **File Path**) The file criteria consisting of the file path for each file used as an indicator of the existence of the component. At least one file criterion is required. To manage the file criterion in the text box: ● **Add a file path**— Click the Add icon + and enter the file's path in the new row. You can provide the file's absolute or relative path or enter a path pattern. A *path pattern* consists of the asterisk symbol * within the path, denoting any number of directories or files. For example, the following path pattern indicates that any file with the extension `.h` under the directory `root` will be considered detection criteria for the rule. `**/root/*.h` ● **Edit a file path**— Click within the path row and make the textual changes. ● **Remove a file path**—Click within the path row, and then click the Remove icon - .
Actions	The following are actions conclude your update session.	
	Save	Click **Save** to save the rule updates to the Code Insight Data Library. You will be asked for confirmation to proceed with the creation.
	Cancel	Click **Cancel** to cancel your updates. You will be asked for confirmation to proceed with the cancellation.

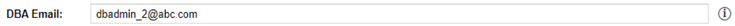
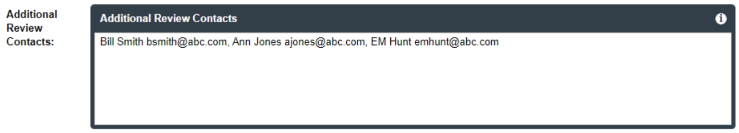
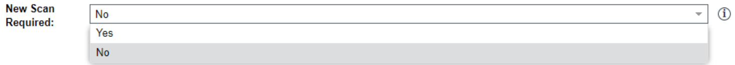
See Also
Managing Custom Detection Rules
Finalizing the Notices Text for the Notices Report

Edit Project: Custom Fields Tab

The **Custom Fields** tab on the **Edit Project** window lists the fields that were defined specifically for your site to provide information about projects in addition to the standard Code Insight fields. For any or all of the fields on this tab, enter the requested information as it pertains to the current project.

***Note ▪** If no custom fields for projects have been configured for your site, the following message is displayed on the tab: "There are no custom fields configured." However, if custom fields have been defined for your site but are currently not available for display, this tab is blank (that is, shows no message or fields).*

Table 8-23 ▪ Edit Project: Custom Fields Tab

Column/Field	Description
Field(s)	Complete any or all of the custom fields with information describing the current project. Consider the following: ● If an information ⓘ icon is available in the top right corner of a given field, click it to obtain more information about the field. ● For a regular text entry field, you can enter a value up to 128 characters.  DBA Email: dbadmin_2@abc.com ● For the larger text-area field, you can enter a value up to 512 characters.  Additional Review Contacts: Additional Review Contacts Bill Smith bsmith@abc.com, Ann Jones ajones@abc.com, EM Hunt emhunt@abc.com ● For a dropdown list, select one option only. You might need to scroll down to see the complete list of options.  New Scan Required: No Yes No ● For information about completing the **SBOM Bucket Name** field (if it is available), see Assigning the Project to an SBOM Insights Bucket. The field values you provide here are also viewable on the project's **Summary** tab.
Save	Click this button to save the updates to the field values.

See Also

Completing Custom Fields for the Project

Edit Project: General Tab

The **General** tab on the **Edit Project** window displays information about the selected project that you can edit. The tab contains the following fields:

Table 8-24 ▪ Edit Project: General Tab

Column/Field	Description
Project Name	The name of the selected project. You can change the name by typing over the current project name.
Description	A freeform text field in which you can enter a description for the project. This field provides enough space to add as much detail about the project as necessary.
Project Visibility	The visibility status—**Public** or **Private**—of the project. ● **Public**—A project that provides read-only access to any user in the system. To what degree a user can interact with the project depends on whether the user has a project role and what the role is—Project Administrator, Analyst, or Reviewer. (Default) ● **Private**—A project that is hidden from all users except the Project Contact and those users assigned as Project Administrators, Analysts, Reviewers, or Observers of the project. Additionally, project and vulnerability ID searches will not return private projects unless the user performing the search has the permissions to see a given private project.
Project Risk	The current vulnerability risk value (**Low**, **Medium**, or **High**) for the project. To edit, select another value from the dropdown list. (Default is **Medium**)

Table 8-24 ▪ Edit Project: General Tab (cont.)

Column/Field	Description
Project Status	The current status of the project. The following statuses are available and their suggested definitions are provided here. However, you can apply these statuses as appropriate for your site: ● **Not Started**—Indicates that the project scan results are not available for manual analysis. This value automatically defaults when the initial project scan has not been run or when the initial scan fails. However, you can manually change this status. (Default) ● **Analysis in Progress**—Indicates that the project scan results are available for manual analysis, or a manual analysis is underway. This value is automatically set when the initial project scan is complete. (For a project import, however, the status of the target project is retained.) You can manually change this status. ● **Analysis Completed**—Indicates that manual analysis is finished; and the published inventory is ready for legal, security, or software-engineering review. (You must manually set this value.) ● **Project Complete**—Indicates that the project analysis and review processes are complete, and notices content for all OSS and third-party software is finalized. (You must manually set this value.)
On the data import or rescan, delete inventory with no associated files	This option determines whether “empty” system-generated inventory items are deleted in the target project during project imports and rescans. Empty inventory items have no associated files. ● When selected, this option deletes empty inventory items from the target project during project imports and rescans. Only inventory items with associated files are retained/created. ● When *not* selected, this option retains/creates all inventory items—with or without matching associated files in the target codebase—in the target project during imports and rescans. For example, you might want to retain inventory items to save their analysis details. (You will need to manually delete inventory that is not applicable to the current project.) Additionally, do not select this option when importing a scanned codebase into a project for which no codebase has been uploaded or obtained through synchronization. The unselected option ensures that inventory is generated in the target project. By default, this option is not selected.

Table 8-24 ▪ Edit Project: General Tab (cont.)

Column/Field	Description
Expand Source and Uber jar files	This option determines whether uber and sources jars are expanded during a codebase upload to the project. ● When selected, this option enables the expansion of the uploaded top-level uber or sources jar and any uber or sources jars contained in the uploaded jar, according to the expansion level defined for the upload. ● When not selected, this option does expand any uber or sources files in the uploaded codebase. For more information, see Expansion of a Sources or Uber Jar.
Project Folder	The folder in the list of projects under which the project is currently grouped. To edit the project location in the list, select one of the following: ● **Clear Project Folder button**—Click this button to remove the project from the current folder in the project list and place it in the root folder. ● **Select a New Folder dropdown**—Click the down arrow to locate and select an available folder to which to move the project.

See Also
Editing the Project Definition and General Settings

Edit Project: Project Hierarchy Tab

The **Project Hierarchy** tab on the **Edit Project** window for a given Code Insight project enables you to manage project's hierarchy. A project hierarchy provides a means to keep track of projects related each other. It is created by simply identifying one or more projects as *child projects* of the current project on which the **Project Hierarchy** tab is opened (called the *parent project*). Once the hierarchy is created, links are established in Code Insight between the parent project and the associated child projects so that you can easily move between projects to assess scan results and review inventory.

A project hierarchy is useful when your product application contains one or more modules, each with a codebase for which you want to set up a separate Code Insight project to track and assess the open-source or third-party software. By setting up a project hierarchy, you can easily switch between the main project for your application (the parent project) and the projects for the modules (the child projects) to complete the work needed to build a composite Bill of Materials.

Note that a child project, in turn, can be identified as a parent project to other child projects. Likewise, a given parent project can be identified as a child project to another parent project. Since hierarchies are created as needed, projects might have no association with a hierarchy.

For complete information about creating and managing project hierarchies, see Identifying Child Projects for a Project.

Once a project hierarchy is established for a given project, you can do the following:

- From the **Summary** page for the project, view and link to any of its child and parent projects (see Summary Tab).

- From the **Inventory** view showing inventory across all projects, examine the inventory of its child projects as well as link to any these projects (see Inventory View).

The following table describes the fields and button available on the **Project Hierarchy** tab.

Table 8-25 ▪ Project Hierarchy Tab

<table>
<tr><th>Category</th><th>Column/Field</th><th>Description</th></tr>
<tr><td rowspan="4"><b>Child project entry</b></td><td colspan="2">The following columns show the properties of each child project in the hierarchy for the project currently open and describe the actions available for the child project.</td></tr>
<tr><td><b>Project Name</b></td><td>The name of the project identified as child project of the project currently open (parent project).</td></tr>
<tr><td><b>Project Contact</b></td><td>The main contact of the child project (initially the project creator).</td></tr>
<tr><td><b>Action</b></td><td>Click <b>X</b> to disassociate the child project from the current parent project. Once you confirm to disassociate the project, child project is removed from the hierarchy. The links associated with this parent-child relationship are also removed from the <b>Summary</b> pages for the parent project and the project that was disassociated. The links are also removed from the <b>Inventory</b> view.</td></tr>
<tr><td><b>Add Child Project</b></td><td colspan="2">Click this button to add a new child project to the current project. The <b>Add Child Project</b> dialog is opened, enabling you to select the new child project. (If necessary, navigate the project list more quickly by using the page navigation tools at the bottom of the dialog; or search the project list by entering a project name string in the search box.)
After you select a project, click <b>Add Project</b> to return to the <b>Project Hierarchy</b> tab, which now lists the new child project.
<i><b>Note ▪</b> To avoid cyclical parent-child relationships, the <b>Add Child Project</b> dialog does not list projects that are parents, parents of parents, children, or children of children of the current project.</i></td></tr>
<tr><td rowspan="3"><b>Actions for overall project settings</b></td><td colspan="2">These buttons control whether changes to project settings are saved across all <b>Edit Project</b> tabs.</td></tr>
<tr><td><b>Save</b></td><td>Click this button to save all your project edits and return to the <b>Summary</b> tab.</td></tr>
<tr><td><b>Cancel</b></td><td>Click this button to return to the <b>Summary</b> tab without saving your project edits on other tabs.</td></tr>
</table>

Edit Project: Review and Remediation Settings Tab

The **Review and Remediation Settings** tab on the **Edit Project** window enables you to overwrite default settings that configure the automation of the review, remediation, and status notification processes for published inventory in your project. These settings, which work in conjunction with the set of policies in the project's policy profile, are used to set up the following in your project:

- The policy profile to associate with the project. The policies in the selected profile work in conjunction with the review, remediation, and notification configuration defined on this tab.
- Automatic creation of manual review tasks for inventory items not reviewed by policy during publication performed as part of a scan.
- Automatic assignment of review tasks to the default legal or security contact that you specify.
- Automatic creation of remediation tasks and associated external work items for inventory that is rejected either automatically by policy or during manual publication by an analyst.
- Automatic assignment of remediation tasks to the default engineering contact that you specify.
- Automatic rejection of published inventory impacted by new vulnerabilities detected in the latest scan, Electronic Update, or Library Refresh.
- The automatic generation of email notifications that alert the Project Contact of rejected or non-reviewed published inventory items that need attention.

See the following field descriptions for more information.

Table 8-26 ▪ Edit Project: Review and Remediation Settings Tab

Category	Section/Field	Description
Automated Review Options	**Policy Profile**	Select policy profile you want to associate with your project. The policy profile contains a set of policies that use vulnerability scores and severities, license types, and component versions as criteria to automatically reject or approve inventory items during a codebase scan (or post-scan). For more information about policy profiles in general, see Managing Policies to Automatically Review Inventory.
	Automatically reject inventory items impacted by a new vulnerability that violates your policy	Determine what action the system should take for published inventory affected by a new security vulnerability discovered during a post-publication scan, an Electronic Update, or Library Refresh. The selected action applies to both non-reviewed and previously approved inventory items on the **Project Inventory** tab. ● Select this checkbox to automatically reject those project inventory items impacted by a new security vulnerability only if this vulnerability has a CVSS score or severity *greater than* the thresholds configured as policy for the Code Insight project. For each inventory item rejected due to a new security vulnerability, the ⚑ icon and a tip are added to indicate the status change and its reason. If a new vulnerability does not exceed policy thresholds, the current status of the inventory item is not affected. ● Leave the checkbox unselected to retain the current status of inventory items impacted by the new vulnerability. For information about setting policies that define CVSS-score and severity thresholds used to reject or approve inventory items automatically, see Policy Page and Policy Details Window. For information about associating these policies with a project, see Managing Policies to Automatically Review Inventory.

Table 8-26 ▪ Edit Project: Review and Remediation Settings Tab (cont.)

Category	Section/Field	Description
Manual Review Options	**What should happen if inventory items are not reviewed by policy?**	Determine what action should be triggered for those inventory items that are *not* affected by policy (and therefore have a **Not Reviewed** status) during the publication of inventory either as part of a scan or manually by a user: ● **Do nothing**—Simply show the status of the inventory item as **Not Reviewed** on the **Project Inventory** tab. ● **Send an email notification to the project contact**—Automatically send an email to the Project Contact, stating the need for a manual review of the item. The value for **Select the minimum priority...** (described in the next table entry) affects this option. ● **Automatically create a manual review task**—Automatically create a manual review task assigned to the default security or legal contact, and send an email, notifying the contact about the assigned task. (Information about managing such a task to track the progress of a manual review is found in Creating Inventory from the Project Inventory Tab.) The value for **Select the minimum priority...** (described in the next table entry) affects this option. The Project Contact is automatically designated as the creator of manual review task.
	Select the minimum priority to perform the action selected above	(Enabled when an option other than **do nothing** is selected for the previous field.) Select the minimum inventory priority (**P1**, **P2**, **P3**, or **P4**) to which the value for the previous field applies. For example, if the previous field is set to **send an email notification to the project contact** and minimum priority is set to **P3**, then the email notification will be sent for only those non-reviewed inventory items with a P1, P2, or P3 priority. No email notification will be sent for P4 inventory items. ***Note ▪*** *This option has no effect on the* ***do nothing*** *value.*

Table 8-26 • Edit Project: Review and Remediation Settings Tab (cont.)

Category	Section/Field	Description
	What type of manual reviews will be performed on this project?	Determine the type of manual review tasks to be generated: • **Legal Only**—Review tasks are generated for those non-reviewed inventory items that meet no policy criteria. The tasks are automatically assigned to the default Legal reviewer. • **Security Only**—Review tasks are generated for only those non-reviewed inventory items that have security vulnerabilities. The tasks are automatically assigned to the default Security reviewer. • **Both Legal and Security**—Review tasks are generated for all non-reviewed inventory items meeting no policy criteria and are assigned to the default Legal reviewer. Additionally, review tasks are generated for those non-reviewed inventory tasks associated with security vulnerabilities and are assigned to the default Security reviewer. With this value, a single inventory item might have both a legal review task and security review task generated. However, if the default reviewers are the same user, a single task is created, describing the requirement for both a legal and security manual review.
	Select reviewers for this project	If desired, designate a new default reviewer to which to assign manual review tasks. (The available reviewer types—Legal or Security or both—depend on the type of manual reviews your site performs, as defined for the previous option.) If your site generates both Legal and Security review tasks, Code Insight determines which reviewer—Legal or Security—is assigned the task and then notified of the task by email. (See the previous option description for “both Legal and Security” for more information about how this determination is made.) The reviewer can then manage the task accordingly, possibly reassigning it to another user. For details about managing and reassigning tasks, see Creating and Managing Tasks for Project Inventory. To select a new reviewer, click **Change User** next to the name of the current **Legal reviewer** or **Security reviewer** assignee, select a user from the **Select new...contact** dialog, and click **Apply**. When a new default reviewer is selected, that user is automatically given the role of project “reviewer” if the user does not currently have this role. However, if a specific task is reassigned to another user, that user is not automatically given the “reviewer” role and must be given that role manually (if the user is not already have it).

Table 8-26 ▪ Edit Project: Review and Remediation Settings Tab (cont.)

Category	Section/Field	Description
Remediation Options	These options determine the next step in the remedial process for rejected project inventory.	
	What should happen if inventory items are rejected?	Determine what action should be triggered for those inventory items that are automatically rejected by policy during an Electronic Update, Library Refresh, or the publication of inventory either as part of a scan or manually by a user: ● **Do nothing**—Simply show the status of the inventory item as **Reject** on the Project Inventory tab. ● **Send an email notification to the project contact**—Automatically send an email to the Project Contact stating the need for remediation work on the inventory item. ● **Automatically create a remediation task**—Automatically create a remediation task assigned to the default development contact (see the **Assignee for remediation work** option) and send an email, notifying the contact about the assigned task. (Information about managing such a task to track the remediation progress is found in Creating and Managing Tasks for Project Inventory.) The Project Contact is a automatically designated as the task creator. ● **Automatically create a remediation task and an external work item**—Automatically do the following: ● Create a remediation task assigned to the default development contact (see the **Assignee for remediation work** option) and send an email, notifying the contact about the assigned task. (Information about managing such a task to track the remediation progress is found in Creating and Managing Tasks for Project Inventory.) The Project Contact is automatically designated as the task creator. ● Create a work item and associate it with the task. The work item is created in your Application Lifecycle Management (ALM) system by using the settings defined for the ALM instance with which the Code Insight project is associated. For more information about configuring an ALM instance for the project, see Associating the Project with an Application Life Cycle System to Create Work Items. ***Note ▪*** *Currently Code Insight supports only Jira as an ALM system and Jira issues as work items.*

Table 8-26 ▪ Edit Project: Review and Remediation Settings Tab (cont.)

Category	Section/Field	Description
	Assignee for remediation work	If desired, designate a new default development contact to which to assign remediation tasks. This contact can then manage the task accordingly—for example, reassigning it to another user or manually creating an external work item and assigning it to someone on the development team. For details about managing and reassigning tasks, see Creating and Managing Tasks for Project Inventory. To select a new contact, click **Change User** next to the name of the current assignee, select a user from the **Select new...contact** dialog, and click **Apply**.
Actions	These buttons control whether changes to project settings are saved across all **Edit Project** tabs.	
	Save	Click this button to save your project edits and return to the **Summary** tab.
	Cancel	Click this button to return to the **Summary** tab without saving your project edits.

See Also

Editing the Project Definition and General Settings
Policy Page
Policy Details Window
Project Defaults Tab
Managing Policies to Automatically Review Inventory
Creating Inventory from the Project Inventory Tab
Creating and Viewing External Work Items for a Project Inventory Task
Updating Inventory Review and Remediation Settings for a Project
Associating the Project with an Application Life Cycle System to Create Work Items

Edit Project: Scan Settings Tab

The **Edit Project: Scan Settings** tab on the **Edit Project** window displays information about the scan settings defined for the selected project. You can edit the following information on this tab. (See also the Edit Project: General Tab to configure the project setting that determines whether the scan retains inventory that has no files associations.)

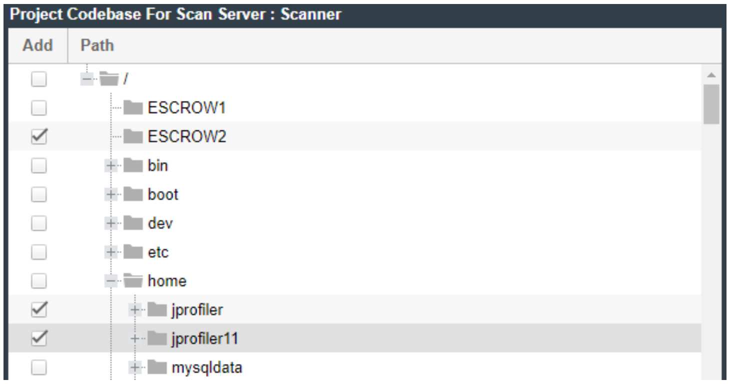
Table 8-27 ▪ Edit Project: Scan Settings Tab

Category	Column/Field	Description
Scan Server Options	These fields identify the Scan Server and profile used to run a project scan.	
	Scan Profile	The scan profile associated with the project. You can select a different scan profile from the dropdown list. Click ⓘ to view the properties of the currently selected scan profile.
	Scan Server	The Scan Server assigned to this project. This field is not editable. Click ⓘ to view the properties of the currently selected Scan Server.
Auto-Publish	These options enable and configure the automatic publication of project inventory as part of the project scan process. If the **Auto-publish system-created inventory items meeting this minimum Confidence Level** is selected to enable auto-publication, the other auto-publish options are made available.	
	Auto-publish system-created inventory items meeting this minimum Confidence Level	Select this option to enable the auto-publication of system-generated inventory items. (By default, the option is selected.) Then select the minimum Inventory Confidence level required to determine which items to auto-publish: ● **Low**—Auto-publish all system-generated inventory. ● **Medium**—Auto-publish only those system-generated inventory items with Medium and High confidence levels. (This is the default value.) ● **High**—Auto-publish only those system-generated inventory items with a High confidence level. For a description of the Confidence levels and how they are used, see Inventory Confidence. ***Note ▪*** *The scan ignores this rule for the following system-generated inventory and never automatically publishes this inventory:* ● *Inventory detected by the File Name Analyzer only* ● *Inventory automatically designated as Work in Progress*

Table 8-27 ▪ Edit Project: Scan Settings Tab (cont.)

Category	Column/Field	Description
	Do not auto-publish inventory items with an undetermined license	Select this option to *not* auto-publish any system-generated inventory item with an undetermined license (that is, an inventory item whose **License** value is **I don’t know**). An undetermined license can occur under the following conditions: ● The scan was not able to identify a license for the given component during the scan and therefore set the **I don’t know** license value. ● The inventory item has multiple possible disjunctive licenses (for example, “GPLV2 or MIT”). However, the scan could find no evidence of the desired selected license and therefore set the **I don’t know** license value. ● The inventory item has multiple possible conjunctive licenses (for example, “GPLv2 and MIT”). However, since Code Insight currently supports only a single selected license, the scan automatically set the **I don’t know** value for the inventory item. By default, this option is *not* selected, allowing the auto-publication of inventory with undetermined licenses. The option is available only if **Auto-publish system-created inventory items meeting this minimum Confidence level** is selected.
	Mark associated files as reviewed	Select this option to automatically mark the files associated with each auto-published inventory item as “reviewed”. (By default, the option is selected.) This option is available only if **Auto-publish system-created inventory items meeting this minimum Confidence level** is selected.
Project Codebase for Scan Server	These settings enable you to limit which directories are scanned in your codebase.	

Table 8-27 ▪ Edit Project: Scan Settings Tab (cont.)

Category	Column/Field	Description
	Path	From the interactive directory tree representing the project’s codebase on the Scan Server, select the checkbox next to one or more top-level directories that you want to scan. To scan only specific subdirectories in a top-level directory, drill down in that directory and select the desired subdirectories.  Project Codebase For Scan Server : Scanner Add Path ☐ / ☐ ESCROW1 ☑ ESCROW2 ☐ bin ☐ boot ☐ dev ☐ etc ☐ home ☑ jprofiler ☑ jprofiler11 ☐ mysqldata If the Scan Server is down, no project tree is displayed.
	Selected Paths	The pane showing the path for each directory currently selected for the scan. As a quick method for removing a given directory from the scan without having to drill down in the tree to locate it, simply click the **X** next to the directory in this pane. If the Scan Server is down, this pane is blank.
Actions	These buttons control whether changes to project settings are saved across all **Edit Project** tabs.	
	Save	Click this button to save your project edits and return to the **Summary** tab.
	Cancel	Click this button to return to the **Summary** tab without saving your project edits.

See Also
Editing the Project Definition and General Settings
Updating Scan Settings for a Project
Setting Policies to Publish Inventory Automatically

Edit Token Dialog

The **Edit Token** dialog appears when you click the **Edit Token** icon on the **Preferences** page. It lets you edit an authorization token (that is, a JSON Web Token known as a JWT) used to authenticate calls to Code Insight REST APIs. The token is associated with the user account under which you are logged in or whose password you have changed in the **Change Password** fields on the **Preferences** page.

This dialog also allows you to copy the token value to the Clipboard so that you paste it whenever needed for Code Insight REST access (for example, when configuring the Code Insight plugins, importing or exporting project data, or running REST API directly).

The dialog has the following fields:

Table 8-28 ▪ Edit Token Dialog

Column/Field	Description
Name	Enter a name for the token you are creating.
Token	Displays the actual characters of the system-generated token.
Select Token Text	Click this button to highlight the token characters displayed in the **Token** field. To copy the toke to the clipboard, press **CTRL-C**.
Expiration	A read-only field that displays the expiration date of the token, or the text “Token has no expiration date.”
Save	Click this button to save your edits.
Cancel	Click this button to exit the **Edit Token** dialog without saving your edits.

See Also
Preferences Page
Add Token Dialog
Exporting and Importing Project Data
Performing Remote Scans

Edit User Dialog

The **Edit Users** dialog is where you can edit users who are already in the Code Insight system. The dialog contains the following columns and fields:

Table 8-29 ▪ Edit User Dialog

Column/Field	Description
Login	Displays the login of the selected user. This field is read-only and cannot be changed.
First Name	Displays the first name of selected user. To change the user’s first name, type over the existing name.
Last Name	Displays the last name of selected user. To change the user’s last name, type over the existing name.
Email	Displays the email address of the user associated with the login. To change the user’s email address, type over the existing email address.

Table 8-29 ▪ Edit User Dialog (cont.)

Column/Field	Description
Password	The current password is not displayed in this field. A user with Code Insight System Administrator permissions can type a new password in this field.
Password Confirm	The current password is not displayed in this field. A user with Code Insight System Administrator permissions can type a new password in this field.
Question	The prompt that the user must answer to retrieve a forgotten password.
Answer	The answer to the question in the previous field.
Submit	Select **Submit** to have the system save your user edits. A prompt appears to notify you that your edits have been saved.
Cancel	Select **Cancel** to return to the Administration Users tab without saving your changes.

See Also
Users/Permissions Tab

Electronic Updates Tab

An initial full Electronic Update is run automatically after your initial startup of Code Insight. It provides the basis of a local Data Library used by Code Insight to identify OSS and third-party code in your codebase. The **Electronic Updates** tab on the **Administration** page enables you to configure how and when subsequent Electronic Updates are run to keep this library up to date. Refer to the following topics for more information:

- Overview of Electronic Update Setup
- Field Descriptions

For detailed instructions on how to schedule and run Electronic Updates, see “Configuring Code Insight” in the *Code Insight Installation & Configuration Guide*.

Overview of Electronic Update Setup

The following describes the basics for configuring electronic updates:

- Specifying an Update as Server or Local
- Scheduling Electronic Updates
- Notification of an In-Progress Electronic Update

Specifying an Update as Server or Local

The **Electronic Update Type** field (see Field Descriptions below) on the **Electronics Update** tab enables you to configure the Electronic Update to run as either a server or local update. The difference between the two methods is the means by which the Code Insight server obtains the files required to run the update:

- During a **server** Electronic Update, the most recent Electronic Update files are automatically downloaded from Revenera to the Code Insight server prior to processing the update.
- For a **local** Electronic Update, you must manually download the Electronic Update files from Revenera to a location that is locally accessible to the Code Insight server, such as a shared drive or a local USB drive. Then, when an update is triggered, the Code Insight server automatically uploads the files and proceeds with the update. This type of Electronic Update is useful when the Code Insight server has no external Internet access or when a specific Electronic Update version is needed for testing or demo purposes.

By default, the file download from Revenera is performed using HTTPS. However, you can configure the download process to use SFTP instead. For more information, see “Configuring an SFTP Connection for Downloading Update Files” in the *Code Insight Installation & Configuration Guide*. This configuration must be performed before running updates.

Scheduling Electronic Updates

The remaining fields (see Field Descriptions below) on the **Electronic Updates** tab allow you either to schedule an Electronic Update to run automatically at regular intervals or to manually request an update as needed. By default, an update is *incremental* (that is, the update applies on changes from the previous update). However, you have the option force a *full* Electronic Update, which replaces all data from the previous update. (A full update might be necessary, for example, if the most recent update did not complete properly.)

In summary, you can schedule the following:

- An incremental Electronic Update (server type only) that runs automatically at a regular frequency that you define.
- An incremental Electronic Update (server or local type) that you manually run as needed.
- A full Electronic Update (server or local type) that you manually run when necessary. Use this option with caution as forcing a full update to run will take several hours to complete, similar to the initial update run when Code Insight was first installed.

Note ▪ *Codebase scans cannot be performed during the Electronic Update process, but a scan that is already underway will not be interrupted when an update is scheduled to begin. The Electronic Update will be queued and automatically run based on queue order.*

Notification of an In-Progress Electronic Update

Whenever an Electronic Update is in progress, a banner is displayed at the top of the Code Insight UI, indicating that an update is running and that scheduled scans will resume once the update completes. The banner is shown for any Electronic Update—whether server or local, forced or automatically run by schedule—and persists across all Code Insight pages.

The banner is automatically closed once the Electronic Update completes. (Users cannot manually close the banner.)

Field Descriptions

The tab contains the following columns and fields:

Table 8-30 ▪ Electronic Updates Tab

Category	Column/Field	Description
General	The initial step in running an Electronic Update is to determine whether you are running it as a local or server update. For more information about these two types of updates, see Specifying an Update as Server or Local.	
	Electronic Update Type	Select the type of Electronic Update to run based how the Code Insight server obtains the Update Manifest and Update Data files required to perform the update: ● **Local**—This type of Electronic Update requires that you have manually downloaded these files from Revenera to a locally accessible location prior to the update. During the update, the Code Insight server uploads each of these files from this location, which you identify in the **Update Manifest File** and **Update Data File** fields. ● **Server**—The most recent Electronic Update files are automatically downloaded from Revenera to the Code Insight server as part of the Electronic Update process. ***Note ▪*** *By default, downloading files from Revenera is performed using HTTPS. However, you can configure the download process to use SFTP instead. For more information, see “Configuring an SFTP Connection for Downloading Update Files” in the Code Insight Installation & Configuration Guide. This configuration must be performed prior to running updates.*

Table 8-30 ▪ Electronic Updates Tab (cont.)

