revenera

Code Insight 6.14.2 SP2

User Guide

Legal Information

Book Name: Code Insight 6.14.2 SP2 User Guide

Part Number: RCI-6142_SP2-UG00

Product Release Date: November 2021

Copyright Notice

Intellectual Property

Restricted Rights Legend

Contents

1

Code Insight Overview

The following content provides an overview of FlexNet Code Insight, support resources, and contact information:

Intended Audience

The Code Insight 6.14.2 SP2 User Guide is intended for the following team members:

- Developers
- Technical leads, senior engineers
- Engineering managers
- IT security officers
- Project managers
- Release managers
- Program managers
- Legal and patent attorneys

What It Does

Code Insight is the industry's first application security solution targeting today's widespread use of Open Source Software (OSS). Code Insight leverages the industry-leading detection and analysis capabilities of Revenera products by integrating a policy-based software component-management system with inventory, for example, intellectual property in the form of published groups of files via Detector.

Role-based features and functions help engineering and project management streamline daily code management activities for various team members. Hierarchical organization creates an effective and accurate view of team projects and software use policies.

Why It's Needed

If your company is concerned about any of the following, you need Code Insight:

- GPL and other Viral Licenses (especially v3.0)
- Affero GPL
- Commercial Content and Libraries
- Restrictions on commercial use or field of use (e.g. no Military use)
- Cryptography
- Code with Unknown Licenses
- Percent (%) of undisclosed content

How It Works

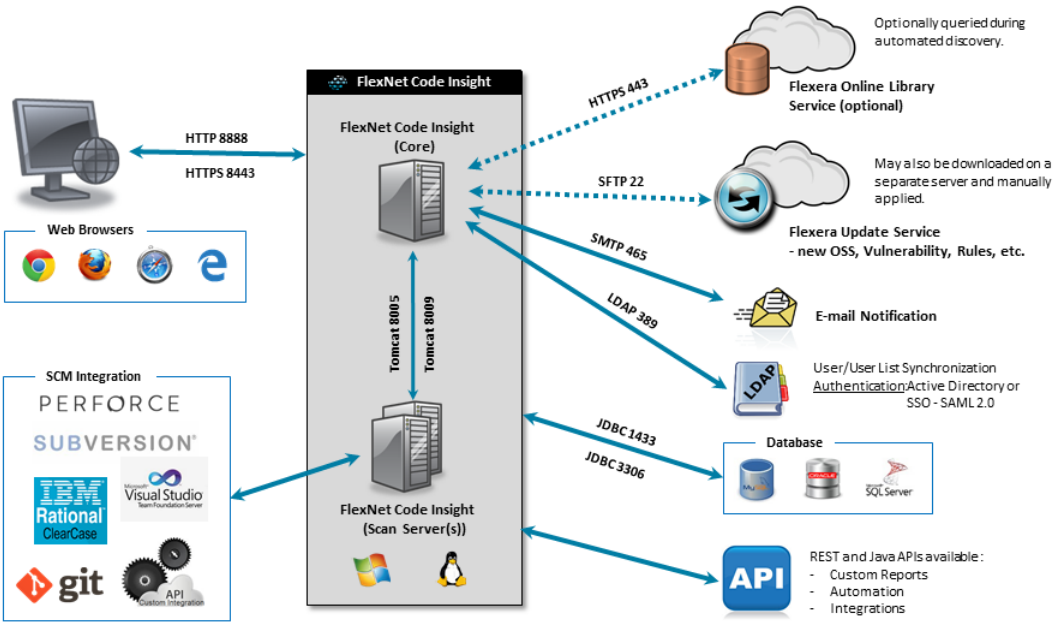
Revenera's unique software composition analysis technology captures the composition of a code base, and provides your team with an inventory of Open Source software (OSS) component usage. The resulting inventory identifies security vulnerabilities and intellectual property (IP) issues associated with the inventoried OSS components.

The policy and code management workflow of Code Insight enables development organizations to cost effectively manage and secure mission critical applications and products.

Component management tasks are assigned and completed by a cross-functional team of users in line with the policies of Code Insight.

Technical Architecture Overview

The following diagram shows how the various clients work in conjunction with the workspaces, databases, and the Code Insight application. Code Insight implements the code management workflow on top of inventory detection, analysis, and generation capabilities.



Common Terminology

The following are common terms that appear in Code Insight documents and on reports and product pages:

Table 1-1 ▪ Common Terminology

Terms	Definitions
AutoWriteUp - Ibiblio Maven2	Uses component information from the bulk collection of metadata from a Maven mirror to create inventory items for jars.
CL	Compliance library, a collection of data that Code Insight uses to generate an inventory of OSS and third-party components and associated vulnerabilities. This library is large and is provided on a separate disk.
Component Analyzers	The analyzers identify source distributions of common open source projects, especially targeting those in the C/C++ where package managers and metadata files are not standard. These analyzers are attuned to regularities across the entire history of the open source project, so generally even a version released tomorrow can be identified reliably. Other analyzers identify packages and components from widely used package managers and repositories.
Component Analyzer - Abstracted	This Analyzer leverages the high-quality data of a component analyzer when that component is found in other, non-source-code contexts (such as binary distributions).

Table 1-1 ▪ Common Terminology (cont.)

Terms	Definitions
Composer (php) Package Analyzer	An analyzer for Composer (php) packages. If there is a composer.lock file, those defined dependencies and versions will be used. If there is no lockfile (e.g. if the package has not been 'installed'), then composer.json files are parsed for uninstalled 'required' dependencies (Uninstalled Composer Dependency). In this case, versions are set at the most recent version in the Packagist registry (packagist.org) that satisfies the given composer.json's semantic versioning restrictions (Composer/Packagist API).
CPE	Common Platform Enumeration, a widely-used, standard methodology to define and distinguish such items as application classes, operating systems, and hardware present in a organization's computing environment.
CRAN Analyzer	Uses metadata within a CRAN source code package to create an inventory item.
CVE	Common Vulnerabilities and Exposures, a list of known information-security vulnerabilities and exposures in commercial and open-source software. Hackers can use software errors and software code errors to gain access to a company's computer systems. For more information, visit http://cve.mitre.org/. Code Insight supports only the CVSS v2 scoring system when reporting security vulnerability details.
Data Services - Component Lookup	Uses Data Services to look up the best FlexeNet Code Insight component match.
Data Services - Digest	Use a file's md5 digest to determine whether the file has an Auto-WriteUp rule or other library data associated with it.
Data Services - Maven Component Lookup	Uses GroupId and ArtifactId to select the best FlexeNet Code Insight component.
Data Services - Release Name	Uses Data Services to analyze the origin of the file based on library collection data.
GPL	General Public License. These licenses for open source code expose commercial code that contains open source to public copyright.
Intellectual Property (IP)	Refers to creations of the intellect for which a monopoly is assigned to designated owners by law. In the context of Code Insight and this guide, IP refers to programming code created internally or externally.
Intellectual property rights (IPRs)	Legal protections, such as trademark, copyright and patents, given to the owners and/or creators of IP.

Table 1-1 ▪ Common Terminology (cont.)

Terms	Definitions
Inventory	A list of all available OSS components. Inventory takes the form of published groups from the Detector, the result of auditing analysis.
JDBC Driver Connector File	A file required to establish a connection with your data source. You must specify this file during installation.
Jenkins	A Java open source automation server that supports continuous integration.
License Lookup - Component Level	Identifies the license(s) of a component across versions. Typically this will be an aggregate of the licenses for the project over time or the license at the time of data collection.
License Lookup - Version Level	Identifies the license(s) of a component at a specific version. This data is sensitive the possibility of a change in license over time.
LOC	Lines of Code.
NPM Analyzer	Identifies node_modules from npm and other components using a package.json file to store metadata.
OSS	Open Source Software.
Policies	Rules and conditions assigned globally or to specific teams or projects which automate the review process of inventory items and/or requests to use OSS components.
POM	Project Object Model
POM Analyzer	An analyzer that analyzes pom.xml files to determine dependencies and resolve versions based on the project's declared or inherited properties. The Analyzer will attempt to retrieve pom information about dependencies from the local maven repositories (as defined in the pom.xml) or Maven Central. To become an inventory item, dependencies must: ● have a version that can be resolved ● not be in scope Test ● not be set to Optional
POM Ancestry License	Indicates a license determined by the declared license in a parent POM file.
POM Dependency	A GroupId ArtifactId pair identified as a dependency within a pom.xml file. See POM Analyzer for the conditions under which a dependency will become an inventory item.
POM License Declaration	Indicates a license determined by the declared license in the POM.

Table 1-1 ▪ Common Terminology (cont.)

Terms	Definitions
POM Transitive Dependency	A dependency not declared directly in a project's POM, but inferred from the retrieved POM of a dependency.
POM <type> Copyright	Indicates various methods for building a copyright for the purpose of creating a notice.
Projects	Specific development goals, usually a software application or element that a project team is developing. Projects are comprised of workspaces, policies, requests, inventory, and tasks.
Related Component Lookup	Improves component information, such as CVEs and License data, by finding different FlexeNet Code Insight components for the same Open Source project as found in different forges (e.g. the component that corresponds to the RPM's centos distribution and the SourceForge source repository).
Reports	Project- or inventory-based lists of code, license citations filtered by global, team, or project parameters.
Requests	Inquiries from users asking if certain third-party code or components are allowable for use in their software development projects.
RPM Metadata	Uses metadata store inside an RPM to create an inventory item.
RubyGem Analyzer	Uses metadata within a RubyGem package in source code form to create an inventory item.
RubyGems API	Indicates that the RubyGems.org API (rubygems.org) was used to get metadata about a .gem package to create an inventory item.
SCA	Software Composition Analysis.
Tasks	Individual actions in the code management workflow.
Uninstalled Node Module Dependency Inventory	When there is a package.json with dependencies but the corresponding node modules are not installed. The NPM API uses the npm registry (registry.npmjs.org) to get package information and resolve semantic versioning. Versions are set at the most recent version in the npm registry that satisfies the semantic versioning restrictions in the given package.json file.
Users	Those accessing the application on a frequent or infrequent basis with roles and responsibilities that have been assigned to them by the application administrator and project owner.

Table 1-1 ▪ Common Terminology (cont.)

Terms	Definitions
Viral license	A derogatory term synonymous with copyleft license, itself a play on the term "copyright." In general, if your codebase contains open source code, you are required to retain a copyright statement that contains license terms and author acknowledgment in any application or software derived from your codebase. In addition, any codebase containing code covered by a viral license must be provided for free.
Workspace	A container for codebase scan settings and scan results. A project may contain multiple workspaces which together comprise the entire codebase for that project.

Data Organization: Teams and Projects

Global data is comprised of teams. Teams are comprised of projects, and projects are comprised of workspaces:

- Global assignment: The policy or rule is applied to any and all teams within the corporation that use this application; for example, a corporate-wide standard.
- Team assignment: An organizational unit is influenced by a policy or rule; for example, all software development teams working within the corporation, regardless of the project.
- Project assignment: Only a specific project is influenced by a policy or rule; for example, only the widget API version 1.14 release is influenced by the policy or rule.

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

For customers who have purchased a maintenance contract for their product(s), you can submit a support case or check the status of an existing case by making selections on the **Get Support** menu of 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

Roles and Responsibilities

The following Set Up and Code Management Tasks table provides a summary of the Code Insight process and shows which user roles are eligible for which activities.

Code Insight requires some application administrative setup tasks. Then all of the ongoing code management cycle tasks occur in parallel. Users can perform any function appropriate to their assigned roles and the progress of the project.

Code Insight includes automatic approval and rejection features to facilitate efficient responses to requesters, as well as a manual review feature.

- **Automatic Acceptance**: The automatic acceptance feature of the policy engine gives developers and other requesters an immediate answer as to which components are already approved for use. This is useful, for example, for setting up pre-approved versions of a corporate repository.
- **Automatic Rejection**: This feature gives requesters an immediate answer when users request known unacceptable components. Automatic rejection is useful for quickly communicating inappropriate license/usage combinations and unsecured components.
- **Manual Review**: This feature allows employees with the appropriate permissions to review component requests, which may have varying restrictions and need to be reviewed on a case-by-case basis. We suggest that the policy administrator set the Review policy before allowing users to access the application's features.

These features allow users to see the policies and rules in real time as they make requests and perform reviews related to specific projects. However, you can also do this after users have accessed the system.

Table 2-1 ▪ Set Up and Code Management Tasks

Task	Category	Milestone Target	Role	Activity
Setting up and Configuring a Project	**Create Project**	Project start	Application Admin	Create users, teams, projects
	Review Project Policy	Project start, Ongoing	Policy Admin	• Define conditions of use • Add attachments to policies • Manage project policies
	Configure Project	Project Start	Participant (selected as the Project Owner)	Set project users roles, such as assign auditor and security analyst Define request review process Associated workspaces with project
Code Management Cycle	**Conduct Audit**	Ongoing	Auditor	Audit codebase associated with project and publish groups to create inventory items
	Create Component Requests	Ongoing	Requester	Create requests, review and acknowledge conditions of use imposed by reviewers
	Review a Request	Ongoing	Reviewer	Review requests and define conditions of use (technical, business, legal, etc.)
	Reconcile Inventory	Ongoing	Owner or project participant	Review project inventory detected via audit and associate with request approving its use
	Security Review for Vulnerabilities	Ongoing	Project Security Analyst	Review vulnerabilities associated with project inventory detected via audit
	Report Generation	Ongoing	Various	Capture a summary of inventory or security cycle status to share with project team

3

Application Administrator Tasks

The application administrator can enter user, team, and project data into Code Insight. The user with this role has to possess a data-entry skill set. The application administrator is responsible for completing the following activities at the start of a project:

- Creating New Users
- Creating New Teams
- Creating New Projects
- Importing Bulk Data from an Excel Workbook
- Using the Scripting Feature
- Running an Electronic Update
- Email Notifications
- Downloading Code Insight Logs

Creating New Users

The following are the ways to add users to projects:

- My Shortcuts Tab: Create a New User
- Administration Menu: Users Option

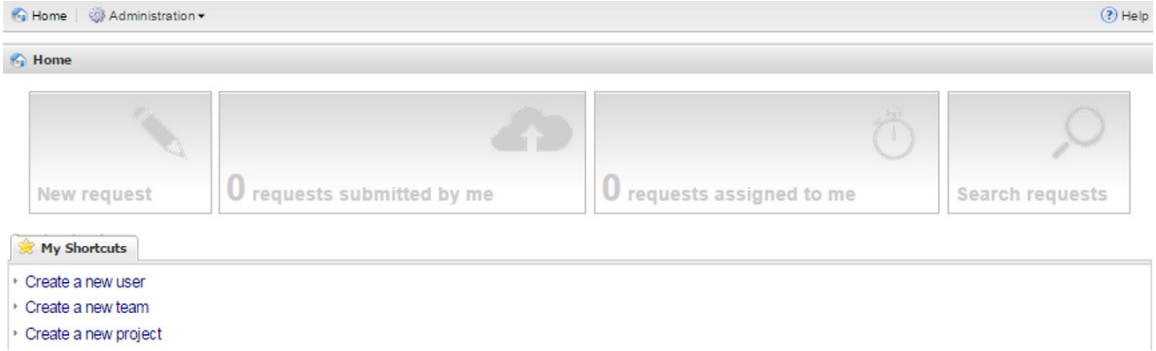
My Shortcuts Tab: Create a New User

You can create new users from the **My Shortcuts** tab.

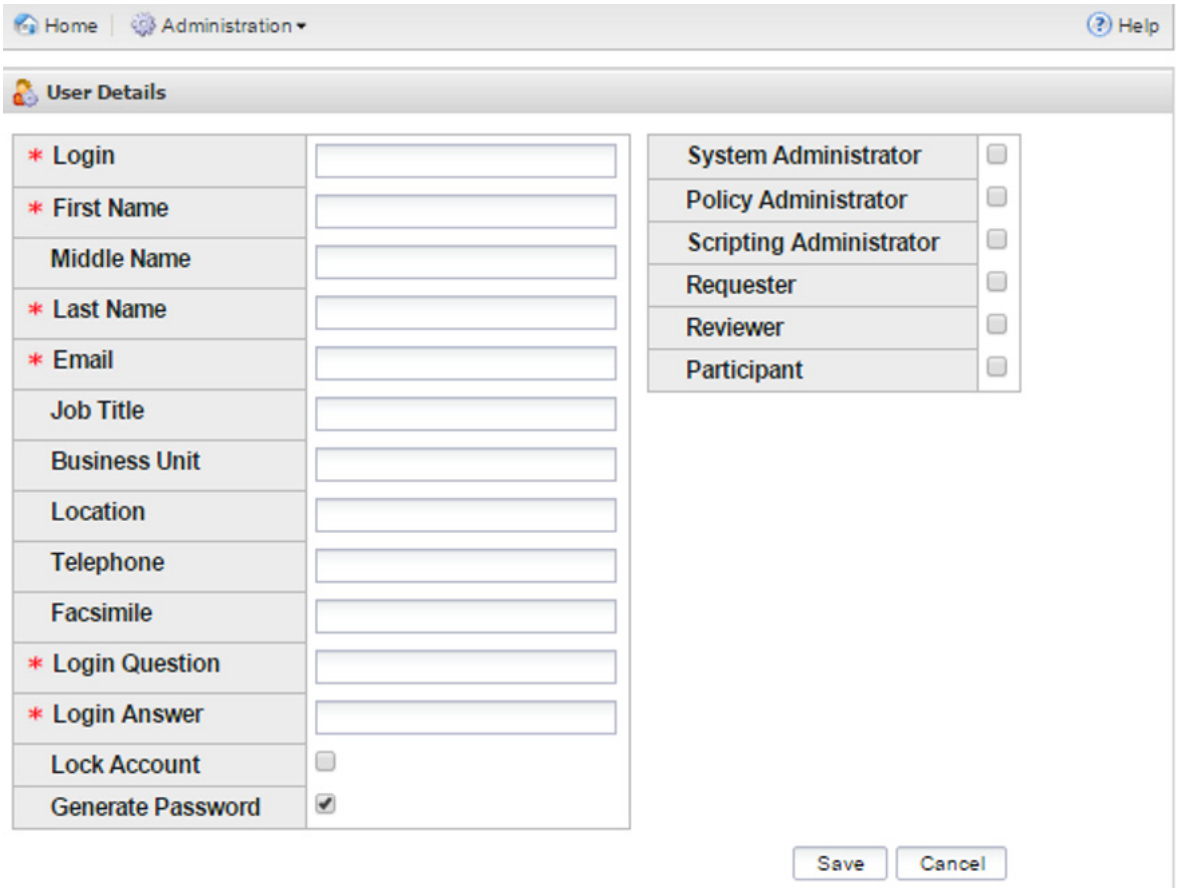
Task ***To create a new user, do the following:***

1. Log into Code Insight as administrator:

The **Administration Home** page appears. On this page you can create users, teams, and projects.



2. You may also use the **Main** menu bar to navigate the administration pages.
3. To create new users, click **Create a new user**. The **User Details** page appears.



4. Enter all necessary user information and set the role of the user by checking one of the possible project roles in the shaded box to the right. Note the following:

   - If you choose to have the user's password system-generated, Code Insight sends automatic emails to new users notifying them of the system-generated temporary password. New users will be prompted to change this password the first time they log in. Ensure that you enter a valid email address for each user.
   - If you choose to explicitly provide a password, uncheck the **Generate Password** option to display the **Password** field and the associated confirmation field.

   Note ▪ *Do not include a left angle-bracket (<) character in the password. When you attempt to save the user profile, an error message is displayed, stating that the password is invalid due to the bracket character. You must re-enter a password that does not include this character.*

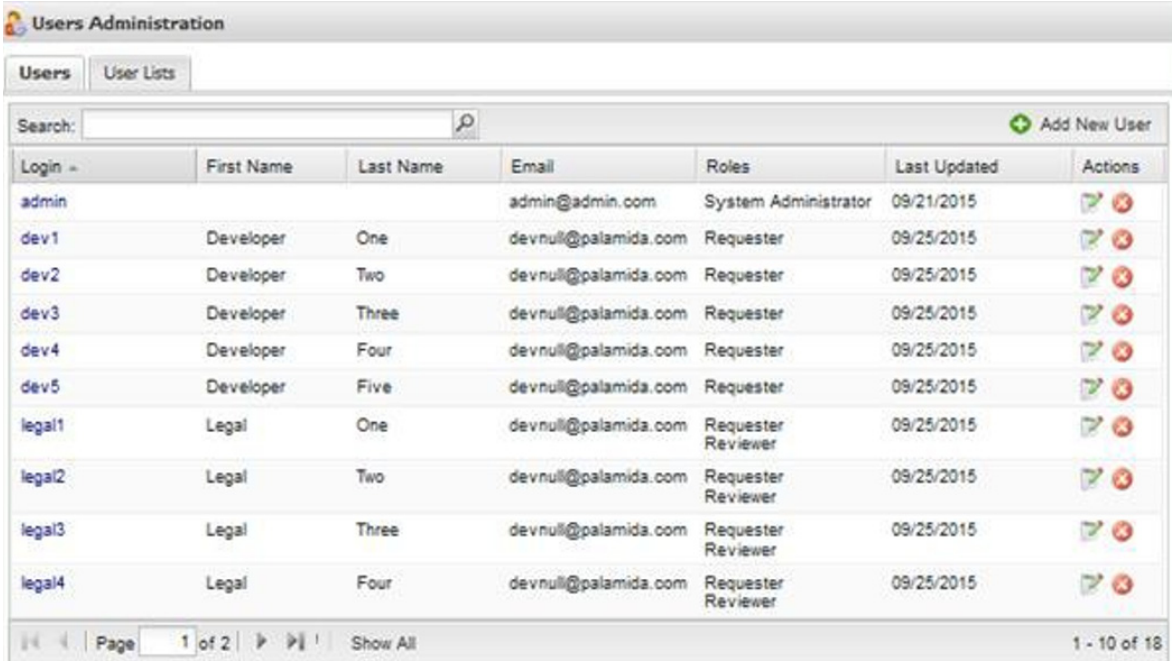
5. Click **Save** to save your changes.