Category	Column/Field	Description
Local Electronic Update configuration	If you intend to run a *local* Electronic Update, you must specify the location of the two required “update” files that you manually downloaded from Revenera. The location of each file must be locally accessible. To run the update, use the **Run Update Now** option.	
	Update Manifest File	Click **Select File** to search for and select the Update Manifest file (`update_manifest.txt`) to upload to the Code Insight server. The manifest file contains the following: ● Information that Code Insight uses to determine whether to perform the update. ● The expected hash value for each data file stored in the `update.zip` file (see the **Update Data File** field description). This information will be compared with the hash values of actual files in the archive to ensure that the files have not changed or been tampered with.
	Update Data File	Click **Select File** to search for and select the data archive (`update.zip`) file to upload to the Code Insight server. This archive contains data files that provide the CVSS information used by Code Insight to perform update. Code Insight uses the hash information in the manifest file (see the **Update Manifest File** field description) to ensure that the data files are the expected ones and have not changed or been tampered with.

Table 8-30 ▪ Electronic Updates Tab (cont.)

Category	Column/Field	Description
Configuration for automatic Electronic Updates (server update only)	Code Insight enables you to configure *server* incremental updates to run automatically at a frequency you define, as described below. Note that you can always manually force an incremental or full update between the scheduled updates, or you can disable scheduled automatic updates altogether and manually run updates as needed. In either case, you would need to use the **Run Update Now** option to run an Electronic Update.	
	Update Frequency	Select from one of the available frequencies for running an incremental Electronic Update automatically: ● **Never**—If you select **Never**, Electronic Updates will not be run automatically. (Selection of this option hides any additional dropdown lists.) You can always manually schedule an incremental or full update as needed using the **Run Update Now** option. ● **Daily**—If you select **Daily**, a second dropdown list is displayed to choose the time of day when you want the Electronic Update to occur. ● **Weekly**—If you select **Weekly**, both the “time of day” and **Select a day...** dropdown lists are displayed. Select the time of day and the day of the week when you want the Electronic Update to occur.
	Save Schedule	Click this button to save the schedule. Your future incremental updates will run automatically according to the frequency you defined.

Table 8-30 ▪ Electronic Updates Tab (cont.)

Category	Column/Field	Description
Configuration for manually running an Electronic Update	You can manually run an Electronic Update at any time. The update is run immediately or placed in queue and initiated once all pending scans have completed. Currently, this is the only way to schedule a local update. If automatic server updates are also configured, a manually-run update is in addition to the automatic updates.	
	Run Update Now	Select the scope of the manually-run update: ● **Incremental Update**—Schedule an incremental Electronic Update. However, if no changes have occurred in the Electronic Update data since it was last run, the update is not executed. ● **Full Update**—Force a full Electronic Update to run whether or not changes have occurred in the Electronic Update data since it was last run. Use this option judiciously as a full Electronic Update takes several hours to complete, similar to the time required to run the initial update when Code Insight was installed.
	Update	Click this button to initiate the Electronic Update immediately or once pending scans are completed.

See Also
“Configuring Code Insight” chapter in the *Code Insight Installation & Configuration Guide*

Email Server Tab

The **Email Server** tab on the **Administration** page allows you to enable email notifications and set email options. The tab contains the following columns and fields:

Table 8-31 ▪ Email Server Tab

Column/Field	Description
Enable Email Server	Select **Yes** to enable Code Insight to use the email server or **No** to leave it disabled. The default is **No**. The rest of the fields on this page are not available until you select **Yes**.
Sender’s Email Address	Enter the email address of the sender.
SMTP Host Name	Enter the Simple Mail Transfer Protocol (SMTP) host name.
SMTP Host Port	Enter the port number of the SMTP host.
SMTP User Name	Enter the SMTP user name. This field is optional. Leave it blank if you are using anonymous SMTP.

Table 8-31 ▪ Email Server Tab (cont.)

Column/Field	Description
SMTP User Password	Enter the SMTP user password. This field is optional. Leave it blank if you are using anonymous SMTP.
Enable SMTP over TLS	Select **Yes** to use Transport Layer Security (TLS) to secure email over SMTP or select **No** to leave this option disabled.

Evidence Details Tab in the Analysis Workbench

The **Evidence Details** tab provides details about the inventory, component, and the files in the inventory.

Note ▪ *For files scanned by a Code Insight scan-agent plugin on a remote system, only license evidence is currently reported in Code Insight.*

The tab has the following fields:

Table 8-32 ▪ Evident Details Tab

Column/Field	Description
Expand All/Collapse All	Click to toggle between expanded and collapsed display.
Search field	Enter search criteria.
Tree view	Click to change the display to a tree view.
List view **o**	Click to change the display to a list view.
Select Evidence Types	Click to select evidence types to display.

File Search Results Pane

The **File Search Results** pane displays the results of your file search. The **File Search Results** pane has the following fields.

***Note** ▪ Some panes do not contain data until you choose a file in another pane.*

Table 8-33 ▪ File Search Results Pane

Column/Field	Description
↻	Click to refresh the search.
Advanced Search	Click to open the **Advanced File Search** dialog on which you can choose a standard search or add a new one.
Clear Search Results	Click to clear the results of the search.
Current Search	Displays the criteria for the current search.
Results Tree	The results of the current search.

Global Component & License Lookup Tab

The **Global Component & License Lookup** tab on the **Data Library** page provides a means for users to explore OSS and third-party components and licenses in the Code Insight Data Library outside the context of project inventory. From this tab, users perform a filtered search on components or licenses in the library (on the **Components** or **Licenses** tabs, respectively) and then, from the list of results, can delve into the details about the individual components or licenses.

A global search might be useful, for example, when a current inventory component is associated with security vulnerabilities. Users can perform a global search on the Data Library to look for components and their versions that might be associated with less severe or no vulnerabilities, thereby helping the user to decide whether to replace the current inventory component with another more secure one.

For more information about accessing and using this tab, see Exploring Components and Licenses in the Data Library.

The **Global Component & License Lookup** tab comprises the following tabs:

- Components Tab
- Licenses Tab

Components Tab

The **Components** tab on the **Global Component & License Lookup** tab enables you to perform a filtered search for specific OSS or third-party components (both those that are standard in the Code Insight Data Library and those that are custom) and then explore the components that end up in the search results.

During your component search, you might discover that a component is missing from the Data Library. From the **Components** tab, you can create a custom component for the missing component either from scratch or based on the **Search By** criterion you entered. The component is saved to the Code Insight database for immediate lookups and, in the background, automatically indexed in the Data Library to make it globally available. You can also edit a custom components from the **Components** tab.

For additional information about the tab, see the following:

- For instructions on how to access the **Components** tab, see Exploring Components and Licenses in the Data Library.
- For more information on how to use the **Components** tab, see Exploring Components Globally.
- For a description of how to use the **Components** tab to create a custom component either from scratch or based on the **Keyword**, **URL**, or **Forge** criterion that you entered for **Search By**, see Creating a Custom Component.

The following table describes the fields and mechanisms available on this tab.

Table 8-34 ▪ Components Tab

Section	Field/Column	Description
Search By	Select one of the following criterion by which to search components (including custom components) in the Code Insight Data Library. Then click **Search**. If no matches are found, a pop-up message is displayed with the message “No results found. Please check and try again.”	
	Key	Search for components by one or more strings in the component name. Enter the string(s) in the associated **Keyword** field. Multiple strings must be separated with spaces (not commas) and can be listed in any order, as in the example: **tomcat apache** The search will filter to only those component names that contain *all* the strings entered for the **Keyword** field. ***Note ▪*** *The search is case-insensitive, so it filters to all such components, no matter the upper or lower case of these strings in the* ***Keyword*** *field or in the actual component name.*

Table 8-34 ▪ Components Tab (cont.)

Section	Field/Column	Description
	URL	Search for components by the URL of the forge in which the components are found. For the **URL** value, you can enter the complete forge path, such as **https://github.com/jquery/jquery**, or a string in the path, such as **jquery**. ***Note*** ▪ *The search is case-insensitive, so the results will include all components with the matching forge path or path string (whichever criterion you entered in the **URL** field), no matter the upper or lower case used in the criterion or in the actual component path.*
	Forge	Search for a specific component by the name of its forge and third-party project or repository. First, select the forge name from the **Forge** dropdown list. Then provide a value for each additional field required to identify the project or repository within the forge.
	Component ID	Search for a component by its ID. Enter the complete ID for a component in the Code Insight Data Library or for a custom component. Ensure that the value is a positive integer of no more than 19 characters. If you enter a value in any other format, the field is immediately bordered in red and the **Search** button is disabled. (You can hover over the value for an explanation of the error.) If no match for the ID exists, a pop-up message stating “No results found...” is displayed
Search	Click this button to search the Code Insight Data Library based on the criterion provided for your **Search By** selection.	
Create New Component	Click this button to open the **New Custom Component** window. From here, you can create a custom component either from scratch or based on the **Keyword**, **URL**, or **Forge** criterion that you entered for **Search By**. For complete instructions, refer to Creating a Custom Component.	

Table 8-34 ▪ Components Tab (cont.)

Section	Field/Column	Description
Search results grid	The results of the component search are displayed in a grid list below the search criterion. The list contains the following columns to describe the attributes of each component. Certain attributes provide mechanisms that enable you to examine more information about the component.	
	Component Name	The name of the component. Click the ⓘ icon next to the component name to open the Component Details Window. This window provides publicly available details about the component, such as whether it has security vulnerabilities, supports your product's encryption capabilities, and is a custom component.
	Forge	The name of the forge that contains the component. No link to the web page of the forge itself is provided here. However, see the next field, **URL**, for a link to the component's project/repository location within the forge.
	URL	The URL of the component's third-party project or repository within the forge. When you click the hyperlinked text, the external web page of the project or repository is opened in a separate browser tab.
	Possible License(s)	Licenses that might associated with the component. Click the ⓘ icon next to a license to open the License Details Window and view details about the license.
	Actions	Click an icon in this column to perform the given action (described below) on the component: ● **View Versions** icon—Opens the Versions for <component> Window, enabling you to view information about each version of the component, including the licenses and vulnerability totals (by severity) associated with each version. You can also create a new version for the component from this window. ● **Edit Component** icon—(Available for custom components only) Opens the **Edit Custom Component** window, enabling you to update the properties of the custom component.

Grid Control

You can do the following to manage the grid.

- Control the column presentation in the grid:
    - Click the up ↑ (or down arrow) in the **Component Name** column header (or select the appropriate sorting order from the header's dropdown list) to sort the component list in ascending or descending alphabetical order.
    - From the dropdown list in any column header, select the columns you want to display or hide in the grid.
- Use the navigation icons at the bottom of the grid to move between next or previous pages or to a specific page number in the search results.
- Click the refresh ↻ icon to keep the data in the results current.

Licenses Tab

From the **Global Component & License Lookup** > **Licenses** tab, users can search the Code Insight Data Library for a specific OSS or third-party license.

Additionally, if you discover a license missing from the Data Library, you can create a custom license for the missing license from the **Licenses** tab. The custom license is added to the Code Insight database and made immediately available in any of the dropdown lists for a license lookup and from the **Licenses** tab on the **Global Component & License Lookup** tab.

For more information about this tab, see the following:

- For instructions on how to access the **Licenses** tab, see Exploring Components and Licenses in the Data Library.
- For more information on how to use the **Licenses** tab, see Exploring Licenses Globally.
- For a description of how to use the **Licenses** tab to create a custom license from scratch, see Creating a Custom License.

The following table describes the search mechanism used to look up a license from the **Licenses** tab and provides a description of the attributes displayed for the license found.

Table 8-35 ▪ Licenses Tab

Field	Description
License	Select the desired license from the **License** dropdown list. Alternatively, with the list opened, enter a string found in the license name to filter the list. Then select the license from the filtered list.
Id	The internal identification number generated for the license in the Code Insight Data Library.
Name	The name of the license, along with its short name (usually the SPDX short identifier) in parentheses.

Table 8-35 ▪ Licenses Tab (cont.)

Field	Description
Family	The family of licenses to which this license belongs, if applicable. The following are examples of family licenses: Apache, BSD, EPL, MIT, and MPL.
Priority	The priority ranking of the license as determined by Code Insight. For more information, see License Priority.
URL	The URL link to the license website.
Description	A short description of the license.
External ID	The ID that corresponds to the license in an external system (such as your site's license-tracking system). This mapping enables you to easily locate the license in the external system. If the license is not mapped to an external ID, this field is blank.
Category	Currently not in use.
Policy	The **Yes** or **No** indicator specifying whether a Code Insight review policy is defined for the license. (License policies are used to approve or reject inventory items when they are published.).
Custom License	The **Yes** or **No** indicator specifying whether this license was manually created in Code Insight.
Family License	The **Yes** or **No** indicator specifying whether this is a family license from which other OSS licenses are derived. The following are examples of family licenses: Apache, BSD, EPL, MIT, and MPL.
Commercial	Currently not in use.
Copyleft	Currently not in use.
Free Software License	The **Yes** or **No** indicator specifying whether the license is considered to be a free software license.
GPL V2 Compatible	The **Yes** or **No** indicator specifying whether the license is compatible with GPL v2.0.
Patent License Grant	Currently not in use.
License Text	The complete text of the selected license as stored in the Code Insight Data Library. This text represents the external forge license text.
Create New License	Click this button to open the **New Custom License** window. From here, you can create a custom license from scratch. For complete instructions, refer to Creating a Custom License.

Import Project Data Dialog

The **Import Project Data** dialog is displayed when you select the **Import Project Data** option from the **Manage Project** menu on the **Summary** tab for a project. This dialog enables you to import data from another Code Insight project or source into the current Code Insight project (called the *target project*) from which you are invoking the import. The project data to be imported must be in a properly formatted and archived JSON file, called the *import data file* (such as the output of a Code Insight project data export).

Complete details on import process—how to prepare for it, how to run the import, and what results to expect—are provided Exporting and Importing Project Data.

An additional setting, not shown on the **Import Project Data** dialog but used during the import process, is defined at the project level. The setting (**On data import or scan, delete inventory...**) determines whether the import should create “empty” inventory on the target—that is, inventory in the import data file that either has *no* associated files or *has* associated files that have no match in the target project codebase. For a description of this setting and how to change its value, if necessary, before running the import, see Edit Project: General Tab.

The following table describes the import options available on the **Import Project Data** dialog.

Table 8-36 ▪ Import Project Data Dialog

Column/Field	Description
Choose File to Import	Click **Browse** next to the **Choose File to Import** field to search for and select the `.zip` file containing the JSON project data you are importing.

Table 8-36 ▪ Import Project Data Dialog (cont.)

Column/Field	Description	
Add Files to Inventory	Select **Yes** to enable the import to create new file associations with target inventory. The **File Matching Criteria** field is displayed (if it is not already). Select **No** to disable the creation of any new file associations with target inventory during the import process.	
	File Matching Criteria	Select the option that identifies the criteria (file path, file MD5, or both) used to determine whether a file associated with an inventory item in the import data file has a match in the target project codebase. Only those files in the import data file that have matches in the target project codebase can be associated with target inventory. For a description of the criteria options, see About the File-Matching Criteria for the Import.
	...to *n* directories above the file	If you have selected **Check only the file partial path** or **Check the file MD5 and the partial path** in the **File Matching Criteria** field, provide the directory depth that defines the partial path. That is, specify the number of directories *above the file name* that must be the same before two paths are considered as matches. For example, a value of 2 indicates that file paths must match 2 directories above the file name. Suppose a file in the import data file has the path `/ePortal-1.3/copy1/src/gettext.c` and a file in the target codebase has the path `/ePortal-2.0/copy1/src/gettext.c`. With a directory depth of 2, only `/copy1/src/gettext.c` needs to match in the two file paths to meet the partial-file criterion. You can enter a directory-depth value between 1 and 20, inclusively. For more information, see About the File-Matching Criteria for the Import.

Table 8-36 ▪ Import Project Data Dialog (cont.)

Column/Field	Description	
Mark Files as Reviewed	Select **Yes** to enable the import to flag files with the “Reviewed” status in the target project codebase if they do not already have this status. The **File Matching Criteria** field is displayed (if it is not already). Select **No** to disable the import’s ability to mark files as reviewed during the import process. ***Note ▪*** *For this option, the import compares only those files in the import data file that are marked as reviewed with files in the target codebase.*	
	File Matching Criteria	Select the option that identifies the criteria (file path, file MD5, or both) used to determine whether a file in the import data file has a match in the target project codebase. Only those files in the import data file that have matches in the target project codebase can be marked as reviewed. For a description of the criteria options, see About the File-Matching Criteria for the Import.
	...to *n* directories above the file	If you have selected **Check only the file partial path** or **Check the file MD5 and the partial path** in the **File Matching Criteria** field, provide the directory depth that defines the partial path. That is, specify the number of directories *above the file name* that must be the same before two paths are considered as matches. For example, a value of 2 indicates that file paths must match 2 directories above the file name. Suppose a file in the import data file has the path `/ePortal-1.3/copy1/src/gettext.c` and a file in the target codebase has the path `/ePortal-2.0/copy1/src/gettext.c`. With a directory depth of 2, only `/copy1/src/gettext.c` needs to match in the two file paths to meet the partial-file criterion. You can enter a directory-depth value between 1 and 20, inclusively. For more information, see About the File-Matching Criteria for the Import.

Table 8-36 ▪ Import Project Data Dialog (cont.)

Column/Field	Description	
Inventory Notes Handling	Select the option that defines how the import should handle custom fields and “notes” fields for imported inventory items. The “notes” fields include: ● **Notices Text** ● **Audit Notes** ● **Usage Guidance** l ● **Remediation Notes** ***Note*** ▪ *For a custom inventory field to be imported along with its associated inventory item, the name of the field defined for the inventory item in the import data file must match the name of a custom field defined for the target inventory item.*	
	Overwrite existing notes with imported notes	(Default) Select this option to overwrite the value of each “notes” and custom field in the target inventory item with the value of the corresponding field for the inventory item in the import data file. If any of these fields is blank in the import data file, the existing value for that same field in the target inventory item is retained.
	Append imported notes to existing notes	Select this option to append the value of each “notes” and custom field for the inventory item in the import data file to the value of the corresponding field in the target inventory item. The appended value in the target field is separated from the existing value with a line break and this heading: `Copied during import from <ProjectName>:<InventoryName> (TimeStamp)` If the value for any of these fields is the same in both the import data file and the target inventory item, no value is appended in the target field.
Inventory Usage Handling	Select either of these options to define how the import should handle the Usage attributes for imported inventory items. For a description of the inventory Usage fields, refer to Usage tab in the **Project Inventory Details Pane** topic.	
	Reset usage field values to system default	Do not copy existing Usage values for inventory items from the source project to the target project. Instead, in the target project, reset all Usage fields for imported inventory to the system default value: **Unknown**. (Default)
	Copy existing usage field values	Copy the existing Usage values for inventory items from the source project to the target project.

About the File-Matching Criteria for the Import

When configuring the import to create new file associations in target inventory or to mark files in the target project codebase as reviewed, you must define the file-matching criteria needed by the import to compare files in the import data file with the target codebase files. Target files that match files in the import data file are eligible for either of these import functions.

The file path and the file MD5 value are the key criteria used to locate target codebase files that match files in the import data file. When the MD5 value is used as a criterion, the MD5 for a file in the import data file must have an exact MD5 match in the target codebase. However, when the file path is used as a criterion, the file-matching process can apply various rules.

For more information, see the following sections:

- About File-Path Processing During the Import
- Available File-Matching Criteria

About File-Path Processing During the Import

When the file path is used as a criterion for locating a matching file, the import process internally subtracts the scan root path from the absolute path of codebase files in the import data file and in the target project. The result is the *complete file* path for a given file, as illustrated in these examples:

- **Absolute file path**—`/home/fnci/scanRoot/1/ePortal-1.3/src/gettext.c`
- **Root path**—`/home/fnci/scanRoot/1/`
- **Complete File path**—`/ePortal-1.3/src/gettext.c`

Then, based on the file-path criterion selected by the user, the import locates matching files by searching complete file paths, partial paths, or simply file names. The following examples illustrate a complete path in comparison with a partial path or file name:

- **Complete path**—`/ePortal-1.3/copy1/src/gettext.c`
- **Partial path**—`/copy1/src/gettext.c` **or** `/src/gettext.c`
- **File name**—`gettext.c`

Available File-Matching Criteria

The following describes the available options for the **File Matching Criteria** field used by the import to locate file matches between the import data file and the target project codebase.

Table 8-37 ▪ Options to Define File-Matching Criteria

Option	Description
Check only the file MD5	A file's MD5 value in the import data file must match an MD5 in the target project codebase.
Check only the file name	The name of the file in the import data file must match a file name in the target project codebase. (No path is compared in the file-matching process.)

Table 8-37 ▪ Options to Define File-Matching Criteria (cont.)

Option	Description
Check only the complete path	The complete path of a file (including the file name) in the import data file must match a complete file path in the target project codebase.
Check only the partial path	A file's partial path (including of the file name) in the import data file must match the partial path of a file in the target project codebase. For this criterion, you must also specify the directory depth of the partial path. See the **...to *n* directories above the file** field description in the previous table, Import Project Data Dialog. The partial-path depth enables the import to match files when the codebase location is different between the target project codebase and the project codebase whose scanned results are included in the import data file.
Check the file MD5 and file name	The MD5 and name of a file in the import data file must match the MD5 and name of a file in the target project codebase.
Check the file MD5 and the complete file path	A file's MD5 value and complete path (including the file name) in the import data file must match the MD5 and complete path of a file in the target project codebase.
Check the file MD5 and the partial file path	A file's MD5 value and partial path (including the file name) in the import data file must match the MD5 and partial path of a file in the target project codebase. For this criterion, you must also specify the directory depth of the partial path. See the **...to *n* directories above the file** field description in the previous table, Import Project Data Dialog. The partial-path depth enables the import to match files when the codebase location is different between the target project codebase and the project codebase whose scanned results are included in the import data file.

Inventory Details Tab in the Analysis Workbench

The **Inventory Details** tab in the **Analysis Workbench** contains a sub-tab for each inventory item you have opened from the **Inventory Items** pane. Each sub-tab contains the following fields describing a given inventory item:

Table 8-38 ▪ Inventory Details in the Analysis Workbench

Category	Column/Field	Description
Header information	The **Inventory Details** tab header shows buttons that enable you take actions on the inventory item and lists attributes about the item and its associated component.	
	Recall or **Publish**	A toggle button that shows **Publish** when viewing an unpublished or recalled item or **Recall** when viewing the details for a published inventory item. ● Click **Publish** to publish a currently unpublished (or recalled) inventory item. In the **Inventory Items** pane, the item is re-listed with a filled box icon next to its name. Additionally, the inventory item is now visible on the **Project Inventory** tab. Upon publication, the inventory item is automatically reviewed by the review policy currently associated with the project and is either approved, rejected, or kept it in a **Draft** state (**Not Reviewed** on the **Project Inventory** tab). On the **Project Inventory** tab, users can then further review the inventory item's security or legal issues and, if appropriate, take steps to remediate and prepare the item for inclusion in the final Third-Party Notices report for the project. ● Click **Recall** to unpublish a currently published inventory item in if it does not fit the criteria for a published item. In the **Inventory Items** pane, the item is re-listed with a clear box icon next to its name. (However, the item is removed from the **Project Inventory** tab.) A recalled inventory item retains the status it had before the recall (until it is re-published).
	View History	Click open the Inventory History Window, which shows a list of all updates made to the inventory item up to the current date and provides details for each update.
	Create Custom Rule	(Available when inventory **Type** is **Component**) Click to open the **Custom Detection Rule** dialog to define an new detection rule for codebase files that are associated with a third-party component but not associated with inventory. For details, see Managing Custom Detection Rules.

Table 8-38 ▪ Inventory Details in the Analysis Workbench (cont.)

Category	Column/Field	Description
	Save	Click to save any changes you have made to the inventory details. This action can trigger an automatic review of the inventory item. See Managing Policies to Automatically Review Inventory for details.
	Close	Click to close the **Inventory Details** pane without saving changes. You are asked to save changes before the actual closure.
	Review Status	The status of the inventory item: ● **Approved**—The item is approved for use in the software project. ● **Draft**—This item has not yet been reviewed by automatically by policy or manually by a user. ● **Rejected**—The item is not approved for use in the software project. Instead, the item needs further review and remediation before being used in the software project.
	Alerts	Notifies you whether or not security alerts exist for this item. If alerts exist, click the ***x* Open Alerts** or ***x* Closed Alerts** link to view their details. If no alerts exist, **None** is displayed. You can access the **Alerts** dialog from this pane to change the status or priority of an alert. For more information, see Managing Security Vulnerability Alerts.
	Priority	A dropdown list showing the priority level given to this inventory item by the system, with P1 as the highest priority and P4 as the lowest. ***Note*** ▪ *During a scan, the priority for auto-published inventory is automatically assigned based on the associated license.* You can change the priority for this inventory item by selecting a different priority from the dropdown list and clicking **Save**. For more information about priorities, see Inventory Priority.

Table 8-38 ▪ Inventory Details in the Analysis Workbench (cont.)

Category	Column/Field	Description
	Vulnerabilities	A bar graph showing the count of known vulnerabilities by severity color for the inventory item. Click the graph to view the list of vulnerabilities and their details. For details about the graph and vulnerabilities in general, see Security Vulnerabilities Associated with Inventory. The counts in this graph do not include vulnerabilities that are currently suppressed. If no vulnerabilities have been found for the inventory item, the value **No** is displayed in place of the graph. Additionally, if the **Type** value for the inventory item is **Work in Progress** or **License Only**, the value **N/A** is displayed.
	Created By	The creator of the inventory item as either: ● **System**—Code Insight automatically generated the item per one of these detection techniques (as designated in the **Notes** for the inventory item) during a scan: ● High Confidence Custom Auto-WriteUp Rule ● High Confidence Auto-WriteUp Rule ● Medium Confidence Auto-WriteUp Rule ● Automated Finding ● Low Confidence Auto-WriteUp Rule ● High Confidence MID Rule ● Low Confidence MID Rule ● Audit Import ● **<user_name>**—The first and last name of the user who manually created the item.
	Confidence	A simple three-segment graph representing the Confidence level (High, Medium, or Low) of the inventory item. The Confidence level is the measure of the strength of the discovery technique used to generate the inventory item. The graph shows three dark-shaded segments for High confidence, two for Medium, and one for Low. For more information about the Confidence levels, see Inventory Confidence.
	Created On	The date and time that the inventory item was created.
	Updated On	The date and time that the inventory item was updated. If the item has not been updated since its creation, the date and time shown here will be the same as the **Created On** date and time.

Table 8-38 ▪ Inventory Details in the Analysis Workbench (cont.)

Category	Column/Field	Description
Inventory details	The following attributes describe the inventory item. You can update these attributes as needed from this pane. For a description of the inventory creation or editing process, see Creating an Inventory Item from the Analysis Workbench or Editing Inventory from the Analysis Workbench.	
	Name	The name of the inventory item.
	Type	The type of inventory item based on the codebase files that point to evidence of a third-party or OSS component. ● **Work in Progress**—A set of files with evidence in common. The work in progress will become a component or license only via manual audit work. ● **Component**—Files from a specific component version with a known or unknown license. If this type is selected, the **Lookup Component** button becomes active, enabling you to select a new component instance (component, version, license) for the inventory item or create a new component for the item (see Using “Lookup Component” to Search for Components to Associate with Inventory). ● **License Only**—Files under a specific license without a known component.
	Component	(Available for **Component** inventory types) The name of the component. Click ⓘ to view publicly available information about the component. (See Component Details Window for details.) You can also do the following: ● Click ✎ to select a new version (or license) for the component. ● To help you make an informed decision about the version selection, you can click the **View all versions** link to open the **Versions for** <**component**> window. From here, view the list of all versions for the component and, for each version, its associated licenses and security vulnerability totals (by severity). You can also delve into more detail for each associated vulnerability. For more instructions, see Versions for <component> Window.

Table 8-38 ▪ Inventory Details in the Analysis Workbench (cont.)

Category	Column/Field	Description
	License	(Available for **Component** and **License Only** inventory types) The name of the license associated with the component or the **License Only** inventory item. Click ⓘ to view additional information about the license. See License Details Window for further information. For an inventory item of the type **Component**, you can click ✎ next to the **License** field to select a new license or version from respective **License** or **Version** dropdown lists. If you select a new license from a group of licenses under the **System Suggested License** category in the **License** dropdown or select the license from the **Other Licenses** category in the dropdown, the **Update License Mapping** window is displayed. This window gives you the option to save the license mapping at the system level. See Specifying a User-Preferred License Mapping for complete details.
	Description	A description of the inventory item. You can update the description as needed.
	URL	The URL of the forge repository for this inventory item. You can update the URL as needed.
	purl	The package URL for the component represented by the inventory item. This non-editable value is retrieved from the Code Insight data library and is applicable to a component associated with a non-custom version only. If no purl value is available in the data library or the version is custom, the value for this field is **N/A**. The use of package URLs is an attempt to standardize the way in which software packages and their locations are identified so that this information is more universal and uniform across programming languages, packaging conventions, tools, APIs, and databases.

Table 8-38 ▪ Inventory Details in the Analysis Workbench (cont.)

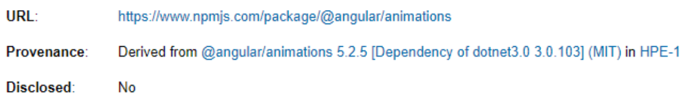
Category	Column/Field	Description
	Provenance	The source project from which the current inventory item was derived. ***Note ▪*** *You cannot update this property from the Code Insight Web UI in general, but you can edit it when creating or updating inventory using the Inventory REST API.* If the inventory item is not derived from another project, the value **Originated in this project** is displayed. However, if the inventory item is derived from another project (for example, the inventory item was imported, copied, or branched to the current project), the origin of the inventory is displayed with the inventory name and project name:  URL: https://www.npmjs.com/package/@angular/animations Provenance: Derived from @angular/animations 5.2.5 [Dependency of dotnet3.0 3.0.103] (MIT) in HPE-1 Disclosed: No If the source project and inventory item still exist, this value is hyperlinked so that you can open the source project directly to the **Project Inventory** tab, with focus on the **Inventory Details** page for the original inventory item. The linked inventory item enables you to trace the origin of the item through its chain of predecessors. You can explore the auditing and review details of the each preceding inventory item to determine inventory history—for example, the reason the item was previously approved or rejected. If the source inventory item or its project no longer exists, the link to the previous inventory item is provided is permanently disabled (once the link in initially clicked).

Table 8-38 ▪ Inventory Details in the Analysis Workbench (cont.)

Category	Column/Field	Description
	Dependency Scope	(Not editable) The dependency scope of the inventory item: ● **Runtime**—The inventory item is a dependency required at runtime. ● **Non-Runtime**—The inventory item is a dependency not required at runtime. ● **N/A**—The inventory item cannot be classified as a runtime or non-runtime dependency. Such items include top-level inventory, dependencies for which Code Insight does not currently support the reporting of scope, and migrated inventory for which a scan has not been run. For more information, see Dependency Scopes in the **Automated Analysis** section. ***Note ▪*** *Your access to inventory of a specific scope in a project can change if a certain reconfiguration has previously occurred—for example, a change to the scan profile or a re-upload of updated runtime and non-runtime dependencies—and a rescan or full rescan has subsequently taken place.*
	Disclosed	The **Yes** or **No** option indicating whether the third-party component or artifact represented by the inventory item known third-party dependency in your code before it was discovered by the scan or you. This field is used most often by analysts to denote information about the state of the inventory item.

Table 8-38 ▪ Inventory Details in the Analysis Workbench (cont.)

Category	Column/Field	Description
	Workflow URL	The URL (or a text reference such as a Jira issue number) that points to the request data pertaining to this inventory item as found in your site's external workflow system. When you view this value on the **Inventory Details** tab in **Project Inventory**, the URL displays as a link (labeled as **View Associated Request**), enabling the reviewer to easily access to the workflow data that tracks the status of open tasks for the inventory item. A text reference entered here is not converted to a link on the **Inventory Details** tab, but it still provides direction in locating the appropriate data in the workflow system. The value is **None** if you enter no URL or reference. Additionally, when you view the **Inventory Details** tab in **Project Inventory**, an ⓘ icon will be displayed next to the URL if additional request-related details are available for the inventory item. The reviewer can then click the icon for a quick review of pertinent details about the request without having to access the workflow system.
Notices Text tab	The **Notices Text** tab is used to finalize the exact content to include in the Notices report. You can edit the notices content as needed from this tab when editing an existing inventory item or creating a new one. For more information, see Finalizing the Notices Text for the Notices Report.	
	As-Found License Text	The **As-Found License Text** field shows the license text or license references found in the scanned codebase. You cannot edit this field. However, if you want to use this content in the Notices report, click **Copy to Notices Text** to copy the text to the **Notices Text** field and modify it if necessary. If content already exists in the **Notices Text** field, you can choose either to append the **As-Found License Text** content to the existing notices content or to replace the existing notices content. This field is blank if no license text or references were found in the scanned codebase. If this field contains information and the **Notices Text** field remains blank, the Notices report uses the content in this field. If both fields are empty, the report uses the license content from Code Insight Data Library (see License Details from the Code Insight Data Library).

Table 8-38 ▪ Inventory Details in the Analysis Workbench (cont.)

Category	Column/Field	Description
	Notices Text	The exact content to include in the Notices report. You can edit any license text previously saved to this field or manually add your own license text, such as license information for rules that you developed during your manual research on the inventory item. You can also copy the **As-Found License Text** content (see the previous description) to the **Notices Text** field and modify it as needed. As a third option, you can click the **Update Notices Text** button to pull a copy of the current license content from the Code Insight Data Library into the **Notices Text** field and modify it as needed. Or you can leave this field empty. If you provide information in this field, the Notices report pulls the content of only this field into the report. If this field is empty, the content of the **As-Found License Text** field is used in the report. If both fields are empty, the report uses the license content from Code Insight Data Library (see License Details from the Code Insight Data Library). For more information, see Finalizing the Notices Text for the Notices Report.
	Copy to Notices Text button	(Located within the **As-Found License Text** field) Click this button to copy content the text in this field into the **Notices Text** field and modify it as necessary. If the **Notices Text** field already contains content, you are given the option either to append the **As-Found License Text** content to the existing **Notices Text** content or to replace all the existing **Notices Text** content with the **As-Found License Text** content. Appended text starts on a new line after the existing content in the **Notices Text** field.
	Update Notices Text button	(Located within the **Notices Text** field) Click this button to copy content from the Code Insight Data Library into the **Notices Text** field. You can then modify the content as needed. If the **Notices Text** field already contains content, you asked whether to overwrite the content. If you select **No**, the copy operation is ended. If you select **Yes**, the operation proceeds. Refer to Using License Text from the Revenera Data Library in the Notices Report for the prerequisites needed to perform this copy and the types of issues you can encounter.
Notes tab	The **Notes** tab provides information about the automated and manual analysis of codebase as it relates to an inventory item.	

Table 8-38 ▪ Inventory Details in the Analysis Workbench (cont.)

Category	Column/Field	Description
	Detection Notes	System notes that can specify the following: ● The automated detection technique that was used to locate the component ● License information in the case that the license has changed from one version to another or if the component has multiple licenses ● Attributes extracted from a POM or manifest file containing project and configuration details
	Audit Notes	Any notes added to the inventory item by the auditor or reviewer, based on findings during the analysis. You can edit these notes as needed from this pane when editing an existing inventory item or creating a new one. See Viewing and Updating Detection and Auditing Notes in the Analysis Workbench.

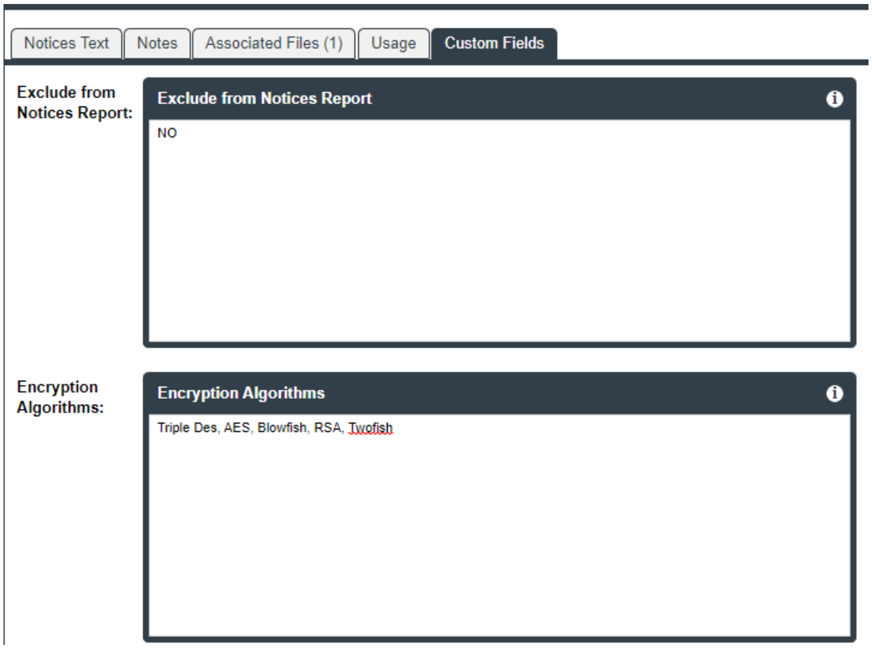
Table 8-38 ▪ Inventory Details in the Analysis Workbench (cont.)

Category	Column/Field	Description
Associated Files tab		Click this tab to view a list of the files that are part of the inventory for this project. Each file entry shows the following: - **Action**—Icons that you can click to perform certain actions on the file. Currently, only the ✖ icon shows, enabling you to disassociate the file from the inventory item. - **Alias**—The unique user-defined alias that was defined for the scanner (Scan Server or remote scan agent) to represent its scan-root path containing the codebase in which the file is located. The alias provides a name that is more meaningful than the scan-root path. (The actual absolute scan-root path for each scanner associated with the project is available on the project’s **Summary** tab.) - **File Path**—The file’s path relative to the scan-root path on instance hosting the scanner. You can click the file-path link to open the **File Details** tab for that file. - **Evidence**—The color-coded icons representing the types of open-source or third-party evidence found in the file (see Using the Filter Legend Options to Filter the Codebase for a description of the icons). A check mark indicates that the file has been reviewed. ***Note ▪*** *You cannot sort the file list.* Optionally, you can right-click a file entry for options that enable you to perform additional operations on the file, such as marking it as reviewed, reverting its reviewed status to unreviewed, and other operations. See Managing the Codebase Files for details about these same options that are also available from the **Codebase Files** and **File Search Results** panes in the **Analysis Workbench**. To add associated files to this list, see Adding Files to Inventory From the Codebase List.
Usage tab		The **Usage** tab provides details on how your product uses the OSS or third-party software. You can update this information as needed from this pane when editing an existing inventory item or creating a new one. See Viewing or Editing Inventory Usage Information from the Analysis Workbench.

Table 8-38 ▪ Inventory Details in the Analysis Workbench (cont.)

Category	Column/Field	Description
	Distribution Type	The option indicating how you are distributing the OSS or third-party component associated with the inventory item. The distribution type can affect license priority and obligations: ● **Internal**—The component is distributed internally only (for example, as an internal test framework included in the codebase but not distributed publicly with the software package). ● **External**—The component is a separate entity from your software package. It might be shipped as a separate component along with the software package or deployed through some method, such as a private cloud at the customer site. ● **Hosted**—The component is hosted in your company's data center (for example, as a SAAS application) ● **Unknown**—The distribution type is unknown.
	Part of Product	The option indicating whether the item is part of the core product or an infrastructure piece such as a build or test tool. This can affect whether third-party notices are required for this item. The value can be **Yes**, **No**, or **Unknown**.
	Linking	The option identifying how your software package links to the OSS or third-party component libraries. This method can affect license priority and obligations. ● **Not linked**—The software package uses no links to the component libraries. ● **Statically linked**—The component libraries are included in the software materials and thus linked statically. ● **Dynamically linked**—The component libraries are brought in at runtime. ● **Unknown**—The type of linking is unknown.
	Modified	The option indicating whether code from the OSS or third-party package has been modified for use by your organization. The value can be **Yes**, **No**, or **Unknown**.
	Encryption	The option indicating whether the component provides the encryption capabilities used in the product. Encryption can affect export controls. The value can be **Yes**, **No**, or **Unknown**.

Table 8-38 • Inventory Details in the Analysis Workbench (cont.)

Category	Column/Field	Description
Custom Fields tab	The **Custom Fields** tab displays fields that were defined specifically for your site to provide information that standard Code Insight fields on the **Inventory Details** tab do not capture about the inventory.  Notices Text Notes Associated Files (1) Usage Custom Fields Exclude from Notices Report: Exclude from Notices Report NO Encryption Algorithms: Encryption Algorithms Triple Des, AES, Blowfish, RSA, Twofish If no custom fields have been defined, the tab displays the message “There are no custom fields configured”. Use the following guidelines for entering (or editing) a value in a custom inventory field: - If available, click the ⓘ icon in the upper right corner of a field to obtain help on completing the field. - You can enter a value up to 64k (64000 characters) in size. - To save the value, click the **Save** button next to **Create Custom Rule** button (in the **Inventory Details** tab header.	

Inventory History Window

Users can view the history of updates made to a specific inventory item within a project by clicking the **View History** button on either the Inventory Details Tab in the Analysis Workbench or the Project Inventory Details Pane. The **Inventory History** window is opened, listing the updates in a grid, each row representing a specific update made to the inventory item. The updates are grouped by revision IDs, each representing a *revision session*—that is, group of updates that were saved together at a specific point in time.

Refer to the following topics for procedures related to this window:

- Viewing the Update History for an Inventory Item in the Analysis Workbench
- Viewing the Update History for an Inventory Item in Project Inventory

The following describes the information recorded for each update.

Table 8-39 ▪ Attributes of Each Update Listed in Inventory History

Field/Column	Description	
Revision header	Inventory updates are grouped chronologically by the revision session in which they were made, each session with its own header.	
	Revision ID	Identifies a single revision session consisting of one or more updates that were saved together at a specific point in time.
	Revision details	The date and time at which the revision was saved and the number of updates made during the revision session.
Details of update	The following details describe each update made to the inventory item.	
	Date	The date and time at which the update was saved. You can sort on this column. By default, the updates are listed in descending order by date so that users view the most recent updates first.
	Event	The type of update: **Inventory Created**, **Inventory Updated**, **Inventory Published**, **Full Rescan**, **Inventory Recalled**, or **Inventory Reviewed**.
	Action	The user action that resulted in the update: ● **Scan**—The initial scan the created the inventory item. ● **Project Copy**—A Code Insight project copy. ● **Project Rescan**—A full rescan on the project. Although a full rescan usually involves updates to project inventory, no separate records are listed for these updates in the history. ● **Project Import**—A Code Insight project import. This can be either a standard project import or the import performed during a project-branching process. ● **Manual Analysis**—A manual update made to the inventory item through the Code Insight Web UI or a REST API. ● **Policy**—An update to the policy profile that impacted the status of the inventory item. ● **hyphen (-)**—The value that defaults when the inventory was migrated from a pre-2021 R3 version of Code Insight.

Table 8-39 ▪ Attributes of Each Update Listed in Inventory History (cont.)

Field/Column		Description
	User	The name of the user who performed the update. The name is hyperlinked to open an email draft addressed to this user. If the system performed the update, the value **System** is displayed with no hyperlinked text.
	Field	Either of the following: ● **<field_name>**—The name of the inventory field that was updated. You can review its previous and new value in the **Old Value** and **New Value** columns. ● **Associated Files**—Term indicating that one or more files were added to or removed from the inventory item's list of **Associated Files** tab before the inventory item was saved. (No files names are listed in the **Old Value** and **New Value** fields.) Each **Associated Files** record reflects an inventory save (and thus a separate revision session). For example, if the user saved the inventory item each time a file association was added or removed, a separate **Associated Files** record in a separate revision session is added for each association update. If the user added or removed multiple file associations before saving the inventory item, a single **Associated Files** record reflecting all the association updates is added to the revision session.
	Old Value	The previous value of the inventory field. Note the following: ● The value for the **URL** field is hyperlinked and, when clicked, opens to the linked site on a new browser tab. ● If the value for the **Description** field or a notes field (such as **Audit Notes**, **Notices Text**, and others) exceeds 150 characters, a **Show more...** link is displayed to expand the row to show the entire text. You can then click **Show less...** to collapse the text. ● When you click the ⓘ icon next a **Component** or **License** value, a pop-up dialog shows read-only details about that item.

Table 8-39 ▪ Attributes of Each Update Listed in Inventory History (cont.)

Field/Column		Description
	New Value	The value of the inventory field after the change. Note the following: ● The value for the **URL** field is hyperlinked and, when clicked, opens to the linked site on a new browser tab. ● If the value for the **Description** field or a notes field (such as **Audit Notes**, **Notices Text**, and others) exceeds 150 characters, a **Show more...** link is displayed to expand the row to show the entire text. You can then click **Show less...** to collapse the text. When you click the ⓘ icon next a **Component** or **License** value, a pop-up dialog shows read-only details about that item.
Actions	The following buttons and icons enable you to navigate and manage the history view.	
	↻	Refresh the history view.
	Page controls	Move to the next or previous page or to the first or last page in the view; or enter a specific page number in the **Page** field. Note that the default page size is 25 revision records. (Page size is based on the number of revisions, not updates.)
	Close	Exit the **Inventory History** window.

Inventory View

Code Insight enables you to view published inventory of open-source (OSS) or third-party components found across the projects in your Code Insight system. This inventory, displayed in a single scrollable window called the **Inventory** view, provides the means to make overall assessments of the OSS or third-party code used in your company's software deliverables.

Table 8-40 ▪ Inventory View

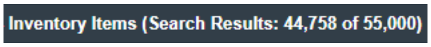
Category	Column/Field	Description
Search and filter fields and buttons	Use these fields and buttons (which display at the top of the **Inventory** view) to filter and modify the inventory list in the view. For your reference, the total number of filtered inventory items currently displayed compared to the total number of items in the full **Inventory** view is tracked in the **Inventory** view header:  Inventory Items (Search Results: 44,758 of 55,000)	

Table 8-40 ▪ Inventory View (cont.)

Category	Column/Field	Description
	Enter Inventory Name	Use this field to filter the inventory list by inventory name. Enter a string by which to filter the inventory names. If necessary, click the search icon next to the field to initiate search. To remove the string and restore the full list of inventory items, click the **X** in the field.
	Advanced Search button	Click this button to open the **Advanced Inventory Search** dialog. From this dialog, you can set search criteria (based on inventory attributes) by which to filter the inventory list. For details about the criteria available on this dialog, see Advanced Inventory Search Dialog.

Table 8-40 ▪ Inventory View (cont.)

Category	Column/Field	Description
	Context for the view (dropdown list)	From this dropdown list, select the major context for the **Inventory** view: ● **My Projects**— Show all published inventory across those Code Insight projects in which you are assigned a role. You might use this context to show areas where you need to provide review or remedial work, or you might want to review the overall state of inventory found in your projects. (This is the context enabled by default when you open the **Inventory** view.) ● **All Projects**—Show all published inventory across all projects in your Code Insight system. This context is helpful in visualizing trends in your company's use of open- source and third-party code in its software projects. ● **Select Project**—Show all published inventory for a selected Code Insight project. Since projects represent versions of a particular software product, this view allows you to see all inventory items for that product. Furthermore, you can also opt to list the inventory for all child projects of the selected project. These child projects represent modules used by your top-level product. You can directly link to the inventory item of the child project or to the child project itself. You can also view the parent hierarchy of the child project to understand the provenance of the inventory items. (See Including the Inventory of Child Projects on the Inventory View for details.) If you choose this option, a dialog is displayed from which to select a project. Once you choose the project, the inventory list is refreshed with the inventory for that project only.
	Change Project	(Displayed once a specific project is selected for **Select Project** in the previous field) Click this button to select a different project whose inventory you want to display in the **Inventory** view.

Table 8-40 ▪ Inventory View (cont.)

Category	Column/Field	Description
	Show All Items button	Click this button to remove all current criteria configured on the **Advanced Inventory Search** dialog and switch the focus of the **Inventory** view to show all projects. ***Note ▪*** *This button does not display if the* ***Inventory*** *view is already using the* ***All Projects*** *focus.*
	Include inventory items from child projects	If child projects have been identified for project currently in context in the **Inventory** view, select this option to refresh the view to include the inventory for these child projects. In this way, you can examine the inventory found across the project codebases for all parts of your software project, including its dependencies and sub-modules For more information, see Including the Inventory of Child Projects on the Inventory View. Note that by selecting to include inventory from child projects, all child-projects associated recursively to the current top-level project will be included in your inventory items list. Each child project is identified by the icon next to its name in the list.

Table 8-40 ▪ Inventory View (cont.)

Category	Column/Field	Description
Inventory columns	The following columns identify and provide information about each inventory item listed in the **Inventory** view. To manage column content, hover over the right side of a specific column header, and click its dropdown menu. From this menu, you can re-sort column values in ascending or descending order, as well as display or hide any column in the **Inventory** view. (By default, the **#Files** column is hidden.) ***Note ▪*** *Currently you can re-sort the values in the **Project**, **Inventory Name**, **Priority**, **#Files**, **Status**, **Created On**, and **Updated On** columns.By default, the **Inventory** view is sorted alphabetically by **Inventory Name** in ascending order.* To open a read-only version of the details for the given inventory item, click anywhere in the row for the item (except on linked text or a linked icon). A slide-out is displayed, showing most of the details that are also available for the item on its **Project Inventory Details** pane in the actual project. However, unlike the **Project Inventory Details** pane, the values on the slide-out are not editable. (While these values are read-only, certain ones are hyperlinked, enabling you to still explore and maintain the inventory item if you want.) For more information, see Opening a Read-Only Version of Inventory Details on the Inventory View. For a description of the inventory details available on the slide-out, see Project Inventory Details Pane. Otherwise, you can use links directly on the **Inventory** view to open an inventory item's associated project to examine the item within the context of its actual project and to edit its details as your permissions allow. See the Project and Inventory Name column descriptions.	
	Project	The name of the Code Insight project to which the given inventory item belongs. If a project is a child project of the current project, the icon displays next to the child project name. Click this icon to view the recursive hierarchy of the child project's parents. To open the project to its **Project Inventory** tab, click the hyperlinked project name. From here, you can explore and edit all published inventory (including the given inventory item) as your permissions allow. For more information, see Open the Associated Project to the List of All Inventory in the Project. You can sort the **Inventory** view by this column alphabetically in ascending or descending order.

Table 8-40 ▪ Inventory View (cont.)

Category	Column/Field	Description
	Inventory Name	The name of the inventory item in *component version (license)* format. To open the project to which the given inventory item belongs, click the hyperlinked inventory name. The project opens to the **Project Inventory Details** pane on the **Project Inventory** tab, providing access to all information available for the given inventory item within the project. From here you can explore and edit this inventory item as your permissions allow. For more information, see Open the Associated Project Directly to the Details for a Given Inventory Item. Alternatively, instead of opening the project, you can view a read-only version of the inventory details within the **Inventory** view. See the Inventory columns description. By default, the **Inventory** view is sorted by this column alphabetically in ascending order.
	Priority	The inventory priority of the item (**P1**, **P2**, **P3**, or **P4**). For more information about this attribute, see Inventory Priority. You can sort the **Inventory** view by this column in ascending (**High** to **Low**) or descending (**Low** to **High**) order.
	Component	The name and version of the open-source or third-party component on which the inventory item is based. For more information about the component, click ⓘ to open the **Component Details** window. This window shows publicly available details for the component as found in the Code Insight Data Library of third-party and OSS component information. When a component is not known, **N/A** is displayed.
	License	The license associated with the open-source or third-party component. For more information about the license, click ⓘ to open the **License Details** window. See License Details Window for a description of the available details. When a license is not known, the value **I don't know** is displayed.

Table 8-40 ▪ Inventory View (cont.)

Category	Column/Field	Description
	Vulnerabilities	A bar graph showing the count of known security vulnerabilities by severity color for the inventory item. Click the graph to view a list of these vulnerabilities and their CVSS details. For more information about security vulnerabilities, see Working with Security Vulnerabilities. The counts in this graph do not include vulnerabilities that are currently suppressed. If the inventory item has no known vulnerabilities, **None** is displayed.
	Tasks	Access to the open tasks for the inventory item: ● If open tasks exist for the inventory item, the ☰ icon is displayed. Click this icon to open a **Tasks** window, listing the open tasks specific to the inventory item. From here, you can view or edit details for each open task, close the task, or create new tasks for the inventory item if needed. ● If no open tasks exist for the inventory item, no icon is displayed.
	Alerts	Access to any security alerts for the inventory item. An alert is generated if the Electronic Update or Library Refresh detects a new security vulnerability for the inventory item since the last scan. ● If alerts exist for the inventory item, the 🔔 icon is displayed. Click this icon to open an **Alerts** window, listing the new security vulnerabilities and their CVSS information. From here, you can change the priority or status of the alert. See Managing Security Vulnerability Alerts for details. ● If no alerts exist for the inventory item, no icon is displayed.
	# Files	***Note ▪*** *This column is hidden by default. For instructions on how to display the column, see Focusing Column Content.* The number of codebase associated with the inventory item. You can sort the **Inventory** view by this column in ascending or descending numeric order.

Table 8-40 ▪ Inventory View (cont.)

Category	Column/Field	Description
	Status	The status of the inventory item: ● ✓ **Approved**—Approved for inclusion in the final notices of open-source and third-party components (such as a Bill of Materials or a similar document). ● ✕ **Rejected**—Rejected for inclusion in the final notices. ● ? **Ready for Review**—Not yet reviewed. You can sort the **Inventory** view by this column in ascending (**Ready for Review**, **Approved**, **Rejected**) or descending order (**Rejected**, **Approved**, **Ready for Review**).
	Created By	***Note ▪*** *This column is hidden by default. For instructions on how to display the column, see Focusing Column Content.* The creator of the inventory item as either: ● **System**—Code Insight automatically generated the item per one of these detection techniques (as designated in the **Notes** for the inventory item) during a scan: 　● High Confidence Custom Auto-WriteUp Rule 　● High Confidence Auto-WriteUp Rule 　● Medium Confidence Auto-WriteUp Rule 　● Automated Finding 　● Low Confidence Auto-WriteUp Rule 　● High Confidence MID Rule 　● Low Confidence MID Rule 　● Audit Import ● **<user_name>**—The first and last name of the user who manually created the item. The name is hyperlinked, enabling you to send an email to the user. You can sort the **Inventory** view by this column alphabetically in ascending or descending order.

Table 8-40 ▪ Inventory View (cont.)

Category	Column/Field	Description
	Created On	The date on which the inventory item was created. You can sort the **Inventory** view by this column in ascending (earliest to latest) or descending (latest to earliest) order.
	Updated On	The date on which the inventory item was last updated. If the item has not been updated since its creation, its **Updated On** date is the same as its **Created On** date. You can sort the **Inventory** view by this column in ascending (earliest to latest) or descending (latest to earliest) order.

Jobs Queue

The **Jobs** queue (in the **Jobs** window) enables you to monitor all scheduled and active jobs and view historical jobs in your Code Insight system. The queue is in a navigable grid, enabling you locate and view details for jobs more easily. The following table describes the queue columns that provide details for each job. It also describes the fields that help you navigate the queue or control the queue view.

For complete information on how to access and search the **Jobs** queue, see Monitoring the Code Insight Jobs Queue.

Note *▪ The "the specified project" in the following descriptions refers to the project identified in the* ***Project Name*** *column for the job.*

Table 8-41 ▪ Jobs Queue

Field/Column	Description
Show jobs for	The filter that configures the **Jobs** queue to show only those jobs that were added to the queue in the last number of days specified, enabling you to monitor a manageable number of jobs. You can also select to view all jobs. Select one of these values: ● **15 days**—Show only jobs queued in the last 15 days. (Default) ● **30 days**—Show only jobs queued in the last 30 days. ● **All**—Show all jobs.

Table 8-41 ▪ Jobs Queue

Field/Column	Description
The following columns in the **Jobs** queue describe each job.	
Job ID	The internally generated ID for the job. By default, the **Jobs** queue is sorted in descending order by this column. However, you change the sort to ascending order or choose to sort by another column. For more information, see Sorting the Jobs Queue.
Job Type	The type of event that triggered the job: ● **Project Branching**—The branching of the specified project to another project. ● **Project Copy**—The copying of the specified project to another project. ● **Project Scan**—A Code Insight scan of the specified project. ● **Project Re-Scan**—A rescan of the specified project. ● **PDL Update**—An Electronic Update of the local Code Insight Data Library. ● **Library Refresh**—A daily Code Insight Data Library refresh of new vulnerability alerts generated since the previous day. ● **Project Deletion**—The deletion of the specified project. ● **Export to SBOM Insights**—The export of a project's inventory data to SBOM Insights. ● **Update Notices**—The automatic update of the Notices content across all inventory in a project. ● **Project Import**—The import of exported project data into a project. ● **Project Export**—The export of project data to a JSON file so that it can be imported into another project. ● **Remote Scan**—The processing of the scan results sent to Code Insight by a Code Insight scan-agent plugin residing on a remote server. Currently, the **Jobs** queue supports the processing of scan results from the generic or Docker Images scan-agent plugin only. For all other scan-agent plugins, the results-processing phase begins as soon as Code Insight receives the results; no job is added to the **Jobs** queue to run in the background. For more information, see Support for Processing Remote Scan Results in the Background. **Report-Audit Report**—The generation of an Audit report for the specified project. ● **Report-Notices Report**—The generation of a Notices report for the specified project. ● **Report-Project Report**—The generation of a Project report for the specified project.

Table 8-41 ▪ Jobs Queue

Field/Column	Description
	● **Report-<customReportName>**—The generation of a custom report for the specified project.**Apply Policy - Global**— ● **Apply Policy - Project**—The application of current criteria in a review policy profile to inventory in a specified project associated with the profile. ● **Apply Policy - Global**—The application of current criteria in a given review policy profile to inventory across all projects associated with the profile. You can use this column to filter the **Jobs** queue to one of the job types. For more information, see Filtering the Jobs Queue by Column.
Project Name	The name of the project associated with the job. If you click the hyperlinked name, the project is opened to its **Summary** tab. (If the project is private, only administrators and those users assigned roles on the project can open it. All other users receive the message “Unable to access the project”.) Note the following: ● If the project has been deleted, this column shows ***Deleted Project*** (with no hyperlink). However, in most cases, the name of the deleted project is still referenced in the **Details** column. ● For a **Project Copy** job, this column shows the name of the source project from which data is copied. The target project to which the data is copied is identified in the **Details** column. ● For a **PDL Update**, **Library Refresh**, or **Apply Policy - Global** job, this column shows **N/A** since the job is a system event (and not related to one specific project). You can use this column to filter the **Jobs** queue to a specified project name string. For more information, see Filtering the Jobs Queue by Column.

Table 8-41 ▪ Jobs Queue

Field/Column	Description
Status	The current status of the job: ● **New**—The job has just been added to the queue, and its current status is pending. ● **Active**—The job is now executing. ● **Canceled**—A user manually canceled the job while it was in a **Scheduled** state. ● **Completed**—The job has successfully finished. ● **Failed**—The job encountered an error. Usually a description of the error is provided in the **Errors** column for the job. ● **Scheduled**—The job is waiting for a scan or rescan to complete for one or more projects other than the one specified for the job. ● **Terminated**—A system timeout or loss of connection caused the job to end prematurely. This status is also used if the job was canceled while in an **Active** state. ● **Waiting on update**—The job is waiting for an Electronic Update to complete on your Code Insight instance. ● **Waiting on library refresh**—The job is waiting on a Library Refresh to complete on your Code Insight instance. You can use this column to filter the **Jobs** queue to a single status. For more information, see Filtering the Jobs Queue by Column.
Scan Server	The alias of the Scan Server that performed the scan (even if the Scan Server has since been disabled). For jobs related to the Code Insight Core Server, the value is **N/A**. You can use this column to filter the **Jobs** queue to one or more Scan Server aliases. You cannot filter using the **N/A** (Core Server) value. For more information, see Filtering the Jobs Queue by Column.
Triggered By	The login name of the user who initiated the event that triggered the job. Clicking the hyperlinked name lets you can send an email to the user. You can use this column to filter the **Jobs** queue to a specified user-name string. For more information, see Filtering the Jobs Queue by Column.