Administration Menu: Users Option

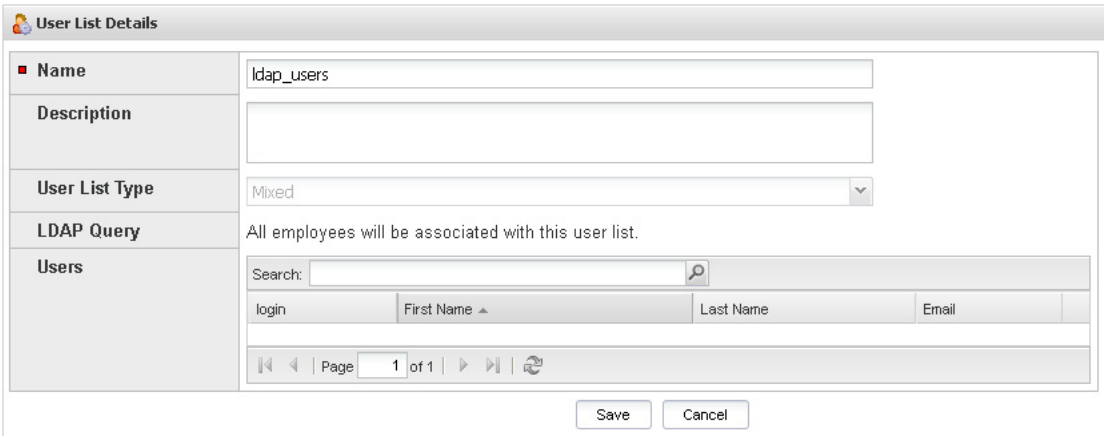
The other way to add and manage users is via the Administration pull-down menu. This allows you to manage users in your system as well as integrate with an LDAP server to dynamically populate a user list.

Task ***To add and manage users via the Administration pull-down menu, do the following:***

1. Log into Code Insight as an *administrator*.
2. Click **Administration** button on the **Main** menu bar.
3. Select **Users** from the pull-down menu. The **Users Administration** page appears. The **User** tab allows you to manage users, while the **User Lists** tab allows you to manage user lists. Follow the steps in My Shortcuts Tab: Create a New User to manage a single user.



4. To manage user lists, select the **User List** tab, and then click the **Add New User List** icon. The **User List Details** page appears:



5. Enter the name and description for the User list.

6. To filter the available users to only ones with the required role, select one of the following user types from the User List Type pull-down menu:

Note ▪ *You cannot change the user list type after creation so be sure to select the right one.*

- **Mixed**: All users can be associated with this type of user list. User list cannot be associated with a project role on the project configuration page.
- **Project Participant**: Only users with the project participant role can be associated with this type of user list. This user list can be associated as project observers, auditors, security analysts, or QuickReview facilitator on the project configuration page.
- **Requester**: Only users with the requester role can be associated with this type of user list. This user list can be associated with the requester role on the project configuration page.
- **Reviewer**: Only users with the reviewer role can be associated with this type of user list. This user list can be associated with any review level on the project configuration page.

7. If LDAP has been enabled, and if the User List LDAP properties have been properly defined in the core.ldap.properties configuration file, the user list is dynamically updated during each LDAP sync to contain users returned by the LDAP query. If a user list has been configured to utilize an LDAP query rather than be manually populated, the description of the associated LDAP query will be shown, and the add/delete users buttons will no longer be available to manually manage users for this user list. Users with mismatched roles (based on the user list type) will be removed from the list returned by the LDAP query.

8. The following parameter must be configured to set user role assignment during the LDAP sync. The available roles are: requester, reviewer and participant. All other roles must be set manually. See the "LDAP Configuration" section of the *Installation and System Administration Guide* for more detailed instructions on configuring this property.

```
ldap.user.role = requester, reviewer, participant
```

9. The following two parameters must be configured for each user list that is associated with an LDAP query. The LDAP query will be executed each time an LDAP sync occurs.

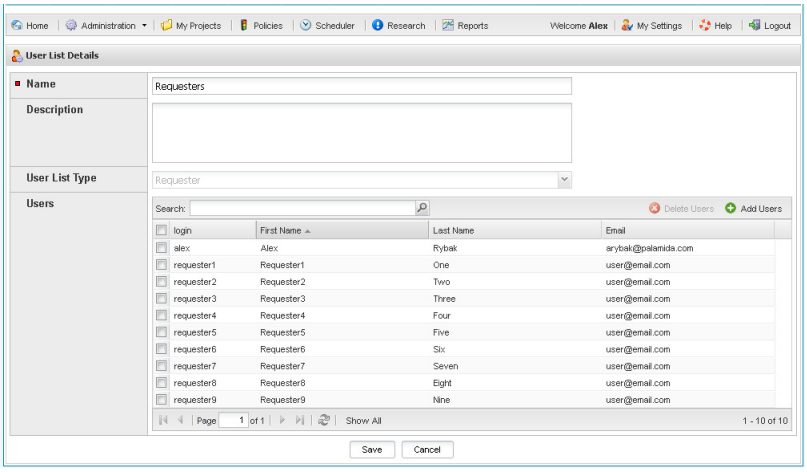
```
<userlist_name>.ldap.description
```

A human-readable description of the LDAP query. This will be shown on the **User List Details** page to indicate to the user which LDAP query is dynamically being used to populate the user list.

```
<userlist_name>.ldap.query
```

The LDAP query to execute for fetching users to dynamically populate the associated user list.

10. You can also manually associate any user with the necessary roles to a user list via the Users table. Click in Add Users to select the Users to add to the User List or remove existing users via the checkbox to the left of the name and then click on the **Delete Users** button.



11. If you have LDAP configured, you can sync Code Insight with LDAP at any time by going to the User Sync tab and pressing the Start User Sync button. This will dynamically update all users and user lists in the system. For more information on how to configure LDAP, see the "Integrating with LDAP for Authentication (Optional)" section in the *Code Insight Installation and System Administration Guide*.

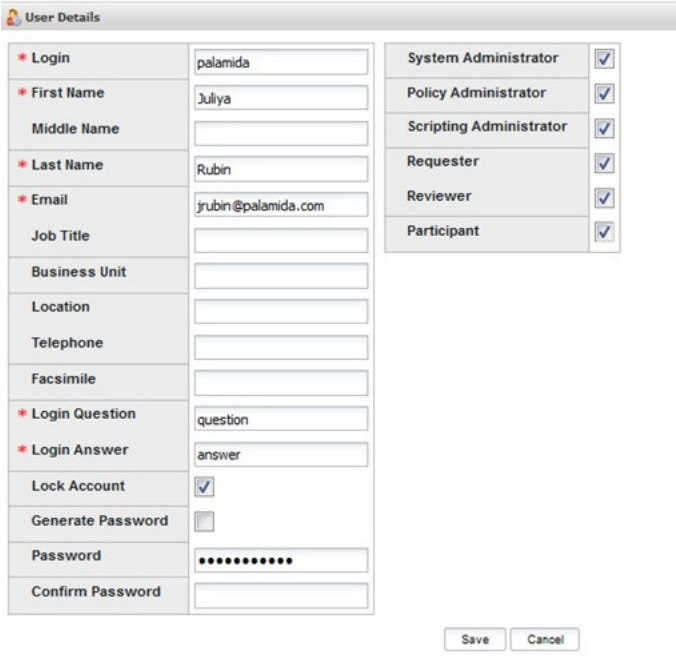
Adding and Removing User Account Locks

The system locks out someone who repeatedly attempts to access the system with incorrect login and password information. If this occurs, users will see the following message:

As an administrator, you may also lock a suspicious user out of the system.

Task ***To lock or unlock a user account, do the following:***

1. Log in to Code Insight as *administrator*.
2. Select **Users** from the **Administration** pull-down.
3. Click the **pencil and paper** icon associated with a user name or on the appropriate link in the **Login** column.
4. Lock or unlock a user account:
   - To lock a user account, check **Lock Account**.
   - To unlock a user account, uncheck **Lock Account**.



Note ▪ *The lock/unlock options are not available for external users (i.e., LDAP users)*

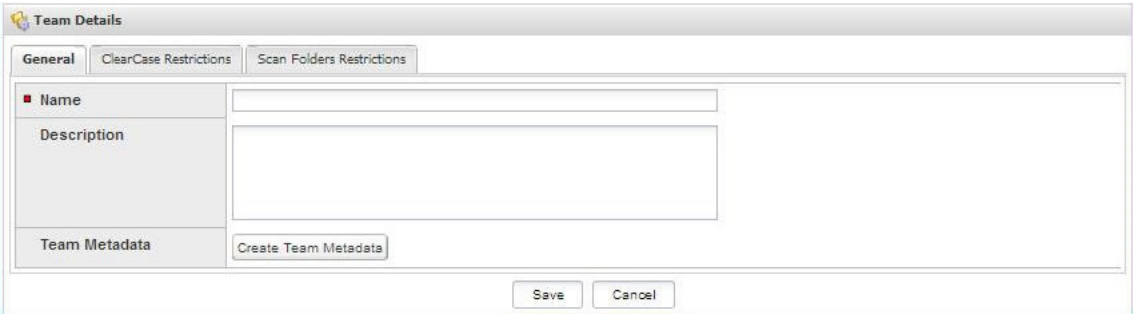
5. Click **Save**.

Creating New Teams

Use the following procedure to create a new team.

Task ***To create a new team, do the followings:***

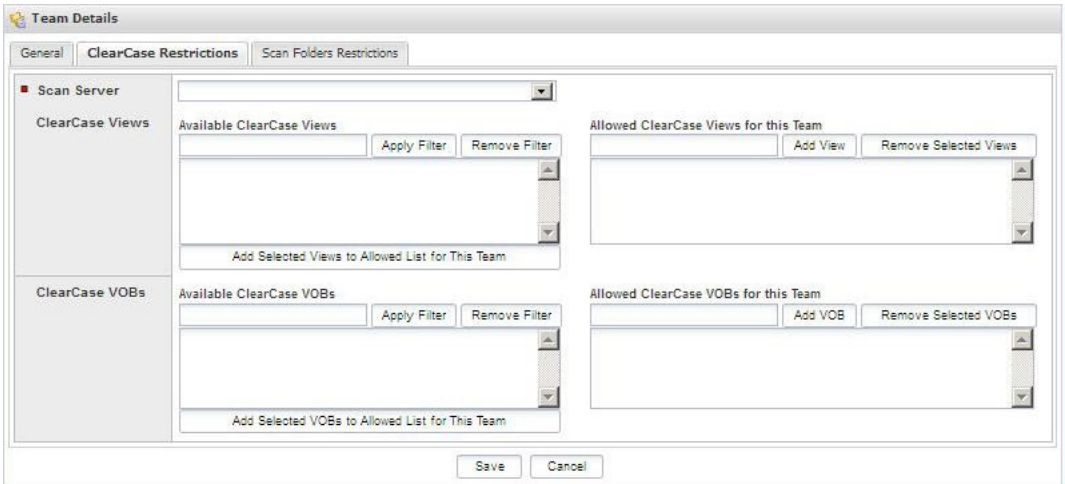
1. Click **Home** in the top left-hand corner to return to the **Administrator Home** page.
2. Click the **Create a new team** link. The **Team Details** page opens. You may also use the **Administration > Teams** command on the Main Menu bar to navigate the administration pages.



3. Enter a team name and a team description in the **General** tab.

Note ▪ *To use the Create Team Metadata button, you must have created Metadata fields for Teams. To create metadata, log in as Administrator, and see the Metadata command on the Administration pull-down menu. Metadata is typically created during the system configuration phase. However, it can also be created any time thereafter.*

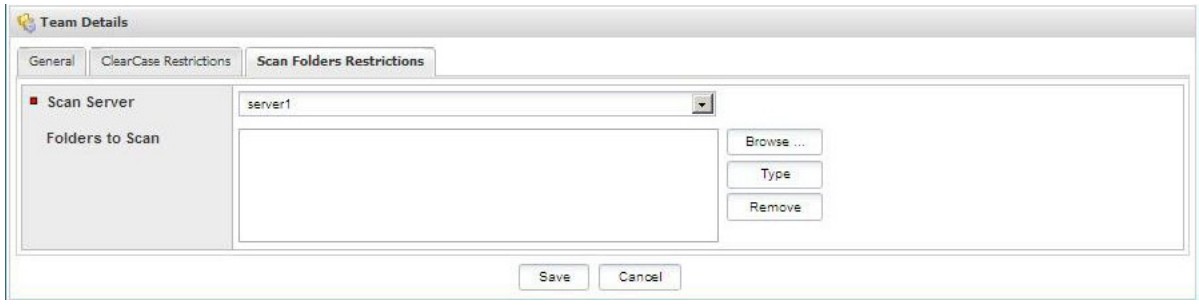
4. If ClearCase has been enabled for any Scan Servers, a **ClearCase** restrictions tab will be shown on the **Team Details** page:



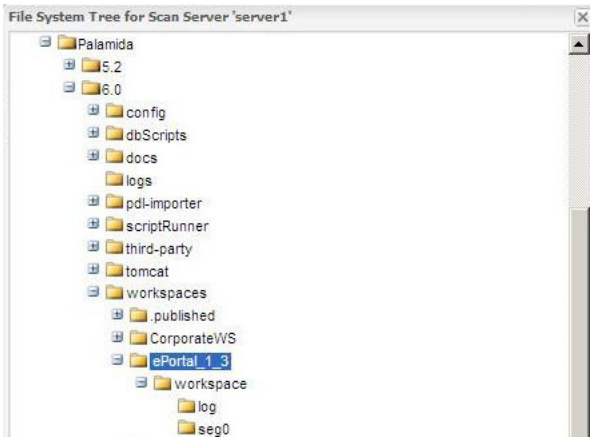
This tab is used to restrict which ClearCase views and VOBs may be used for scanning by all projects under this team.

- A filter may be used to find the desired views and/or VOBs.
- Selected views and VOBs will be saved with the team, and will restrict available values in the Workspace Settings page for the Project Owner and Auditor.
- The `clearcaseRestrictionsShowAllDataByDefault` property in the `scm.properties` file controls whether all views and VOBs are shown to the Project Owner and Auditor on the Workspace Settings page if no ClearCase restrictions have been defined for the team. By default, the property is set to true, so all ClearCase views and VOBs are available unless restricted by the Application Administrator.

5. The **Scan Folder Restrictions** tab is used to restrict which locations on the Scan Server may be used for scanning by all projects under this team.
6. Select the **Scan Server** from the pull-down menu.
7. Click **Browse** to select folders from the server file system tree, or manually type in the full path, as shown in the following figure.



8. In the server file system tree, select the Scan Server to display that server's file system tree.



9. Click on the folder(s) to add them to the list of locations that can be selected in the workspace settings for any project under this team. This will restrict which folders can be scanned for all projects under this team.

10. Click the **Save** button.

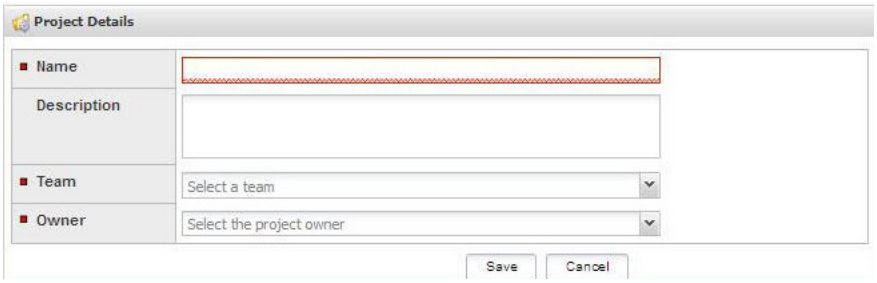
Creating New Projects

Use the following procedure to create a new project.

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

1. On the **Administrator Home** page click the **Home** button in the top left-hand corner.

2. Select the **Create a new project** link. The **Project Details** page appears:



You can also use **Administration > Projects** option on the **Main** menu bar to navigate to the administration pages.

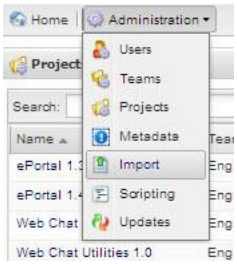
3. On this page, enter the following:
   - **Name**: Name of the project
   - **Description**: Brief description of the project
   - **Team pull-down menu**: Select the team to which this project belongs
   - **Owner pull-down Menu**: The project owner. Project owners have permission to create, delete, and modify certain types of data.

Importing Bulk Data from an Excel Workbook

The Bulk Import feature allows you to import data globally. For example, you can bulk import all requests as completed.

Task ***To bulk import user, simple policy, or completed request data via an Excel Workbook (spreadsheet), do the following:***

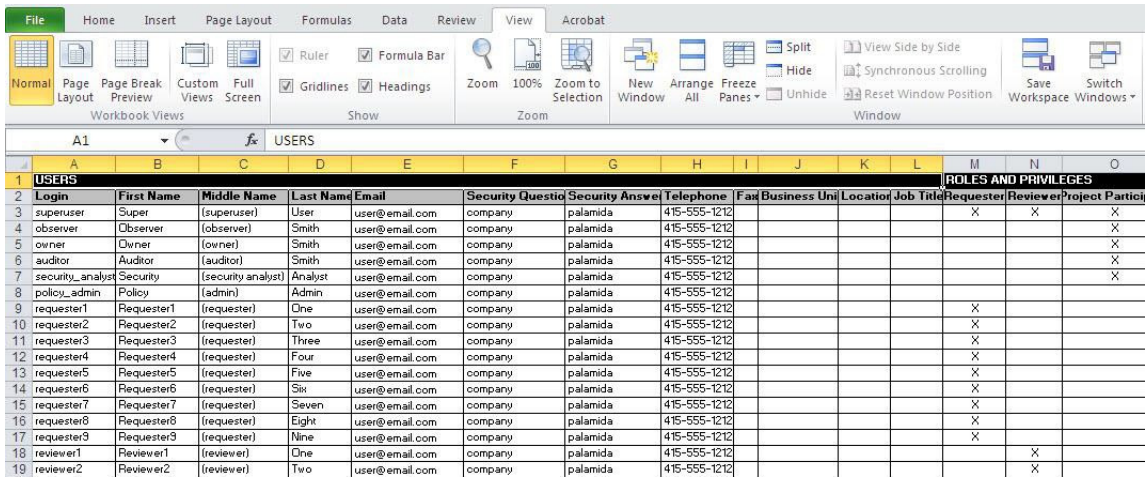
1. Access the **Admin Home** page by clicking on the **Home** button in the top left-hand corner.
2. Click the **Administration** pull-down menu.



3. Select the **Import** option. The **Import** page appears.

On this page, click **Browse** to select a system data file to import.

4. If you click **Download**, a sample workbook for bulk user import, an Excel workbook sample page opens populated with sample Code Insight administration data.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1	USERS												ROLES AND PRIVILEGES		
2	Login	First Name	Middle Name	Last Name	Email	Security Questio	Security Answe	Telephone	Fax	Business Uni	Location	Job Title	Requester	Reviewer	Project Partici
3	superuser	Super	(superuser)	User	user@email.com	company	palamida	415-555-1212					X	X	X
4	observer	Observer	(observer)	Smith	user@email.com	company	palamida	415-555-1212							X
5	owner	Owner	(owner)	Smith	user@email.com	company	palamida	415-555-1212							X
6	auditor	Auditor	(auditor)	Smith	user@email.com	company	palamida	415-555-1212							X
7	security_analyst	Security	(security analyst)	Analyst	user@email.com	company	palamida	415-555-1212							X
8	policy_admin	Policy	(admin)	Admin	user@email.com	company	palamida	415-555-1212							
9	requester1	Requester1	(requester)	One	user@email.com	company	palamida	415-555-1212					X		
10	requester2	Requester2	(requester)	Two	user@email.com	company	palamida	415-555-1212					X		
11	requester3	Requester3	(requester)	Three	user@email.com	company	palamida	415-555-1212					X		
12	requester4	Requester4	(requester)	Four	user@email.com	company	palamida	415-555-1212					X		
13	requester5	Requester5	(requester)	Five	user@email.com	company	palamida	415-555-1212					X		
14	requester6	Requester6	(requester)	Six	user@email.com	company	palamida	415-555-1212					X		
15	requester7	Requester7	(requester)	Seven	user@email.com	company	palamida	415-555-1212					X		
16	requester8	Requester8	(requester)	Eight	user@email.com	company	palamida	415-555-1212					X		
17	requester9	Requester9	(requester)	Nine	user@email.com	company	palamida	415-555-1212					X		
18	reviewer1	Reviewer1	(reviewer)	One	user@email.com	company	palamida	415-555-1212						X	
19	reviewer2	Reviewer2	(reviewer)	Two	user@email.com	company	palamida	415-555-1212						X	

Using the Scripting Feature

Use the following procedure to access the scripting feature.

Task ***To access the scripting feature, do as follows:***

1. Access the **Admin Home** page by clicking the Home button in the top left-hand corner. Access the scripting feature from the **Administration** pull-down menu.

The **Scripting** page appears.

2. Select a script from the pull-down menu.
3. To execute the selected script, click **Execute Script**.
4. The scripts are registered in the database (MySQL/Oracle) using the following sample insert statement. The corresponding groovy script file should be placed in the scripts directory under the following:

```
$Code Insight_ROOT_DIR/config/core/scripts/.
INSERT INTO pas_system_script (id_, name_, display_name_, description) VALUES
(1,'sample_script','Sample Groovy Script','This is a sample Groovy script.');
```

Note ▪ *You need a commit; statement if executing this on Oracle.*

Running an Electronic Update

Code Insight supports electronic updates of product data in several ways. An update can be applied either automatically on a recurring basis, or manually via the Web UI. In case of a manual update, the Code Insight Update HTTPS service is used whereby an update archive is downloaded and processed by the system, or in case of a setup where internet access is not available, a local update archive can be used.

Running an Electronic Update Manually

Use the following procedure to run an electronic update manually.

Task ***To run an electronic update manually, do the following:***

1. Log in to Code Insight as *Application Administrator*.
2. Navigate to the Administration à Updates menu option.

Using the Electronic Update Server (HTTP)

On the **Last Update** tab, click the **Check for Electronic Update** button to connect to the Code Insight Update HTTPS server and check for updates. If one is available, follow the prompts to process the update. The following types of information may be provided in updates:

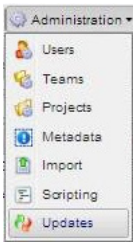
- New components, versions, licenses, and vulnerabilities
- License by version mappings
- License families
- License detection data
- License obligations data
- License compatibility data
- AutoExpert™ rules

Using a Local Electronic Update Archive

Use the following procedure to use an electronic update archive.

Task ***To use a local electronic update archive, do the following:***

1. Switch to the **Manual Update** tab.



2. Browse for the manifest (update_manifest.txt) and data file (update.zip). Follow the prompts to process the update.

Scheduling Automatic Electronic Updates

Use these steps to schedule an electronic update.

Task ***To schedule automatic electronic updates, do the following:***

1. Set the following settings in the <Code Insight_ROOT_DIR>\config\core\core.properties file:

   `auto.update.enabled = true auto.update.jobFrequency = 0 0 12 ? * SUN`

2. Use the cron expression format to define the frequency for processing the update data. By default, the update is set to run every Sunday at noon (12pm).

   See http://www.quartz-scheduler.org/documentation/quartz-2.x/tutorials/crontrigger for more information about the cron expression format.

Viewing Electronic Update History

Select the **Update History** tab to view and sort the number of affected projects and the number of affected inventory items based on data changes via the electronic update.

Email Notifications

Code Insight sends email notifications for various events in the application. Email notifications use HTML templates to generate the appropriate email subject line and body text. The email templates can be found in the <CODEINSIGHT_ROOT_DIR>\config\core\email\ directory, and they can be modified as necessary.

Email Event Triggers and Notifications

The following table lists events that trigger an email notification, the users who receive the email notification, and whether any configuration option exits for the given event.

Table 3-1 ▪ Email Triggers and Notifications

Email Triggering Event	Email Notification To	Configuration Option
Administrative		
Code Insight license close to expiration	Application administrators	None
New user account created	New user	None
Forgot password	Current user	None
Project		
New project created	Project owner Policy administrators	None
Project copy completed	Project owner for new project	None
Project summary per selected frequency	Project owner	Frequency
Task is reassigned to another user	New task owner	None
Policy		
Policy created, updated, or deleted	Policy administrators	Which roles get notified
Electronic Updates		
Update service completed with failed status	Application administrator	None
Scanning		
Scan server down	Application administrators	None
Scan task completion	User that scheduled tasks	None

Table 3-1 ▪ Email Triggers and Notifications (cont.)

Email Triggering Event	Email Notification To	Configuration Option
Inventory		
Inventory scheduled for full review	First pool of reviewers	None
Inventory with vulnerabilities scheduled for full review	Security Analysts for project	None
Vulnerability alert (inventory item with new vulnerabilities)	Project owner Security Analysts for project	None
Inventory checklist item created, edited, or deleted	User assigned to checklist item	None
Inventory checklist item marked as completed	Project owner Posted by user Marked complete by user	None
Inventory question created, edited, or reassigned with specified assignCode Insight	User assigned to question	None
Inventory question answered or answer updated	Project owner Posted by user Answered by user	None
Requests		
New request submitted	Requester and first review level users	None
Existing request renewed and resubmitted	Requester and first review level users	None
Request field value(s) updated by reviewer	All request participants who have touched request	Which project participants (by role) get notified
Comment added to request	All request participants who have touched request	Which project participants (by role) get notified
Request recalled	All request participants who have touched request	None
Request in pending state for long time	Current actors (requester or reviewer)	Frequency

Table 3-1 ▪ Email Triggers and Notifications (cont.)

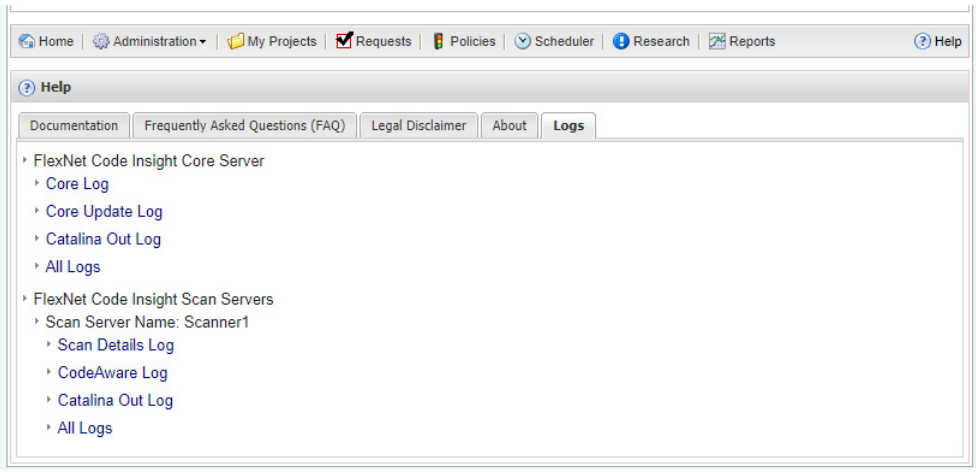
Email Triggering Event	Email Notification To	Configuration Option
Request reached terminal state (approved/rejected)	All request participants who have touched request	None

Downloading Code Insight Logs

Code Insight allows you download selected 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 administrator.
2. Click the **Help** in the top right-hand corner. The **Help** page is displayed.
3. Click the **Logs** tab.



4. Click the option for the Code Insight log that you want to download. See Log Files Downloaded Per Option for a list of log files downloaded for each option on the this tab.

   The log file or files are downloaded as a single `.zip` file to your default download location. (To preview the log file contents from the Code Insight Web UI, use the download-preview mechanism available with your browser.)

Log Files Downloaded Per Option

The following table lists the log files that are downloaded in the resulting `.zip` archive when a given option is selected on the **Logs** tab.

Table 3-2 ▪ Log Files Downloaded Per Option

Code Insight Component	Log Option	Log File(s) Downloaded in .zip File
Code Insight Core Server	The following options enable you to download the logs associated with the Core Server.	
	Core Log	`core.log1`
	Core Update Log	`core.update.log1`
	Catalina Out Log2	`catalina.out`

	All Logs	`core.debug.log`
		`core.log`
		`core.log.x (if generated)`
		`core.update.log`
		`core.update.log.x (if generated)`

Code Insight Scan Servers	The following options download the logs associated with a given Scan Server configured in your Code Insight system. (In the option list, each Scan Server is identified by its name, followed by its own set of the following options.) Note that the name of the output`.zip` file includes the name of the Scan Server whose logs are contained in the archive.	
	Scan Details Log	`scanEngineDetail.log1`
	CodeAware Log	`codeaware.log2`
	Catalina Out Log3	`catalina.out`

	All Logs	`codeaware.log`
		`codeaware.log.<date&sequence> (if generated)`
		`scanEngineDetail.log`
		`scanEngineDetail.log.x (if generated)`
		`scanEngineRule.log`
		`scanEngineSupport.log`

1 If multiple files have been generated for the log, the single file listed here is downloaded. This file contains data merged from the two most recent log files.

2 If multiple files have been generated for the CodeAware log, only the most recent log file is downloaded.

3 If the `catalina.out` log is not present, this option is not displayed. For example, `catalina.out` is not present (and hence this option is not available) if, when you launched Code Insight, you directed the Tomcat logs to a location other than `catalina.out`.

4

Owner Setup Tasks

The owner of a project has specific tasks to complete before a project launch. These tasks include the following:

- Configuring and Editing Your Project
- Adding and Deleting Requesters
- Editing, Adding, and Deleting Project Reviewers
- Viewing, Adding, and Deleting Policies
- Viewing Scan Server Queues
- Generating Reports
- Conducting Research

Configuring and Editing Your Project

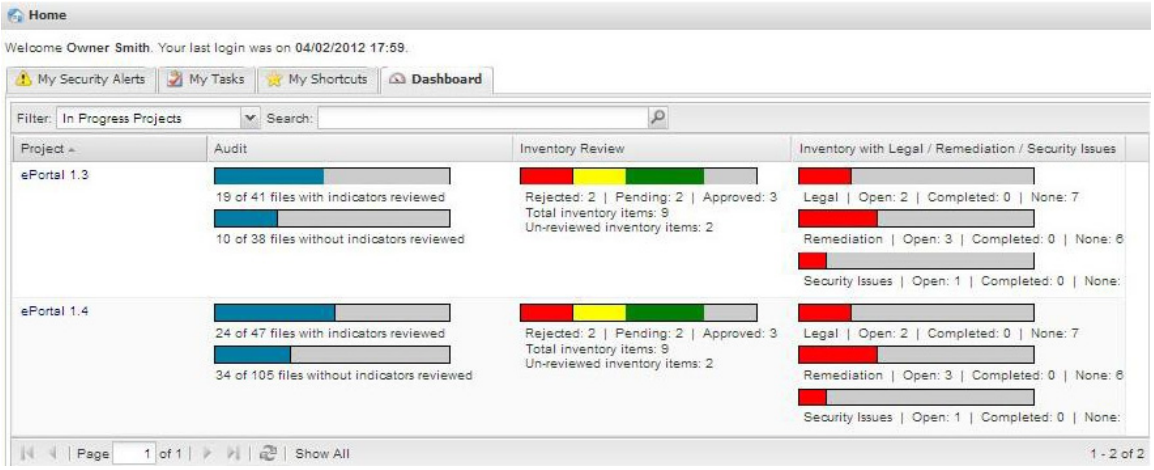
General project information contains the following items:

- Name
- Description
- Team
- Advanced Options
- Enable Inventory Quick Review
- Auto-Publish System-Detected Inventory
- Apply Policies to Inventory
- Request Form
- Owner
- Auditor

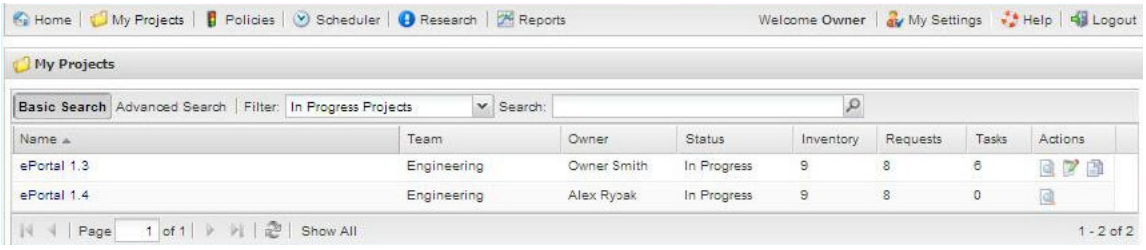
- Security Analyst
- Workspaces
- Review levels (along with any conditions required to enable them, by default, all review levels are always enabled)

Task ***To edit, delete, or add information to a project, do as follows:***

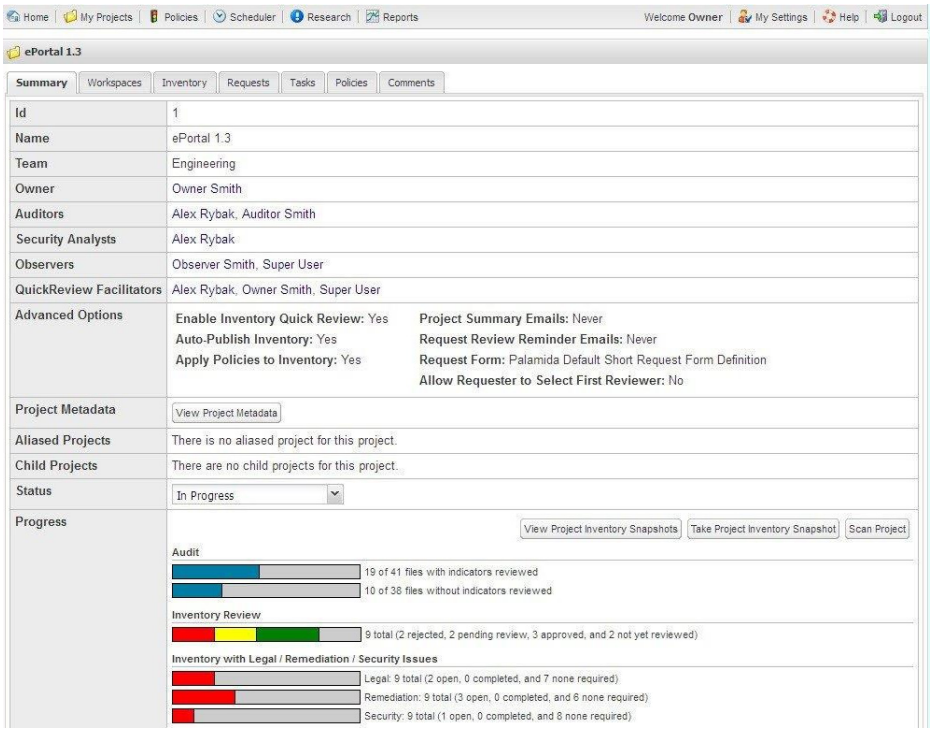
1. Log in to Code Insight as an *Owner*.
2. Click **Home** in the **Main** menu bar. The Home page appears:



3. On the top left of the page, click the **My Projects** button to open the **My Projects** page, which lists all projects:



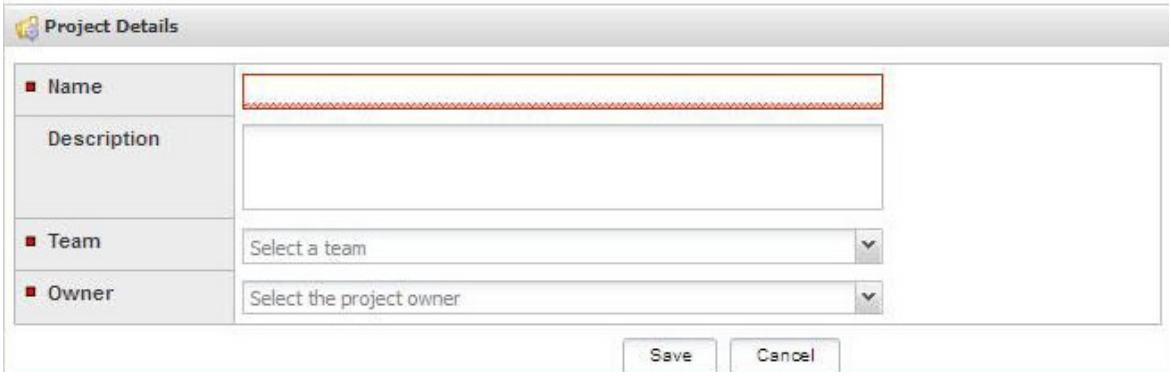
4. Click the project name. The **Project Summary** page appears.



On the **Summary** tab you can view and scan the project using the buttons in the **Progress** section.

5. Click the **Back** button to return to the **My Projects** page.

6. Click the **Pencil and Paper** icon to the right of the project you want to edit. The **Project Details** page appears.



7. For each project, you can edit general project information, as well as the lists of users associated with each project:

   - **Observers**: Read-only view of project data
   - **QuickReview™ Facilitators**: Approve/reject inventory items
   - **Auditors**: Launch Detector to review scan results and conduct audit
   - **Security Analysts**: Review security vulnerabilities associated with inventory items
   - **Requesters**: Submit request for use of OSS/TP software
   - **Reviewers**: Review requests

8. Enter text in the **Name** and **Description** fields.

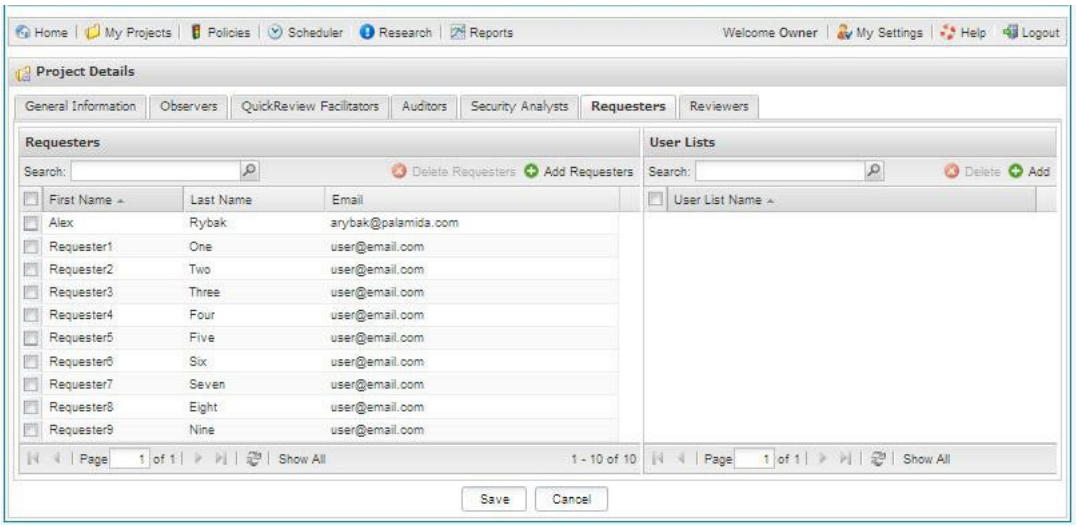
Note ▪ *The selected **Team** is not editable via the **Edit Project** page. Only an application administrator can change the team to which the project is assigned via the **Administrator > Projects > Project Details** page.*

9. Select the **Owner** and **Project Summary Email Frequency** using the pull-down menu.
10. Choose the **Request Review Reminder** email delivery frequencies from the pull-down menu.
11. Select either a long or short request form type via the pull-down menu.
12. (Optional) Check the **Advanced Options** checkboxes:
    - **Enable Inventory Quick Review**: Enables an **Approve/Reject** inventory option for the project owner on the **Inventory Details** page. If either of these options is selected, then the request workflow is not required.
    - **Auto-Publish System-Detected Inventory**: Automatically publishes any system-detected groups from Detector as inventory.
    - **Apply Policies to Inventory**: Applies any always approve/reject policies to published inventory and sets the **Review Status** to **Approved** or **Rejected** if a matching policy exists. All other inventory items receive a **Review Status** of **Ready** for **Review**.
13. Add or delete **Review Levels** and **Child Projects** in the associated fields if you wish. To add a review level, for example, if you want to differentiate between specific technical review levels in a software development project, click on the **Add A Review Level** (+) icon. This opens a text field into which you can enter a name of your choosing for the new review level. This also adds a new tab to the **Project Details** page on which you must select both the appropriate reviewers for a given level, and optionally define the conditions to activate this level if it should not always be active. By default, all review levels are created as enabled, but levels may be changed to conditionally enabled if desired.
14. Click **Alias Project** to search aliased projects.
15. To edit the name of a review level, click the **Pencil and Paper** icon associated with that particular review level.
16. To delete a review level, click the **X**.
17. To add the name of a child project, click on the **Add Child Projects** button in the **Child Projects** section.
18. To edit or delete a child project, click on **Pencil and Paper** icon next to the child project to edit, or click the X icon to delete.
19. Select a Scan Server from the **Scan Server** pull-down menu. This feature allows all workspaces to reside on one server.
20. To manage the scan root path containing the files to scan for the project, click **Browse**, **Type**, or **Remove** in the **Scan Root Path** section.
21. Click **Save** to save your work.

Adding and Deleting Requesters

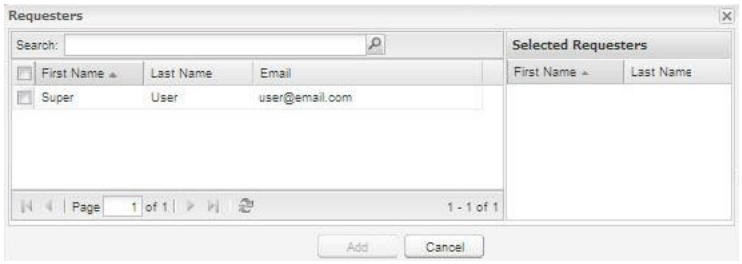
Task ***To edit, add, or delete requesters associated with a specific project, do as follows:***

1. Click the **My Projects** button in the top left corner.
2. Click the Pencil and Paper icon next to the project you want to edit.
3. Select the **Requesters** tab:



4. To delete a requester, select the name you want to delete and click **Delete Requesters**.
5. Add a requester in one of the following ways (a or b):

   a. Click the green **Add Requester** button in the left pane. The **Add Requester** dialog box opens with names of possible requesters.



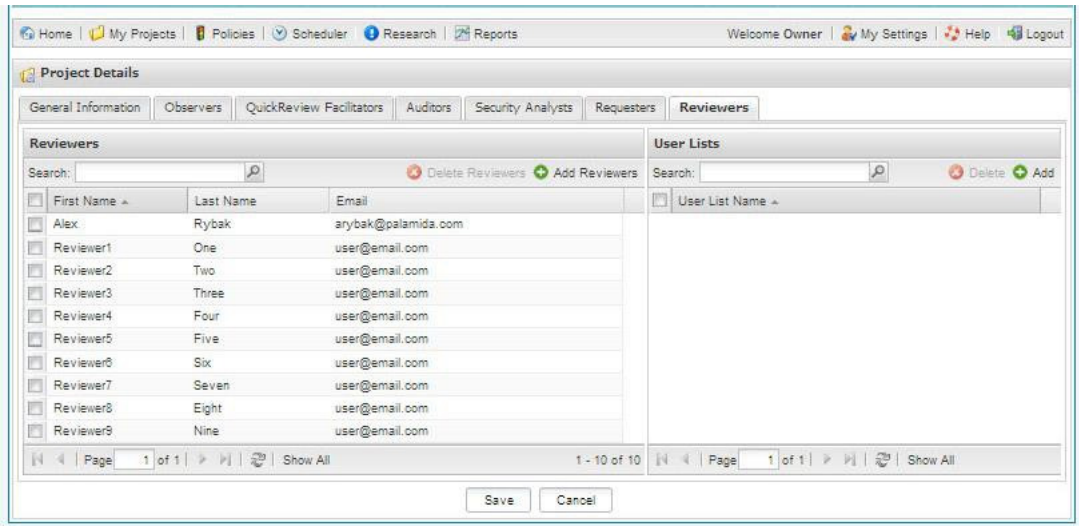
   - Select the name of the requester you want to add. The name appears in the **Selected Requester** list to the right of the name.
   - Click **Add** to add this person to the project in the role of a requester.

   b. Click **Add Requestor from User List** or a user list name in the **Search** field. A dialog box opens from which you can choose a specific user list. You can create user lists that filter users by project roles.