Table 8-41 ▪ Jobs Queue

Field/Column	Description
Queued On	The date and time at which the job was added to the queue. You can sort the **Jobs** queue by this column. For more information, see Sorting the Jobs Queue. *Note ▪ A* ***Remote Scan*** *job is added to the queue when the scan results from a scan-agent plugin residing on a remote server are successfully sent to Code Insight for processing. Currently, the* ***Jobs*** *queue supports the processing of scan results from the generic or Docker Images scan-agent plugin only. For all other scan-agent plugins, the results-processing phase begins as soon as Code Insight receives the results; no job is added to the* ***Jobs*** *queue to run in the background.*
Activated On	The date and time at which the job began to execute.
Completed On	The date and time at which the job ended. (Check the **Status** column to determine the final status of the job when it ended.) You can sort the **Jobs** queue by this column. For more information, see Sorting the Jobs Queue.
Details	In general, a description of the job and, if applicable, the name of the project on which it was run. However, for the following jobs, the column provides different information: ● **Project Copy** job—Details include both source project from which data was copied and the target project to which data was copied. (The source project is also listed under **Project Name**.) ● **Apply Policy - Global** and **Apply Policy - Project** jobs—The description provides the name of the review policy profile applied to inventory.
Errors	Details of the error(s) that caused a job to end with a “Failed” status. If no errors were generated for the job, this value is empty.
These buttons at the bottom of each page in the **Jobs** queue enable you to navigate easily between pages. A page holds a maximum of 25 jobs.	
« »	Navigate to the first or last page in the queue.
‹ ›	Navigate to the previous or next page in the queue.
Page 2 of 19	Navigate to the specified page in the queue.
These buttons refresh the **Jobs** queue or close the **Jobs** window.	
↻	Refreshes the **Jobs** queue with the latest job information. All search mechanisms currently applied to the queue are retained.

Table 8-41 ▪ Jobs Queue

Field/Column	Description
Close	Closes the **Jobs** window. When you reopen the window, the **Jobs** queue is displayed in its default view. (Search mechanisms applied previously to the queue are not retained.)

LDAP Tab

Code Insight supports user authentication and authorization through LDAP (Lightweight Directory Access Protocol). The **LDAP** tab on the **Administration** page configures the synchronization of user identification data from LDAP to Code Insight, thus enabling LDAP user authentication for Code Insight. For detailed information about the fields on this tab and about the configuration in general, see “Configuring Code Insight for LDAP” in the *Code Installation and Configuration Guide*.

The tab contains the following columns and fields:

Table 8-42 ▪ LDAP Tab

Category	Column/Field	Description
LDAP enablement	This option enables the use of LDAP for your Code Insight system. When LDAP is enabled, the settings used to configure Code Insight for LDAP are made available for editing on this tab. You can use this option to turn off LDAP whenever necessary.	
	Enable LDAP	Select **Yes** or **No** to determine if LDAP will be used for user authentication. The default is **No**.

Table 8-42 ▪ LDAP Tab (cont.)

Category	Column/Field	Description
LDAP Connection Details	These settings configure the Code Insight connection to the LDAP server. This connection is required for each synchronization process of LDAP user information to Code Insight and for authentication each time a user logs into Code Insight.	
	LDAP URL	Specify the URL of the LDAP server in the following format: `ldap://<ldap_server_host>:<ldap_port>` where `<ldap_server_host>` is either the hostname or IP address of the LDAP server; and `<ldap_port>` is the port on which the server listens for requests. The following is an example URL, which uses the standard LDAP server port 389: **`ldap://acme.com:389`** If using SSL to provide data encryption security for user information passed over the network, specify the `ldaps://` protocol with the port 636, which is the default dedicated port for SSL: **`ldaps://acme.com:636`** ***Note ▪*** *When the LDAP directory service is Active Directory, requests from users located outside the global catalog's base domain will fail to authenticate if you use the port specified above. This occurs because requests sent to the default LDAP port 389 (or 636 if SSL is used) search for objects only within the global catalog's base domain. To authenticate users from outside the base domain, change the LDAP port to 3268 (or 3269 if SSL is used). Requests sent to this port search for objects in the entire forest.*

Table 8-42 ▪ LDAP Tab (cont.)

Category	Column/Field	Description
	Authentication Type	Select the type of LDAP authentication used to establish a connection with the LDAP server: ● **Anonymous**—Code Insight will establish a connection with the LDAP server without the use of user credentials. (When this option is selected, the **LDAP Username** and **LDAP Password** fields in this section are disabled.) This authentication type is generally used for testing purposes. ● **Authenticated**—Code Insight requires the user credentials provided in the **LDAP Username** and **LDAP Password** fields to authenticate and establish a connection with the LDAP server.
	LDAP Username	Depending on your LDAP setup, enter either of the following to identify the user used connect to the LDAP server: ● The user's login ID, such as `mburns` ● The user's Distinguished Name (DN), such as: `CN=Monty Burns,OU=usa,DC=acme,DC=com` For more information about providing the DN, see "Distinguished Name for an Object" in the *Code Installation and Configuration Guide*. This identification, along with the associated password (see the next field), is used to authenticate the connection to the LDAP server. Note that the user must have READ permissions to query the LDAP server (and therefore does not need to be an administrator). This field is disabled if **Anonymous** is selected for **Authentication Type**.
	LDAP Password	Enter the password associated with the user specified for **LDAP Username**. This field is disabled if **Anonymous** is selected for **Authentication Typ**e.

Table 8-42 ▪ LDAP Tab (cont.)

Category	Column/Field	Description
LDAP Query Details	The following fields define the query that identifies the subset of users on the LDAP server to be synchronized to Code Insight. This query is used for the initial synchronization process and for each subsequent synchronization performed per the **LDAP User Sync Frequency** value.	

	LDAP Base	Specify the Distinguished Name (DN) of the LDAP base domain in the Directory Information Tree (DIT) on your LDAP server. This domain is the top-level directory to which all other objects in the directory structure belong; it typically represents your organization. The base domain is identified by domain controller objects (DCs), which make up its DN. For example, the base domain in the example DIT in Figure 2-1 is the following:
		`DC=acme,DC=com`
		In some cases, a sub-domain can be a part of the base domain:
		`DC=software,DC=acme,DC=com`
		For more information, see “LDAP Base” in the *Code Installation and Configuration Guide*.

	LDAP Search Base	Specify the DIT directory, relative to the LDAP base directory, under which you store all Code Insight objects on the LDAP server and from which you search for Code Insight users. In reference to the example DIT in Figure 2-1, if you enter `OU=usa` for the search base, all searches for user information will be performed below the directory “usa”. (LDAP internally identifies the DN for this directory as the **LDAP Base** + **LDAP Search Base** value.) If you leave this field blank, the search is performed at the LDAP base level. For more information, see “Setting Up a User Search” in the *Code Installation and Configuration Guide*.
	LDAP Search Query	Specify the search query used to retrieve the users from **LDAP Search Base** directory to synchronize to Code Insight. Each attribute in a query is listed in parenthesis in the format (*attribute=value*), such as in the following, which searches for only those users belonging to the “engineering” group under the “usa” node: `(&(objectClass=person)(memberOf=CN=engg,OU=usa,DC=acme,DC=com))` For other search query examples, see “Setting Up a User Search” in the *Code Installation and Configuration Guide*.

Table 8-42 ▪ LDAP Tab (cont.)

Category	Column/Field	Description
	Use Paging	Select **Yes** if the LDAP server has paging enabled for synchronization results. If you select **Yes**, the **LDAP Page Size** field is enabled, enabling you to customize the page size. Select **No** if the server does not have paging enabled. If you select **No**, the server sends 1000 elements per page by default unless this behavior is changed at the organization level on the LDAP server.
	LDAP Page Size	Indicate the page size you want for the synchronization results. The default page size is 1000 elements.
	LDAP User Sync Frequency	Specify the frequency at which Code Insight will synchronize user data with the LDAP server: ● **Never**—Select this option to disable the automatic user synchronization. A synchronization occurs only if the user clicks the **Sync Now** button. For all other values, automatic user synchronization is enabled per the configured frequency. (This is the default value.) ● **Hourly**—Enter an integer value representing the number of hours between user synchronizations. ● **Daily**— Select a time at which the user synchronization will run every day. ● **Weekly**—Select a day of the week and a time of the day when the user synchronization will run each week.
	Search Sub-tree	Select this checkbox to enable deep searches through the subtree of the path defined by **LDAP Base** + **LDAP Search Base**. Note that, while helpful in locating users in certain cases, a deep search can negatively affect performance (and therefore, by default, is not enabled). For more information, see “Setting Up a User Search” in the *Code Installation and Configuration Guide*.

Table 8-42 ▪ LDAP Tab (cont.)

Category	Column/Field	Description
LDAP User Property Mappings	The following information maps LDAP attribute labels to their corresponding labels in Code Insight (the field names shown below). These mappings are used for LDAP synchronization to Code Insight and for user authentication each time a user logs into Code Insight.	
	Login	Enter the user attribute label on your LDAP server corresponding to the user **Login** field in Code Insight. This is the same attribute that the user will use to log into Code Insight.
	First Name	Enter the user attribute label on your LDAP server corresponding to the user **First Name** field in Code Insight.
	Last Name	Enter the user attribute label on your LDAP server corresponding to the user **Last Name** field in Code Insight.
	Email	Enter the user attribute label on your LDAP server corresponding to the user **Email** field in Code Insight. ***Note ▪*** *Only those users with a valid email address specified as a user attribute on the LDAP server will be synchronized. Therefore, ensure that you have entered the correct label here for the email attribute on your LDAP server and that each user has valid email for this attribute on the server. See "Setting Up a User Search" in the "Code Installation and Configuration Guide" for more information.*
	Login Filter	Specify a filter for the user-login search performed in the LDAP search base location. For example, the value `(sAMAccountName={0})`, when used against the **LDAP Search Query** results, searches for each entry where the `sAMAccountName` is equal to the user login name.

License Details Window

The **License Details** window displays the details and license text as found in the Code Insight Data Library for a selected license. You can access this window by clicking the information icon ⓘ displayed next to a license name. The window has the following fields.

Table 8-43 ▪ License Details Window

Tab		Description
General Information tab	The attributes of the current license.	
	Id	The internal identification number generated for the license in the Code Insight Data Library.

Table 8-43 ▪ License Details Window (cont.)

Tab		Description
	Name	The name of the license, along with its short name (usually the SPDX short identifier) in parentheses.
	Family	The family of licenses to which this license belongs, if applicable. The following are examples of family licenses: Apache, BSD, EPL, MIT, and MPL.
	Priority	The priority ranking of the license as determined by Code Insight. For more information, see License Priority.
	URL	The URL location where the license is available.
	Description	A short description of the license.
	External ID	The ID that corresponds to the license in an external system (such as your site's license-tracking system). This mapping enables you to easily locate the license in the external system. If the license is not mapped to an external ID, this field is blank.
	Category	Currently not in use.
	Policy	The **Yes** or **No** indicator specifying whether a Code Insight policy is defined for the license. (License policies are used to approve or reject inventory items when they are published.)
	Custom License	The **Yes** or **No** indicator specifying whether this license was manually created in Code Insight.
	Family License	The **Yes** or **No** indicator specifying whether this is a family license from which other OSS licenses are derived. The following are examples of family licenses: Apache, BSD, EPL, MIT, and MPL.
	Commercial	Currently not in use.
	Copyleft	Currently not in use.
	Free Software License	The **Yes** or **No** indicator specifying whether the license is considered to be a free software license.
	GPL V2 Compatible	The **Yes** or **No** indicator specifying whether the license is compatible with GPL v2.0.
	Patent License Grant	Currently not in use.

Table 8-43 ▪ License Details Window (cont.)

Tab	Description
License Text tab	The complete text of the selected license as stored in the Code Insight Data Library. This text represents the external forge license text.

Lookup Component Window

The **Lookup Component** window is displayed when you click **Lookup Component** within the context a inventory item, with the purpose of letting you search for a new component-version-license instance to associate with the inventory item. The search is performed against the Code Insight Data Library to locate components that meet your criteria. The search results in a list of components, each component displayed with a set of details and a list of its available version-license instances.

Once you locate the desired component, you can select the appropriate version-license combination to associate with your inventory item. Alternatively, you can create your own instance. (Any custom version-license instances created for a component are made available at the system level for association with inventory in other projects.) If no component meets your criteria for the inventory item, the **Lookup Component** window provides access to a feature that lets you create a custom component.

Table 8-44 ▪ Lookup Component Window

Category		Column/Field	Description
Search controls	Use one of these fields to enter the criterion by which to search for a component to associate with an inventory item or to serve as a basis for creating a custom component.		
	Search by	Select the method by which to search components or to create a new component.	
		Keyword	Select this option to search by component name. In the **Keyword** field, enter a single string within the component name. ***Note ▪** The search is case-insensitive and thus filters to all component names containing the **Keyword** criterion, no matter the upper or lower case used in the criterion or in the actual component name.* If you are creating a new component, the string is used to pre-populate certain fields in the **New Custom Component** window. See the **Create New Component** description.

Table 8-44 ▪ Lookup Component Window (cont.)

Category	Column/Field	Description
	URL	Select this option to search by the URL of the third-party forge where the component is found. For the **URL** value, enter the complete forge path, such as **https://github.com/jquery/jquery**, or a string in the path, such as **jquery.** ***Note ▪*** *The search is case-insensitive, so the results will include all components with a matching forge path or path string (whichever criterion you entered in the* ***URL*** *field), no matter the upper or lower case used in the criterion or in the actual component path.* If you are creating a new component, the URL is used to pre-populate certain fields in the **New Custom Component** window. See the **Create New Component** description.
	Forge	Select this option, and then select the forge (and project repository) by which to search components. If you are creating a new component, the selected forge is used to pre-populate certain fields in the **New Custom Component** window. See the **Create New Component** description.
	Search	Click this button obtain the search results.
	Create New Component	Click this button to open the **New Custom Component** window. Certain fields in this window are pre-populated with values based on the criterion you entered on the **Lookup Component** window. For information on creating a custom component, see Creating and Editing Custom Components.

Table 8-44 ▪ Lookup Component Window (cont.)

Category	Column/Field	Description
Search results	The results of the search is a list of components, each component with a set of details (see **Component details**) and a list of available version-license instances to which you can associate with the current inventory item (see **Version-license instances**). The following describes the information shown for each component listed.	
	Component details	The details for a given component can include the component's product logo, vendor content describing the component, and a link to the actual OSS or third-party product. It also includes the following component details from the Code Insight Data Library.
	Component	The name of the OSS or third-party component and its internal ID, as identified in the Code Insight Data Library.
	Possible Licenses	License candidates that can be associated with this component.
	Custom Component	The **Yes** or **No** value, indicating whether the component is custom (created by a user) or provided as part of the Code Insight Data Library.
	CPE	The list of CPE names—from the National Vulnerability Database—that are mapped to the component. CPE (Common Platform Enumeration) is a structured naming scheme that includes the component's vendor and product names in the following format: **cpe://<part>:<vendor>:<product>** where <part> is either **a** (applications), **h** (hardware platforms), or **o** (operating systems). ***Note ▪*** *The data provided represents only the part, vendor, and product; the version information is truncated from the CPE string.*

Table 8-44 ▪ Lookup Component Window (cont.)

Category	Column/Field	Description
	Version-license instances	The information for each component includes a list of its available version-license instances. (To toggle between showing or hiding the list, click **Show Versions/ Instances** or **Hide Instances**.) From this list, you can do any of the following: ● Select a given version-license instance to associate with the current inventory item. ● Select a new license for a given instance. ● Register a new version-license instance for the component. ● Designate that the license newly selected for an existing instance (or for one being registered) be mapped to all future inventory created by the system for the component version. This type of license is known as a "user-preferred license". Instances mapped to a user-preferred license are displayed with the 👤✓ icon. (See Specifying a User-Preferred License Mapping for more information.) ● If the component is custom, edit the component as needed. A bar graph is included with each instance to show its current security-vulnerability counts by severity level (if any). See Security Vulnerabilities Associated with Inventory for details.
	Use This Instance	Click this button to associate the version-license instance with the inventory item you are currently creating or editing. You are directed back to the inventory item, now showing the new component-version-license association. You can also select a different license for the instance from the **Selected License** dropdown. See the **Register New Instance** description below for further details.

Table 8-44 ▪ Lookup Component Window (cont.)

Category	Column/Field	Description
	Register New Instance	Click this button to add a new version-license instance to the component. From the **Version** dropdown list, select an existing version associated with this component (as stored in the Code Insight Data Library), or create your own version. From the **Selected License** dropdown list, select a license to associate with this component. See Specifying a User-Preferred License Mapping for information about what happens depending on the type of license you select. ***Note ▪** You cannot create a custom license from the **Lookup Component** window to associate with a component version. However, you can create a custom license for the inventory item after you have selected an instance or when you editing the item. Alternatively, you can create custom licenses from the **Policy Details** window or from the **Licenses** tab on the **Global Component & License Lookup** page. For more information, see Creating a Custom License.* New instances are made available at the global level for use by inventory in other projects.
	Edit Custom Component	(Available if the component is custom) Click this button to open the **Edit Custom Component** window to update the component properties. For information on editing a custom component, see Creating and Editing Custom Components.

See Also
Using “Lookup Component” to Search for Components to Associate with Inventory
Specifying a User-Preferred License Mapping
Security Vulnerabilities Associated with Inventory
Creating and Editing Custom Components
Creating and Editing Custom Licenses

Policy Details Window

The **Policy Details** window is displayed when you select to create, edit, or view a policy profile from the Policy Page. A policy profile contains a set of policies used to perform an automatic review of inventory items upon their publication. Each policy defines criteria based on OSS or third-party component versions, licenses, or security vulnerabilities. Inventory items that meet any of the profile’s policy criteria can be automatically approved or rejected (or flagged for a manual review). Any one policy that results in a rejection causes the inventory item to be rejected despite any approvals.

The following topics explain how to use the **Policy Details** window to define policies within a policy profile:

- Policy Fields
- Fields Specific to Maintaining License Policies
- Interface for Adding Reviewer Content to Policies
- Impact on Policies When Code Insight's CVSS Configuration Changes

Only users who have Policy Manager permissions can create, edit, or copy a policy profile. All other users can view the policy profile only.

See Also
Policy Page
License Details Window
Lookup Component Window
Managing Policies to Automatically Review Inventory

Policy Fields

The Policy Details Window provides the following fields to define policies that automatically approve or reject an inventory item when it is published. If no policy applies to an inventory item, the item's status is **Not Reviewed**, requiring the item to be reviewed manually. Only users with Policy Manager permissions can edit these fields.

When you select to *view* a policy profile from the Policy Page, the following fields are read-only. Any user can view a profile, including those users who do not have Policy Manager permissions.

Table 8-45 ▪ Policy Details Window

Category	Column/Field	Description
General	These fields identify the policy profile you are creating or editing.	
	Name	The name of the policy profile that you are editing or copying. If you are copying a profile, the name of the copy will be `Copy of` *`selected_policyProfile`*, where *`selected_policyProfile`* is the name of the original profile. To change the name of the profile copy, type over the generated name with the new name in this field.
	Description	The policy profile description, if it exists. You can edit or add a description.
	Created	(Available in the Edit and View versions of the profile) The name of the user who created the policy profile, and the date and time the profile was created. You can click the hyperlinked name to send an email to the user who created the profile.
	Updated	(Available in the Edit and View versions of the profile) The name of the user who last updated the policy profile, and the date and time the profile was updated. You can click the hyperlinked name to send an email to the user who updated the profile.

Table 8-45 ▪ Policy Details Window (cont.)

Category	Column/Field	Description
Vulnerabilities	The following define policies that automatically approve or reject inventory items with security vulnerabilities. ***Note ▪*** *These policies ignore suppressed vulnerabilities when making decisions whether to automatically approve or reject published inventory items. (A change in policy due to the suppression of a vulnerability does not change the existing approval/rejection status of an inventory item unless the item is manually recalled and then republished.)*	
	📄	Click this icon next to a vulnerability policy to provide (or edit or view) meaningful content intended for inventory reviewers about the impact of the given policy. For example, this content might include reasons why the specific security vulnerabilities identified in the policy pose a risk to your intellectual property. This information is propagated to those project inventory items that are actually rejected by the policy, providing reviewers with context about the inventory's status. For more information, see Interface for Adding Reviewer Content to Policies.
	Only auto-approve inventory items if there are no associated security vulnerabilities	Select this checkbox to have Code Insight skip any matching license-based or component policies if the inventory item has any associated security vulnerabilities.
	Reject inventory items if any associated security vulnerabilities have a CVSS score above <score>	Select this checkbox to have Code Insight automatically reject any inventory items with any associated security vulnerabilities that have a CVSS score above the value you enter. (The scores available for this field are based on the CVSS version currently used by Code Insight. For information, see Security Vulnerabilities Associated with Inventory.) This policy takes precedence over any other automated approval policy. ***Note ▪*** *If the Code Insight System Administrator changes the CVSS version used by Code Insight, the value you selected for this field might change. See Impact on Policies When Code Insight's CVSS Configuration Changes for details.*

Table 8-45 ▪ Policy Details Window (cont.)

Category	Column/Field	Description
	Reject inventory items if any associated security vulnerabilities have a severity equal to or higher than <severity level>	Select this checkbox to have Code Insight automatically reject any inventory items with any associated security vulnerabilities that have a severity equal to or higher than severity you select. (The severities available for this field are based on the CVSS version currently used by Code Insight. For information, see Security Vulnerabilities Associated with Inventory.) This policy takes precedence over any other automated approval policy. ***Note ▪*** *If the Code Insight System Administrator changes the CVSS version used by Code Insight, the value you selected for this field might change. See Impact on Policies When Code Insight's CVSS Configuration Changes for details.*
Licenses	The following fields describe and manage the policies that automatically approve or reject inventory associated with a given license.	
	Add License	Click this button to add a new license policy based on a selected license and inventory usage criteria. (The **Edit** [or **Add**] **License and Usage Criteria** window is opened to enable you to do this.) See Fields Specific to Maintaining License Policies for further details. Once you create the license policy, its entry is added to the **Licenses** list. For the entry, you can then select the review status (under **Action**) that this policy automatically assigns an inventory item if the policy's criteria are met.
	📄	Click this icon to the left of each license policy to provide (or edit or view) meaningful content intended for inventory reviewers about this policy. For example, the content might list requirements for using the licenses identified in the policy's criteria or reasons why these licenses pose a legal risk. This information is then propagated to those project inventory items that are actually approved or rejected by the policy, providing reviewers with context about the inventory's status. For more information, see Interface for Adding Reviewer Content to Policies.
	ⓘ	Click this icon to the left of each license policy to open the License Details Window, enabling you to view information about the license, including its attributes and license text.

Table 8-45 ▪ Policy Details Window (cont.)

Category	Column/Field	Description
	Licenses (list)	The list of license policies (in a grid format) currently used by this profile for automatically reviewing inventory items. Each license policy entry contains the license name, inventory usage criteria that can impact the obligations incurred by the use of the license, and actions you can perform on the policy. • **Name**—The name of the license on which the policy is based. The following read-only criteria are currently defined for the given license policy and describe how a software package developed in your organization uses the OSS or third-party component associated with an inventory item. (This usage can have an impact on your license obligations and conditions of use.) To define or edit these criteria for a license policy, see for Fields Specific to Maintaining License Policies. • **Distribution type**—The criterion specifying how the OSS or third-party component associated with the inventory item is distributed with your software package. • **Linking**—The method that your software package uses to link to libraries in the OSS or third-party component associated with the inventory item. • **Modified**—The criterion specifying whether code from the OSS or third-party package has been modified for use by your organization. The following field specifies the review status automatically assigned to inventory items based on their meeting the criteria for this license policy: • **Action**—Select one of the following to indicate what review status to automatically assign an inventory item that meets the criteria in this license policy: ◦ **Approve** ◦ **Reject** ◦ **No Action** (New inventory is assigned the **Not Reviewed** status. The status for existing inventory remains as is.) The following icons at the right of each license policy are used to manage the policy: • ✕ **(delete)**—Click this icon to delete the license policy. • ✎ **(edit)**—Click this icon to edit the license policy criteria. See Fields Specific to Maintaining License Policies.

Table 8-45 ▪ Policy Details Window (cont.)

Category	Column/Field	Description
Components	The following fields define and manage policies that automatically approve or reject inventory based the component version associated with the inventory.	
	Add Component	Click this button to select a component on which to create the policy, or create a new component from the **Lookup Component** window. (See Lookup Component Window for information about how to use this window.) Once you select a component, its entry is added to the **Components** policy list.
	🗎	Click this icon to the left of each component policy to provide (or edit or view) meaningful content intended for inventory reviewers about this component. For example, this content might include “need to know” information about why the component versions identified in the policy pose a risk. This information is then propagated to those project inventory items that are actually approved or rejected by the policy, providing reviewers with context about the inventory’s status. For more information, see Interface for Adding Reviewer Content to Policies.
	ⓘ	Click this icon to the left of each component policy to open the Component Details Window window, enabling you to view relevant information about the component, including its forge, possible licenses, CPE names, and more.

Table 8-45 ▪ Policy Details Window (cont.)

Category	Column/Field	Description
	Components (list)	The list of current components and versions (in a grid format) currently used as criteria for automatically reviewing inventory items. ● **Name**—The name of the component. ● **Versions**—Select a specific version or a range of versions for the given component. (The **Versions from** and **to** drop-down lists are populated with available versions for the component.) Here are some example ways to specify a version or version range: ● To enter a specific version, select the same version in the **Versions from** and **to** fields. ● To enter an explicit range, select a minimum version in the **Versions from** field and the maximum version in the **to** field. ● To specify any version for the given component, select the wild card * in both **Versions from** and **to** fields. ● To specify any version up to a specific version, enter the wild card * in the **Version from** field and the maximum version in the **to** field. ● To specify any version after a specific version, select the specific version in the **Versions from** field and the wild card * in the **to** field. The **unknown** option applies to certain components that were collected without a version value. To specifically handle unknown versions, set both **Versions from** and **to** fields to **unknown**. ● **Action**—Select one of the following to indicate what status to automatically assign to inventory items associated with this component-version: ● **Approve** ● **Reject** ● **No Action** (same as the **Not Reviewed** inventory status, thus requiring a manual review) Click ✖ to the right of each component to delete the component from the policy.
Actions	These actions manage the entire policy profile.	
	Save	Click to save the changes you have made to this policy profile.
	Close	Click to close the **Policy Details** window. If you have made changes the profile, be sure that you have clicked **Save** before closing the page; otherwise, changes are lost.

Fields Specific to Maintaining License Policies

On the Policy Details Window window, when you click **Add License** to create a license policy or click the ✎ icon (at the end of a license policy entry) to edit a license policy, the **Add** (or **Edit**) **License and Usage Criteria** window is displayed. From here you can add or modify the following criteria used to define the license policy.

Note that a license policy must be unique. That is, you cannot have two license policies for the same license when all their usage criteria are the same (such as all are **Any**). However, you can have two license policies for the same license if they define different usage criteria.

Table 8-46 ▪ License and Usage Criteria Used in a License Policy

Category	Column/Field	Description
Select License	The license on which the policy is based.	
	License	Do either: ● Select the license from the dropdown list of available licenses. ● Click **Create Custom License** to create a license to assign to this policy. See the next description.
	Create Custom License	Click this button to open the **Create Custom License** window and create your own license. See Step 2: Create the Custom License for instructions. Once you save the custom license, you are returned to the current **Add** (or **Edit**) **License and Usage Criteria** window, where the custom license has now been added to the **License** dropdown list and is in focus for your immediate selection.
Select Usage Criteria	These properties are defined for an inventory item to describe how a software package developed in your organization *uses* the OSS or third-party component associated with the inventory item. Because usage can have an impact on your license obligations and conditions of use, the following usage properties are available as criteria in a given license policy.	
	Distribution Type	Specify the criterion that identifies how the OSS or third-party component associated with the inventory item is distributed with your software package. Distribution type can affect license priority and your license obligations. ● **Internal**—The component is distributed internally only (for example, as an internal test framework included in the codebase but not distributed publicly with the software package). ● **External**—The component is a separate entity from your software package. It might be shipped as a separate component along with the software package or deployed through some method, such as a private cloud at the customer site. ● **Hosted**—The component is hosted in your company's data center (for example, as a SAAS application) ● **Any**—The policy ignores this criterion.

Table 8-46 ▪ License and Usage Criteria Used in a License Policy (cont.)

Category	Column/Field	Description
	Linking	Specify the criterion that identifies how your software package links to the OSS or third-party component libraries. This method can affect license priority and obligations. ● **Not linked**—The software package uses no links to the component libraries. ● **Statically linked**—The component libraries are included in the software materials and thus linked statically. ● **Dynamically linked**—The component libraries are brought in at runtime. ● **Any**—The policy ignores this criterion.
	Modified	Specify the criterion that identifies whether code from the OSS or third-party package has been modified for use by your organization. ● **No**—The component software has not been modified. ● **Yes**—The component software has been modified. ● **Any**—The policy ignores this criterion.
Actions	These actions manage the license policy creation or update.	
	Add	Click this button to add the license policy to the **Licenses** list on the **Policy Details** window. Once added, you can select the review status that the policy automatically assigns when an inventory item meets the policy criteria.
	Save (for updates)	Click this button to save the policy changes and return the to the **Policy Details** window.
	Cancel	Click this button to discard any policy changes and return to the **Policy Details** window.

Interface for Adding Reviewer Content to Policies

When reviewers (and other users) examine published inventory that has been approved or rejected automatically by the Code Insight policy, they likely do not know or have access to the policy that resulted in the approved or rejected inventory. Without this information, they might not know what factors were involved in the rejection of inventory, what issues need to be addressed for rejected inventory, or what guidelines or special notes are available for approved inventory.

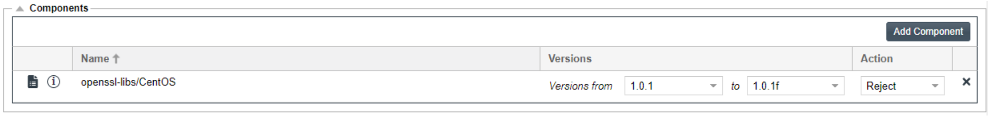
From the Policy Details Window, users with Policy Manager permissions can provide such information for reviewers by adding guidance content for any given policy in the policy profile currently open in the window. (To add this information, simply click 🗎 at the left of the policy row, and enter the content in the **Usage Guidance** pop-up.) Then, if an inventory item is automatically approved or rejected by the policy, this content is propagated to the **Usage Guidance** pane for the item on the **Project Inventory** tab, providing reviewers with context about the inventory's status. (While users can generally edit content in the **Usage Guidance** pane for project inventory, they cannot edit the specific content propagated from policies to this pane.)

Refer to the following topics for more information:

- Usage Guidance Scenario
- Samples of Usage Guidance Content

Usage Guidance Scenario

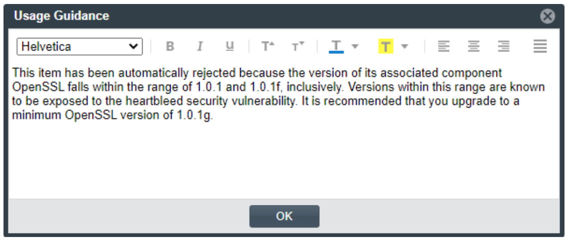
Suppose you define the following policy for a component version on the **Policy Details** window.



Components

Name	Versions	Action
openssl-libs/CentOS	Versions from 1.0.1 to 1.0.1f	Reject

You also provide the following content in the **Usage Guidance** pop-up for the policy. (Click 🗎 to the left of the policy to open the pop-up.)

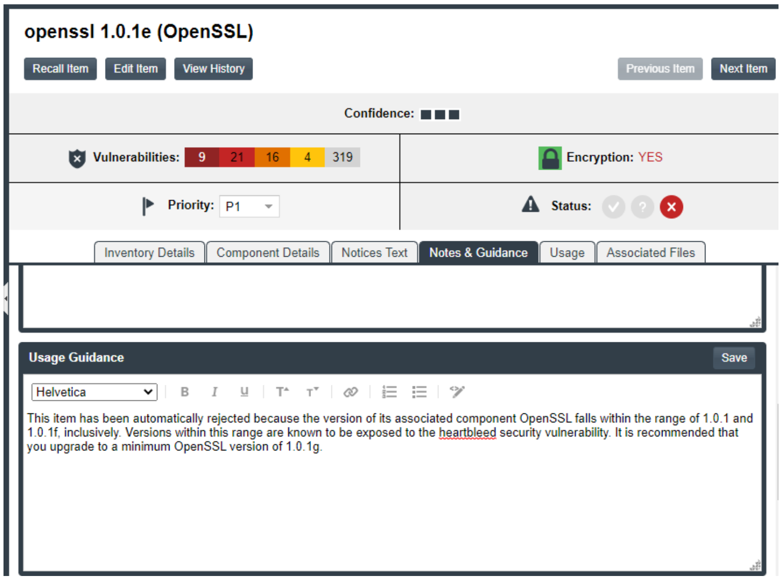


Usage Guidance

Helvetica

This item has been automatically rejected because the version of its associated component OpenSSL falls within the range of 1.0.1 and 1.0.1f, inclusively. Versions within this range are known to be exposed to the heartbleed security vulnerability. It is recommended that you upgrade to a minimum OpenSSL version of 1.0.1g.

When an inventory item is rejected because its component and version meet the criteria of this policy, the **Usage Guidance** pane on the **Notes & Guidance** tab for this inventory item shows your explanation.



This item has been automatically rejected because the version of its associated component OpenSSL falls within the range of 1.0.1 and 1.0.1f, inclusively. Versions within this range are known to be exposed to the heartbleed security vulnerability. It is recommended that you upgrade to a minimum OpenSSL version of 1.0.1g.

If one or more policies approve an inventory item, the **Usage Guidance** content from each of the policies is listed in the **Usage Guidance** pane for the inventory.

If one or more policies reject an inventory item, the **Usage Guidance** content from the only the first policy that rejected the inventory item is displayed in the **Usage Guidance** pane for that item.

User cannot edit the specific content propagated from policies to the **Usage Guidance** pane. However, they can edit and add other information in this pane.

Samples of Usage Guidance Content

The following shows examples of **Usage Guidance** content that can be provided for policies:

- For Vulnerability Policies
- For License Policies
- For Component Policies

For Vulnerability Policies

These are samples of **Usage Guidance** content that can be provided for policies listed in the **Vulnerabilities** section.

Rejection based on CVSS score:

> *This item has been automatically rejected due to one or more associated security vulnerabilities with a CVSS score greater than 7.0. Please consult with your security team for further guidance.*

Rejection based on severity:

> *This item has been automatically rejected due to one or more associated security vulnerabilities with a high severity. Please consult with your security team for further guidance.*

For License Policies

These are samples of **Usage Guidance** content that can be provided for policies listed in the **Licenses** section.

Approval based on license:

License Name: Apache License 2.0 (Apache-2.0)
License Priority: 3 - Permissive / Public Domain

Usage Guidance:
- Registration required before use
- Include in third-party notices if shipped
- Retain copyright notices

Rejection based on license:

This item has been automatically rejected based on a combination of the associated weak-copyleft license (LGPL-2.1) and the fact that the items has been modified and is being distributed.

For Component Policies

This is a sample of **Usage Guidance** content that can be provided for a policy listed in the **Components** section.

Rejection based on a component with version range:

This item has been automatically rejected due to the component being OpenSSL versions 1.0.1 through 1.0.1f. These are known to be exposed to the heartbleed security vulnerability. We recommend you upgrade to a minimum OpenSSL version of 1.0.1g.

Impact on Policies When Code Insight's CVSS Configuration Changes

If the Code Insight System Administrator changes the CVSS version for Code Insight, the following describes the impact on policies related to vulnerabilities on the Policy Details Window.

When CVSS v2.0 is switched to CVSS v3.x

Code Insight makes the following changes:

- If the severity level for the **Reject inventory items if any associated securities vulnerabilities have a severity level equal to or higher than...** field was **Unknown** previously, it is now **None**.
- An additional severity, **Critical**, is available for this same field.

When CVSS v3.x is switched to CVSS v2.0

Code Insight makes the following changes:

- If the severity level for the **Reject inventory items if any associated securities vulnerabilities have a severity level equal to or higher than...** field was previously **None**, it is now **Unknown**.
- If the severity level for this same field was previously **Critical**, note that this severity is no longer available. To handle the conversion, Code Insight checks to see if a score was previously entered in the **Reject inventory items if any associated security vulnerabilities have a CVSS score above**... field. If a score less than **9** was entered, that value is retained in the field (since the previous **Critical** severity started with the score **9**). If a value greater than **9** or no value was entered, the value for this field is now **9**.

Policy Page

The **Policy** page lists all current policy profiles available for use by Code Insight projects. A policy profile contains a set of policies used to perform an automatic review of a project's inventory items upon publication of the inventory. The policies within a given policy profile are defined and maintained on the associated Policy Details Window for the profile.

From the **Policy** page, users who have Policy Manager permissions can access functionality to create new policy profiles, as well as update or copy existing profiles, as described in Managing Policies to Automatically Review Inventory. Users without Policy Manager permissions can view policies only.

To open the **Policy** page, use either of these methods:

- From the Code Insight dashboard (which is displayed when you start Code Insight), click **view policy**. See Opening Code Insight for details on accessing the dashboard.
- Click the ☰ icon in the upper right corner of the Code Insight Web UI to open the Code Insight main menu. Select **POLICY** from the menu.

The **Policy** page contains the following information and functionality:

Table 8-47 ▪ Policy Page

Column/Field	Description
Policy	The list of current policy profiles in a grid format. Each entry shows the profile name and its description, the user who last updated the profile, and date of the last update for the profile. Select a policy profile to edit, copy, or view or to globally apply to inventory across all projects associated with the profile. ● **View icon**—Click 👁 to view the selected policy profile in a read-only format. The **Policy Details** window is opened, listing all the policy details without providing editing capabilities. (This is the only option available to users who do not have Policy Manager permissions.) ● **Edit icon**—Click ✎ to edit the selected policy profile. The **Policy Details** window is opened, showing the profile details. ● **Copy icon**—Click 🗍 to copy the selected policy profile. The **Policy Details** window is opened, showing the new copy. (The selected profile is always saved first.) This new copy is named `Copy of` *`selected_policyProfile_name`*. ● **Apply Policy - Global icon**—Click ⚐ to initiate a job that applies the selected profile to inventory across all projects associated with the profile. This provides a convenient way to force an automatic review/ re-review of inventory if the policy has changed. Refer to Forcing an Automatic Review of Inventory Across All Projects for details. Note the following: ● The policy is applied to all inventory, whether or not the current status of an inventory item was manually set by a user. ● This option is available only when the Code Insight instance is configured with a MySQL database.
Add Policy	Click the **Add Policy** button to create a new policy profile. The **Policy Details** window is opened.

See Also
Policy Details Window
License Details Window
Managing Policies to Automatically Review Inventory

Preferences Page

The **Preferences** page appears when you select **Preferences** from the main menu. From this page, you can change your Code Insight user account password. In addition, you can view and add authorization tokens—that is, JSON Web Tokens known as JWTs—for use with Code Insight REST APIs. (The authorization token are associated with the current user account.) The page has the following fields:

Table 8-48 ▪ Preferences Page

Column/Field	Description
Change Password	
New Password	Enter a new password for the selected authorization token. The password must be a minimum of 8 characters, one of which must be numeric and one of which must in upper case. No spaces are allowed in the password.
New Password Confirm	Reenter the password you entered in the **New Password** field.
Update Password	After entering the password in both fields, click **Update Password** to save your changes.
AUTH Tokens	
Add Token	Click this button to display the **Add Token** dialog.
Name	A list of the names of previously created tokens.
Token	The system-generated token associated with the name.
Create Date	The date on which the token was created.
Actions	A group of icons that indicate actions you can take on each token: ● **Edit (✎)**: Click to open the **Edit Token** dialog. ● **Delete (✖)**: Click to delete the selected token. The token is deleted immediately. ● **Copy to clipboard (📋)**: Click to copy the selected token to the clipboard. You can use this option to copy tokens so that you paste them whenever needed for Code Insight REST access (for example, when configuring the Code Insight plugins, importing or exporting project data, or running REST API directly).

See Also
Add Token Dialog
Edit Token Dialog
Exporting and Importing Project Data
Performing Remote Scans

Project Defaults Tab

The settings on **Project Defaults** tab on the **Administration** page work provide a convenient way to pre-populate fields used to configure new projects to ensure consistency and enable an easier project creation experience for users. Although the settings you define here are global across all projects, they can be overridden at the project level as needed. See the following field descriptions for more information.

Table 8-49 ▪ Project Defaults Tab

Category	Field	
General Options	These options set defaults for project creation and assign default users to project roles. Users can change these defaults when creating a project or when editing a project or its users using **Manage Project | Edit Project | General** or **Manage Project | Edit Project | Edit Project Users** on the project **Summary** tab.	
	Project Visibility	Select the default for visibility status—**Public** or **Private**—for projects. (The initial system default is **Public**.) Any user in the system read-only access to a public project. To what degree a user can interact with the project depends on whether the user has a project role and what the role is—Project Administrator, Analyst, or Reviewer. However, private projects are hidden from all users except the Project Contact and those users assigned as Project Administrators, Analysts, Reviewers, or Observers of the project. Additionally, project and vulnerability ID searches will not return private projects unless the user performing the search has the permissions to see a given private project.
	Project Risk	Select the default risk value (**Low**, **Medium**, or **High**) for projects. To edit, select another value from the dropdown list. The initial system default is **Medium**.
	Project Users	Click the **Edit Project Users** link to open the **Edit Default Project Users** page. From here you assign project roles—Analysts, Reviewers, and Observers—that will default for any new project created (but which can then be edited at the project level). See Edit (Default) Project Users Page for details.

Table 8-49 ▪ Project Defaults Tab (cont.)

Category	Field	
	On the data import or rescan, delete inventory with no associated files	This option determines whether “empty” system-generated inventory items are deleted in the target project during project imports and rescans. Empty inventory items have no associated files. ● **Selected**—Deletes empty inventory items from the target project during project imports and rescans. Only inventory items with associated files are retained/created. ● **Unselected**—Retains/creates all inventory items—with or without matching associated files in the target codebase—in the target project during imports and rescans. For example, you might want to retain inventory items to save their analysis details. (Users will need to manually delete inventory that is not applicable to the current project.) This configuration (unselected) is required when importing a scanned codebase into a project for which no codebase has been uploaded or obtained through synchronization. This option ensures that inventory is generated in the target project.
	Expand Source and Uber jar files	This option determines whether uber and sources jars and are expanded during a codebase upload to the project. ● When selected, this option enables the expansion of the uploaded top-level uber or sources jar and any uber or sources jars contained in the uploaded jar, according to the expansion level defined for the upload. ● When not selected, this option does expand any uber or sources files in the uploaded codebase. For more information, see Expansion of a Sources or Uber Jar.
Scan Settings	These options identify the default Scan Server and scan profile for projects. Users can change these settings at the project level by navigating to **Manage Project | Edit Project | Scan Settings** from the project **Summary** tab.	
	Scan Profile	Select the scan profile to default for projects. Click ⓘ to view the details of the scan profile.
	Scan Server	Select the Scan Server to default for projects. Note that only those Scan Servers in an “enabled” state are available for selection. If only one Scan Server has been identified to the system, this server is automatically selected as the default.

Table 8-49 ▪ Project Defaults Tab (cont.)

Category	Field	
Automated Inventory Publish Options	These options enable and configure the automatic publication of project inventory as part of the project scan process. Users can change these settings at the project level by navigating to the project **Summary** tab and selecting **Manage Project | Edit Project | Scan Settings**. If the **Auto-publish system-created inventory items meeting this minimum Confidence Level** is selected to enable auto-publication, the other auto-publish options are made available.	
	Auto-publish system-created inventory items meeting this minimum Confidence Level	Select this option to enable the auto-publication of system-generated inventory items. (By default, the option is selected.) Then select the minimum Inventory Confidence level required to determine which items to auto-publish: ● **Low**—Automatically publish all system-generated inventory. ● **Medium**—Automatically publish only those system-generated inventory items with Medium and High confidence levels. ● **High**—Automatically publish only those system-generated inventory items with a High confidence level. For a description of the Confidence levels and how they are used, see Inventory Confidence.
	Do not auto-publish inventory items with an undetermined license	Select this option to *not* auto-publish any system-generated inventory item with an undetermined license (that is, an inventory item whose **License** value is **I don’t know**). An undetermined license can occur under the following conditions: ● The scan was not able to identify a license for the given component during the scan and therefore set the **I don’t know** license value. ● The inventory item has multiple possible disjunctive licenses (for example, “GPLV2 or MIT”). However, the scan could find no evidence of the desired selected license and therefore set the **I don’t know** license value. ● The inventory item has multiple possible conjunctive licenses (for example, “GPLv2 and MIT”). However, since Code Insight currently supports only a single selected license, the scan automatically set the **I don’t know** value for the inventory item. This option is available only if **Auto-publish system-created inventory items meeting this minimum Confidence level** is selected. By default, when you first open Code Insight instance after it has been installed or migrated, this option is unselected, allowing the auto-publication of inventory with undetermined licenses.

Table 8-49 ▪ Project Defaults Tab (cont.)

Category	Field	
	Mark associated file as reviewed	Select this option if you want Code Insight to automatically mark the files associated with each automatically published inventory item as “reviewed”. This option is available only if **Auto-publish system-created inventory items meeting this minimum Confidence level** is selected.
Automated Review Options	These options configure defaults for enabling policies that automatically accept or reject inventory when it is published. Users can change these settings at the project level by navigating to **Manage Project | Edit Project | Review and Remediation Settings** from the project **Summary** tab.	
	Policy Profile	Select the default policy profile to associate with all new projects. (The system default is **Default License Policy Profile**.) The policy profile contains a set of policies that use components, versions, licenses, and vulnerability scores and severities as criteria to automatically reject or approve inventory items during a codebase scan (or post-scan). For more information about policy profiles in general, see Managing Policies to Automatically Review Inventory.
	Automatically reject inventory items impacted by a new vulnerability that violates your policy	Indicate the default action to take for published inventory affected by a new security vulnerability downloaded as part of an Electronic Update or Library Refresh. The selected action applies to both non-reviewed and previously approved inventory items on the **Project Inventory** tab. ● Select this checkbox to automatically reject those project inventory items impacted by a new security vulnerability only if this vulnerability has a CVSS score or severity *greater than* the thresholds configured as policy for the Code Insight project. For each inventory item rejected due to a new security vulnerability, the ⚑ icon and a tip are added to indicate the status change and its reason. If a new vulnerability does not exceed policy thresholds, the current status of the inventory item is not affected. ● Leave the checkbox unselected to retain the current status of inventory items impacted by the new vulnerability. For information about setting policies that define CVSS-score and severity thresholds used to reject or approve inventory items automatically, see Policy Page and Policy Details Window. For information about associating these policies with a project, see Managing Policies to Automatically Review Inventory.

Table 8-49 ▪ Project Defaults Tab (cont.)

Category	Field	
Manual Review Options	These options configure defaults for project inventory not automatically reviewed by policy. Users can change these settings at the project level by navigating to **Manage Project | Edit Project | Review and Remediation Settings** from the project **Summary** tab.	
	What should happen if inventory items are not reviewed by policy?	Indicate the default action to trigger for those inventory items that are *not* affected by policy (and therefore have a **Not Reviewed** status) during the publication of inventory either as part of a scan or manually by a user: ● **Do nothing**—Simply show the status of the inventory item as **Not Reviewed** on the **Project Inventory** tab. ● **Send an email notification to the contact**—Automatically send an email to the Project Contact, stating the need for a manual review of the item. The value for **Select the minimum priority...** (described in the next table entry) affects this option. ● **Automatically create a manual review task**—Automatically create a manual review task assigned to the default legal or security reviewer (or both reviewers), and send an email, notifying the reviewer(s) about assigned task. Information about managing such a task to track the progress of a manual review is found in Creating and Managing Tasks for Project Inventory. The value for **Select the minimum priority...** (described in the next table entry) affects this option.
	Select the minimum priority to perform the action selected above	(Enabled when an option other than **do nothing** is selected for the previous field.) Select the default minimum inventory priority (**P1**, **P2**, **P3**, or **P4**) to which the value for the previous field applies. For example, if the previous field is set to **send an email notification to the project contact** and minimum priority is set to **P3**, then the email notification will be sent for only those non-reviewed inventory items with a P1, P2, or P3 priority. No email notification will be sent for P4 inventory items. ***Note ▪** This option has no effect when the **do nothing** value is selected.*

Table 8-49 ▪ Project Defaults Tab (cont.)

Category	Field	
	What type of manual reviews will be performed on this project?	Set the default type of manual review tasks to be generated: ● **Legal Only**—Review tasks are generated for those non-reviewed inventory items that so not meet legal policy criteria. The tasks are automatically assigned to the default Legal reviewer. ● **Security Only**—Review tasks are generated for only those non-reviewed inventory items that have security vulnerabilities. The tasks are automatically assigned to the default Security reviewer. ● **Both Legal and Security**—Review tasks are generated for all non-reviewed inventory items that do *not* meet legal policy criteria; these are assigned to the default Legal reviewer. Additionally, review tasks are generated for those non-reviewed inventory tasks associated with security vulnerabilities and are assigned to the default Security reviewer. With this value, a single inventory item might have both a legal review task and security review task generated. However, if the default reviewers are the same user, a single task is created, describing the requirement for both a legal and security manual review.
	Select reviewers for this project	If desired, designate a new default reviewer to which to assign manual review tasks. (The available reviewer types—Legal or Security or both—depend on the type of manual reviews your site performs, as defined for the previous option.) If your site generates both Legal and Security review tasks, Code Insight determines which reviewer—Legal or Security—is assigned the task and then notified of the task by email. See the previous option description for “both Legal and Security” for more information on how this determination is made. For details about managing and reassigning tasks, see Creating and Managing Tasks for Project Inventory. To select a new default reviewer, click **Change User** next to the name of the current **Legal reviewer** or **Security reviewer** assignee, then select a user from the **Select new...contact** dialog, and click **Apply**. (To reset the reviewer to the Project Contact, click **Reset**.) When a new default reviewer is selected, that user is automatically given the role of project “reviewer” should the user not currently have this role. If “Project Contact” is specified as a default reviewer, the Project Contact’s actual user name is displayed for the reviewer in the project.

Table 8-49 ▪ Project Defaults Tab (cont.)

Category	Field	
Remediation Options	These options configure defaults for rejected project inventory. Users can change these settings at the project level by navigating to **Manage Project | Edit Project | Review and Remediation Settings** from the project **Summary** tab.	
	What should happen if inventory items are rejected?	Indicate the default action to trigger for those inventory items that are automatically rejected by policy when inventory is published during a scan or manually published by a user: • **Do nothing**—Simply show the status of the inventory item as Reject on the Project Inventory tab. • **Send an email notification to the project contact**—Automatically send an email to the Project Contact, stating the need for remediation work on the inventory item. • **Automatically create a remediation task**—Automatically create a remediation task assigned to the default development contact (see the **Assignee for remediation work** option) and send an email, notifying the contact about the assigned task. • **Automatically create a remediation task and an external work item**—Automatically do the following: • Create a remediation task assigned to the default development contact (see the **Assignee for remediation work** option) and send an email, notifying the contact about the assigned task. (See the previous bulleted item for more information.) • Associate a work item with the task, creating the work item in an Application Lifecycle Management (ALM) system (such as an issue in Jira). The work item is created and assigned using the settings defined for the ALM instance to which the Code Insight project is associated. For more information about configuring an ALM instance for the project, see Associating the Project with an Application Life Cycle System to Create Work Items.
	Assignee for remediation work	If desired, designate a new default development contact—for example, an engineering manager—to which to assign remediation tasks. (The Project Contact is the initial system default.) This contact can then manage the task accordingly—for example, reassigning it to another user or manually creating an external work item and assigning it to someone on the development team. For details about managing and reassigning tasks, see Creating and Managing Tasks for Project Inventory. To select a new contact, click **Change User** next to the name of the current assignee, select a user from the **Select new...contact** dialog, and click **Apply**. (To reset the reviewer to the Project Contact, click **Reset**.) If “Project Contact” is specified as the default, the Project Contact’s actual user name is displayed as the remediation assignee in the project.

See Also

Policy Page
Policy Details Window
Edit Project: General Tab
Edit Project: Scan Settings Tab
Edit Project: Review and Remediation Settings Tab
Edit (Default) Project Users Page
Managing Policies to Automatically Review Inventory
Assigning and Removing Project Users
Creating Inventory from the Project Inventory Tab
Creating and Viewing External Work Items for a Project Inventory Task
Associating the Project with an Application Life Cycle System to Create Work Items

Project Inventory Details Pane

The **Project Inventory Details** pan is located on the **Project Inventory** tab for the current project. It is populated with details about the currently selected inventory item on the **Project Inventory** tab. (For a description of the **Project Inventory** tab and how to access it, see Project Inventory Tab.)

The **Project Inventory Details** pane enables legal and security experts to review the published inventory as needed and either approve items for inclusion in the Bill of Materials or reject them until further review or remediation efforts are performed. The reviewers can create tasks for the additional reviews or for the remediation work required by software engineering to fix security or legal risks in the code. They can also finalize the third-party Notices content that can be used in the Bill of Materials.

The following table describes the **Project Inventory Details** pane.

Table 8-50 ▪ Project Inventory Details Pane

Category	Column/Field	Description
Header information	The header on the **Project Inventory Details** pane provides buttons that enable you take actions on the inventory item. It also lists attributes about the item and its associated component.	
	Recall Item	Click to recall (remove) a published inventory item from **Inventory Items** list if it does not fit the criteria for inclusion. The selected items are removed from the **Project Inventory** view and are only visible in the **Analysis Workbench**. The recalled item retains the status it had before the recall (until it is re-published).
	Edit Item	Click to open the **Edit Inventory** window where you can update inventory attributes, including selecting a new component, version, or license. See for Editing Inventory from the Project Inventory Tab details.
	View History	Click open the Inventory History Window, which shows a list of all updates made to the inventory item up to the current date and provides details for each update.

Table 8-50 ▪ Project Inventory Details Pane (cont.)

Category	Column/Field	Description
	Previous Item/ Next Item	Show the details for the previous or next inventory item in the **Inventory Items** list.
	Confidence	A simple three-segment graph representing the Confidence level (High, Medium, or Low) of the inventory item. The Confidence level is the measure of the strength of the discovery technique used to generate the inventory item. The graph shows three shaded segments for High confidence, two for Medium, and one for Low. For more information about the Confidence levels, see Inventory Confidence.
	Encryption	The **Yes**, **No**, or **N/A** value indicating whether the component associated with the inventory item provides the encryption capabilities used in your product. Encryption can affect export controls. This attribute can be updated on the **Edit Inventory** dialog (see Editing Inventory from the Project Inventory Tab).
	Vulnerabilities	A bar graph showing the count of known vulnerabilities by severity color for the component associated with the inventory item. Click the graph to view the list of vulnerabilities and their details. For details about the graph and vulnerabilities in general, see Working with Security Vulnerabilities. The counts in this graph do not include vulnerabilities that are currently suppressed. If no vulnerabilities have been found for the inventory item, the value **No** is displayed in place of the graph.
	Priority	A dropdown list showing the priority level given to this inventory item by the system, with P1 as the highest priority and P4 as the lowest. ***Note ▪** During a scan, the priority for auto-published inventory is automatically assigned based on the associated license.* You can change the priority for this inventory item by selecting a different priority from the dropdown list and clicking **Save**. For more information about priorities, see Inventory Priority.

Table 8-50 ▪ Project Inventory Details Pane (cont.)

Category	Column/Field	Description
	Status	The status of the inventory item: ● **Approved**—The item is approved for use in the software project. ● **Not Reviewed**—The item has not been automatically reviewed by policy (and therefore requires a manual review). ● **Rejected**—The item is not approved for use in the software project. Instead, the item needs further review and remediation before being used in the software project.
Inventory Details tab	The **Inventory Details** tab lists attributes of the inventory item.	
	Name	The name of the inventory item. This attribute can be updated on the **Edit Inventory** dialog (see Editing Inventory from the Project Inventory Tab).
	Description	A description of the inventory item. This attribute can be updated on the **Edit Inventory** dialog (see Editing Inventory from the Project Inventory Tab).
	URL	The URL of the forge repository for this inventory item. You can click the URL link to open the component website. This attribute can be updated on the **Edit Inventory** dialog (see Editing Inventory from the Project Inventory Tab).
	purl	The package URL for the component represented by the inventory item. This non-editable value is retrieved from the Code Insight data library and is applicable to a component associated with a non-custom version only. If no purl value is available in the data library or the version is custom, the value for this field is **N/A**. The use of package URLs is an attempt to standardize the way in which software packages and their locations are identified so that this information is more universal and uniform across programming languages, packaging conventions, tools, APIs, and databases.

Table 8-50 ▪ Project Inventory Details Pane (cont.)

Category	Column/Field	Description
	Created By	The creator of the inventory item as either: ● **System**—Code Insight automatically generated the item per one of these detection techniques (as designated in the **Notes** for the inventory item) during a scan: ● High Confidence Custom Auto-WriteUp Rule ● High Confidence Auto-WriteUp Rule ● Medium Confidence Auto-WriteUp Rule ● Automated Finding ● Low Confidence Auto-WriteUp Rule ● High Confidence MID Rule ● Low Confidence MID Rule ● Audit Import **<user_name>**—The first and last name of the user who manually created the item.
	Created On	The date and time that the inventory item was created.
	Updated On	The date and time that the inventory item was updated. If the item has not been updated since its creation, the date and time shown here will be the same as the **Created On** date and time.

Table 8-50 ▪ Project Inventory Details Pane (cont.)

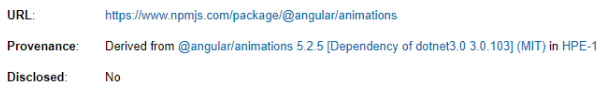
Category	Column/Field	Description
	Provenance	The project from which the inventory is immediately derived. You cannot update this property from the Code Insight Web UI in general, but you can edit it when creating or updating inventory using the Inventory REST API. If the inventory item is not derived from another project, the value **Originated in this project** is displayed. However, if the inventory item is derived from another project (for example, the current inventory item was imported or copied to the current project), the inventory and project name of the inventory item’s predecessor is displayed:  **URL**: https://www.npmjs.com/package/@angular/animations **Provenance**: Derived from @angular/animations 5.2.5 [Dependency of dotnet3.0 3.0.103] (MIT) in HPE-1 **Disclosed**: No If the preceding project and inventory item still exist, this value is hyperlinked so that you can open this project directly to the **Project Inventory** tab, with focus on the **Inventory Details** page for the preceding inventory item. This direct link enables you to trace the origin of the item through its chain of predecessors, exploring the audit and review details of the preceding inventory items to determine inventory history—for example, the reason the item was previously approved or rejected. If the preceding inventory item or project no longer exists, no link is provided.

Table 8-50 ▪ Project Inventory Details Pane (cont.)