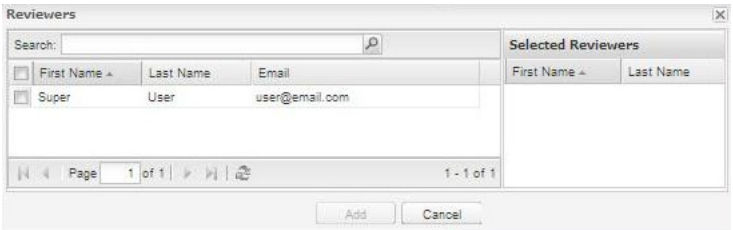
Editing, Adding, and Deleting Project Reviewers

Task ***To edit, add, or delete reviewers associated with a specific project, do the following:***

1. Click **My Projects** in the top left corner.
2. Click the Pencil and Paper icon next to the project you want to edit. The **Project Details** page opens.
3. Select the **Reviewers** tab:



4. Add or delete a reviewer:
   - To delete a reviewer, select the name you want to delete, and click **Delete Reviewer**.
   - To add a reviewer, click **Add Reviewer** and select the name of the reviewer you want to add from the **Reviewers** dialog:



   The name appears in the **Selected Reviewers** list box.

5. Click at the bottom of the box to add this person to the project in the role of a reviewer.

Viewing, Adding, and Deleting Policies

Policies are used by Code Insight to automate the review of inventory items (in a QuickReview™ scenario) or requests to use OSS components (in a full review scenario). Policies can be defined up-front and re- evaluated each time a new project is created. Policies can also be built off of requests that are being reviewed by someone with the policy administrator permission. Creating a policy during the time of request review will result in a very specific policy at the current project scope and will automate the action taken by the reviewer at the time of the policy creation. The purpose of policies is two-fold:

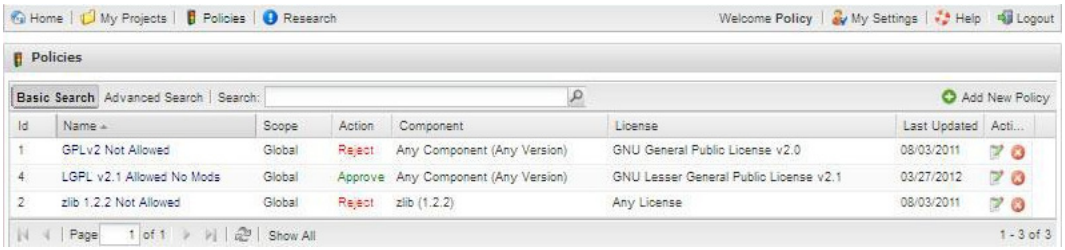
- Policy notifications (flags in the web UI and Detector auditing client) inform the user whether the component and/or license has an associated policy. This information may be used to increase the priority of reviewing particular items in cases where the associated components and/or licenses have an auto-reject policy.
- Policies are also used to automate manual reviews in cases where a particular component or licenses has previously been reviewed and future manual reviews are no longer necessary.

Anyone with policy administrator permissions can create, add, delete, and edit policies. The system administrator allows any user to have these permissions by checking Policy Administrator in the role box when setting up or editing a new user.

Viewing and Editing Policies

Task ***To view or edit policies, do the following:***

1. Log in as policy administrator or make sure your administrator checked the role of **Policy Admin** in your **User Details**.
2. Click **Policies** on the **Main** menu bar. The **Policies** page appears:

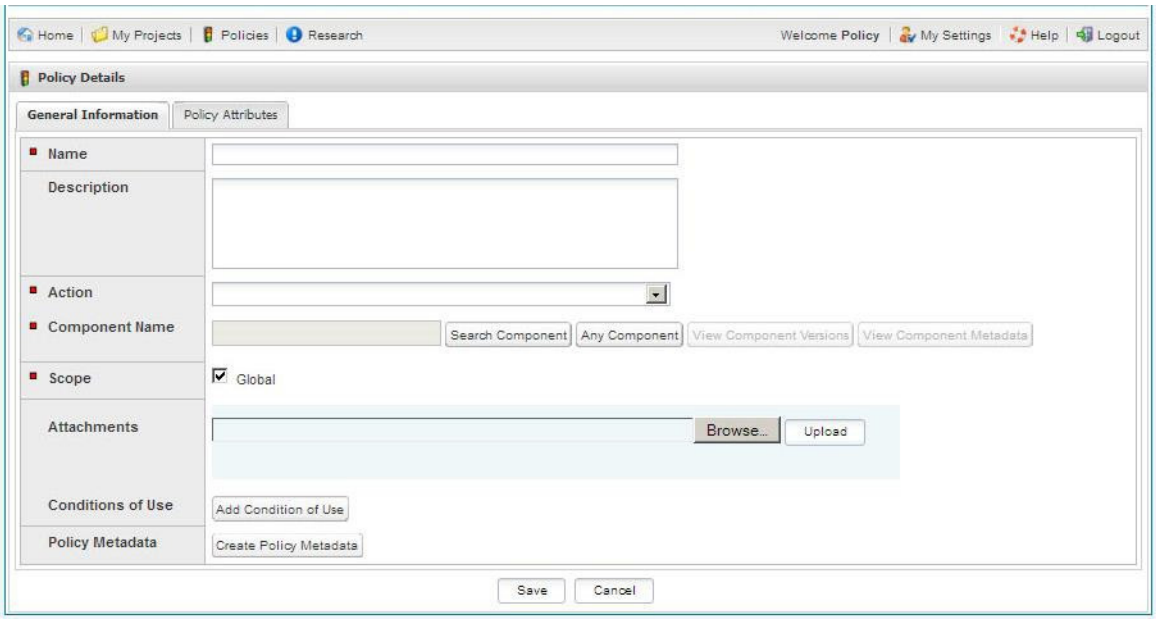


3. View or edit policies by clicking the **Pencil and Paper** icon associated with an existing policy.

Adding a Policy

Task ***add a policy, do the following:***

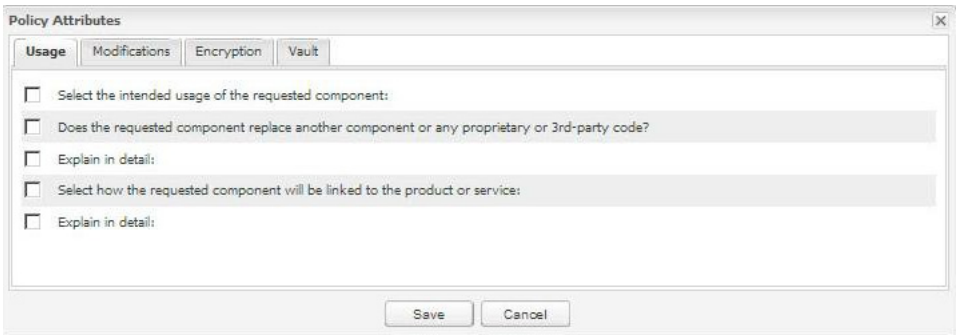
1. Click Add New Policy on the right side of the page to create a new policy. The **General Information and Attributes** tab opens.



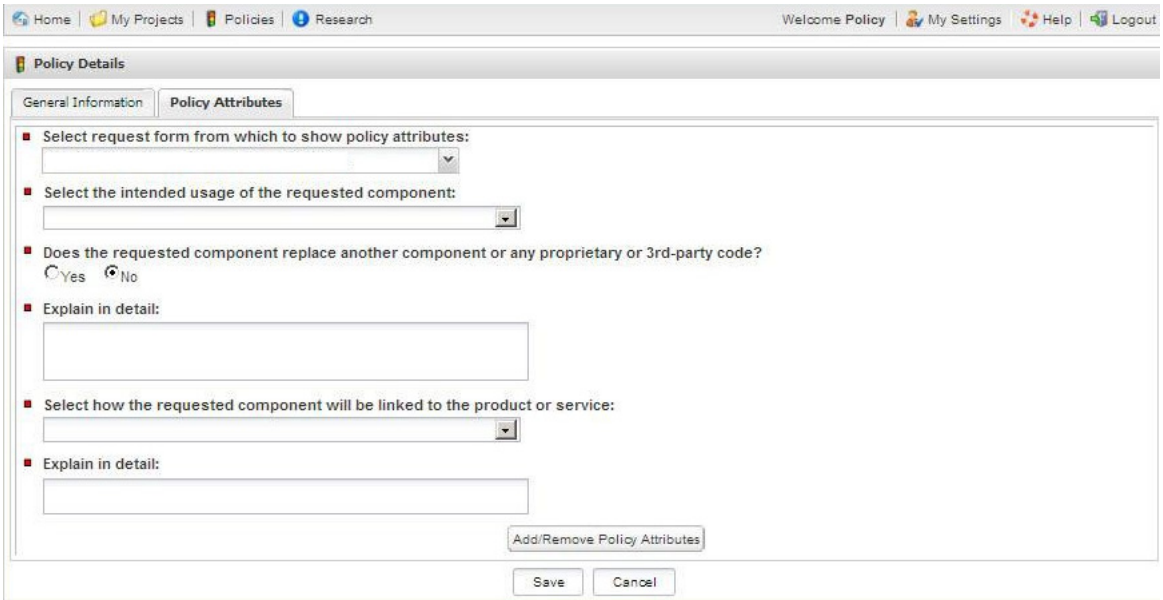
2. On the **General Information** tab, select Approve, Reject, or Manual Review from the **Action** pull-down menu.
3. Determine the scope. Global is the default. Uncheck the Global checkbox if you want to apply the policy to a specific team. Select the appropriate team by double-clicking on the name of the team in the left column, thereby moving it to the right column.
4. Add an attachment if necessary.
5. Click **Add Condition of Use**. The **Add Conditions of Use** dialog appears:

6. Enter text explaining specific conditions of use for this policy.
7. Check the **Required** checkbox if you want the requester to be required to acknowledge the policy conditions of use.
8. Click **Save** to save your work.

9. Select the **Policy Attributes** tab.
10. Select a Request Form using the pull-down menu to inform the system as to which request form attributes to load. (You must select a request form type to set policy attributes.)
11. Click the **Add/Remove Policy Attribute** button. The **Policy Attribute** checkbox opens.



Any boxes you check on the **Usage**, **Modifications**, **Encryption**, and **Vault** tabs become part of the viewable policy attributes. When you click **Save**, the **Policy Attribute** tab is populated with those checked boxes.:



Note about Policy Conflicts

Code Insight does not support conflict resolution in cases where multiple policies apply to a given inventory item or request. There are conflicting policy validations in place to ensure that more than one identical policy is not created with alternate actions. However, policies of various specificities can be defined and more than one can apply at a given time. In such a scenario, the policy that was defined first (by creation date/time) will take precedence. Below is an example:

- Policy #1 – Automatically approve component: Mozilla Firefox
- Policy #2 – Manually review license: Mozilla Public License (MPL), Version 1.1

If an inventory item with Mozilla Firefox under the Mozilla Public License (MPL), Version 1.1 is evaluated by policies, the first policy will take precedence, since it was created before the second policy, and the inventory item would be automatically-approved.

Copying a Policy to Create a New Policy

Instead of creating a new policy from scratch, you can copy and modify an existing policy.

Task ***To copy and modify an existing policy, do the following:***

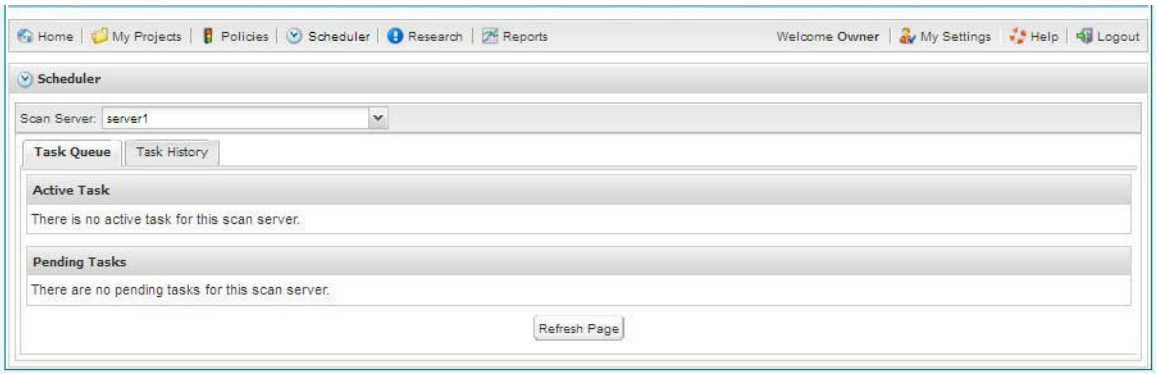
1. Select an existing policy, and then click the Copy Policy icon ( ). The copied policy appears in the **Policy Details** window.
2. To create a new policy from the copy, modify the copied policy and save it as a new policy.

Viewing Scan Server Queues

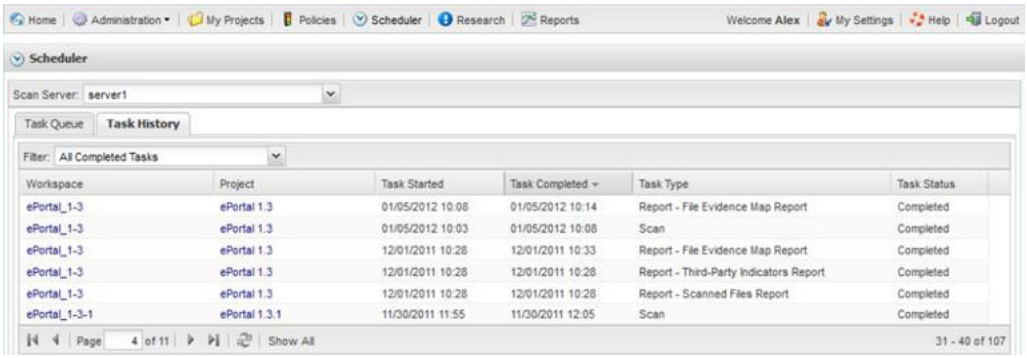
To view the Scan Server task queue and task history for a workspace Scan Server, log in as someone other than an admin or a policy admin, and do as follows:

Task ***To view the Scan Server queue, do the following:***

1. Click **Scheduler** on the **Main** menu bar. The **Scheduler** page appears with **Task Queue** and **Task History** tabs.



2. Select a server from the **Scan Server** pull-down menu, and you can view the progression of the task queue and the task history associated with that server.
3. Click **Refresh Page** to see the latest active and pending tasks.
4. Select the **Task History** tab to view a list of tasks and dates. To sort the columns in ascending or descending order, click the **Task Status** column heading or any other column heading.



Generating Reports

You can generate many types of reports. To see available report types, click the **Reports** button on the **Main** menu bar.

- Reports are defined in the database. Refer to the `default_reports.sql` script for a sample definition of the default reports.
- Reports are generated on the server using a groovy script for the report logic and velocity templates for the rendering logic. You can modify both of these sets of files. Your modifications will influence the report the next time it is generated.
- Although the Web UI lets you preview and download a report that you generate, you can also access the report (and the reports generated by other users) at the following location:

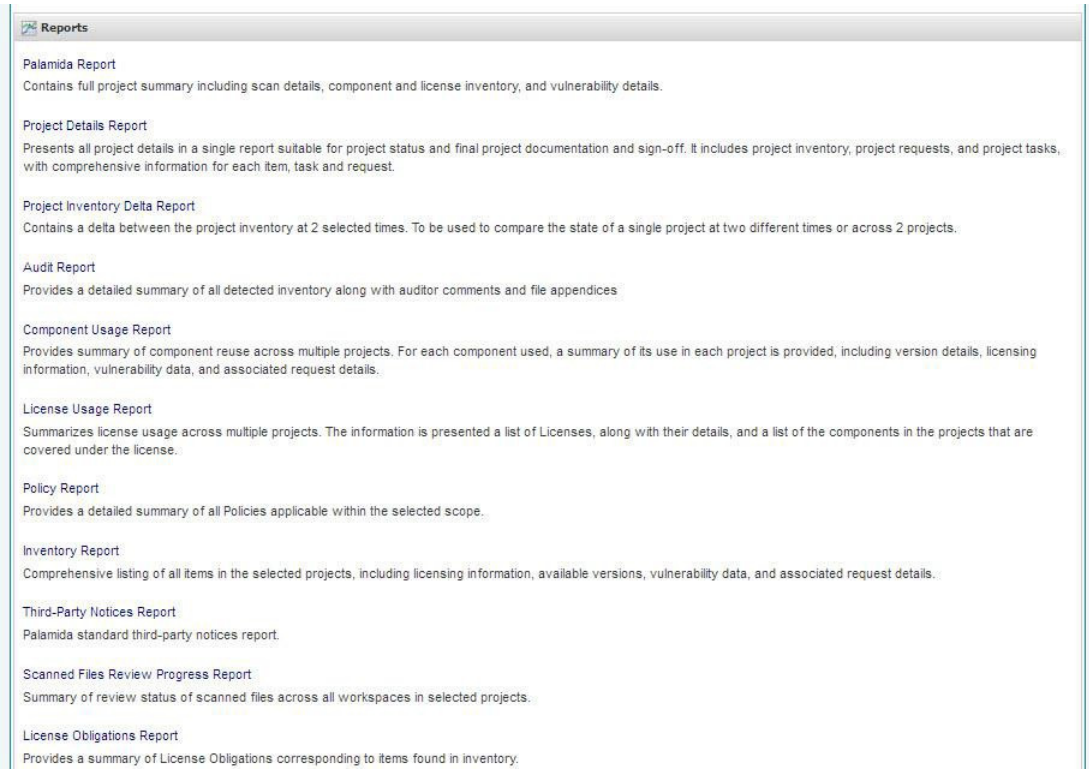
  ```
  <Code Insight_ROOT_DIR>/config/core/output/<productReleaseName>/<reportName>/<loggedInUserName>/
  <timestamp>
  ```

- Code Insight offers a Third-Party Notices report as well as the Code Insight report, which is an extensive project detail summary report.

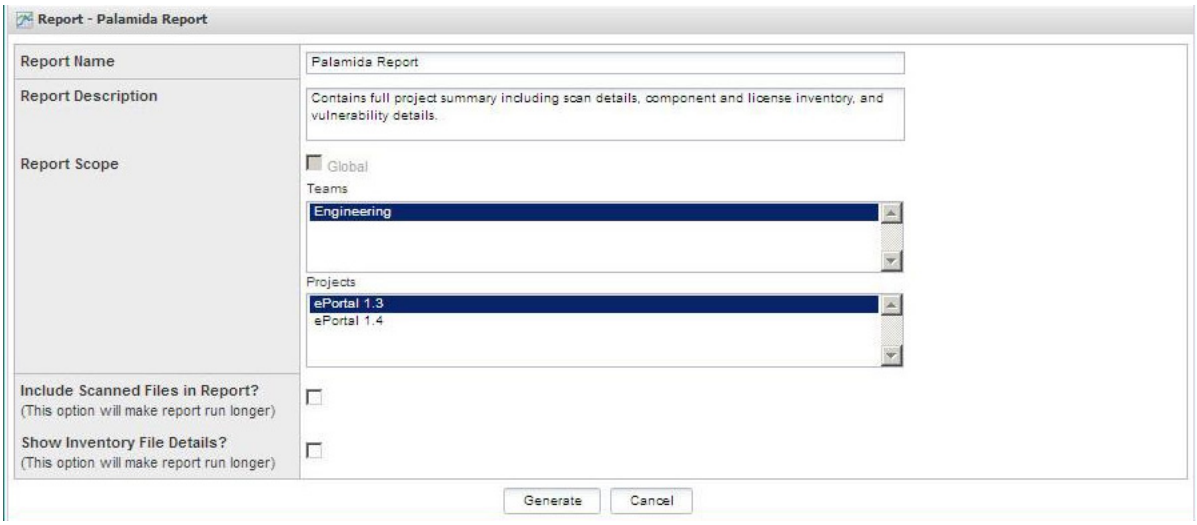
Generating a Project Details Report

Task ***To generate a project details report, do as follows:***

1. Click **Reports** on the **Main** menu bar. The **Reports** page appears.

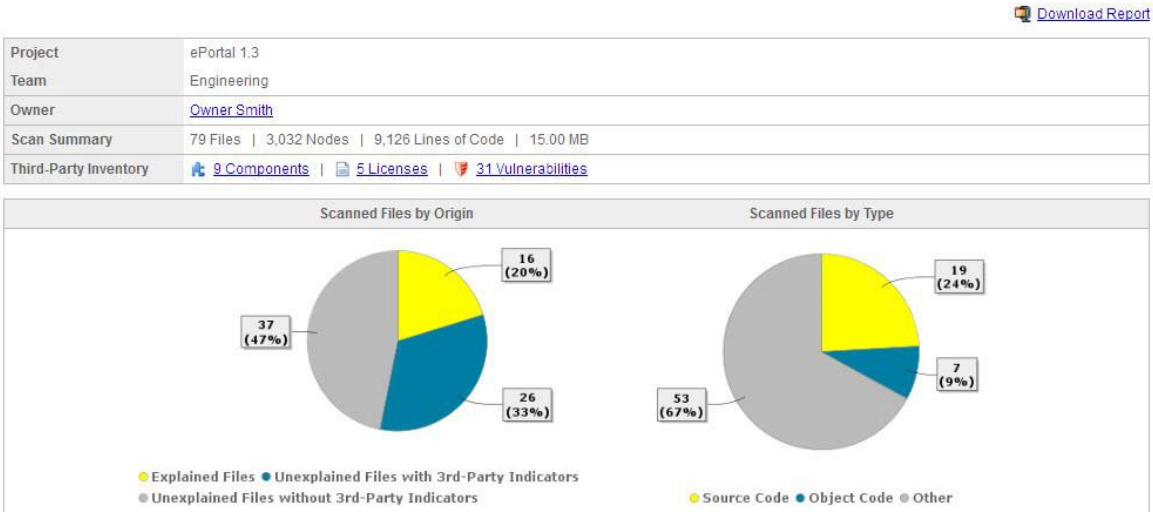


2. Select the **Palamida Report** link. The **Palamida Report** page appears.

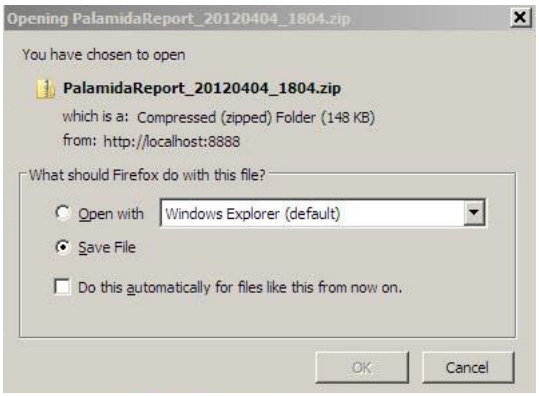


3. Enter or edit a report name and description on this page as needed.

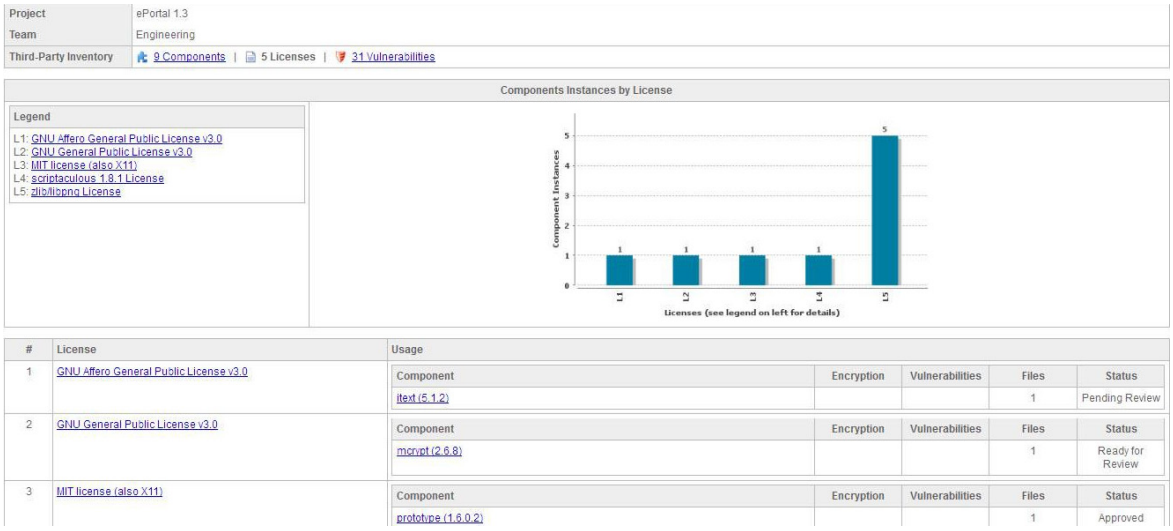
4. Uncheck the **Global** checkbox.
5. Select a team name in the **Teams** column to enable and view associated project choices in the **Project** pane.
6. Select the project in the **Projects** pane.
7. Click **Generate** to create the report. When the report is ready to view, a box opens with a link to view the report.
8. Click the report link and the report opens.



9. Click the **Download Report** link to download the report as a compressed file in the default **Report** directory.



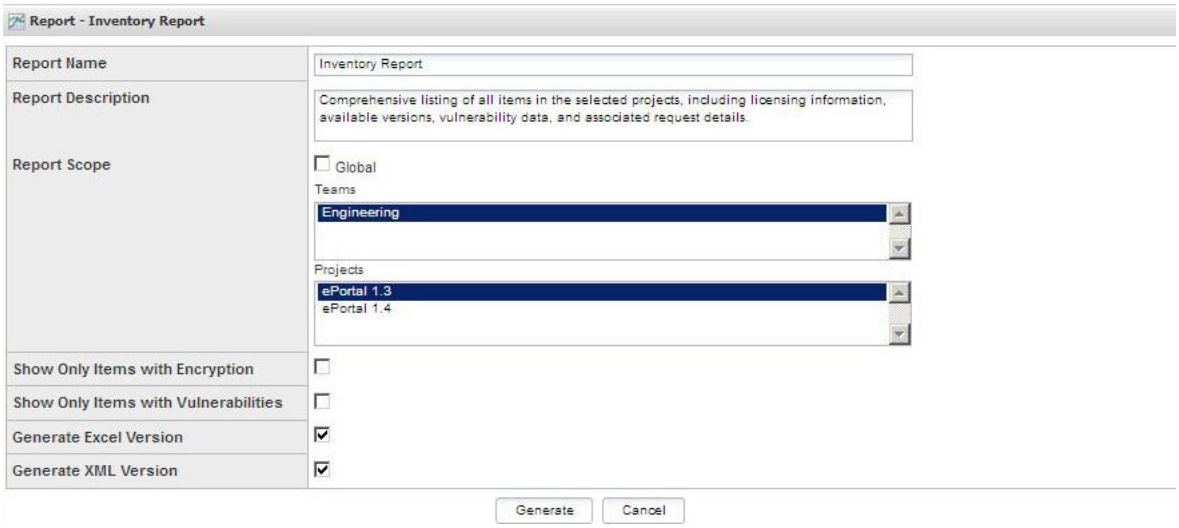
10. To view a specific aspect of the project report, click one of the links in the **Third-Party Inventory** section. For example, click on the **Licenses** link. Reports contain a variety of graphical representations and detailed information.



Generating Inventory Reports

Task ***To generate an Inventory report, do the following:***

1. Click **Reports** on the **Main** menu bar. The **Reports** page appears.
2. Select the **Inventory Report** link. The **Inventory Report** page appears.

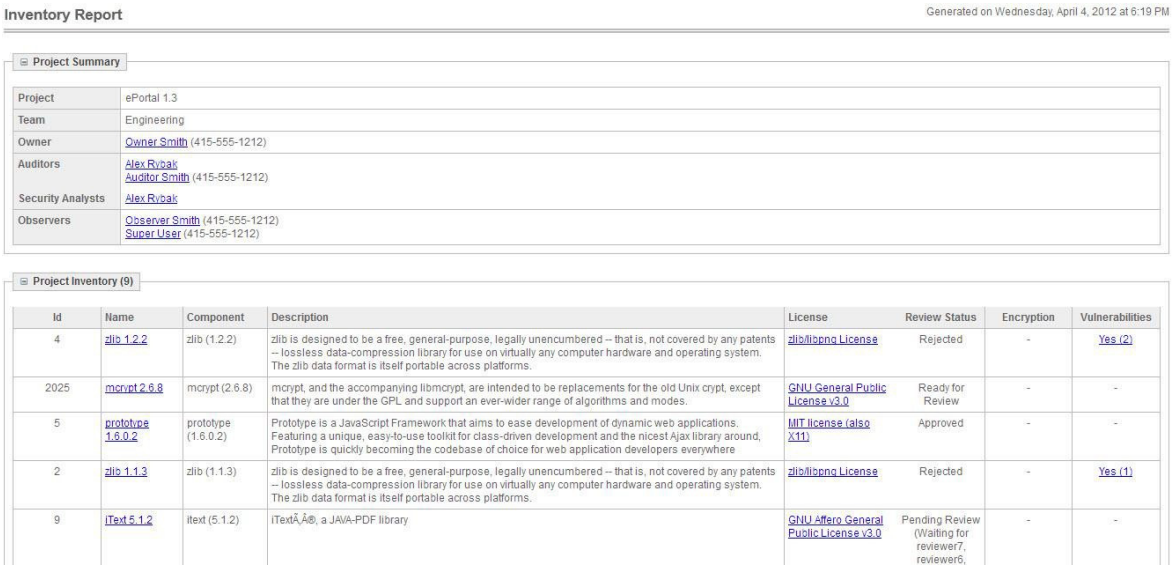


3. (Optional) Enter or edit a report name and description on this page.
4. Select a team name in the **Teams** column to enable and view associated project choices in the **Project** pane.
5. Select a project in the **Project** pane.
6. Click the **Global** checkbox or double-click on a team in the **Teams** column to view associated projects in the Projects pane. Then, select the project in the Projects column.

7. To create versions of the report in Excel or XML format, check the associated checkboxes.

8. Click **Generate**. When the **Inventory Report** is available for viewing, a dialog box opens with a link to the report. The report is presented for the project you selected. Each **Inventory** report has two sections. The **Summary** section provides general project information. The **Inventory** section lists ID, name, component, description, license, review status, encryption, and vulnerabilities. Certain elements in the report are linked to additional information.

Note ▪ *The scores and severities displayed for security vulnerabilities (when you click **Yes** in the **Vulnerabilities** column to view the security vulnerabilities associated with the inventory) are based on the CVSS v2 scoring system.*



Inventory Report — Generated on Wednesday, April 4, 2012 at 6:19 PM

Project Summary

Project	ePortal 1.3
Team	Engineering
Owner	Owner Smith (415-555-1212)
Auditors	Alex Rybak Auditor Smith (415-555-1212)
Security Analysts	Alex Rybak
Observers	Observer Smith (415-555-1212) Super User (415-555-1212)

Project Inventory (9)

Id	Name	Component	Description	License	Review Status	Encryption	Vulnerabilities
4	zlib 1.2.2	zlib (1.2.2)	zlib is designed to be a free, general-purpose, legally unencumbered -- that is, not covered by any patents -- lossless data-compression library for use on virtually any computer hardware and operating system. The zlib data format is itself portable across platforms.	zlib/libpng License	Rejected	-	Yes (2)
2025	mcrypt 2.6.8	mcrypt (2.6.8)	mcrypt, and the accompanying libmcrypt, are intended to be replacements for the old Unix crypt, except that they are under the GPL and support an ever-wider range of algorithms and modes.	GNU General Public License v3.0	Ready for Review	-	-
5	prototype 1.6.0.2	prototype (1.6.0.2)	Prototype is a JavaScript Framework that aims to ease development of dynamic web applications. Featuring a unique, easy-to-use toolkit for class-driven development and the nicest Ajax library around, Prototype is quickly becoming the codebase of choice for web application developers everywhere	MIT license (also X11)	Approved	-	-
2	zlib 1.1.3	zlib (1.1.3)	zlib is designed to be a free, general-purpose, legally unencumbered -- that is, not covered by any patents -- lossless data-compression library for use on virtually any computer hardware and operating system. The zlib data format is itself portable across platforms.	zlib/libpng License	Rejected	-	Yes (1)
9	iText 5.1.2	itext (5.1.2)	iTextÃ‚Â®, a JAVA-PDF library	GNU Affero General Public License v3.0	Pending Review (Waiting for reviewer7, reviewer6,	-	-

Conducting Research

The Research feature allows you to search the Code Insight Data Library for components and licenses, as well as to create and delete components, component versions, licenses, and vulnerabilities (provided that you have the correct permissions).

Note ▪ *Only custom data may be deleted.*

See the following topics for more information:

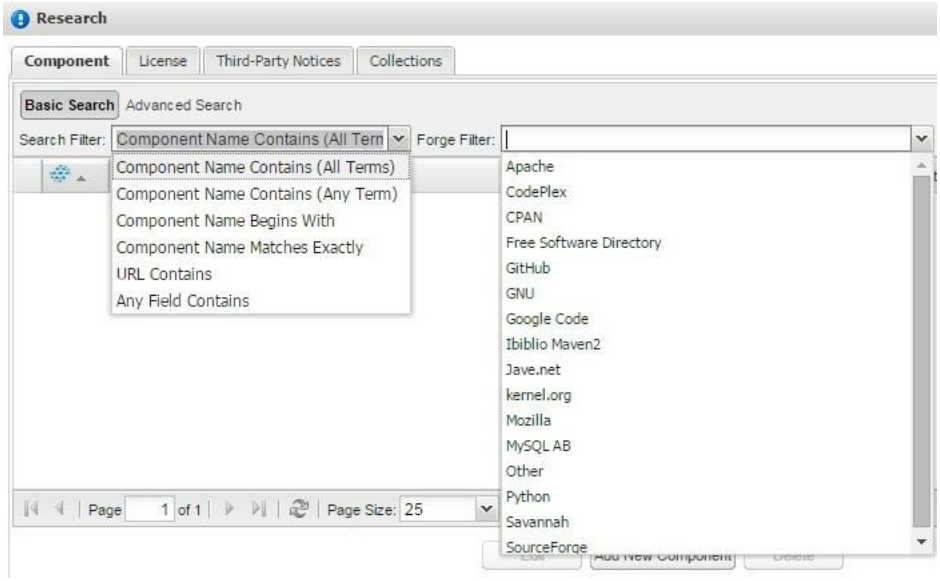
- Searching for Components
- Component Search Tips
- Component Fields
- Component Search FAQ
- Creating a New Component

- Deleting a Custom Component
- Adding an Existing Vulnerability to a Component Version
- Adding a New Vulnerability to a Component Version
- Disassociating a Custom Vulnerability from a Component Version
- Searching for Licenses
- Editing a License
- Viewing, Adding, or Editing a New Obligation
- Creating a New License
- Adding New Licenses to Components
- Deleting a License from Components

Searching for Components

Task ***To search for a component in the Code Insight Data Library, do the following:***

1. Click Research on the **Main** menu bar. The **Research** page appears.



The default for component search is a **Basic Search**. It includes a Search Filter and Forge Filter to help you limit your search results.

Note ▪ *The defaults for these filters are configured by the Code Insight Administrator and may be edited at any time.*

2. Select a search filter from the pull-down menu:

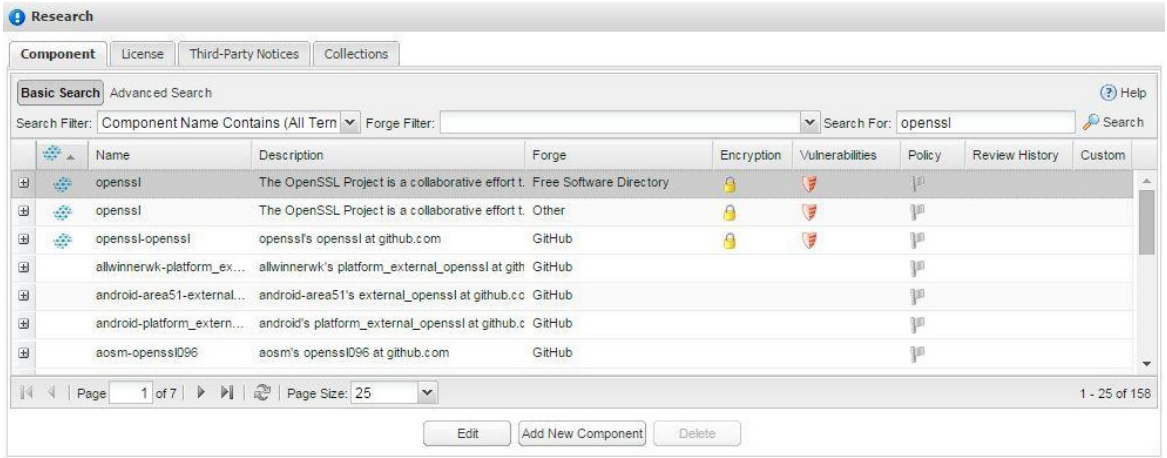
   - **Component Name Contains (All Terms)**: searches all component names and returns all components whose name contains all of the submitted search terms.

     Note ▪ *Component Name refers to the name of the component assigned by Code Insight or by the user in the case of a custom component. Component Names typically do not contain spaces.*

   - **Component Name Contains (Any Terms)**: searches component names and returns all components in which the component name contains any of the submitted search terms.
   - **Component Name Begins With**: searches component names and returns all components in which the component name begins with a submitted search term.
   - **Component Name Matches Exactly**: searches component names and returns all components whose name matches the submitted search term exactly.
   - **URL Contains**: searches component URLs and returns all components whose URL matches the submitted search term URL in part or in full.
   - **Any Field Contains**: searches all the fields outlined above, as well as the component Description field, and returns all components that contain text in one of these fields that matches the submitted search term.

3. Select a forge from the **Forge Filter** or leave it blank to search across all forges. The Forge Filter allows you to limit your search to a specific forge, also known as a "repository" or "source" of code. For example, the Free Software Directory forge references all components in the Code Insight Data Library that were obtained from https://directory.fsf.org, and the GitHub forge references all projects obtained from https://github.com/.

4. Enter the name of a component in the **Search For** box and click the **Search** button (Search) to search for the component. Every component that matches the specified search criteria will be listed in the results. For example, a search for the term "openssl" without a specified forge filter, results in multiple entries for the OpenSSL component, including an entry for the component obtained from "The Free Software Foundation", an entry from the "OpenSSL Project" website (forge is listed as "Other") and an entry for the component obtained from "GitHub".



The result set is sorted first by all components marked as *Important*, indicated by the icon to the left of the component name. A component is designated *Important* if it meets one or more of the following criteria:

- **Code Insight Top Component**: this popular OSS component was manually inspected and edited by the Code Insight Library Team for accuracy. Information for Code Insight Top Components is delivered to the system via Electronic Update. Ensure that you have the latest update for the most accurate information.
- **Referenced Component**: the component is associated with a Request (including Draft and Rejected request), Policy, Group (including unpublished group), Inventory Item or third-party notice. These components are classified as Important because they are currently in use by one or more Code Insight users.
- **Custom Component**: the component was manually created by a Code Insight user and marked as Important via the Important checkbox in Component Details. Note: custom components have negative component IDs. A custom component can be unmarked as Important by unchecking the Important checkbox in the Edit Component menu.

5. To re-sort the result set, select any column to merge the results for Important and all other components into a single list.
6. Click expand ( ) to the left of the component entry to expand to view component details such as Title, Forge, URL, Description, Versions, Licenses, Vulnerabilities, Encryption information, and Custom information.
7. Click one of the **Manage** buttons to view detailed information for Versions, Licenses and other data available for the component entry.

The following list summarizes other features that may be available in **Component Details**.

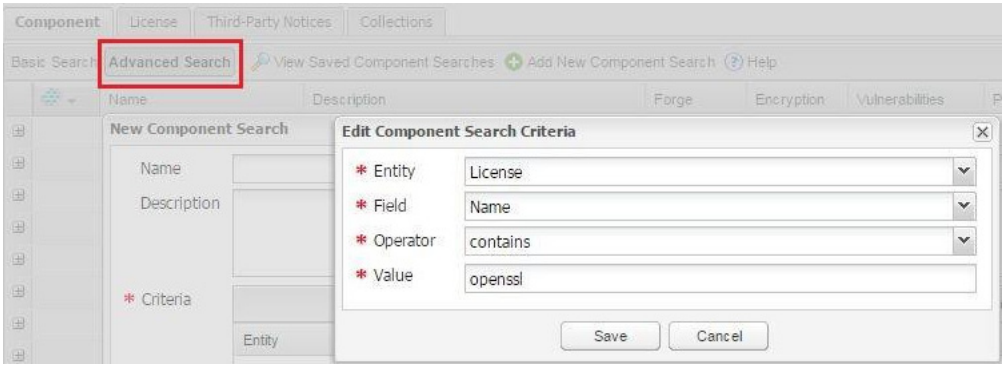
Component	Icon	Description
Encryption Details		Details related to whether or not encryption is required.
Security Vulnerability Details icon		Indicates that security vulnerabilities exist for one or more component versions. Click to view details about the vulnerabilities. ***Note*** ▪ *The scores and severities displayed in the security vulnerability details are based on the CVSS v2 scoring system.*
Component Policy Flag icons		Component always approved for use.
		Component always rejected for uses.
		Component has unknown policy since it depends on use.
		Component does not have a matching policy.

Component	Icon	Description
Approval Status icons		Component has been previously Approved.
		Component has been previously Rejected.

Component Search Tips

The following are some tips to maximize your success when searching:

- To get more results, use one of the Component Name Contains search filters and substitute spaces for dashes.
- Use the **URL Contains** search filter to search for the URL. For better results, use only essential elements of the URL. For example, if the URL is https://www.github.com/facebook/facebook-ios-sdk try searching with just github.com/facebook/facebook-ios-sdk, or for more results, try github.com/facebook/.
- To narrow the search, click the **Advanced Search** tab. The **Edit Component Search Criteria** dialog appears:



- Add a new component search; try searching based on the component license or other important field. Consult the Advanced Search section for more information on how to configure and use the feature.

Component Fields

The following fields are available for every component in the Code Insight system. Additional component data may be stored in custom metadata fields. See Metadata Framework for more information.

Table 4-1 ▪ Fields and Descriptions

Field	Description
Id	The unique ID of the component. Custom components have negative IDs while non-custom have positive IDs.

Table 4-1 ▪ Fields and Descriptions (cont.)