Category	Column/Field	Description
	Dependency Scope	The dependency scope of the inventory item: ● **Runtime**—The inventory item is a dependency required at runtime. ● **Non-Runtime**—The inventory item is a dependency not required at runtime. ● **N/A**—The inventory item cannot be classified as a runtime or non-runtime dependency. Such items include top-level inventory, dependencies for which Code Insight does not currently support the reporting of scope, and migrated inventory for which a scan has not been run. For more information, see Dependency Scopes in the **Automated Analysis** section. This field is not editable. ***Note*** ▪ *Your access to inventory of a specific scope in a project can change if a certain reconfiguration has previously taken place—for example, a change to the scan profile or a re-upload of updated runtime and non-runtime dependencies—and a rescan or full rescan has subsequently taken place.*
	Disclosed	The property indicating whether the third-party component or artifact represented by the inventory item known third-party dependency in your code before it was discovered by the scan or you. The value is either **Yes** or **No**. This field is used most often by analysts to denote information about the state of the inventory item. This attribute can be updated on the **Edit Inventory** dialog (see Editing Inventory from the Project Inventory Tab).
	Modified	The property indicating whether code from the OSS or third-party package has been modified for use by your organization. The value is either **Yes, No,** or **Unknown**. This attribute can be updated on the **Edit Inventory** dialog (see Editing Inventory from the Project Inventory Tab).
	Alerts	The property notifying you whether or not security vulnerability alerts exist for this item. If alerts exist, click the ***x*** **Open Alerts** or ***x*** **Closed Alerts** link to view their details. If no alerts exist, **None** is displayed. You can open the **Alerts** dialog from this pane to change the status or priority of an alert. For more information, see Managing Security Vulnerability Alerts.

Table 8-50 ▪ Project Inventory Details Pane (cont.)

Category	Column/Field	Description
	Tasks	The number of open or closed tasks for this inventory item. Click the **x Closed Tasks** or **x Open Tasks** link to view and update the tasks. If no tasks are associated with this inventory item, **None** is displayed. You can access the Tasks dialogs from this pane to create, edit, and close tasks. See for Creating and Managing Tasks for Project Inventory details.
	Workflow URL	The URL link or a plain text reference (such as a Jira issue number) to request data pertaining to this inventory item in your site's external workflow system. The link enables the reviewer to easily access the workflow data that tracks the status of open tasks for the inventory item. (The plain text reference still helps the reviewer locate the appropriate data in the workflow system.) You can define this attribute when you edit or manually create an inventory item from the **Analysis Workbench** or the **Project Inventory** tab. If no URL or reference has been defined, the value is **None**. If additional request-related details are available for this inventory item, the ⓘ icon is displayed next to the URL. Click the icon to open the **Workflow Request Details** window for a quick review of pertinent details about the request without having to access the workflow system. ***Note ▪*** *These details come from the specific external workflow system associated with your site. The details can vary based on your workflow system.*

Table 8-50 ▪ Project Inventory Details Pane (cont.)

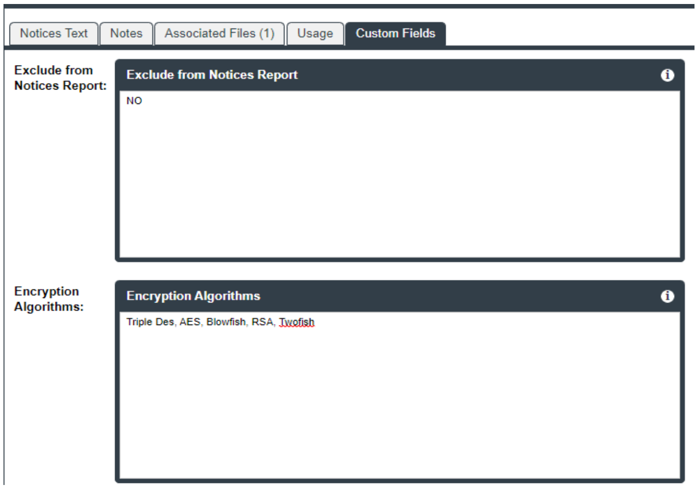
Category	Column/Field	Description
	Custom fields	Any custom fields that were defined specifically for your site display at the bottom of the **Inventory Details** tab (after the **Workflow URL** field). These fields provide information that standard Code Insight fields on the **Inventory Details** tab do not capture about the inventory.  Notices Text Notes Associated Files (1) Usage Custom Fields Exclude from Notices Report: **Exclude from Notices Report** NO Encryption Algorithms: **Encryption Algorithms** Triple Des, AES, Blowfish, RSA, Twofish If no custom fields have been defined, nothing is displayed after the **Workflow URL** field. Use the following guidelines for entering (or editing) a value in a custom inventory field: ● If available, click the ⓘ icon in the upper right corner of a field to obtain help on completing the field. ● You can enter a value up to 64k (64000 characters) in size. ● To save the value, click the **Save** button in the upper right corner of the field. (This button is activated when you begin to type in the field.)
Component Details tab	The **Component Details** tab lists attributes of the OSS or third-party component associated with the inventory item.	
	Component	The name of the OSS or third-party component and internal ID, as identified in the Code Insight Data Library. You can associate the inventory item with a different component using the **Edit Inventory** dialog (see Editing Inventory from the Project Inventory Tab).

Table 8-50 ▪ Project Inventory Details Pane (cont.)

Category	Column/Field	Description
	Version	The component version and its internal ID, as identified in the Code Insight Data Library. You can associate the inventory item to a different version of the component using the **Edit Inventory** dialog (see Editing Inventory from the Project Inventory Tab).
	Forge	The external repository associated with the component. You can click the forge link to open the forge website.
	Selected License	The name of the license selected for this component. Click ⓘ to view additional information about the license. See License Details Window. You can switch to a different license from the **Edit Inventory** dialog (see Editing Inventory from the Project Inventory Tab).
	Possible Licenses	Other licenses that can be associated with the component.
	Custom Component	The **Yes** or **No** value indicating whether the component is custom (created by a user) or provided as part of the Code Insight Data Library.
	Vulnerabilities	A bar graph showing the count of known vulnerabilities by severity color for the component. Click the graph to view the list of vulnerabilities and their details. For details about the graph and vulnerabilities in general, see Security Vulnerabilities Associated with Inventory. If no vulnerabilities have been found for the inventory item, the value **No** is displayed in place of the graph.
	Encryption	The **Yes**, **No**, or **N/A** value indicating whether the component provides the encryption capabilities used in your product. Encryption can affect export controls. This attribute can be updated on the **Edit Inventory** dialog (see Editing Inventory from the Project Inventory Tab).

Table 8-50 ▪ Project Inventory Details Pane (cont.)

Category	Column/Field	Description
	CPE	The list of CPE names—from the National Vulnerability Database—that are mapped to the component. CPE (Common Platform Enumeration) is a structured naming scheme that includes the component's vendor and product names in the following format: **cpe://<part>:<vendor>:<product>** where <part> is either **a** (applications), **h** (hardware platforms), or **o** (operating systems). Note that the data provided only represents the part, vendor and product; the version information is truncated from the CPE string.
Notices Text tab	The **Notices Text** tab is used to finalize the exact content to include in the Notices report. You can edit the notices content as needed from this tab when editing an existing inventory item or creating a new one. For more information, see Finalizing the Notices Text for the Notices Report.	
	As-Found License Text	The **As-Found License Text** field shows the license text or license references found in the scanned codebase. You cannot edit this field. However, if you want to use this content in the Notices report, click **Copy to Notices Text** to copy the text to the **Notices Text** field and modify it if necessary. If content already exists in the **Notices Text** field, you can choose either to append the **As-Found License Text** content to the existing notices content or to replace the existing notices content. This field is blank if no license text or references were found in the scanned codebase. If this field contains information and the **Notices Text** field remains blank, the Notices report uses the content in this field. If both fields are empty, the report uses the license content from Code Insight Data Library (see License Details from the Code Insight Data Library).

Table 8-50 ▪ Project Inventory Details Pane (cont.)

Category	Column/Field	Description
	Notices Text	The exact content to include in the Notices report. You can edit any license text previously saved to this field or manually add your own license text, such as license information for rules that you developed during your manual research on the inventory item. You can also copy the **As-Found License Text** content (see the previous description) to the **Notices Text** field and modify it as needed. As a third option, you can click the **Update Notices Text** button to pull a copy of the current license content from the Code Insight Data Library into the **Notices Text** field and modify it as needed. Alternatively, you can leave this field empty. Click **Save** at the top of the field if you make any changes to this field. If you provide information in this field, the Notices report pulls the content of only this field into the report. If this field is empty, the content of the **As-Found License Text** field is used in the report. If both fields are empty, the report uses the license content from Code Insight Data Library (see License Details from the Code Insight Data Library). For more information, see Finalizing the Notices Text for the Notices Report.
	Copy to Notices Text button	(Located within the **As-Found License Text** field) Click this button to copy content the text in this field into the **Notices Text** field and modify it as necessary. If the **Notices Text** field already contains content, you are given the option either to append the **As-Found License Text** content to the existing **Notices Text** content or to replace all the existing **Notices Text** content with the **As-Found License Text** content. Appended text starts on a new line after the existing content in the **Notices Text** field.
	Update Notices Text button	(Located within the **Notices Text** field) Click this button to copy content from the Code Insight Data Library into the **Notices Text** field. You can then modify the content as needed. If the **Notices Text** field already contains content, you asked whether to overwrite the content. If you select **No**, the copy operation is ended. If you select **Yes**, the operation proceeds. Refer to Using License Text from the Revenera Data Library in the Notices Report for the prerequisites needed to perform this copy and the types of issues you can encounter.

Table 8-50 ▪ Project Inventory Details Pane (cont.)

Category	Column/Field	Description
	Save button	Click this button to save any changes you made to the **Notices Text** field. The information saved to this field will be used for the inventory item in the Notices report.
Notes & Guidance tab	The **Notes & Guidance** tab provides information about the automated and manual analysis of codebase as it relates to an inventory item.	
	Detection Notes	System notes that can specify the following: ● The automated detection technique that was used to locate the component ● License information in the case that the license has changed from one version to another or if the component has multiple licenses ● Attributes extracted from a POM or manifest file containing project and configuration details
	Audit Notes	Any notes added to the inventory item by the auditor or reviewer, based on findings during the analysis.
	Usage Guidance	Notes helpful provided by a reviewer to assist other reviewers or to provide guidance to software engineers assigned tasks to fix or modify the use of the OSS or third-party software in the product code.
Usage tab	The **Usage** tab provides details on how your product uses the OSS or third-party software. You cannot update these items on the **Usage** tab, but you can update them on the **Edit Inventory** dialog (see Editing Inventory from the Project Inventory Tab).	
	Distribution Type	The option indicating how the inventory item is distributed: ● **Internal**—Internally only (such as test framework that might be included in the codebase but is not distributed with the product). ● **External**—Externally with the product, shipped to customers (outside of your organization, including a private cloud deployment at the customer's site) ● **Hosted**—Hosted in your company's data center (such as a SAAS application). ● **Unknown**—Unknown distribution type.

Table 8-50 ▪ Project Inventory Details Pane (cont.)

Category	Column/Field	Description
	Part of Product	The option indicating whether the item is part of the core product or an infrastructure piece such as a build or test tool. This can affect whether third-party notices are required for this item. The value can be **Yes**, **No**, or **Unknown**.
	Linking	The option identifying how your software package links to the OSS or third-party component libraries. This method can affect license priority and obligations. ● **Not linked**—The software package uses no links to the component libraries. ● **Statically linked**—The component libraries are included in the software materials and thus linked statically. ● **Dynamically linked**—The component libraries are brought in at runtime. ● **Unknown**—The type of linking is unknown.
	Modified	The option indicating whether code from the OSS or third-party package has been modified for use by your organization. The value can be **Yes**, **No**, or **Unknown**.
	Encryption	The option indicating whether the component provides the encryption capabilities used in the product. Encryption can affect export controls. The value can be **Yes**, **No**, or **Unknown**.
Associated Files	Click this tab to view a list of the files that are part of the inventory for this project. Each file entry shows the following: ● **Alias**—The unique user-defined alias that was defined for the scanner (Scan Server or remote scan agent) to represent its scan-root path containing the codebase in which the file is located. The alias provides a name that is more meaningful than the scan-root path. (The actual absolute scan-root path for each scanner associated with the project is available on the project's **Summary** tab.) ● **File Path**—The file's path relative to the scan-root path on instance hosting the scanner. You can click the file-path link to open the **File Details** tab for that file. If you have Analyst permissions, the path is hyperlinked to open to the file's **File Details** tab in the **Analysis Workbench**, where you can view file evidence. If necessary, while in the **Analysis Workbench**, you can also add or remove files associated with the inventory. If you do not have Analyst permissions, the path remains in plain text.	

Project Inventory Tab

The **Project Inventory** tab lets you search and review the published inventory of open-source and third-party components found in the source code and artifacts scanned in a Code Insight project. To access the **Project Inventory** tab, see Displaying Project Inventory.

About Published Project Inventory

An inventory item can be placed in a *published* state automatically by a scan (based on policy criteria) or manually by an analyst if enough evidence exists to ensure that this component is actually used by your code. At publication, those inventory items that are not automatically approved (based on policy criteria) will need to be reviewed to validate that they should be included in the Bill of Materials for your product.

The **Project Inventory** tab provides the means of finalizing the inventory to include in your Bill of Materials. The tab enables legal and security experts to review the published inventory as needed and either approve items for inclusion in the Bill of Materials or reject them until further review or remediation efforts are performed. The reviewers can create tasks for the additional reviews or for the remediation work required by software engineering to fix security or legal risks in the code.

Field Descriptions

The **Project Inventory** tab consists of two panes: the left pane showing the list of published inventory for the project and the **Project Inventory Details** pane on the right showing the details of the inventory item currently selected in the list. The following table describes the information on this tab.

Table 8-51 ▪ Project Inventory Tab

		Field/Panes	Description
Inventory Items (x) pane	This pane shows the list of inventory items that have been published in the project. The pane title includes the number of inventory items currently displayed in the list.		
	Inventory item fields	The following properties describe each inventory item in the list.	
		Name	The inventory item name in *componentName version (license)* format, as in the following: apache-activemq 5.4 (Apache-2.0) If the inventory item is a dependency of another inventory item, the relationship is shown within brackets, as in this example: activemq-optional 5.4.0 [Bundled with apache-aapache-activemq 5.4] (Apache-2.0) You can sort the inventory list on this column in ascending or descending alphabetical order. To view or edit information about an inventory item, click the hyperlinked inventory name to open its **Project Inventory Details** pane.
		Priority	The inventory priority, indicating how the item ranks in importance in the review process, with P1 as the highest priority and P4 as the lowest. For more information about inventory priority, see Inventory Priority. You can sort the inventory list on this column in ascending (P1 to P4) or descending numeric order (P4 to P1).

Table 8-51 ▪ Project Inventory Tab (cont.)

	Field/Panes	Description
	Vulns	The total number of security vulnerabilities associated with the inventory item. Vulnerability details for the inventory item are available in the Project Inventory Details Pane when you select the item.
	Status	The review status of the inventory item: **Not Reviewed**, **Approved**, or **Rejected**. For more information about the inventory status, see Review Status of Inventory. You can use this column to sort the inventory list by review status in ascending (**Not Reviewed**, **Approved**, **Rejected**) or descending (**Rejected**, **Approved**, **Not Reviewed**) order.
Inventory search options	If you need to filter the inventory list to locate the inventory items to review, use either or both of these options.	
	Enter Inventory Name	To filter the inventory list by a string contained in the item name, enter this string in the **Enter inventory name** box. As you type each character in the string, the list is automatically filtered according to the entered characters. This current name filter is copied to the Advanced Inventory Search Dialog should you later perform an Advanced Inventory Search (by clicking **Advanced Search**). Likewise, if you enter a name filter on the **Advanced Inventory Search** dialog, it is copied to this field. This behavior enables you to keep the name filter persistent. However, you can always change or remove the filter in either location.
	Advanced Search	To filter the inventory list by one or more inventory properties, click the **Advanced Search** button. For complete details about performing an advanced inventory search, see Advanced Inventory Search Dialog.
Add Item	This button is visible only if you have Analyst permissions. To add a new inventory item to the list, click this button to open the **New Inventory** dialog. (The new item is automatically placed in the published state.) For more information creating an inventory item, see Creating Inventory from the Project Inventory Tab.	

Table 8-51 ▪ Project Inventory Tab (cont.)

Field/Panes	Description
Project Inventory Details pane	When you select an inventory item, the **Project Inventory Details** pane on the right is populated with details about the item. From this pane, you can edit the item's properties, set up review and remediation tasks for the item, provide audit, usage-guidance, and remediation notes, edit the item's third-party Notices content, and ultimately approve or reject the item for its inclusion in the Bill of Materials. For complete information, see Project Inventory Details Pane.

See Also
Displaying Project Inventory
Advanced Inventory Search Dialog
Project Inventory Details Pane
Editing Inventory from the Project Inventory Tab
Reviewing Published Inventory for a Project

Projects Pane and Associated Dashboard

The **Projects** pane in the **Projects** view displays the projects available in your Code Insight instance. You can configure this display as needed to help you locate projects and open them. You can also create, delete, and rename projects from this pane. When you select a project from this pane, its associated dashboard is displayed. The **Projects** pane and associated project dashboard contain the following fields:

Table 8-52 ▪ Projects Pane Page

Column/Field	Description
Tree view	Click to toggle the project display to tree view.
List view ☰	Click to toggle the project display to a plain list view. (By default, all projects are in tree view.)
+	(Tree view only) Expand all folders. (By default, all folders are collapsed.)
−	(Tree view only) Collapse all folders.
Add New	Click to add a new folder or project to the list. (This button is displayed only if you have permission to create projects.)
My Projects	Click to show only those projects with which you are associated, either as Project Contact or with a project role (Analyst, Reviewer, or Observer).

Table 8-52 ▪ Projects Pane Page (cont.)

Column/Field	Description
Projects (x)	The number of projects in the system. If the list is filtered, the filtered count is shown in relation to the full count (for example, "(19 of 123)").
Project Search fields	From the search filters on the left, select the filter based on the type of search you want to perform (**Project Name**, **Project Inventory**, or **Security Vulnerability**). In the field on the right, enter the string criterion for the search: • When searching for a project name, enter a partial string or full project name. • When searching for project inventory, enter the inventory name or the inventory's component name, license name, or license SPDX short identifier. The characters must be consecutive in the search string. A partial string is supported. • When searching for a security vulnerability, enter the exact vulnerability ID. Press Enter to view the filtered projects display. If no inventory items meet the specified criterion, the projects display shows "No Projects". To clear the search filter and display of all projects, click ✕ in the criterion field. See for Searching Across All Projects in the System for full details.
Project tree or plain list	The display format (tree or plain list) of listed projects based on the current filter. You can perform the following from this list (depending on your permissions): • To open a project, either click the **Open Project** icon next to the project entry in the project tree or list, or click the project's name link in the upper left corner of the project's dashboard (to the right of the **Projects** pane). The project opens to its **Project Inventory** tab. • To display the dashboard for a project, either click the project in the project tree or list, or click the **Load Project Dashboard** icon in the project entry. • To rename a project, double-click the project name and overwrite the current name with the new name. • To move a project to a different location in the project tree, drag and drop the project to the desired folder (or to the root location). • To create or delete a project or folder from the project tree or list, use the right-click menu. See Managing Items in the Projects Display.

Table 8-52 ▪ Projects Pane Page (cont.)

Column/Field	Description
Selected Project Name	Select a project from the project tree or list to refresh the dashboard on the right with information about the selected project. You can click the project name displayed in the top left of the dashboard to open the project.
Project Contact	In the project dashboard header, the hyperlinked name of the user designated as the Project Contact—the main point of contact for the project. Click the name to open your default email program to send an email to the Project Contact.
Policy	In the project dashboard header, the name of the policy profile associated with the selected project. The policy profile is used to automatically mark inventory items as approved or rejected when they are published—without the need for a manual review. For more information, see Managing Policies to Automatically Review Inventory.
Created	In the project dashboard header, the date on which the project was created.
Project Summary Graphs	On the project dashboard, graphs providing overview statistics about the inventory associated with the selected project. The graphs are interactive; when a section of a graph (or a corresponding legend item) is clicked, the **Project Inventory** tab is displayed, showing a filtered view of the inventory that applies to the section of the graph (or to the legend item) you clicked. See also Using the Project Dashboard.

See Also
About Code Insight Projects
Summary Tab
Showing Only Your Projects
Searching Across All Projects in the System
Using the Project Dashboard

Reports Tab

The **Reports** tab is displayed within the context of a given project and enables you to generate any standard and custom Code Insight reports currently available for the project. (The same reports are available to all projects.) You can generate a single report or multiple different reports simultaneously for the project. Once the generation of a report has completed, links are provided to view an HTML version of the report and to download the report in all its available formats.

For more information about the procedure used to generate a report and for a description of standard and custom reports, see Generating Reports for a Project.

The following table describes the components of the **Reports** tab.

Table 8-53 ▪ Reports Tab

Category	Column/Field	Description
Report list	The list of reports on the **Reports** tab shows the following information for each available Code Insight report—standard or custom.	
	Name	The name of the report. To generate a report, select its name and click **Generate Selected Report**.
	View Report	The **View** link to open the last generated version of the report in HTML format. When you click the link, the report is displayed in your browser. Anytime that you regenerate the report, the link is updated to open the new report version. No link is displayed until the report is generated for the first time.
	Download Report	The **Download** link to download an archive containing all available formats of the last generated version of the report. When you click the link, the archive is downloaded to your system's default download location, where you can then open (and save) the report in any of its formats. Anytime you regenerate the report, the link is updated to download the new report version. No link is displayed until the report is generated for the first time.
	Generated On	The date and time of the last successful generation of the report. No date and time is displayed until the report is generated for the first time.

Table 8-53 ▪ Reports Tab (cont.)

Category	Column/Field	Description
Generate Selected Report	Click this button generate the currently selected report. The following happens: ● A message prompt appears explaining that the report will be generated in the background, enabling you to continue to work in Code Insight as the report generates. Click **OK** to proceed with the report generation. Or ● If the **Include data from Second Project** field is displayed, enter the name of the second project whose data will be included along with the data from the current project for comparison purposes. As you type a string, project names containing that string are listed in a dropdown list from which you can then select the desired project name. (This is a required field.) ● If other fields are displayed, enter the requested values in those fields. Default values can be overwritten. Click the ? icon next to a field for more information about its purpose and possible values. The **Generate Report** button on the pop-up remains disabled until all required fields are completed. (Required fields left blank are outlined in red.) When these additional fields have been properly completed, click **Generate Report** on the pop-up window. Note that the **Generate Selected Report** button is disabled for each report that is currently generating, but *is* enabled for any other report not being generated. While one report is generating, you can select and generate another report, enabling you to generate multiple different reports for the project simultaneously. Once a report is successfully generated, the **View** and **Download** links and the date/time of the generation are provided for that report. If a report generation fails, a message is displayed (in the area where the links ad day/time information for the report would have displayed), stating that the report has failed and to check the logs for further information. (A Code Insight System Administrator can review the contents of the `core.log` to determine the reason for the report failure and relay the information to the appropriate contacts to fix the issue.)	

See Also

Opening a Project

Generating Reports for a Project

Scan History Dialog

The **Scan History** dialog displays a list of previous scans that have been performed on the selected project. The dialog contains the following fields:

Table 8-54 ▪ Scan History Dialog

Column/Field	Description
➕	Click to view messages about the scan. If no messages were generated during the scan, the message field will be blank.
Scheduled On	The date and time that the scan was scheduled.
Started On	The date and time that the scan was started.
Completed On	The date and time that the scan completed.
Duration	The amount of time the scan took.
Scheduled By	The user name of the person who scheduled the scan.
Status	The status of the scan: *Completed* or *Failed*.
Ok	Click **Ok** to exit the **Scan History** dialog and return to the **Scan Summary** page.

See Also
Auditing Scan Results in the Analysis Workbench

Scan Profiles Tab

The **Scan Profiles** tab on the **Administration** page provides the list of currently available scan profiles. From this list, you can select a scan profile to view its settings in read-only format or to edit its settings. The tab also enables you to create a scan profile.

For a detailed description of the settings that define a scan profile, setting, see Create (or Edit or View) Scan Profile Dialog.

Table 8-55 ▪ Scan Profiles Tab

Column/Field	Description
Scan Profiles list	A list (in grid format) of available scan profiles. The following are the predefined scan profiles: ● Standard Scan Profile ● Basic Scan Profile (without CL) ● Comprehensive Scan Profile The list will contain additional profiles if you have added them. The following are key settings shown for each scan profile in the list. ● **Scan Archives**—Whether the Scan Server will perform package discovery and license detection within all archive files in the project codebase. ● **Dependencies**—The level of component-dependency scanning to be performed by the Scan Server. ● **Exact Matches**—Whether Scan Server is enabled to identify those codebase files that exactly match file data in the CL (Compliance Library). ● **Source Code Matches**—Whether the Scan Server is enabled to identify source-code strings (snippets) in the scanned codebase files that match exact strings in the CL.
Edit icon ✎	To edit a scan profile, click this icon in the **Actions** column for the profile. The **Edit Scan Profile** dialog is opened, enabling you to edit profile settings. See Create (or Edit or View) Scan Profile Dialog for a description of each setting. For further instructions, see “Creating or Editing Scan Profiles” in *Code Insight Installation & Configuration Guide*.
View icon 🗎	To view the settings (in read-only format) defined for a given scan profile, click this icon in **Actions** column for the profile. See Create (or Edit or View) Scan Profile Dialog for a description of each setting.
Add Scan Profile button	Select this button to create a new scan profile. The **Create Scan Profile** dialog is opened, enabling you to define a new scan profile. See Create (or Edit or View) Scan Profile Dialog for a description of each setting. For further instructions, see “Creating or Editing Scan Profiles” in *Code Insight Installation & Configuration Guide*.

See Also

Create (or Edit or View) Scan Profile Dialog
About Code Insight Scans
Applying a Scan Profile to the Project

Scan Server Dialog

Before a user can assign project codebases to a Scan Server in order to scan them, the Scan Server must first be installed either on the same instance as the Code Insight Core Server or on a separate instance, as described in the *Code Insight Installation and Configuration Guide*. (The Scan Server must have the same version as the Core Server.) As Code Insight System Administrator, you must then use the **Scan Server** dialog to “add”—that is, identify—the server to the Code Insight system to make it available for scanning purposes.

In addition to adding a new Scan Server, you use the **Scan Server** dialog to edit an existing Scan Server’s properties. For detailed instructions on adding or editing a Scan Server, see “Adding or Editing Scan Servers” in the *Code Insight Installation and Configuration Guide*.

Multiple Scan Servers

If multiple Scan Servers have been installed, you can identify more than one of these servers to the system, thus enabling users to distribute codebase scans. Keep in mind that, when multiple Scan Servers are installed, each should be installed on a different instance with a unique host ID and port. The codebase for a given project can be assigned to only one of the Scan Servers (but multiple project codebases can be assigned to a single Scan Server). All codebases assigned to a given Scan Server are stored on that server in a location that you specify.

Prerequisite for Adding or Editing a Scan Server

Ensure that the Scan Server that you are adding or editing is currently running and that the Scan Server you are adding has the same version as the Core Server.

Dialog Fields

The **Scan Server** dialog contains the following fields:

Table 8-56 ▪ Scan Server Dialog

Column/Field	Description
Alias	Enter a common name for the Scan Server.
Host	Provide the hostname (such as `krl.eng.companyA.com`) or IP address of the instance hosting the Scan Server. If the Scan Server is on the same instance as the Core Server, enter `localhost`. The same host-and-port combination must be unique among the *enabled* Scan Servers. (See **Status** is this table for a description of enabled Scan Servers.)
Port	Specify the port used by the Scan Server on the host instance. By default, the port is `8888`. The same host-and-port combination must be unique among the *enabled* Scan Servers. (See **Status** is this table for a description of enabled Scan Servers.)

Table 8-56 ▪ Scan Server Dialog (cont.)

Column/Field	Description
CL Path	Provide the path for the Code Insight Compliance Library (CL), downloaded from the Product and License Center. The CL is a database used by the Scan Server to perform exact-file and source-code fingerprint (snippet) matching. Code Insight compares elements of scanned codebase files with information contained in the CL to generate file-level evidence on which you can take action. The validity of the entered path is checked when you click **Save**. Alternatively, leave this field blank to scan your codebase without using the CL. (Code Insight provides the scan profile “Basic Scan Profile (without CL)” to perform the scan.) This type of scan generates inventory from Code Insight’s Automated Analysis feature but has limitations, as described in “About Scanning without the Compliance Library” in the *Code Insight Installation and Configuration Guide*. Keep in mind that, when you run a scan using the CL (that is, by specifying a valid CL path), you obtain a deeper, more comprehensive scan on your codebase. For additional information, see the following: ● “Managing Scan Profiles” in the *Code Insight Installation and Configuration Guide* for more information about the “Basic Scan Profile (without CL)” and about creating and managing scan profiles in general. ● Applying a Scan Profile for instructions on associating a scan profile with a project. ● About Code Insight Scans for information about Code Insight scans in general.
Codebase Path	Provide the path on the Scan Server where Code Insight will store and manage all uploaded code for projects that use this Scan Server. Ensure you have adequate disk space to store the codebases. The recommended starting size for this directory is 500GB. The directory must already exist. The validity of the entered path is checked when you click **Save**. Once the Scan Server to added to the Code Insight system, this field cannot be edited.

Table 8-56 ▪ Scan Server Dialog (cont.)

Column/Field	Description
Status	By default, the Scan Server is enabled for scanning. However, if necessary for an existing Scan Server, select **Disabled** to make the Scan Server unavailable for further scans. Once disabled, the server is no longer displayed in the **Scan Server** dropdown list during project creation or when setting global project defaults. Additionally, this field becomes read-only on the **Edit Project** window. Note the following about disabling a Scan Server: ● If this Scan Server is the system default Scan Server (as defined on the **Project Defaults** tab), you must change this default to another server before you can disable the current server. See Project Defaults Tab for instructions on updating the default Scan Server. ● If this Scan Server is associated with one or more projects, a warning is displayed before you can disable the server. Once you click **Yes**, the **Start Scan** and **Upload Project Codebase** options are disabled on the **Summary** page for each project associated with the server. If you attempt to re-enable a disabled Scan Server when another currently *enabled* Scan Server has the same host-and-port combination or alias, you receive an error when you click **Save**.
Save	Click this button to save any changes you made to the Scan Server properties. Errors are generated when the following conditions exist: ● The Scan Server you are adding or editing is not running. ● The version of the Scan Server you are adding is different from the Core Server version. ● The codebase path or CL path is invalid.
Cancel	Click this button to cancel any changes you made to the fields on the Scan Server dialog.