Field	Description
Name	The name of the component as stored in Code Insight. Code Insight components (non-custom) are typically named according to convention based on the forge from which they are obtained. For example, Code Insight components obtained from GitHub include the author name and project name separated by a "-" in the name (OpenSSL from GitHub has the name "openssl-openssl" in Code Insight). Code Insight component names do not contain spaces. Click the **Help** icon (Help) in the **Components Details** tab to see other naming conventions based on forge.
Title	The unique title of the component as stored in Code Insight, the components (non-custom) of which are also named according to the convention based on the forge. For example, Code Insight components obtained from GitHub include the author name and project name separated by a "/" and appended with the term "GitHub" (OpenSSL form GitHub has the title "openssl/openssl – GitHub".
Forge	The downloaded-from URL pointing to the page from which the component was downloaded.
URL	The project URL pointing to the project home page.
Description	The project description.
Encryption	Indicates whether the component uses encryption.
Vulnerabilities	Indicates whether the component contains one or more vulnerabilities. Click the icon to view details for the vulnerabilities associated with the different component versions. ***Note ▪*** *The scores and severities displayed in the security vulnerability details are based on the CVSS v2 scoring system.*
Custom	Specifies whether the component is custom (user-created) or non- custom (Code Insight-created).
Important	Specifies whether the component has been marked for importance by Code Insight or is referenced in the system.
Available Platforms	The list of operating systems for which this component is developed (if any).
Categories	The list of tags to classify the component in a catalog-type structure. For example, a user may choose to label a set of components as "Permissive" based on their license type.

Table 4-1 ▪ Fields and Descriptions (cont.)

Field	Description
CPE Name	The list of CPE names—manually created or pulled from the National Vulnerability Database—that are mapped to the OSS or third-party component. CPE is a structured naming scheme for a component that includes the component's vendor name and product name and uses the following format: **cpe://<part>:<vendor>:<product>** where <part> is either **a** (applications), **h** (hardware platforms), or **o** (operating systems). To add a custom CPE name for the component or remove any of the component's custom or NVD-published CPE names, click **Manage CPE Name**. When you add a CPE name, Code Insight validates that the name is in the correct format.
Programming Language	The list of programming languages that are used in the codebase for the component.
Last Update Date	The date the codebase was last uploaded to the forge for this component.
Registered Date	The date corresponding to the first code upload to the forge for this component.
Versions	The versions associated with this component.
Licenses	The licenses associated with this component.

Component Search FAQ

1. **How can I tell which component is the correct one?**

   It is always recommended to have some familiarity with the component you are requesting in order to be able to correctly identify it online and in the Code Insight Data Library. The URL is a key to identifying components. If it matches the component you are requesting, this is a reliable indicator that they are the same.

   If you or the developer downloaded the package from another URL, look up the component online to verify what the canonical project page is. Often, developers will download a package from a site other than the main project page.

   Do some Web searches to see if the URLs are related (e.g. fsf.org is related to gnu.org). Look at other information including the descriptions. Look at the project websites and see if they are the same project. If they are, then you can use that component.

2. **What if I cannot find the component?**

   There are several reasons you may not be able to find the component in the Code Insight Data Library, for example:

   - The component is under a different name in the Code Insight data library. Try to search for the component using some of the techniques above.

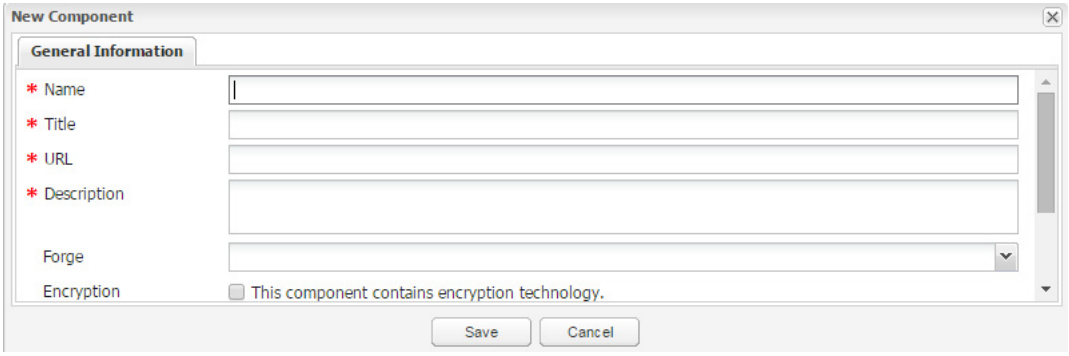
- It is a Commercial component.
- The component is considered to be a subcomponent of a larger project. For example, netkit- base, netkit-ftp, netkit-rsh, are all part of netkit.
- The component has not been collected into data library. Typical reasons for this include:
- It is not traditionally seen as a software library, but is considered to be a module within a larger work.
- It is no longer available on the Internet.
- It is from a non-electronic source (magazine, book, etc.).
- The component is very new, and has not been collected yet. In this case, you may wish to request the Code Insight Library team to collect the component, so the scanner can index the component releases.
- If you cannot find a component in the Code Insight data library, or you wish to categorize them in a different way from how they are in the data library, you can create a custom component.

Creating a New Component

Follow these steps to create a new custom component in the Code Insight system. To limit the number of duplicates, ensure that you create new components only in the case where you are unable to find the component in the system.

Task ***To create a new custom component, do the following:***

1. Click **Research** on the **Main** menu bar. The **Research** page appears.
2. Click **Add New Component**, and then click **Yes** when prompted to create the component. The **New Component** dialog appears.



3. Enter information in the required fields, which are marked with an asterisk (*). The remaining fields are optional. See **Component Fields** for more information.

   The combination of Name and Title must be unique. For example, if there is already a component name “foo” in the Code Insight system, you can create another component named “foo” only if the titles of the two components are not identical.

4. Click **Save** to save the new component. An ID is automatically assigned to the component upon creation. Custom components have negative IDs in the system and are labeled with a **Yes** in the **Custom** field.

Deleting a Custom Component

You can delete only custom components. The Delete feature is disabled for Code Insight components. Follow these instructions to delete a custom component from the system.

Task ***To delete a custom component from the system, do the following:***

Note ▪ *Only custom components can be deleted. The* ***Delete*** *button is disabled for Code Insight components.*

1. Click **Research** on the main menu. The **Research** page appears.
2. In the Search box on the Component tab, enter the name of the component you want to delete. When the **Search Component** box opens, you can choose to search by Component Name or by Filter pull-down menu.
3. Select the component and click **Delete**.
4. Click **Yes** to confirm the deletion.

Creating a New Component Version

Task ***To create a new component version, do the following:***

1. Click **Research** on the **Main** menu bar. The **Research** page appears.
2. In the **Search** field on the **Component** tab, enter the name of the component for which to add a new version.
3. Click the **magnifying glass** icon.
4. Click the left-side plus + icon to expand the information about the desired component.
5. Click **Manage Versions and Vulnerabilities**. The **Version Details** dialog appears.
6. Click **Add New Version**. The **Input** dialog appears.

7. Enter the new component version name and click **OK t**o save the new component version.

Deleting a Custom Component Version

If you create a new component version and decide that you no longer want this component version to be part of the data library, you can delete the component version.

Task ***To delete the component version, do the following:***

1. Click **Research** on the **Main** menu bar. The **Research** page appears.
2. In the **Search** field, enter the name of the custom component version you wish to delete.
3. Select the name of the component, and then click **Delete**. The **Confirmation** dialog appears.
4. Click **Yes** in the **Confirmation** dialog.

Adding an Existing Vulnerability to a Component Version

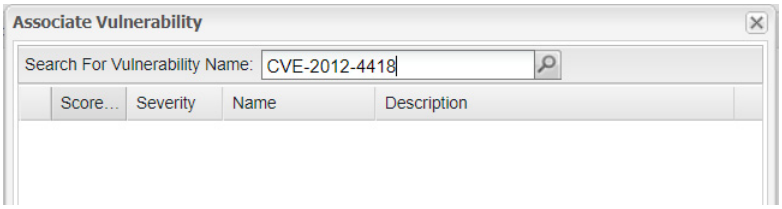
Use the following procedure to manually add an existing security vulnerability to a component version—that is, add a vulnerability already identified in the Code Insight data library but currently not associated with the component version. Once added, this vulnerability is considered a custom vulnerability for the component.

Only users with permission to write to components can perform this task. (See the `component.write.access.user.list` property in `<codeInsightInstallPath>\config\core\core.properties`.)

Task ***To add an existing vulnerability to a component version, do the following:***

1. Click **Research** on the **Main** menu bar. The **Research** page appears.
2. In the **Search** field, enter the name of the component for which you wish to add the vulnerability.
3. Click the **magnifying glass** icon.
4. Locate the desired component, and click the associated **shield** icon in the **Vulnerabilities** column.

   The **Version Details** dialog for the component opens.
5. Locate the component version to which you want to add a vulnerability, and click the **shield** icon in the **Vulnerabilities** column to open the **Security Vulnerabilities** dialog.
6. Click **Associate Vulnerability** to open the **Associate Vulnerability** dialog.
7. In the **Search for Vulnerability Name** field, enter the exact name of the existing vulnerability you want to add.



8. Click the **magnifying glass** icon.
   - If you have entered a vulnerability name that exists in the Code Insight data library, the vulnerability and its details are listed. (Click the plus icon to the left of the vulnerability to show the its description.)

- If you entered a vulnerability name that does not exist in the Code Insight data library, no results are listed. Make sure you have entered the exact vulnerability name and try again. If you continue to see no results, you have the option to create a new vulnerability and associate it with the component version. For details, see the next section, Adding a New Vulnerability to a Component Version.

9. If the security vulnerability displayed is the desired vulnerability, select it and click **Associate** to add it to the component version.

Adding a New Vulnerability to a Component Version

Use the following procedure to manually add a new security vulnerability to the component version—that is, create a vulnerability that has not yet been identified in the Code Insight data library and associate it with the component version. Once the vulnerability is created and associated with the component version, it is added to the data library as a custom vulnerability available for association with other components.

Only users with permission to write to components can perform this task. (See the `component.write.access.user.list` property in `<codeInsightInstallPath>\config\core\core.properties`.)

Task ***To add a new vulnerability to a component version, do the following:***

1. Click **Research** on the **Main** menu bar. The **Research** page appears.
2. In the **Search** field, enter the name of the component for which you wish to add a new vulnerability.
3. Click the **magnifying glass** icon.
4. Locate the desired component, and click the associated **shield** icon in the **Vulnerabilities** column.

   The **Version Details** dialog for the component opens.
5. Locate the component version to which you want to add a vulnerability, and click the **shield** icon in the **Vulnerabilities** column to open the **Security Vulnerabilities** dialog.
6. Click **Add New Vulnerability** to open the **New Vulnerability** dialog.

7. Enter the required vulnerability name and description, and select a severity from the **Severity** pull-down menu. The URL field is optional and can be left blank.
8. Click **Save** to save the new vulnerability and associate it with the selected component version.

Disassociating a Custom Vulnerability from a Component Version

This section describes how to disassociate a custom vulnerability from a component version.

Only users with permission to write to components can perform this task. (See the `component.write.access.user.list` property in `<codeInsightInstallPath>\config\core\core.properties`.)

Note that a custom security vulnerability for a component version is one that was manually added to the version using a public REST or Java API or either of these procedures: Adding an Existing Vulnerability to a Component Version or Adding a New Vulnerability to a Component Version.

Task ***To disassociate a custom vulnerability from a component version, do the following:***

1. Click **Research** on the Main menu bar. The **Research** page appears.
2. In the **Search** box, enter the name of the component.
3. Click the **magnifying glass** icon.
4. Locate the desired component, and click the associated **shield** icon in the **Vulnerabilities** column.

   The **Version Details** dialog for the component opens.
5. Locate the component version that has the custom vulnerability that you want to disassociate, and click the **shield** icon in the **Vulnerabilities** column.

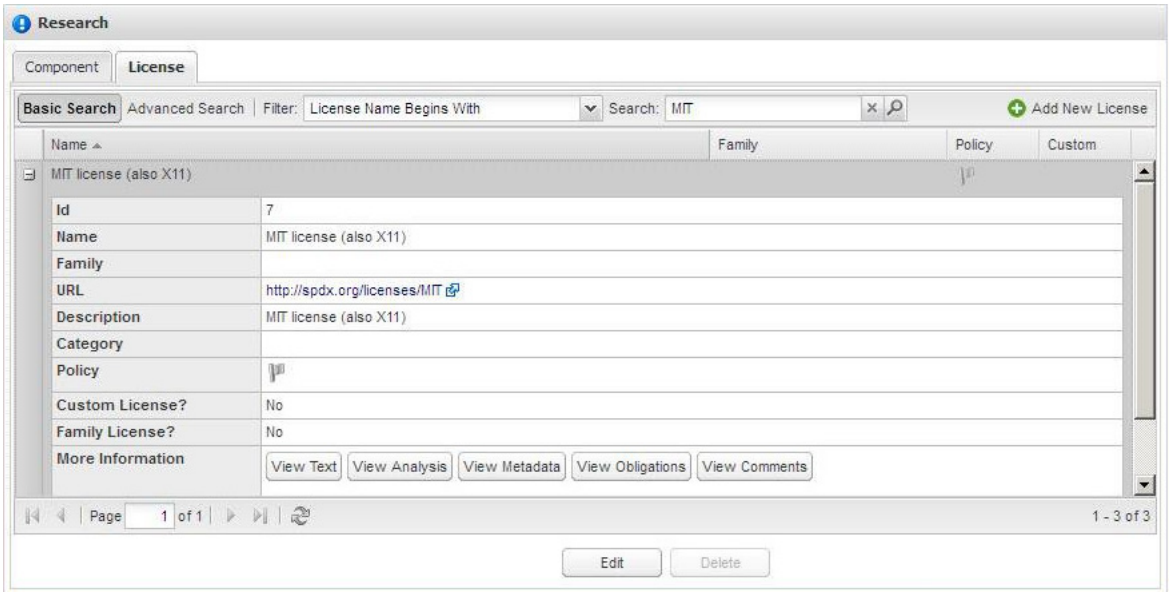
   The **Security Vulnerabilities** dialog opens.
6. Click the red **x** icon next to the custom vulnerability that you want to disassociate from the component version. (Only custom vulnerabilities have the **x** icon.)
7. Click **Yes** to confirm the deletion.

Searching for Licenses

Note ▪ *This process works only with previously created custom vulnerabilities.*

Task ***To see what components are available for a project:***

1. Click **Research** on the Main menu Bar. The **Research** page appears.
2. Select the **Licenses** tab.
3. Enter the name of a license in the Search box, and click the **magnifying glass** icon. Each license that matches the search string appears in the search results with license name, family, policy, and custom information.

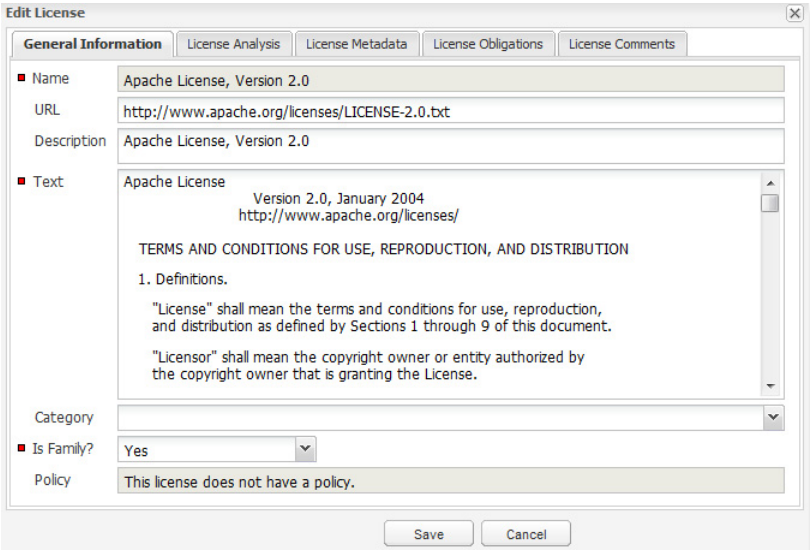


4. To refine your search, use the **Filter** pull-down menu. Using the **Any Field Contains** filter value will extend the license search to any associated metadata fields as well.
5. Click on the license you want to view.
6. Click on the left-side plus + icons to expand the information about the desired license, as shown in Figure 53. If you click on any of the rows, the Edit button toggles on, and you can edit information.
7. You can now view all license text, analysis, metadata, obligations, and comments associated with this license from this page. (Also, the edit button at the bottom of the window is now enabled.)

Editing a License

Task ***To edit a license, do the following:***

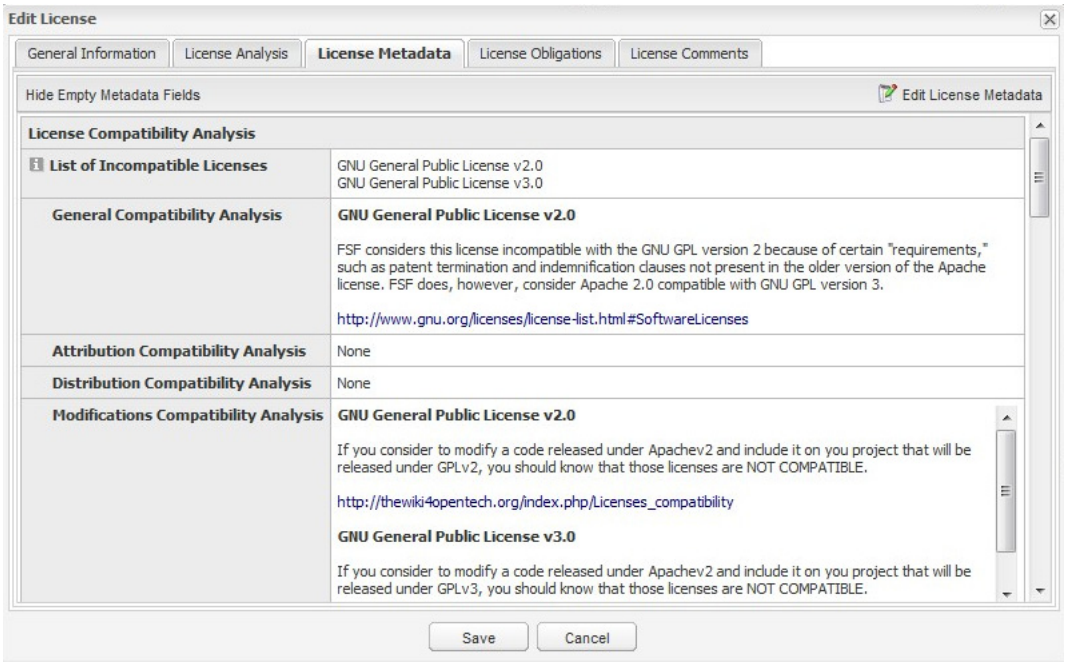
1. Click **Research** in the Main menu bar.
2. Enter a license name in the **Search** field, and click the **magnifying glass** icon.
3. In the search results, click the plus + icon next to the license you want to edit.
4. Click **Edit**. The **Edit License** page appears.



5. Click the appropriate tab to view and edit license information:
   - **General Information**: The license name, URL, description, and license text. The **Category** field can be set so that you skip legal review. You can also choose to alter the workflow routing if you decide you wish to skip review levels. The **Family** pull-down allows you to indicate if a license is in a family. The **Select Family** pull-down menu allows you associate the license with a family and choose what characteristics the license will inherit. The **Policy** field contains relevant policy information.
   - **License Analysis**: This is not editable. Instead you can view the ranking of risk level, license requirements, and descriptions associated with the selected license.
   - **License Metadata**: The license metadata field definitions and value assignments are supported via API and external scripts. The assigned license metadata value fields are visible and can be searched against in the Web UI. see “Metadata Framework” for more information related to the metadata process and supported entities and datatypes.
   - **License compatibility**: On the Metadata tab, at the top, analyses of different license compatibility are provided. These analyses allow you to see which categories of compatibility a license may evoke.

- **License Obligations**: This tab contains the set of license obligations associated with a given license. If a license belongs to the license family and does not have any license obligations, it will inherit the license obligations from the associated license family. License obligations can be defined in the Web UI by clicking on the plus + icon, or they can be bulk loaded by selecting **Import** from the **Administration** menu. Only an Application Administrator can bulk-import license obligations.

The following is an example of the information that appears on the **Metadata** tab.