See Also

Project Defaults Tab
Edit Project: Scan Settings Tab
Scan Servers Tab
About Code Insight Scans

Scan Servers Tab

The **Scan Servers** tab on the **Administration** page lists the Scan Servers that you have identified to your Code Insight system. Each entry in the list shows basic information about the given Scan Server including its status. From a given entry, you can access a separate dialog to edit server properties as well as refresh the entry itself to see the latest server status. The tab also lets you define a new Scan Server. The tab contains the following columns and buttons to identify and manage Scan Servers:

Table 8-57 ▪ Scan Servers Tab

Category	Field		
Scan Server entry	The following columns for each entry in the Scan Server list provide information about the given Scan Server, as well as a means to refresh the Scan Server status and edit server properties.		
	Alias	The user-defined name for the Scan Server, as well as the server's current status. The following icons represent the server status:	
		🟩	The green icon indicates that the Scan Server is "enabled" for scanning and is currently running (turned on). Scans are run in queue order.
		🟥	The red icon indicates that the Scan Server is "enabled" for scanning but is currently not running (that is, it is turned off). Any attempts to associate a project with the Scan Server or upload a codebase to the server generates an error. Additionally, any attempt to initiate a scan will result in the scan's being queued. However, once the server is active, the scan will start based on queue order. (Users can click the **Past Scans** link on the project **Summary** page to view details about the scheduled scan.)
		◼	The gray icon indicates that the Scan Server is "disabled" (that is, cannot be used for scanning). Whether or not the server is running has no effect on this status. If an enabled server is needed for scans on a project assigned to a disabled Scan Server, you must create a new project.
	Host	The `localhost` value is used if the Scan Server is on the same instance as the Core Server.	
	Port	The port used by the Scan Server on the host instance. By default, the port is `8888`.	
	CL Path	The path for the Code Insight Compliance Library (CL). If the path is specified, the CL is accessed as part of the scan to perform exact-file and source-code fingerprint (snippet) matching. Elements of scanned codebase files are compared with information contained in the CL to generate file-level evidence on which you can take action. If the path is not specified, the codebase is scanned without using the CL. This type of scan generates inventory from Code Insight's Automated Analysis feature but has limitations. For more information about the Compliance Library, see About Code Insight Scans.	

Table 8-57 ▪ Scan Servers Tab (cont.)

Category	Field	
	Codebase Path	The path on the Scan Server where Code Insight will store and manage all uploaded code for projects that use this Scan Server. Once the Scan Server is added to the Code Insight system, this field cannot be edited.
	Actions for scan server entry	Select the appropriate icon for the desired action: ● The ✎ (**Edit**) icon to edit the configuration for the given Scan Server. The **Scan Server** dialog is opened, enabling you to make edits. ● The ↻ Refresh icon (visible when you hover over the Scan Server entry) to refresh the Scan Server entry, including its status.
Action	Use the following button to add a new Scan Server to the Code Insight system—that is, that is, identify the server to the Code Insight Core Server to make it available for scanning purposes. To add a Scan Server, ensure that it has been installed and is running. See the *Code Insight Installation and Configuration Guide* for instructions for installing and starting a Scan Server.	
	Add Scan Server	Click this button to add a new Scan Server. The **Scan Server** dialog is displayed.

See Also
Scan Server Dialog

Security Vulnerabilities Window

The **Security Vulnerabilities** window lists all the security vulnerabilities currently associated with a specific inventory item or component version and provides details and lookups for each vulnerability. You access this window by clicking the **Vulnerabilities** bar graph displayed for any of the following entities—if that entity currently has security vulnerabilities:

- A specific inventory item in the **Analysis Workbench** or in **Project Inventory**
- A component version in an inventory item’s **Lookup Component** window
- A component version in the **Global Component & License Lookup** window

The following describes the properties shown for each security vulnerability listed in the **Security Vulnerabilities** window. These properties are not editable.

By default the list is sorted on the **CVSS <version> Score** column in descending order.

Table 8-58 ▪ Description of Security Vulnerability Properties

Property	Description
Source	The security database system or research organization that has reported the security vulnerability (for example, **NVD**, **Secunia**, or another research organization).
ID	The ID of the security vulnerability in the format of the advisory organization that reported it: ● For a vulnerability reported by the NVD, the ID uses the CVE (Common Vulnerabilities and Exposures) format. ● For a vulnerability reported by Secunia Research, the ID uses the SA (Secunia Advisory) format. ● For a vulnerability reported by another research organization, the ID uses the format specific to that organization. Optionally, you can click the hyperlinked CVE ID to open its external third-party web page on a separate tab. The web page can provide referenced CVEs (those not explicitly mapped to the component version but indirectly related to it) and other useful information for researching the vulnerability. The list is sortable on this column. See Grid Control for details.
Description	A description of the security vulnerability pulled from the source. A **More**/ **Less** link enables you to view the full description and then collapse it as needed.
Severity	The severity of the vulnerability (**CRITICAL**, **HIGH**, **LOW**, or other). The level of severity depends on the scoring system used and the vulnerability’s actual CVSS score. For details about the relationship between severity levels and CVSS scoring systems, see Understanding Severity Levels for Security Vulnerabilities.

Table 8-58 ▪ Description of Security Vulnerability Properties (cont.)

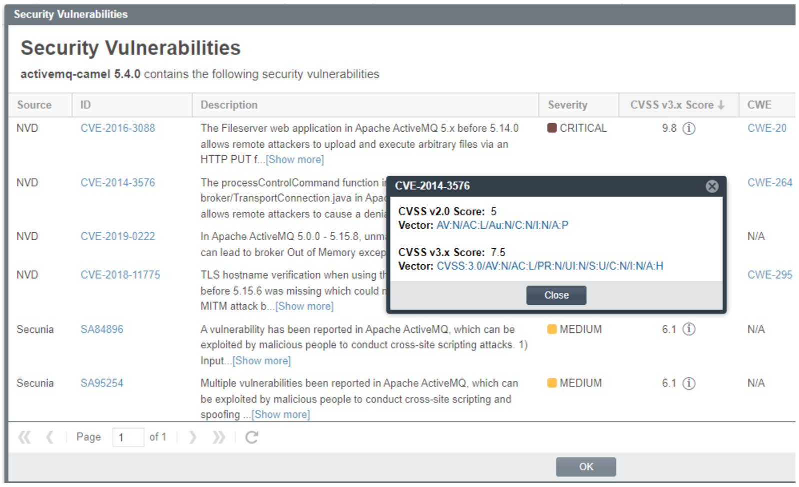
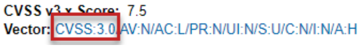
Property	Description
CVSS <version> Score	The vulnerability’s CVSS (Common Vulnerability Scoring System) score, which can have two different values depending on the scoring system used to calculate it—either CVSS v2.0 or v3.x (specified in the property label). For details about scoring system versions, see Understanding Severity Levels for Security Vulnerabilities. In some cases, the advisory CVSS score (or other type of vulnerability score) is unknown for a vulnerability because it has not been provided by the supplier. Code Insight reports the score for such a vulnerability as **N/A**. If you click the ⓘ icon next to the score, the resulting pop-up lists the v3.x and the v2.0 score for the vulnerability, along with the vector value for each score:  Security Vulnerabilities **Security Vulnerabilities** **activemq-camel 5.4.0** contains the following security vulnerabilities Source ID Description Severity CVSS v3.x Score ↓ CWE NVD CVE-2016-3088 The Fileserver web application in Apache ActiveMQ 5.x before 5.14.0 allows remote attackers to upload and execute arbitrary files via an HTTP PUT f...[Show more] CRITICAL 9.8 ⓘ CWE-20 NVD CVE-2014-3576 The processControlCommand function i broker/TransportConnection.java in Apa allows remote attackers to cause a deni CWE-264 NVD CVE-2019-0222 In Apache ActiveMQ 5.0.0 - 5.15.8, unm can lead to broker Out of Memory excep N/A NVD CVE-2018-11775 TLS hostname verification when using th before 5.15.6 was missing which could r MITM attack b...[Show more] CWE-295 Secunia SA84896 A vulnerability has been reported in Apache ActiveMQ, which can be exploited by malicious people to conduct cross-site scripting attacks. 1) Input...[Show more] MEDIUM 6.1 ⓘ N/A Secunia SA95254 Multiple vulnerabilities been reported in Apache ActiveMQ, which can be exploited by malicious people to conduct cross-site scripting and spoofing ...[Show more] MEDIUM 6.1 ⓘ N/A Page 1 of 1 OK CVE-2014-3576 **CVSS v2.0 Score:** 5 **Vector:** AV:N/AC:L/Au:N/C:N/I:N/A:P **CVSS v3.x Score:** 7.5 **Vector:** CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H Close The associated **Vector** value for a v3.x vulnerability has the specific score version—3.0 or 3.1—embedded in the value.  **CVSS v3.x Score:** 7.5 **Vector:** CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:N/A:H The **Vector** value is available only if the vulnerability is found in the NVD. (Otherwise, the **Vector** field shows **N/A**.) This hyperlinked value for this field is a compressed textual representation of the values used to derive the score. When you click the link, the appropriate NVD Common Vulnerability Scoring System Calculator is opened, showing you the environmental and temporal factors that determined the score. You can use the calculator to tweak these factors as necessary to calculate another score that is more realistic for your software product. (Instructions are provided with the calculator.) This adjusted score can then be used internally to direct your review and remediation processes. The list is sortable on this column. See Grid Control for details.

Table 8-58 ▪ Description of Security Vulnerability Properties (cont.)

Property	Description
CWE	The vulnerability's CWE (Common Weakness Enumeration) type. Optionally, you can click the hyperlinked ID to open the CWE web page describing this type. (CWE types are developed by a community of national cyber-security organizations.)
Published	The date on which the vulnerability was originally published, as captured from its source (NVD, Secunia, or another advisory). [1]
Last modified	The date on which the vulnerability was last revised, as captured from its source (NVD, Secunia, or another advisory). If vulnerability has never been revised, the field displays the vulnerability's published date. [1]
Resources	If the **Patches** link displayed, click it to open a pop-up window that lists the patches currently available to fix the vulnerability. From the popup, you can click the hyperlinked URL for any patch listed to open its external third-party web page on a separate tab. The web page provides information about the patch and how to execute it. If no patches are available for the vulnerability, **N/A** is displayed in this column.
Suppress	(Available only to Code Insight System Administrators) Click this button next to a given security vulnerability to suppress—that is, hide—the vulnerability for selected component versions. For more information, see Suppressing/ Unsuppressing Security Vulnerabilities.

[1] If you have migrated from a pre-2021 R3 Code Insight release to the current release, you must run an Electronic Update to obtain the latest date information.

Grid Control

Do the following to manage the grid.

- Control the column presentation:
    - Click the up ↑ (or down arrow) in the **ID** or **CVSS <version> Score** column header (or select the appropriate sorting order from the header's dropdown menu) to sort the vulnerability list in ascending or descending order.

      By default, the list grid is sorted on the **CVSS <version> Score** column in descending order.
    - Click the dropdown menu in any column header to select the columns you want to display or hide in the grid.
- The grid is paginated with each page having 50 records. Use the navigation icons at the bottom of the grid to move between next or previous pages or to a specific page.

Select a New Project Contact Page

The **Select a new project contact** page lets you change the Project Contact of the current project. The Project Contact, initially the project creator, is the default contact for all task-workflow notifications generated during the inventory review process. That is, if a Legal, Security, or Development contact has not been explicitly assigned to the project through system Project Defaults or at the project-settings level, that contact defaults to the Project Contact. Additionally, the Project Contact is the default contact for any “miscellaneous” tasks created during an inventory review.

The project creator is the initial Project Contact. The Project Contact user is initially assigned to all project roles (but can be removed from these roles as needed). This user can also reassign the Project Contact to a different user. When Project Contact is reassigned to another user, that user is assigned to the same roles as the previous Project Contact to ensure a continuation of the same permissions.

The following fields are used to reassign the Project Contact:

Table 8-59 ▪ Select a New Project Contact Page

Column/Field	Description
List of Users	The names of all the users in the system are listed in this field. Select a name and click **Apply** to change the Project Contact.
Apply	Click this button to assign the selected user as Project Contact.
Cancel	Click this button to cancel changes without saving.

Summary Tab

The **Summary** tab for a project allows you to add and edit users who can work in Code Insight, view scan settings and status, generate reports, and manage projects. The page contains the following fields:

Table 8-60 ▪ Project Summary Tab Fields

Category	Column/Field	Description
Start Scan button	Click this button start or schedule a Scan Server scan on the project. If a scan is running on another project, your scan is queued and will automatically run based on queue order. (A temporarily inactive Scan Server will also cause your scan to be queued. The scan will automatically start based on queue order once the server is running again.) If a scan on the your project has already been scheduled or is running, this button is disabled; you must wait until the scan is complete before you can schedule another one for the project. (This button is also disabled if the Scan Server is totally disabled. See Actions to Take When the Start Scan Button is Disabled for details.) For the initial scan of a project codebase, a full scan is run. For subsequent rescans, an incremental scan is run by default (except when a certain event causes a full rescan). However, if necessary, you can force a full rescan by clicking the down arrow on the button and selecting **Full Rescan**. For more information about scans, see Scanning the Codebase (Server Scans) and Rescanning Your Codebase (Server Scans Only)	
Manage Project button	Click this button to view and select from a menu of options used to manage the current project. For more information, see Managing Projects.	
Upload Project Codebase button	Select this button to upload a codebase for the current project. For instructions, see Uploading a Project Codebase (for Server Scans).	
Project Details	These field describe the project attributes. You can edit these details using the **Manage Project | Edit Project** and **Manage Project | Edit Project Users** options available on this **Summary** tab. ***Note ▪*** *For inventory-only projects migrated from Code Insight 2020 R2 or earlier, a legacy attribute,* ***Project Type****, will also display. However, for migrated standard projects, this attribute is no longer required and therefore does not display. See also Legacy Projects.*	
	Name	The name and ID of the selected project.

Table 8-60 ▪ Project Summary Tab Fields (cont.)

Category	Column/Field	Description
	Project Contact	The hyperlinked name of the user who is the main point of contact for the project. Initially, the Project Contact is the project creator but can be assigned to another user (see Changing the Project Contact). You can click the user name to open your default email program to send an email to the Project Contact. By default, all Miscellaneous tasks created for project inventory are assigned to the Project Contact. Additionally, the Project Contact is automatically designated the creator of all manual review and remediation tasks automatically created by project policy.
	Legal Contact	The hyperlinked name of the legal contact assigned to tasks created to review legal issues in the project inventory (for example, inventory that do not meet your site's legal policies). Click the name to open your default email program to send an email to the contact. For details on changing the legal contact for the project, see Updating Inventory Review and Remediation Settings for a Project. If no changes are made to the initial system default for the legal contact (which is set to the Project Contact) or if the automated legal-review process is disabled, this value shows the Project Contact name.
	Security Contact	The hyperlinked name of the default security contact assigned to tasks created to review security issues in the project inventory. Click the name to open your default email program to send an email to the contact. For details on changing the security contact for the project, see Updating Inventory Review and Remediation Settings for a Project. If no changes are made to the initial system default for the security contact (which is set to the Project Contact) or if the automated security-review process is disabled, this value shows the Project Contact name.
	Developer Contact	The hyperlinked name of the default development contact assigned to remediation tasks created to take action on code-related issues in the project inventory. Click the name to open your default email program to send an email to the contact. For details on changing the default development contact, see Updating Inventory Review and Remediation Settings for a Project. If no changes are made to the initial system default for the developer contact (which is set to the Project Contact) or if the automated remediation process is disabled, this value shows the Project Contact name.

Table 8-60 • Project Summary Tab Fields (cont.)

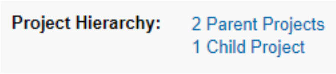
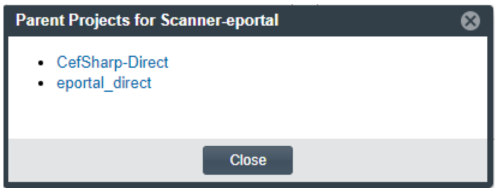
Category	Column/Field	Description
	Description	A description of the project, if provided in the project definition, appears in this field.
	Project Visibility	The visibility of the project: - **Public**—A project visible to all users in the system. Users assigned to roles for the project can perform various maintenance functions based on their project roles. - **Private**—A project visible to and accessible by the Project Contact and those users assigned to roles for the project.
	Project Risk	The project vulnerability risk value (**Low**, **Medium**, or **High**).
	Project Hierarchy	Links to the projects that have been defined as parent and child projects of the current project. These links provide a means to easily navigate to projects directly related to the current project. (Relationships between projects are established by the creation of project hierarchies, as described in Identifying Child Projects for a Project.)  Project Hierarchy: 2 Parent Projects 1 Child Project When you click a **Project Hierarchy** link, a dialog is displayed listing the direct links to the parent or child projects.  Parent Projects for Scanner-eportal - CefSharp-Direct - eportal_direct Click a link on the dialog to open the given child or parent project on its **Project Inventory** tab. From here you can navigate the project as needed.

Table 8-60 ▪ Project Summary Tab Fields (cont.)

Category	Column/Field	Description
	Project Status	The current status of the project that can be manually updated through the **Manage Project | Edit Project** menu option available on this tab. Available status types include: ● **Not Started**—Indicates that the project scan results are not available for manual analysis. This value automatically defaults when the initial project scan has not been run or when the initial scan fails. However, you can manually change this status. ● **Analysis in Progress**—Indicates that the project scan results are available for manual analysis, or a manual analysis is underway. This value is automatically set when the initial project scan is complete. (For a project import, however, the status of the target project is retained.) You can manually change this status. ● **Analysis Completed**—Indicates that manual analysis is finished; and the published inventory is ready for legal, security, or software-engineering review. (You must manually set this value.) ● **Project Complete**—Indicates that the project analysis and review processes are complete, and notices content for all OSS and third-party software is finalized. (You must manually set this value.) For more information, see Editing the Project Definition and General Settings and Edit Project: General Tab.
	Policy Profile	The name of the review policy profile associated with this project. Click the **Information** icon to open a read-only version of the Policy Details Window for the currently assigned policy profile. A policy profile contains a set criteria used to perform an automatic review of inventory items triggered during certain events, such as inventory publication, creation, and updates, project scans, and forced automatic reviews (see below). The criteria is based on OSS or third-party component versions, licenses, or security vulnerabilities. Inventory items that meet any of the profile's policy criteria can be automatically approved or rejected. For more information, see Managing Policies to Automatically Review Inventory. To force an automatic review (based on the policy criteria) of all published inventory in the project, click the **Apply Policy** button. This operation provides a convenient way to apply changed policy criteria across project inventory. Refer to Forcing an Automatic Review of All Inventory in a Project for details.

Table 8-60 ▪ Project Summary Tab Fields (cont.)

Category	Column/Field	Description
	Provenance	If this project is the result of a project-copy or project-branching operation, the hyperlinked name of the source project from which this project is derived. The name of the source project has the following format: **Copied from** <sourceProjectName> **(Id:** <sourceProjectID>**)** ***Note*** **▪** *Projects are copied using the Code Insight Project Copy feature and are branched using the* ***Branch Project*** *wizard.* When you click the link, the **Summary** tab of the source project is opened. (If the source project has been deleted, a message is displayed, stating that the project no longer exists. Once clicked, the link for the deleted project is permanently disabled.) For projects that are not the result of a project-copy or project-branching operation, this field shows **N/A**.
Custom Fields	The **Custom Fields** pane lists the fields that were defined specifically for projects at your site. These fields provide users with helpful information that supplements the information provided by the project fields standard to Code Insight. The values shown for these fields pertain to the current project. If no value has been defined for a given field, a hyphen is displayed in place of the value. While these fields are not editable from this location, you can do the following to manage your view of the field: ● If the value for a specific field is cut off within the **Custom Fields** pane, click the **Show more** link to view the entire value in a pop-up. ● Click the ⓘ icon (if available) in the upper-right corner of a field to obtain more information about the field. ***Note*** **▪** *If no custom fields for projects have been configured for your site or, if custom fields have been defined but are currently not available for display, this pane shows no fields.*	
Scan Settings	The following fields show scan configuration details. You can edit these details on the **Scan Settings** tab accessed using the **Manage Project | Edit Project** option available on this tab.	
	Scan Profile	The name of the scan profile associated with this project. Click ⓘ to view the details of the scan profile.

Table 8-60 ▪ Project Summary Tab Fields (cont.)

Category	Column/Field	Description
	Scan Paths	The absolute path for each scan folder for the project. A given scan folder contains files for either: ● A codebase scanned by the Scan Server. ● A codebase scanned by a Code Insight scan-agent plugin on a remote Engineering system. (The results of the remote scan are sent to the project.) For more information, see Performing Remote Scans. Click ⓘ to view the details about Scan Server or the scan-agent plugin.
Scan Status	The following fields provide information about current and historical scans for this project. For more information about scans, see About Code Insight Scans.	
	Scan Server Status	The server scan status for this project: ● If you have started a server scan for this project, but the scan has been placed in queue, this field shows “Scan scheduled”. The **Scan Progress** field provides a link to view the scan queue and other details (see the “Scan Progress” description below). ● If the server scan you scheduled for this project is running, this field shows “Project being scanned”. The **Scan Progress** field keeps track of the number of files that have been scanned. ● When no scan is currently running for this project, this field shows “No scan scheduled” and provides a hyperlinked **here** to schedule another scan. (However, if the Scan Server is disabled, this link is disabled. See Actions to Take When the Start Scan Button is Disabled.) The **Scan Progress** field is not available when the “No scan scheduled” status is in effect. ***Note ▪*** *Under certain circumstances, the **Scan Server Status** field might not update quickly enough to reflect the “Scan scheduled” or “Project being scanned” status. However, if a scan on the current project is indeed already in queue or running, an attempt to click the field’s “here” link to schedule a scan will result in an error message, stating that you cannot start another scan on the project. For your reference, the message also provides the task ID for the currently queued or running scan. (This ID can be used with the **Get Scan Status** API to check the scan status outside of the UI when necessary.)*

Table 8-60 ▪ Project Summary Tab Fields (cont.)

Category	Column/Field	Description
	Scan Progress	(Available only when a server scan for the project is scheduled or is running) The progress of the scan as follows: ● If the scan has been placed in queue, this field shows "In Scan Queue" and provides a **Show Details** link to open the **Scan Server Status** window. This window identifies the scan server, shows the project currently being scanned by the server, lists the other project scans (if any) currently waiting in queue order (up to 25 scans), and provides an email link for the Project Contact of each project listed. You cannot sort or reorganize the queue list. ● If the scan you scheduled is running, this field keeps track of the number of files that have been scanned against the total number of files to scan in the project.
	Last Server Scan	The final status of the last server scan for the project and a statistical summary of files, disk space, and lines of code scanned. The following are available scan statuses: ● **Completed**—The scan succeeded with no warnings during the scan or the analysis phase. This message appears on the screen in green. ● **Completed with warnings**—The scan succeeded but the analysis phase produced warnings. For more information, check **scanEngineDetail** log for the Scan Server. ● **Failed**—The scan failed. This status appears on the screen in red. For more information, see Scan Failure Reasons and Troubleshooting Measures.
	Past Server Scans	Click the hyperlinked term **here** to view a history of the server scans performed for the project. If a server scan has not yet been performed for the project, the list will be empty.

Table 8-60 ▪ Project Summary Tab Fields (cont.)

Category	Column/Field	Description
	Last Remote Scan	Information about the most recent scans run by remote scan agents. The **Scan Summary** section shows the following combined totals for the most recent scan run by *every remote scan agent* associated with the project: ● The combined total of scanned codebase files across all scan agents associated with the project. ● The combined size of all agent-scanned codebases. The field also shows the following pertaining to the scan results for the *most recently run remote scan only*: ● The status for the import of these results (“Completed” or “Failed”) into Code Insight ● The timestamp for the import For information about remote scans, see Performing Remote Scans. ***Note ▪** If the project is a newly migrated inventory-only project from Code Insight 2020 R2 or earlier, only the statistics for the scan agent that last performed a scan (in the previous Code Insight version) are shown.*
Project Data	This section provides the status of the most recent (or currently running) job to export the current project’s data to a JSON file or to import exported data into the project.	

Table 8-60 ▪ Project Summary Tab Fields (cont.)

Category	Column/Field	Description
	Export Project	The status of the project’s most recent (or currently running) data export job. (For more information about project exports, see Exporting Project Data.) Status statements specify that the export process is either “in queue” or “in progress” or that it “completed” (at the specified date and time) or “failed”. If no data export was ever performed on the current project, the status states “No project data has been exported”. ***Note ▪*** *The “No project data...” message is also displayed for projects when a Code Insight 2023 R4 or later instance has been upgraded, but the `exports` folder has not been copied to the new instance.* For more information about the data export job listed in this field, including who triggered the job and any errors encountered during the job, view the **Jobs** queue. (Use the instructions in Monitoring the Code Insight Jobs Queue to access and monitor the queue.)
	Import Project	The status of the project’s most recent (or currently running) data import job. (For more information about project imports, see Importing Project Data.) Status statements specify that the import process is either “in queue” or “in progress” or that it “completed” (at the specified date and time) or “failed”. If no data import was ever performed on the project, the status states “No project data has been imported”. For more information about the data import job listed in this field, including who triggered the job and any errors encountered during the job, view the **Jobs** queue. (Use the instructions in Monitoring the Code Insight Jobs Queue to access and monitor the queue.)
Buttons	The following buttons are used to upload a codebase for scanning, perform a scan or rescan, and edit current project settings.	
	Start Scan	Initiates the initial scan or a rescan (incremental by default) on the project codebase. Alternatively, click the dropdown and select **Full Rescan** to force a full rescan or the codebase. For more information about scans, see Scanning the Codebase (Server Scans) and Rescanning Your Codebase (Server Scans Only).

Table 8-60 ▪ Project Summary Tab Fields (cont.)

Category	Column/Field	Description
	Manage Project	Opens a menu that provides options to edit the project's settings, manage its users and data, and globally apply Third-Party Notices content across all the project's inventory. The menu options available depend on the project role(s) you have been assigned. For more information, see Managing Projects.
	Upload Project Codebase	Opens the **File Upload** dialog. From here, you select the codebase archive to upload for the project to the Scan Server. You also configure how the upload handles archive expansion and any codebase files already installed for the project on the Scan Server. If the current project is not associated with a Scan Server or if the associated Scan Server is disabled, this button is disabled. For more information about uploading a project codebase, see Uploading a Project Codebase (for Server Scans). ***Note ▪*** *Instead of (or in addition to) uploading a codebase for a project to the Scan Server, you can configure the project to automatically synchronize a codebase from a remote repository to the Scan Server. For more information, see Configuring Source Code Management.*

See Also

About Code Insight Projects
Scanning the Codebase (Server Scans)
Performing Remote Scans
Editing the Project Definition and General Settings
Edit Project: General Tab
Projects Pane and Associated Dashboard
Uploading a Project Codebase (for Server Scans)
Exporting and Importing Project Data

Suppress Vulnerability Window

The **Suppress Vulnerability** window is displayed when you click the **Suppress** button for a given vulnerability on the **Security Vulnerabilities** window. (For more information about accessing this window and about suppressing vulnerabilities, see Suppressing a Security Vulnerability.)

The **Suppress Vulnerabilities** window enables you to suppress the given security vulnerability for one or more (or all) versions of the OSS or third-party component with which the vulnerability is associated. You might want to suppress a vulnerability, for example, if the vulnerability has proven to be a "false positive" (that is, is associated with an incorrect component version) or if remedial steps have been taken to protect your code against the vulnerability.

Vulnerability suppression takes place at the system level in Code Insight. Once suppressed, the vulnerability is no longer published in reports, counted in vulnerability totals at the project, inventory, and component levels, or automatically associated with inventory during future project scans in your Code Insight instance. For a complete description of the impact of suppressing a vulnerability, see Effects of Suppressing a Security Vulnerability.

Vulnerability suppression is performed by a Code Insight System Administrator only, who can also monitor a list of suppressed vulnerabilities and unsuppress vulnerabilities as needed.

The follow describes the fields and features on the **Suppress Vulnerability** window that enable you to suppress a given vulnerability.

Table 8-61 ▪ Suppress Vulnerability Window

Category	Description
Vulnerability Id	(Not editable) The ID assigned to the vulnerability by the source that reported it (see the next field). Optionally, you can click the hyperlinked CVE ID to open its external third-party web page on a separate tab. The web page can provide referenced CVEs (those not explicitly mapped to the component version but indirectly related to it) and other useful information for researching the vulnerability.
Source	(Not editable) The advisory system that reported the vulnerability (for example, NVD or Secunia).
Severity	(Not editable) The level of security risk that this vulnerability can have on your software. The advisory system uses the vulnerability's CVSS score to set the severity. See Understanding Severity Levels for Security Vulnerabilities.
CVSS v3.x (or v2.0) Score	(Not editable) The vulnerability's CVSS score as determined by the advisory system. Depending on your Code Insight configuration, this score is in either CVSS 3.x or CVSS 2.0 format. For more information, see Understanding Severity Levels for Security Vulnerabilities. For a vulnerability found in the NVD, the UI also provides access to a CVSS calculator (provided by NVD). Using this calendar, you can tweak the factors that determined the NVD-based score to calculate another score that is more realistic for your product. This score can then be used internally to direct your review and remediation processes. For information about accessing the CVSS calculator, see the **CVSS <version> Score** description in Security Vulnerabilities Window.
Description	(Not editable) The vulnerability description, as captured from the advisory system.
Affected Component	(Not editable) The OSS or third-party component that is impacted by this security vulnerability.

Table 8-61 ▪ Suppress Vulnerability Window (cont.)

<table>
<tr><th>Category</th><th colspan="2">Description</th></tr>
<tr><td><b>Version Scope</b></td><td colspan="2">(Required) Select the scope of component versions to which the vulnerability suppression will apply.
● <b>Specific Version(s)</b>—One or more component versions that you choose from the <b>Select Version</b> dropdown list (which is enabled only when this option is selected). Note that the dropdown list will show only those versions for which the vulnerability is currently unsuppressed.
By default, this option is initially selected, and the <b>Select Version</b> field shows the component version for the current inventory item.
● <b>All Current Versions</b>—All component versions for which the vulnerability is currently unsuppressed.</td></tr>
<tr><td><b>Select Version(s)</b></td><td colspan="2">(Enabled and required when <b>Version Scope</b> is <b>Specific Version(s)</b>) From the dropdown list (showing all <i>unsuppressed</i> versions currently affected by the vulnerability), select each version for which you want the vulnerability to be suppressed.
By default, the component version for the current inventory item is initially specified.
If necessary, you can remove any of your version selections by clicking the small ✖ icon to the right of the version.</td></tr>
<tr><td><b>Select Reason</b></td><td colspan="2">(Required) Select the reason for suppressing the vulnerability for this component version:
● <b>False-positive</b>—The vulnerability was incorrectly associated with the component version and hence does not apply to the version.
● <b>Remediated</b>—The risk posed by the vulnerability on the component version has been addressed or fixed.
● <b>Other</b>—Another reason.</td></tr>
<tr><td><b>Suppression Remarks</b></td><td colspan="2">(Required) Enter all additional information pertinent to the suppression of the vulnerability for this component version.</td></tr>
<tr><td rowspan="3"><b>Actions</b></td><td colspan="2">The following buttons enact or discontinue the vulnerability suppression process.</td></tr>
<tr><td><b>Suppress</b></td><td>(Enabled when all required fields have been completed) Click to suppress the security vulnerability for the given component version. Then click <b>OK</b> in the pop-up to acknowledge that vulnerability has been suppressed.</td></tr>
<tr><td><b>Close</b></td><td>Close window without saving your input.</td></tr>
</table>

Suppressed Versions of <component> for <vulnerability> Window

This pop-up window is displayed when you click ⓘ next the **Affected Versions** value for a given suppressed security vulnerability listed on the Suppressed Vulnerabilities Tab. The window lists the component versions for which the vulnerability was suppressed and, for each version, provides details about the suppression (who suppressed the vulnerability and when, why the vulnerability was suppressed for this version, and more).

The following table describes the details provided for each component version, as well as describes the navigation features of the window.

Table 8-62 ▪ Vulnerability Suppression Details for Each Impacted Component Version

Category	Column/Field	Description
Suppression details per component version	The following describes details related to the suppression of the security vulnerability for the given component version. These details are not editable.	
	Version Name	A version of the given component for which the vulnerability was suppressed.
	Reason	The reason why the vulnerability was suppressed for this component version. (This information was entered by the System Administrator at the time of suppression.) ● **False-positive**—The vulnerability was incorrectly associated with the component version and hence does not apply to the version. ● **Remediated**—The risk posed by the vulnerability on the component version has been addressed or fixed. ● **Other**—Another reason. (Check the **Remarks** field for a possible reason description.)
	Remarks	Any additional notes about the suppression of the vulnerability for this component version, as entered by the System Administrator at the time of suppression.
	Suppressed By	The user ID of the System Administrator who suppressed the vulnerability for this component version.
	Suppressed Date	The date and time that the vulnerability was suppressed for this component version.

Table 8-62 ▪ Vulnerability Suppression Details for Each Impacted Component Version (cont.)

Category	Column/Field	Description
Actions	The following buttons and icons enable you to navigate and manage the list of component versions on the window.	
	↻	Refresh the data in the window.
	Page controls	Move to the next or previous page or to the first or last page of the window; or enter a specific page number in the **Page** field. Note that the default page size is 100 component version records.
	Close	Exit the **Suppressed Vulnerabilities** tab.

Suppressed Vulnerabilities Tab

The **Suppressed Vulnerabilities** tab on the **Data Library** page lists the security vulnerabilities currently suppressed in your Code Insight instance. (The data is listed in a grid format.) This tab is visible to only Code Insight System Administrators. For more information about accessing this tab, see Viewing Suppressed Security Vulnerabilities.

For a newly installed Code Insight instance or an pre-2021 R3 instance migrated to the current instance, this page initially shows no suppressed security vulnerabilities. (However, the tab will list any vulnerability you subsequently suppress.)

The **Suppressed Vulnerabilities** tab provides the following information and features:

Table 8-63 ▪ Suppressed Vulnerabilities Tab

Category	Column/Field	Description
Filter by	These fields enable you to filter the list of suppressed vulnerabilities. Select the filter type, either **Vulnerability Id** or **Component Name**, from the dropdown list; and then enter the string by which to filter the list. For example, if you select **Component Name** and enter the string open, the list will filter to those suppressed vulnerabilities associated with a component whose name contains "open".	
Details for each suppressed vulnerability	The following describes the details of each suppressed vulnerability listed in the grid. These details are not editable. The **Action** column includes the button used to unsuppress the given vulnerability.	

Table 8-63 ▪ Suppressed Vulnerabilities Tab (cont.)

Category	Column/Field	Description
	Vulnerability ID	The ID assigned to the vulnerability by the advisory system that reported it. Click ⓘ next to the ID to display a pop-up containing details about the vulnerability. The details include: ● **Vulnerability ID**—The ID assigned to the vulnerability by the source that reported it (see the next field). ● **Source**—The advisory system that reported the vulnerability (for example, NVD or Secunia). ● **Severity**—The level of security risk that this vulnerability can have on your software. The advisory system uses the vulnerability’s CVSS score to set the severity. See Understanding Severity Levels for Security Vulnerabilities. ● **Score**—The vulnerability’s CVSS score as determined by the advisory system. Depending on your Code Insight configuration, this score is in either CVSS 3.x or CVSS 2.0 format. For more information, see Understanding Severity Levels for Security Vulnerabilities. For a vulnerability found in the NVD, the UI also provides access to a CVSS calculator (provided by NVD). Using this calculator, you can tweak the factors that determined the NVD-based score to calculate another score that is more realistic for your product. This score can then be used internally to direct your review and remediation processes. For information about accessing the CVSS calculator, see the **CVSS <version> Score** description in Security Vulnerabilities Window. ● **Description**—A description of the vulnerability captured from the advisory system. You can sort on this column alphabetically in ascending or descending order. By default, the IDs are listed in ascending order.

Table 8-63 ▪ Suppressed Vulnerabilities Tab (cont.)

Category	Column/Field	Description
	Affected component	The OSS or third-party component that is impacted by the vulnerability.
	Affected versions	The one or more component versions for which the vulnerability is currently suppressed. If the versions are too numerous list in the grid, the value ends with "...". However, you can always mouse-over the value to see the entire list of versions for which the vulnerability is suppressed. Click ⓘ next to the value to display a pop-up window that shows suppression details for each listed version. For more information, see Suppressed Versions of <component> for <vulnerability> Window.
	Action	Click **Unsuppress** to unsuppress the vulnerability for one or more of the component versions for which it is suppressed. The Unsuppress Vulnerability Window is displayed to walk you through the process. When you return to this window, the component versions for which you unsuppressed the vulnerability are no longer displayed. (If all the component versions for which the vulnerability was previously suppressed are now unsuppressed, the vulnerability is no longer listed on this window.)
Actions	The following buttons and icons enable you to navigate and manage the **Suppressed Vulnerabilities** tab.	
	↻	Refresh the vulnerability data on the tab.
	Page controls	Move to the next or previous page or to the first or last page on the tab; or enter a specific page number in the **Page** field. Note that the default page size is 100 vulnerability records.
	Close	Exit the **Suppressed Vulnerabilities** tab.

System Settings Tab

The **System Settings** tab on the **Administration** page is used to define settings that configure your Code Insight system. The tab provides the following configuration settings:

Table 8-64 ▪ System Settings Tab

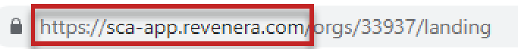
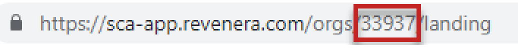
Section	Field	Description
Security Vulnerability Options	Select the CVSS (Common Vulnerability Scoring System) version—**CVSS v3.x** (3.0 and 3.1) or **CVSS v2.0** in which to display security vulnerability scores and severities in the Code Insight Web UI. Initially, **CVSS v 2.0** is the default. If you switch versions, the CVSS scores and severity values displayed for vulnerabilities will be impacted, as will policies based on these values. For more information, see Security Vulnerabilities Associated with Inventory and Managing Policies to Automatically Review Inventory.	
Configure SBOM Insights	If your site intends to export project inventory data to SBOM Insights (a Revenera SCA product), complete the following fields that configure the Code Insight connection to SBOM Insights. Each of these fields is required to create the connection. A **Test Connection** button is available in this section to ensure the connection works. For more information, see "Configuring Code Insight for Exports to SBOM Insights" in the *Code Insight Installation & Configuration Guide*.	
	SBOM Insights URL	Provide the URL for your SBOM Insights instance, as in this example: `https://sca-app.revenera.com` This URL is shown in the address on your browser once you log into SBOM Insights, as shown in this example:  https://sca-app.revenera.com/orgs/33937/landing
	Organization ID	Provide the Organization ID in SBOM Insights to which the Code Insight data will be exported. This ID is shown in the address on your browser once you log into SBOM Insights, as shown in this example:  https://sca-app.revenera.com/orgs/33937/landing
	API Refresh Token	Enter the API refresh token generated in SBOM Insights. This token is required to give Code Insight access to SBOM Insights. Instructions for generating this token are found here in the SBOM Insights user documentation.

Table 8-64 ▪ System Settings Tab (cont.)

Section	Field	Description
	Test Connection	Once you have completed the fields above, click this button to determine whether a connection can be successfully established between Code Insight and SBOM Insights. ● If a connection is established, a “Test Connection Successful” message is displayed in the upper right of the screen. ● If the connection fails, an error message is displayed, explaining the error. Edit the connection information as needed and test the connection again.
	Save	Click this button to store the connection properties in the Code Insight database. This action also tests the connection. If the connection fails, an error message is displayed and the data is not saved. Edit the information as needed and try to save again.
Custom Fields sections	The **Custom Fields for Inventory** and **Custom Fields for Projects** sections provide a means for the System Administrator to manage custom fields for inventory and projects. Within each section, the administrator can create new custom fields, view the list of existing fields, and edit field definitions. By default, a custom field is available for all inventory items or all projects in your Code Insight system. Users with appropriate permissions can then update the field’s value for a specific inventory item or project. (Note, however, that the System Administration can limit the field’s availability in Code Insight.) A maximum for five custom fields for inventory and thirty custom field for projects can be created. A custom field cannot be deleted but can be disabled (and then re-enabled as needed). ***Note** ▪ After a fresh installation of Code Insight, no custom fields exist until the System Administrator creates them.* The remainder of this table describes the mechanisms and attributes used to define and manage custom fields in the **Custom Fields for Inventory** and **Custom Fields for Projects** sections. For details about managing custom fields, see “Creating and Managing Custom Fields for Inventory” and “Creating and Managing Custom Fields for Projects” in the *Code Insight Installation & Configuration Guide*.	
	Add Field button	Within the **Custom Fields for Inventory** or **Custom Fields for Projects** section, click this button to open the **Add Custom Field** window, enabling you to create a custom field. Once the field is defined and saved, it is added to the list of custom fields displayed in the section.

Table 8-64 ▪ System Settings Tab (cont.)

Section	Field	Description
	Update Field icon	To edit the definition of an existing custom field within the **Custom Fields for Inventory** or **Custom Fields for Projects** section, click the ✎ icon in the **Action** column for the field in the list. The **Edit Custom Field** dialog is opened, enabling you to change the field’s attributes. Once saved, the changes are reflected in the attributes for that field in the list.
	Attributes for custom fields	The following attributes are used to define a custom field in either the **Custom Fields for Inventory** or **Custom Fields for Projects** section. These attributes are also displayed for each field in the list of existing custom fields in the section. ***Note ▪** When setting up the custom **SBOM Bucket Name** project field required for exporting inventory data to SBOM Insights, you must specify certain attribute values. See “Configuring Code Insight for Exports to SBOM Insights” in the Code Insight Installation & Configuration Guide for instructions.*
		Enabled The attribute controlling whether the custom field is activated in Code Insight. ● **Yes**—The field will be activated and made available in Code Insight across all projects. Use the **Visible in UI** attribute (see next) to determine whether the field will be displayed in both the Code Insight REST interface and the Web UI or only in the REST interface. ● **No**—The field will not be available in Code Insight. All other attributes defined here are ignored (until the field is enabled).

Table 8-64 ▪ System Settings Tab (cont.)

Section	Field		Description
		Visible in UI	The attribute determining whether the custom field is visible in both the Code Insight Web UI and the REST interface *or* in the REST interface only. (The custom-field locations in the UI for inventory items include the **Inventory Details** tab on the **Project Inventory** pane and the **Inventory Details** tab in the **Analysis Workbench**. For projects, the UI locations include a project's **Summary** tab and its **Summary** > **Manage Project** > **Edit Project** > **Custom Fields** tab.) ● **Yes**—The field is visible in both the Code Insight Web UI and the REST interface, enabling users to use either the UI or REST APIs to view and update the field's value. (Default) ● **No**—The field does not display in the Code Insight Web UI. Users must use REST APIs to view and update the field's value.
		Field Label	(Required) The name of the custom field. The maximum length is 30 characters.
		Field Type	(Available only when defining custom fields for projects) The data type of the custom field: ● **Text Field**—A text field that has a maximum of 128 characters. ● **Text Area**—A large text field that has a maximum of 512 characters. (Default) ● **Drop-Down List**—A dropdown list of multiple options from which a user selects one option. When you choose this field type, the **Drop-Down List Options** field is enabled to define the options (see the next description). ***Note ▪*** *All custom fields for inventory are automatically configured as Text Area fields that have a maximum size of 64K. The field type cannot be changed.*

Table 8-64 ▪ System Settings Tab (cont.)

Section	Field	Description
	Drop-Down List Options	(Available only when defining custom fields for projects *and* the **Field Type** is **Drop-Down List**) The field that defines and manages the options in the dropdown list. ● To add an option to the dropdown list, click **Add Item** next to the field. The **Add Item** pop-up is displayed, allowing you to create an option label up to 30 characters. ● To remove an existing option, click the **X** to the right of the option label. ● To edit an option label, remove the option and re-add the option with the updated label. No limit exists on the number of options the field can have.
	Help Text	Information that is displayed when a user selects the ⓘ icon for the field in the Web UI. Provide content that helps users enter an appropriate value for the field. For example, you might describe the purpose the field and the type of value it requires. If you specify text with **http://** or **https://**, the value will be hyperlinked. The maximum length is 150 characters. If this attribute is left blank, the ⓘ icon will not be available for the custom field in the Web UI.

See Also
Security Vulnerabilities Associated with Inventory
Managing Policies to Automatically Review Inventory
Policy Details Window
“Setting the Common Vulnerability Scoring System” in the *Code Insight Installation and Configuration Guide*
“Creating and Managing Custom Fields for Inventory” in the *Code Insight Installation and Configuration Guide*
“Creating and Managing Custom Fields for Projects” in the *Code Insight Installation and Configuration Guide*

Unsuppress Vulnerability Window

The **Unsuppress Vulnerability** window is displayed when you click the **Unsuppress** button for a given security vulnerability on the Suppressed Vulnerabilities Tab. (For information about accessing this window, see Viewing Suppressed Security Vulnerabilities.)

The **Unsuppress Vulnerabilities** window enables you to unsuppress the security vulnerability for one, some, or all of component versions for which it was previously suppressed. The unsuppression process takes place at the system level of Code Insight, affecting all projects in your Code Insight instance.

Basically this process reverses the effects of the vulnerability's previous suppression so that the vulnerability is once again available for processing. It can be published in reports, counted in vulnerability totals at the project, inventory, and component levels, and automatically associated with inventory during future project scans and rescans in your Code Insight instance. For a complete description of the impact on Code Insight when you unsuppress a vulnerability, see Effects of Unsuppressing a Security Vulnerability.

Only a Code Insight System Administrator only can unsuppress a vulnerability.

The follow describes the fields and features on the **Unsuppress Vulnerability** window that enable you to unsuppress a given vulnerability.

Table 8-65 ▪ Unsuppress Vulnerability Window

Category	Description
Vulnerability Id	(Not editable) The ID assigned to the vulnerability by the source that reported it (see the next field). Optionally, you can click the hyperlinked CVE ID in an entry to view the vulnerability details found on the NVD or other website: Vulnerability Id: CVE-2014-3576
Source	(Not editable) The research system or organization that reported the security vulnerability (for example, **NVD**, **Secunia**, or another advisory entity).
Severity	(Not editable) The level of security risk that this vulnerability can have on your software. The advisory system uses the vulnerability's CVSS score to set the severity. See Understanding Severity Levels for Security Vulnerabilities.
CVSS v3.x (or v2.0) Score	(Not editable) The vulnerability's CVSS score as determined by the advisory system. Depending on your Code Insight configuration, this score is in either CVSS 3.x or CVSS 2.0 format. For more information, see Understanding Severity Levels for Security Vulnerabilities. For a vulnerability found in the NVD, the UI also provides access to a CVSS calculator (provided by NVD). Using this calculator, you can tweak the factors that determined the NVD-based score to calculate another score that is more realistic for your product. This score can then be used internally to direct your review and remediation processes. For information about accessing the CVSS calculator, see the **CVSS <version> Score** description in Security Vulnerabilities Window.
Description	(Not editable) The vulnerability description, as captured from the advisory system.
Affected Component	(Not editable) The OSS or third-party component that is impacted by this security vulnerability.

Table 8-65 ▪ Unsuppress Vulnerability Window (cont.)

<table>
<tr><th>Category</th><th colspan="2">Description</th></tr>
<tr><td><b>Version Scope</b></td><td colspan="2">(Required) Select the scope of component versions for which you want to unsuppress the vulnerability.
● <b>Specific Suppressed Version(s)</b>—The one or more component versions that you choose from the <b>Select Version(s)</b> dropdown list (which is enabled only when this option is selected). Note that the dropdown list will show only those versions for which the vulnerability is currently suppressed.
By default, this option is initially selected.
● <b>All Suppressed Versions</b>—All component versions for which the vulnerability is currently suppressed.</td></tr>
<tr><td><b>Select Version(s)</b></td><td colspan="2">(Enabled and required when <b>Version Scope</b> is <b>Specific Suppressed Version(s)</b>) From the dropdown list, select each version for which the vulnerability should be unsuppressed. The dropdown list shows only those versions for which the vulnerability is currently suppressed.
If necessary, you can remove any of your version selections by clicking the small × icon to the right of the version.</td></tr>
<tr><td><b>Unsuppression Remarks</b></td><td colspan="2">(Required) Enter all additional information pertinent to the unsuppression of the vulnerability for the component version(s).</td></tr>
<tr><td><b>Actions</b></td><td colspan="2">The following buttons enact or discontinue the process of unsuppressing the vulnerability.</td></tr>
<tr><td></td><td><b>Unsuppress</b></td><td>(Enabled when all required fields have been completed) Click to unsuppress the security vulnerability for the specified component version(s). Then click <b>OK</b> in the pop-up to acknowledge that vulnerability has been unsuppressed.</td></tr>
<tr><td></td><td><b>Close</b></td><td>Close window without saving your input.</td></tr>
</table>

Users/Permissions Tab

The **Users/Permissions** tab on the **Administration** page allows you to add and edit users who can work in Code Insight. The tab contains the following columns and fields:

Table 8-66 ▪ Users/Permissions Tab

Column/Field	Description
Add User	Click to display the **Add User** dialog.

Table 8-66 ▪ Users/Permissions Tab (cont.)

Column/Field	Description
Manage Permissions	Click to display the **Manage Permissions** dialog used to assign the following permissions to users: System Administrator, Manage Policy, and Create Projects.
Login	Displays the login of each user that has been added.
First Name	Displays the first name of each defined user.
Last Name	Displays the last name of each defined user.
Email	Displays the email address of the user associated with the login.
Actions	This column contains the **Edit** ✎ icon. Click it to open the **Edit User** dialog, where you can edit information about the selected user.
Enter Search Criteria	Enter a string by which to filter the list of users. A full or partial match to any of the user details is allowed. Click ✕ to remove the filter.

See Also
Add User Dialog
Edit User Dialog
“Configuring Code Insight” in the *Code Insight Installation & Configuration Guide*

Versions for <component> Window

The **Versions for** <**component**> window lists each version for a given component, along with the version’s possible licenses and security vulnerability totals (by severity). You can interact with the vulnerability bar graph to examine details about each associated vulnerability and, if needed, to suppress vulnerabilities for the version. In essence, the window provides a means to determine which component versions have the least security risk. This information can help the user decide whether to replace the current inventory component version with another more secure one.

The **Versions for** <**component**> window is accessed in any of the following ways:

- When editing a given inventory item on the **Inventory Details** tab in the **Analysis Workbench**, click the **View all versions** link next to the **Component** field. (The inventory item must be of **Component** type to see the link.)
- When editing a given project inventory item in the **Edit Inventory** window (opened from the item’s **Inventory Details** tab on the **Project Inventory** tab), click the **View all versions** link next to the **Component** field.
- When performing a global exploration of components in the Code Insight Data Library from the **DATA LIBRARY** > **Global Component & License Lookup** > **Components** tab, click the **View Versions** icon in the **Actions** column for a given component. (The **DATA LIBRARY** option is found on the Code Insight main menu, accessed by clicking ☰ on the Code Insight web page).

The **Versions for** <**component**> window lists (in a grid format) each version for the given component, as identified in the Code Insight Data Library.

The columns in the list provide the following information for each version.

Table 8-67 ▪ Versions for <component> Window

Column	Description
Version	The component version.
Security Vulnerabilities	Security vulnerability totals for the version: ● The value **None** is displayed if no security vulnerabilities are associated with the version. ● If security vulnerabilities are associated with the version, a **Vulnerabilities** bar graph is displayed, showing vulnerability totals by severity. For instructions on how to interact with this graph to examine details about the associated vulnerabilities and, if needed, to suppress vulnerabilities for the version, see Examining Security Vulnerability Details.
License(s)	The licenses associated with this version.

You can also create a new version for the component by clicking the **Create Custom Version** button. SeeCreating Custom Component Versions for instructions.

Grid Control

You can do the following to manage the grid:

- To control column visibility, click the dropdown menu in any column header for an option to select the columns you want to display or hide in the grid.
- Use the navigation icons at the bottom of the grid to move between the next or previous pages or to a specific page number in the search results.
- Click the refresh ↻ icon to keep the data in the window current.

9

Code Insight User Roles and Permissions

The following sections provide a reference to the various user roles and permissions that Code Insight offers as a means to control user access to its functionality at your site:

- System Roles and Permissions
- Project Roles and Permissions
- Roles and Permissions to Manage the Review Task Flow

System Roles and Permissions

The following table lists the roles and associated permissions used to manage Code Insight at the system level. The initial Code Insight System Administrator (and any subsequent System Administrators) manages user accounts and assigns system-level roles to any of these users as needed. For more information, see “Managing Users” in the “Configuring Code Insight” chapter in the *Code Insight Installation and Configuration Guide*.

One user can be assigned to multiple system roles.

Table 9-1 ▪ System Roles and Permissions

			Roles		
			System Admin	Policy Manager	Project Creator
Responsibility	Permissions	Notes			
Administer Code Insight	Manage user accounts and permissions, create other system administrators, create policy managers, and allow all/or specified users to create projects			X	X
	Schedule or force Electronic Updates/Library Refreshes			X	X
	Configure an email server workflow notifications			X	X
	Configure LDAP users			X	X
	Configure Application Lifecycle (ALM) instances to manage inventory review tasks			X	X
	Configure Scan Servers and scan profiles			X	X
	Define global project defaults			X	X
	Determine the CVSS version used for security vulnerability reporting			X	X
	Create and manage custom fields for inventory and projects			X	X
	View Code Insight logs			X	X
	Suppress security vulnerabilities			X	X

Table 9-1 ▪ System Roles and Permissions (cont.)

			Roles		
			System Admin	**Policy Manager**	**Project Creator**
Manage polices for automating inventory review processes	Manage policies		X		X
	Force automatic review of inventory across all projects		X		X
Create projects	Create public and private projects	The user who creates a project automatically becomes the Project Contact for that project. (See Project Roles and Permissions for additional Project Contacts permissions.)	X	X	
	Manage project folders (in **Projects** pane)		X	X	

Project Roles and Permissions

The following table lists the various roles and associated permissions used to manage a given project in Code Insight. The project creator automatically becomes the initial Project Contact and Project Administrator. In turn, a Project Administrator can assign Analyst, Reviewer, and Observer roles to Code Insight users, as well as create other Project Administrators. The Project Administrator can also remove users from any of these roles. For more information, see Assigning and Removing Project Users in this guide.

Users can be assigned multiple project roles.

Table 9-2 ▪ Project Roles and Permissions

			Roles					
			Analyst	Reviewer	Observer*	Proj. Contact	Proj. Admin	Sys. Admin
Responsibility	**Permissions**	**Notes**						
Manage project	Reassign the project contact		X	X	X			
	Manage project users		X	X	X	X		X
	Rename the project		X	X	X	X		X
	Create/edit custom field values for a project (including **SBOM Bucket Name**)		X	X	X	X		X
	Move projects in **Projects** pane		X	X	X	X		X
	Manage scan settings		X	X	X	X		X
	Manage review/ remediation settings		X	X	X	X		X
	Manage Source Control Management (SCM) and Application Lifecycle (ALM) instances		X	X	X	X		X
	Delete the project		X	X	X	X		X
	Branch or copy the project		X	X	X	X		X
Invoke/stop scans				X	X	X		X
Upload codebases				X	X	X		X
Import/export project data				X	X	X		X

Table 9-2 ▪ Project Roles and Permissions (cont.)

	Roles					
	Analyst	Reviewer	Observer*	Proj. Contact	Proj. Admin	Sys. Admin
Assign project to an SBOM bucket	X	X	X	X		X
Export to SBOM Insights		X	X	X	X	X
View project inventory						**

Table 9-2 ▪ Project Roles and Permissions (cont.)

			Roles					
			Analyst	Reviewer	Observer*	Proj. Contact	Proj. Admin	Sys. Admin
Review project inventory	Recall inventory				X	X	X	X
	Approve/reject inventory		X		X	X	X	X
	Set inventory priority		X		X	X	X	X
	Edit/create inventory	Only Analysts have access to the **Add Item** and **Edit Item** buttons to create/ edit project inventory properties.		X	X	X	X	X
	Create and manage work items in the project's associated ALM (application life cycle management) system		X		X	X	X	X
	Update Notices text and notes	This permission refers to inventory's **Notices Text** field (on the **Notices Text** tab) and the information on the **Notes & Guidance** tab (except **Detection Notes**).			X	X	X	X
	Edit custom field values on the **Inventory Details** tab				X	X	X	X

Table 9-2 ▪ Project Roles and Permissions (cont.)

			Roles					
			Analyst	Reviewer	Observer*	Proj. Contact	Proj. Admin	Sys. Admin
	View evidence found in files listed on the **Associated Files** tab and manage the inventory’s file associations	For Analysts only, the file path for an associated file is hyperlinked, enabling them to open to the file’s **File Details** tab in **Analysis Workbench** to view evidence. In **Analysis Workbench**, Analysts can also add/remove files associated with inventory.		X	X	X	X	X
	Force automatic review by policy across all inventory in the project		X		X	X	X	X
Use Analysis Workbench	View/analyze codebase files			X	X	X	X	X
	Edit alerts			X	X	X	X	X
	Create, edit, and recall inventory and manage custom detection rules			X	X	X	X	X
	Edit **Notices Tex**t field on **Notices Text** tab			X	X	X	X	X
	Edit **Audit Notes** field on the **Notes** tab			X	X	X	X	X
	Edit custom field values on the **Custom Fields** tab			X	X	X	X	X

Table 9-2 ▪ Project Roles and Permissions (cont.)

		Roles					
		Analyst	Reviewer	Observer*	Proj. Contact	Proj. Admin	Sys. Admin
Generate reports	Any user (not just one with a project role) can generate reports. For a “private” project, the Observer is considered an “any user”, restricted to viewing project inventory and generating reports.						

* The Observer role is available for only projects defined as “Private”. Private projects are hidden from all users except the Project Contact, the System Administrator (restricted to **Summary** tab only), and those users assigned as Project Administrators, Analysts, Reviewers, and Observers of the project. An Observer is limited to viewing project inventory and generating reports for the “Private Project”.

** In general, a System Administrator has permission to access both public and private projects. However, the **Project Inventory** tab for a private project is visible to a System Administrator only if the user assigned to the System Administrator role is also assigned to a role in the project (Project Administrator, Project Contact, Observer, Analyst, or Reviewer).

Roles and Permissions to Manage the Review Task Flow

The following table lists the project roles and permissions used to manage the tasks to review or remediate inventory items in a project.

Table 9-3 ▪ Project Task-Flow Roles and Permissions

		Roles					
		Analyst	Reviewer	Observer	Project Contact	Task Assignee	Project Admin
Permissions	**Notes**						
Create/edit tasks	Any user assigned to a project role can create and edit tasks.						
Reassign tasks		X	X	X	X		

Table 9-3 ▪ Project Task-Flow Roles and Permissions (cont.)

		Roles					
		Analyst	**Reviewer**	**Observer**	**Project Contact**	**Task Assignee**	**Project Admin**
Close manual review tasks		**X**		**X**	**X**	**X**	**X**
Close remediation tasks		**X**	**X**	**X**	**X**		
Close miscellaneous tasks	Any user assigned to a project role can close a miscellaneous task.						