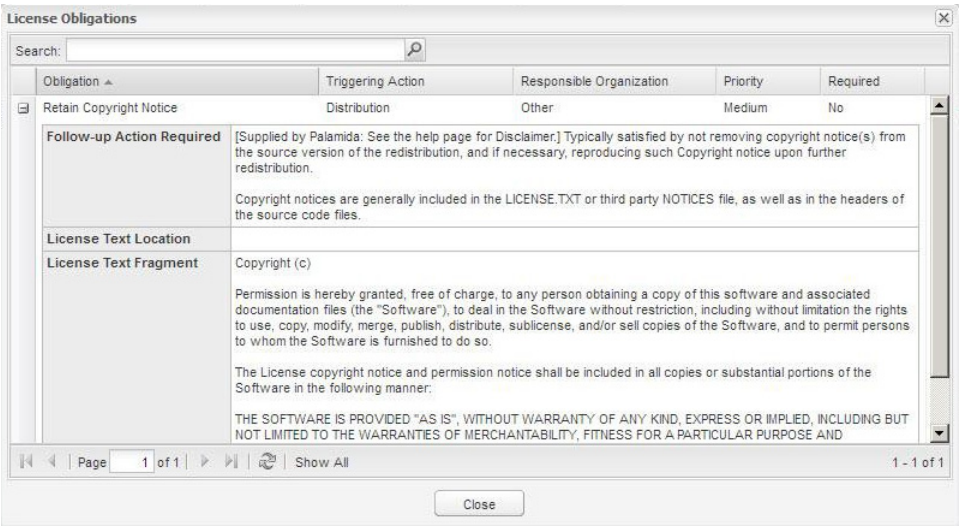


6. When you finish viewing and editing the information, click **Save**.

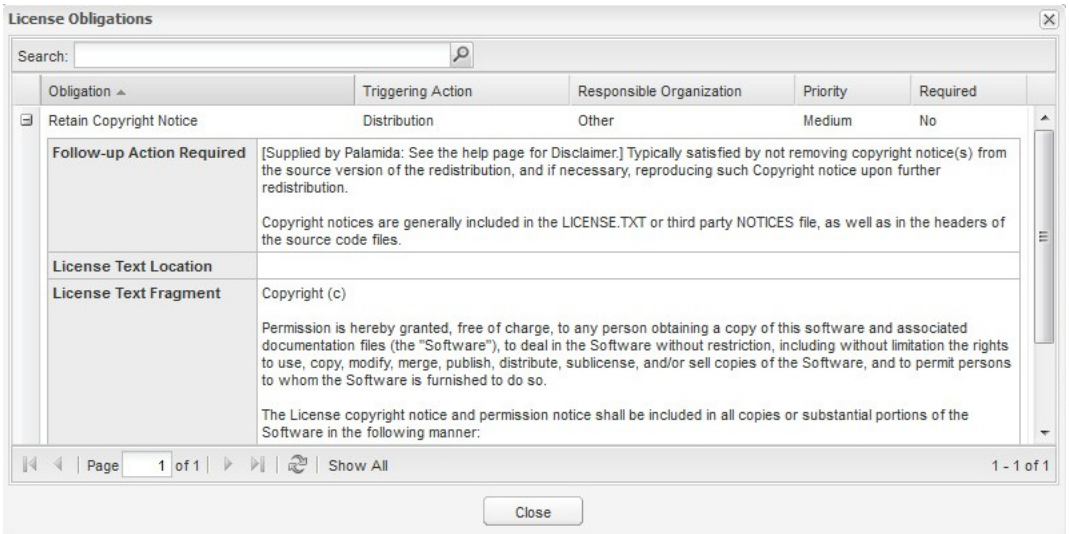
Viewing, Adding, or Editing a New Obligation

Task ***To view, add, or edit an obligation, do the following:***

1. Click the **Research** button in the **Main** menu bar.
2. Enter a license name in the **Search** field, and click the **magnifying glass** icon.
3. Select the license name from the list.
4. Click the **View Obligations** button to see any obligations associated with that license.
5. Click the **Add New Obligation** button to define a new license obligation, or click on the **Pencil and Paper** icon in the **Actions** column.



6. To add a new obligation to a license, go back to the **License Edit** page.
7. Enter the name in the **Search** field of a license for which you want to add a new obligation.
8. Click the **magnifying glass** icon.
9. Select a license from the search results, and click the plus + icon to expand the license information.
10. Click the **Edit** button at the bottom of the page.
11. Click the **License Obligations** tab.
12. Click the **Add New Obligation** icon. The **License Obligation** page appears.



The pull-down menus on the **New Obligation** window can be populated using SQL scripts provided with the product. These menus allow you to specify the following:

- **Obligation**: This is a list of the common types of license obligations provided via a pull-down menu.
- **Follow Up Action**: This text field allows you to enter specific information related to how users are to address the specific obligation for the current license.

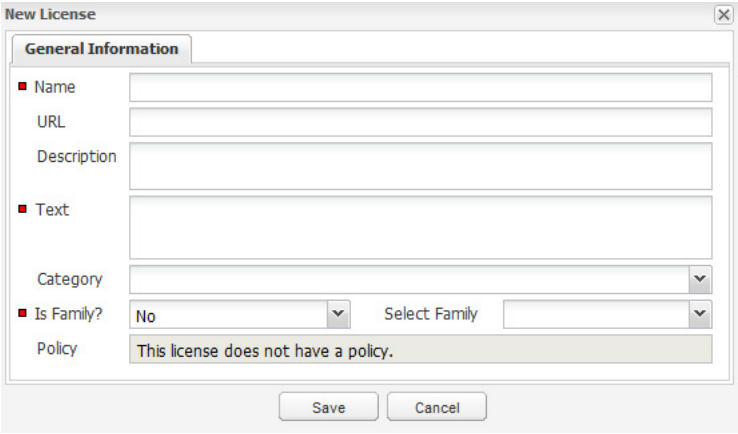
- **License Text Location**: This allows you specify a path.
- **License Text Fragments**: You can cut and paste license version and text into these fields and hyperlink information.
- **Triggering Action**: This is a list of common actions that may trigger a license obligation.
- **Responsible Organization**: This is a list of organizations that typically are responsible for ensuring that required license obligations comply.
- **Priority**: Low, Medium, or High.
- **Required**: Yes or No.

Creating a New License

You create the new license from the License tab. But you add the new license from the Component tab. You can only add licenses to existing components. For details about associating a license with a component, see “Adding New Licenses to Components.

Task ***To create a new license, do as follows:***

1. Click **Research** on the **Main** menu bar. The **Research** page appears.
2. Select the **License** tab.
3. Click **Add New License**. The **New License** dialog appears.



Note ▪ *The Policy field is read-only and contains information regarding any associated policies for this license.*

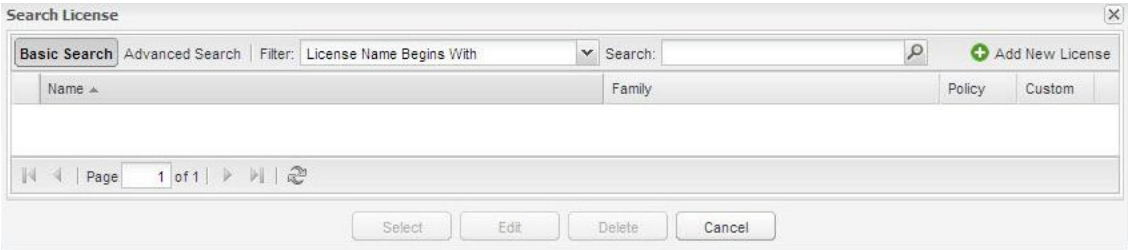
4. Click **Save** to save the new license.

Adding New Licenses to Components

Before you can add or associate a license with a component, you have to create that license. See Creating a New License for more information.

Task ***To add a license to a component, do as follows:***

1. Click the **Research** button on the **Main** menu bar. The Research page appears.
2. Select the **Components** tab.
3. In the **Search** field, enter the name of the component with which the license is associated and click the **magnifying glass** icon.
4. Locate the component to which you wish to add a license, and click the left-hand **Plus** icon to expand the component description.
5. Click the **Manage Licenses** button. The license associated with the component appears.
6. In the **License Details** for that particular component, click the **Add New Component** icon to add a new license. The **Search License** dialog appears.



7. Select the desired license to add. You can also edit or delete a license by using the buttons at the bottom of the **Search License** dialog, or you can create a new license if one does not exist.

Note ▪ *The **Add License** button adds an existing license to the license list. The **Add New License** button opens a dialog to create a new license, which can be added to the list.*

Deleting a License from Components

Task ***To delete a license from a component, do the following:***

1. Click the **Research** button on the **Main** menu bar to open the **Research** page.
2. Select the **Components** tab.
3. In the **Search** field, enter the name of the component with which the license is associated.
4. Click the **magnifying glass** icon.

5. When you locate the component associated with the license, expand the description by clicking on the left-hand plus + icon.
6. Scroll down and click **Manage Licenses**.
7. Click the **Delete** icon to delete the license.

5

Using My Settings and Help Functions

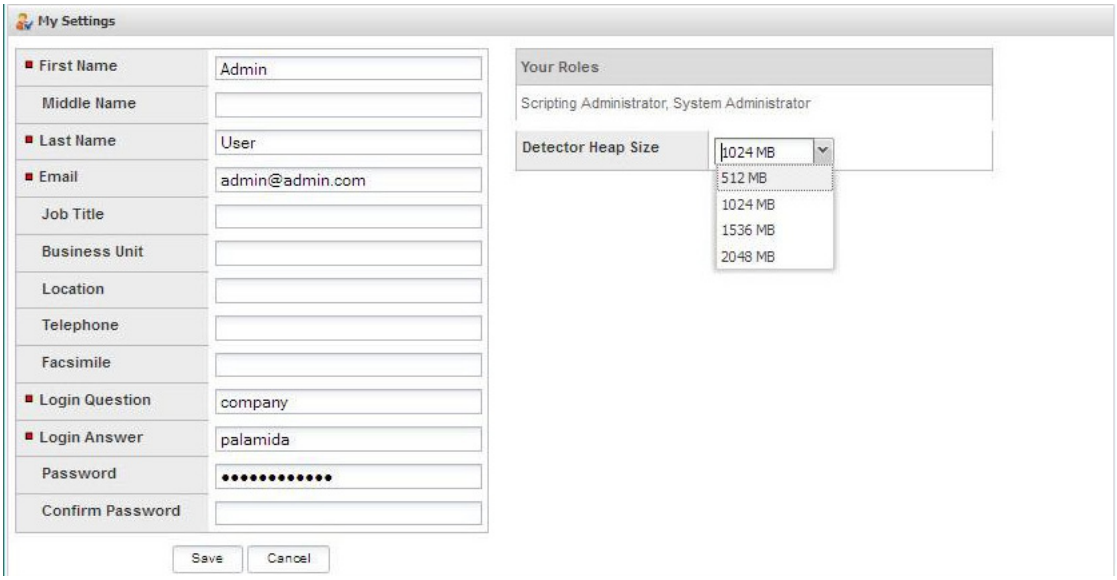
This section provides the steps to access My Settings and Help Functions, which are available to all users:

- My Settings
- Accessing Help

My Settings

Task ***To update your settings, do the following:***

1. Log into Code Insight and click **My Settings** in the top right-hand corner. The **My Settings** page appears.



2. Enter or update the information in any of the fields.

Note ▪ *If you are changing your password, do not enter a password that contains a left angle-bracket (<) character. When you attempt to save your user profile, an error message is displayed, stating that the password is invalid due to this character and that a different password must be entered.*

3. Select the appropriate value from the **Detector Heap Size** pull-down menu.
4. When you finish entering and updating information, click **Save**.

Accessing Help

Task ***To access help, do the following:***

1. Log into Code Insight and click **Help** in the top right-hand corner. The **Help** page appears.
2. Select tabs to view links to current application documentation, release notes, known issues, frequently asked questions (FAQ), licensing, legal disclaimers, and about information.

6

Conducting Audits & Reviewing Inventory

This section contains the following topics:

Conducting Audits

Code Insight provides a process that allows the owner to select a project auditor, and the progress of the audit can be managed as part of the project workflow.

After a project is completely defined by an owner and the policies are reviewed with respect to the new project by the policy administrator, the selected project auditor is assigned a task to conduct an audit for the project codebase.

The auditor is responsible for running scans for the workspaces associated with the project, as well as analyzing results and building project inventory based on forensic evidence and automatic OSS component version detection.

Once the project audit is completed by the auditor, the task is closed allowing the project to obtain a compliant status (assuming all detected inventory is compliant as well).

The activities associated with performing a code audit and analysis, as well as publishing project inventory items are detailed in Auditing and Analysis Overview.

Reviewing Inventory

Code Insight provides a process by which the project owner can review inventory items and determine whether to make an immediate decision or to require a full review including the completion of a request form. We refer to this process as a Quick Review.

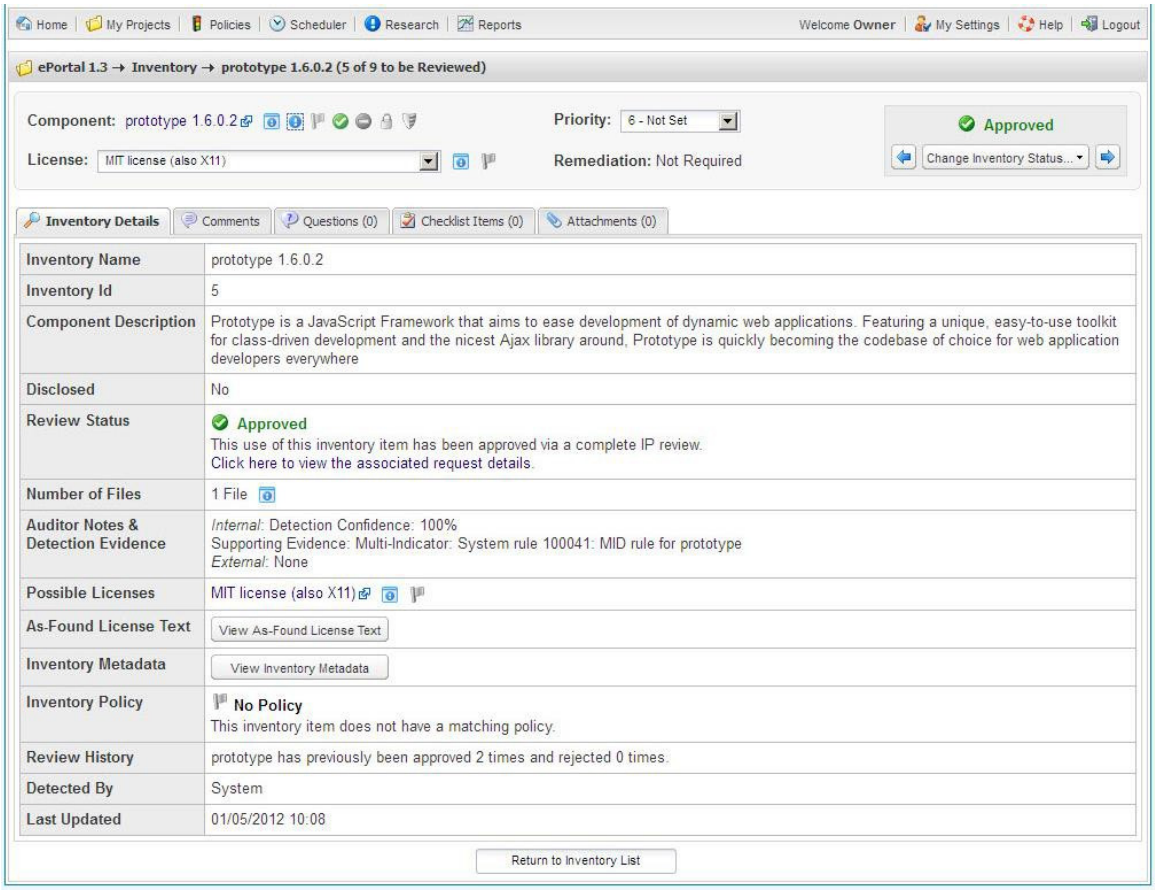
Once the auditor publishes groups from the Detector client, the published groups are available as inventory items in the web client. The project owner can at any time review an inventory item and immediately mark it as approved or rejected for use. Using this operation allows you to skip the full review (workflow) as defined for that project.

Generally, you might use this process as part of a meeting involving all concerned parties: development, legal, security, and any other stakeholders. For example, stakeholders might discuss a set of inventory items ready for review, and then decide if an inventory item should be approved for use, rejected for use, or require a full review including completing the request form associated with the project. If there are any security vulnerabilities associated with the inventory item, a security analyst is required to review these vulnerabilities.

Viewing Inventory Items

Task ***To view inventory item details, do the following:***

1. Log into Code Insight.
2. Click the **My Projects** button in the Main menu bar.
3. Select the view **magnifying glass** icon in the Actions column next to the project name containing the component for which you wish to view version details. This opens the Project Details tabbed page
4. Click on the **Inventory** tab on the **Project Details** page.
5. Select the inventory item you wish to view, and click on the **magnifying glass** icon associated with that item. The I**nventory Details** tab appears.



The **Inventory Details** tab contains all the information necessary to determine whether to approve, reject, or schedule a full review for the item.

Key Inventory Page Items

Key items on the Inventory page are presented in the following tables.

Component Details

Component details and their associated icons are as follows:

Component	Icon	Description
Available Versions icon		List of available versions as well as the count of associated security vulnerabilities.
Component Metadata icon		Show the component metadata dialog;

Component	Icon	Description
Component Policy Flag icons		Component always approved for use.
		Component always rejected for uses.
		Component has unknown policy since it depends on use.
		Component does not have a matching policy.
Approval Status icons		Component has been previously Approved.
		Component has been previously Rejected.
Encryption Details		Details related to whether or not encryption is required.
Security Vulnerability Details icon		Indicates that vulnerabilities are associated with one or more component versions. Click to view details about these vulnerabilities. ***Note*** ▪ *The scores and severities displayed in the security vulnerability details are based on the CVSS v2 scoring system.*

License Details

License details are available for viewing by clicking on associated License icons.

Component	Icon	Description
License Details icon		Includes the license text, as-found license text, license comparison, license analysis (if available), license metadata (including compatibility analysis), license obligations, and license comments.
Component Policy Flag icons		License always allowed.
		License never allowed.
		License has unknown policy since it depends on usage.
		License does not have matching policy.

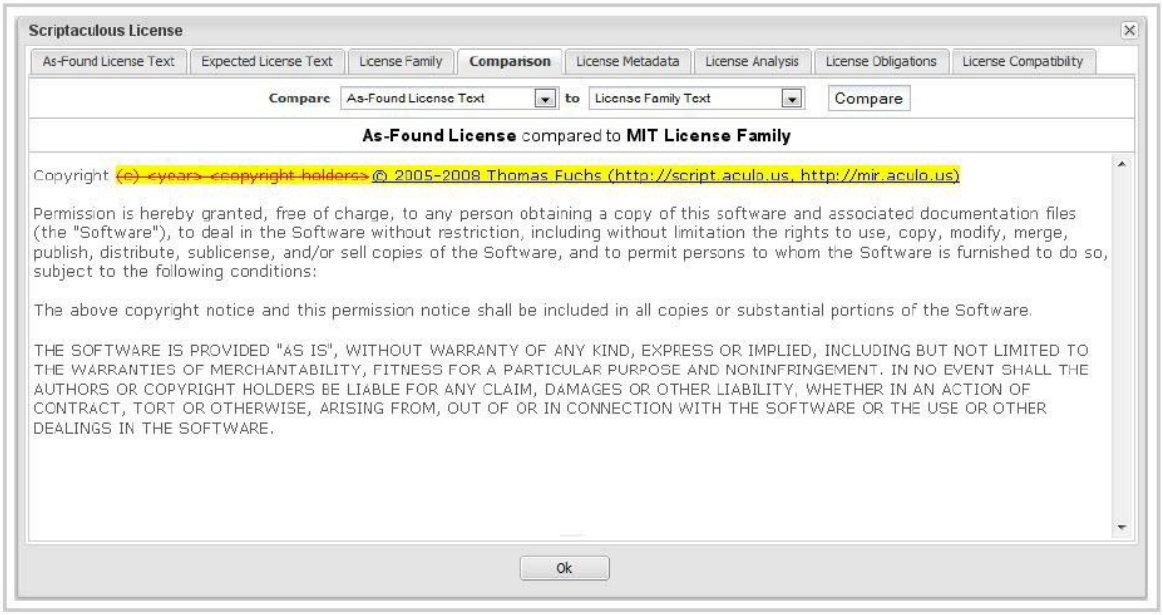
License Text Comparison

The **License text** comparison feature allows you to compare the following types of license text associated with a given inventory item:

- License family
- Expected license
- As-found license

Task ***To compare license types, do the following:***

1. To view an inline comparison of two license texts associated with an inventory item, go to an inventory item.
2. Click the **View License Details** icon ( ) next to the license name. The License Comparison page appears:



3. To compare two different license text types, select the two license text types to compare from the pull-down menus:
   - **License Text**: The license text for the selected license from the Code Insight Compliance Library.
   - **As-Found License Text**: The value of the As-Found License Text group field in Detector that was entered by the auditor.
   - **License Family**: The license text of the license family to which the selected license belongs.
4. Click the **Compare** button.

Note ▪ *If a license text type is empty, it isn't viewable.*

Other Inventory Page Header Items

Items that appear near the top of the Inventory page are as follows:

Table 6-1 ▪ Page Header Items

Item	Description
License	Allows for inventory reviewer to override the selected license for the inventory item. The possible values for the license pull-down menu come from the possible license list as defined by the auditor via the Code Insight client.
Priority pull-down menu	Project owner may change this value, and others will see it as a read-only fields
Remediation status	Denotes the current state of scheduled remediation work for this inventory item
Review Status/Change Inventory Status... Section	Shows the current review status of the inventory item Project owner has the option of marking the inventory item as approved for use, rejected for use, or schedule a full review. If a full review is scheduled, an option to associate an existing request or create a new request will be available. A security analyst may see an option to review security vulnerabilities if any are associated with the component version, and the inventory item requires a full review. Otherwise, a view vulnerabilities button will be available.

Inventory Item Tabbed View

The following tabbed views are associated with Inventory:

Table 6-2 ▪ Tabbed View Items

Item	Description
Comments	List of comments for the current inventory item
Questions	List of questions and answers for the inventory item
Checklist Items	List of check list items of type general, legal, remediation, or security

Table 6-2 ▪ Tabbed View Items

Item	Description
Attachments	List of attached files for the current inventory item

Additional Inventory Details on Tabbed Views

Additional information available for viewing is as follows:

Table 6-3 ▪ Addition Details

Item	Description
Inventory Name	Name of the inventory item (as entered by the system or auditor)
Inventory ID	ID of the inventory item (unique to the system)
Component Description	Description of the associated component
Disclosed	This is a yes or no value that indicates whether or not the inventory item was disclosed before the audit or whether the finding was a surprise.
Review Status	Detailed explanation of the current review status of the inventory item as well as hyperlinks to the policy or request that was used to automatically review the inventory item (if applicable)
Number of Files	List of files associated with the inventory item (ordered by file path or workspace name)
Auditor Notes & Detection Evidence	Auditor notes along with explanation of how the inventory item was detected (if detected by is System)
Possible Licenses	List of all licenses associated with component or version. This list can be modified by the auditor via the Detector client, or by any QuickReview™ Facilitator via the web UI.
As-Found License Text	The license text as found in the codebase by the auditor.
Inventory Metadata	Link to the inventory metadata dialog.
Inventory Policy	Matching policy with a details link including the component name, version, and license
Review History	Number of times, historically, requests for the inventory item were approved and rejected in the past.

Table 6-3 ▪ Addition Details

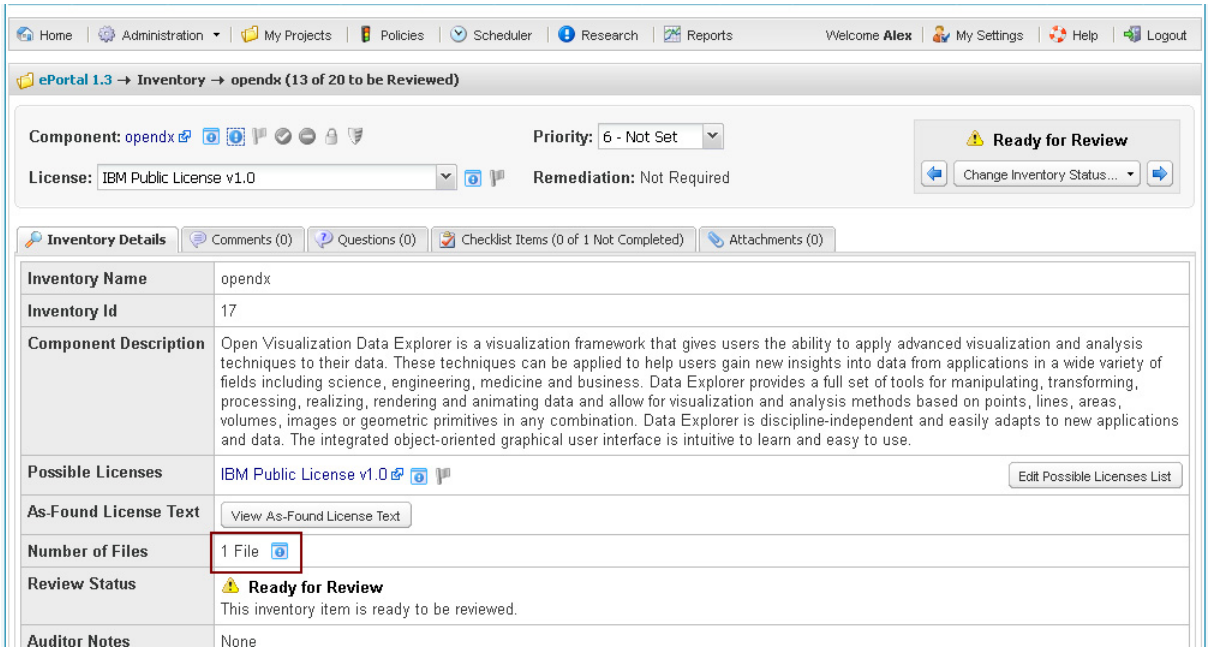
Item	Description
Detected By	Name of auditor who created the inventory item ("System" means that this item was automatically detected by the system.)
Last Updated	Date that the inventory item was last updated

Viewing Source Code Fingerprint Match Highlights

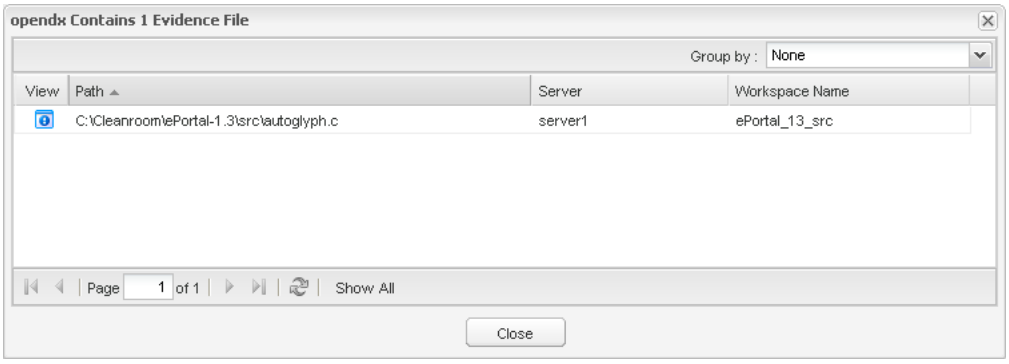
Code Insight provides the ability to view ASCII text file contents associated with a given inventory item as well as source code fingerprint match highlights (if available).

Task ***To view the contents of a file along with source code fingerprint match highlights, do the following:***

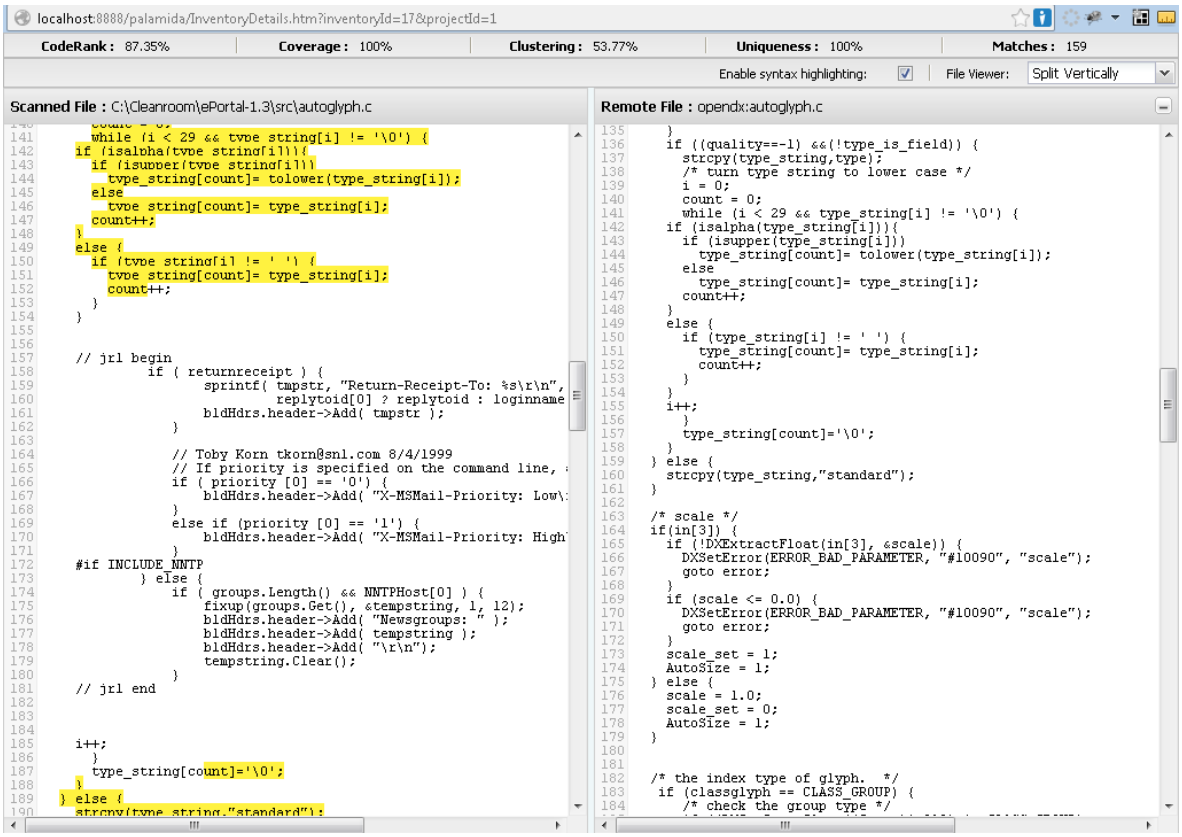
1. Click the **View Inventory Files** icon ( ) in the **Number of Files** field. The **View Inventory Files** page appears:



2. Click the **View** icon next to the desired file.



If the auditor marked several matched files as best source matches, you will see a pull-down menu from which you can select the desired remote matched file for comparison. If the auditor did not mark any matches, the system will show the top match automatically. A new **View Evidence File** window opens with the selected scanned file on the left and the selected (or best) remote matched file on the right.



3. Review the top header for the corresponding source match heuristics for the scanned file. These include Code Insight CodeRank™, Coverage, Clustering, Uniqueness, and Matches, the number of source matches.

4. Below the top header, you can choose to check the Enable syntax highlighting checkbox. This allows for highlighting in programming-language-specific color coding (as you might see in a text editor).

5. Review the File Viewer option. The pull-down allows you to split the dual-pane view horizontally or vertically.

6. Click the **minus** icon in the top-right of the **Remote File** panel to close the panel and show only the contents of the scanned file.

7

Performing Quick Reviews & Approving or Rejecting Inventory

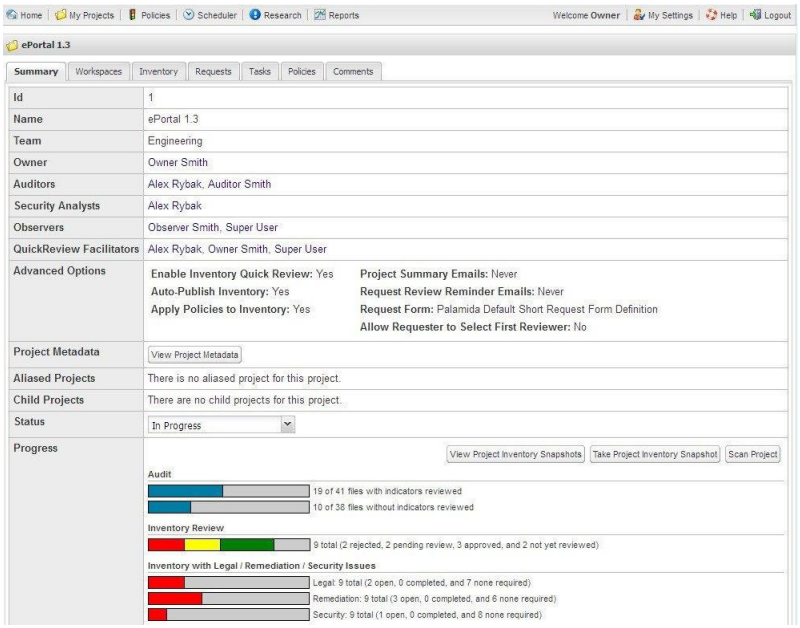
This section explains how to perform a quick review of inventory and mark it as approved or rejected:

- Performing Quick Reviews & Marking Inventory
- Scheduling Inventory for Full Review

Performing Quick Reviews & Marking Inventory

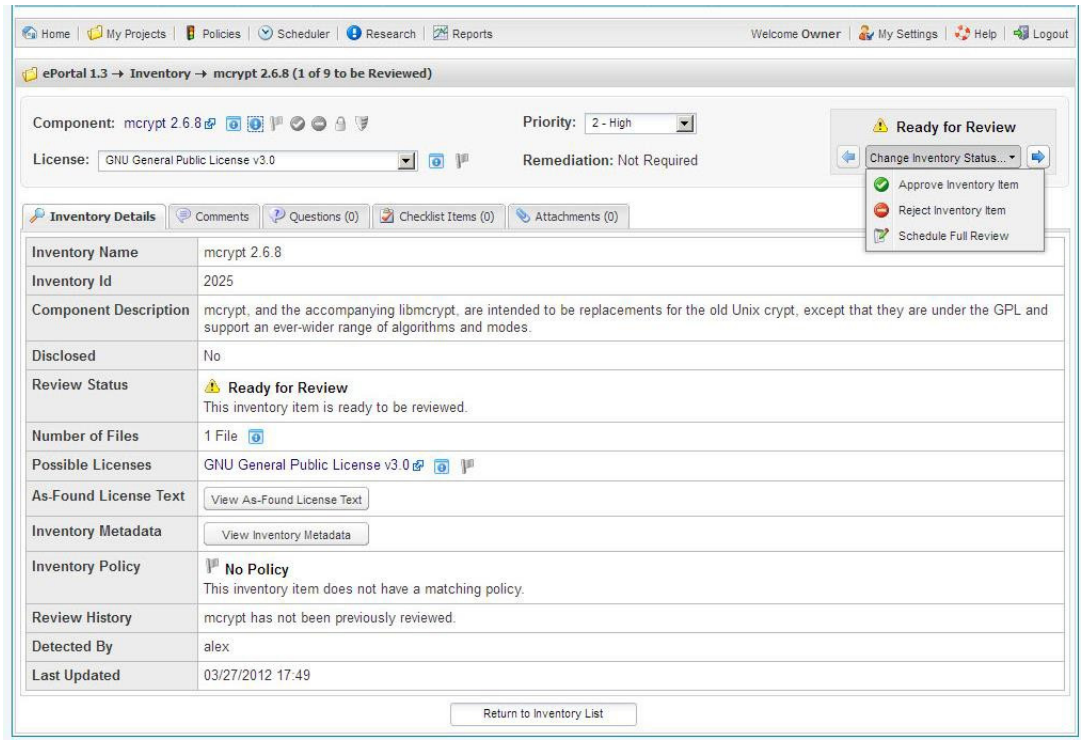
Task ***To perform a quick review and mark inventory as approved or rejected, do the following:***

1. Log into Code Insight as an owner.
2. Click the **My Projects** button in the Main menu bar.
3. Select a project for which you want to review inventory by clicking on the associated view **magnifying glass** icon in the **Actions** column. A Project tabbed view appears that includes the **Summary**, **Workspaces**, **Inventory**, **Requests**, **Tasks**, **Policies**, and **Comments** tabs.



4. Click the **Inventory** tab.

5. Choose the inventory item you wish to review and click the **magnifying glass** icon associated with the component you wish to review. The **Project Component Inventory Details** page appears:



6. Select the **Approve** or **Reject** options in the **Change Inventory Status...** menu in the top-right corner of the page.

Note ▪ *A system-generated comment is added to an inventory item to record when a review status change occurs.*

Scheduling Inventory for Full Review

Task ***To schedule inventory for a full review, do as follows:***

1. Log in to Code Insight as an owner.
2. Click the **My Projects** button in the **Main** menu bar.
3. Select a project for which you want to review inventory by clicking on the associated view **magnifying glass** icon in the **Actions** column. A **Project** tabbed view opens which includes the Summary, Workspaces, Inventory, Requests, Tasks, and Policies tabs.
4. Select the **Inventory** tab.
5. Select the inventory item you wish to review and click on the **magnifying glass** icon associated with this component. The **Project Component Inventory Details** page appears.
6. Select the **Schedule Full Review** option in the **Change Inventory Status...** pull-down menu located at the top-right-hand corner of the page.

Note ▪ *Scheduling an inventory item for full review kicks off the full workflow. It creates an IP review task for the project owner as well as a security review task for the security analyst if the inventory item has any associated security vulnerabilities.*

8

Reconciling Inventory

The following topics appear in this section:

What is Reconciling Inventory?

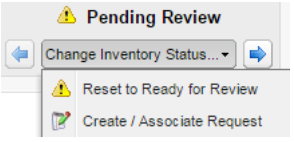
Once the project owner schedules an inventory item for a full review, the item must be associated with a component request to determine its compliance state. The following are the scenarios that may occur as part of this process:

- An approved component request is available to permit the use of the inventory item if the intended use of the item meets the predetermined conditions for use. In this case, the association to this request makes the inventory item IP-Compliant, and the project workflow advances to the next phase.
- A rejected component request is available that matched the conditions under which the inventory item will be used. In this case, an association to this request makes the inventory item IP Non-Compliant. This scenario requires either the requester to change the intended usage of the component, and then submit the request for another review of the amended component request, or a remediation. A remediation may require that the component be modified to a different version or an alternate component altogether.
- There is no existing component request for the inventory item, and a new component request is created to allow for a review of the intended usage of the component. The result of the component request cycle is an approved or rejected component request.

Associating Inventory with Component Requests

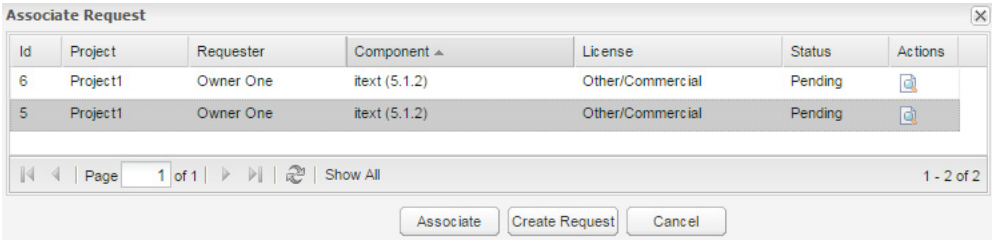
Task ***To associate a compliant inventory item to a request, do as follows:***

1. Log in to Code Insight.
2. Click the **My Projects** button.
3. Select the project for which you would like to review requests and click on the **magnifying glass** icon ( ) in the **Actions** column.
4. Select the **Inventory** tab.
5. Select an inventory item with a status *Pending Review* for which you would like to associate an existing request (option available to Owner or Participant) or create a new request (option available to Requesters only).
6. Click on the **magnifying glass** icon ( ) associated with this inventory item in the **Action** column.
7. Select the **Create/Associate Request** option from the drop down menu.



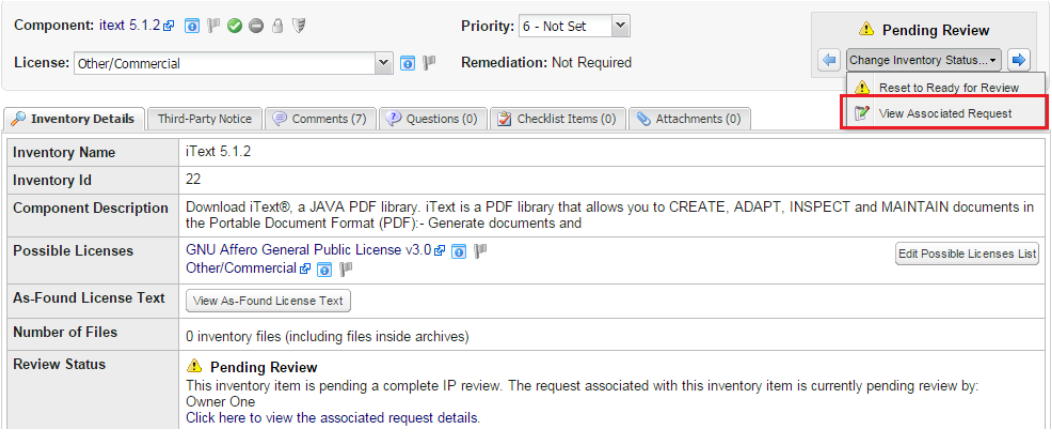
The result will be one matching request, no matching requests or multiple matching requests:

- If there is one matching request in the system, you can click **Associate** to associate the request with the inventory item. If the request is an *Approved* state, the item is automatically approved based on the association.
- If there are no matching requests, a message will be displayed indicating that there are no matches. If you have the role of Requester for the current project, you can click **Create a New Request** to create a request that matches the inventory item.
- If more than one existing request matches the inventory item, a list of possible requests appears:



8. Select the request that matches the inventory item and press the **Associate** button to make the request-inventory item association.

If an inventory item has an associated request, the **Inventory Details** tab will have the **View Associated Requests** option in the upper right-hand corner.



Associating Inventory with Requests across Projects

In some cases, there may be a need to associate an inventory item to a request located in a different project such as the Global Request Project. This type of association can be made if you have Product Catalog configured and the inventory item and request are both associated with the same Product Catalog item.

Task ***To make the associate a request to an item in another Product Catalog product, do the following:***

1. Associate the Product Catalog Item with the given project.
2. Associate the Product Catalog item with the request.
3. From the associated project, create an inventory item that matches the request component, version and license.
4. Schedule a Full Review for the inventory item and select the option to Associate the Request.
5. The request is available in the search results based on the Product Catalog association.

Unassociating a Request from an Inventory Item

Task ***To unassociate a request from an inventory item, follow these steps:***

1. Open the inventory item and select View Associated Request from the **Change Inventory Status...** menu in the upper right-hand corner. The **Request Details** page appears.
2. Press the **Unassociate** button on the bottom of the **Request Details** page to remove the association.

Creating a New Component Request

To create a new component request, see Creating a Component Request.

9

Requesting Permission to Use a Component

This section contains the following topics:

Requests

When a project is set up in the system, the number of review levels and team members responsible for reviewing component requests are established by the project owner. The policy administrator is responsible for setting up automatic policies for component/license combinations, which automatically assign the new request a status of approved, rejected, or determines that the request requires manual review.

If a requester submits a new request for a component/license, and no policy is associated with that new request, a workflow is created based on the project attributes. The system automatically sends email notifications to the users responsible for reviewing the request.

Upon submission, the request is assigned to the first-level reviewer for action. If the request is approved, it is sent to the next person in the workflow responsible for reviewing it. Once all reviewers approve the request, it moves to history. If the component has conditions for use, once the request is approved by all reviewers, the request is automatically sent back to the requester for review and acknowledgment of the conditions for use. If a request is rejected, the workflow stops immediately, and the request is completed and moved to history. For each step in the workflow, the system sends email notifications to the appropriate requesters and reviewers notifying them of the action needed or action taken.

The possible request statuses are as follows:

- Draft: This is a work-in-progress component request saved as a draft and not submitted for review.

- Pending: This is a component request submitted for review, and a workflow is consequently created.
- Approved: This status means that the component request is approved by all reviewers and all required conditions for use are reviewed and acknowledged by the requester.
- Rejected: This means that at least one reviewer rejected the component request.
- Code Insight provides the ability to customize the appearance of the **Request Dashboard**. For more information, see Customizing the Request Dashboard.

Changing the Request Form

Code Insight supports multiple request form definitions. A project can only be associated with a single request form. The product is shipped with two SQL scripts containing a sample definition of a short and long component request form.

Note ▪ *There are several built-in attributes that are required for each request form definition. They are discussed in the following section. Furthermore, the last Additional Information tab is fixed and cannot be changed. An administrator may change the Component Request form input fields (attributes) and/or add additional tabs.*

Via a database SQL script, the following changes can be made to the request form:

- Add or remove tabs.
- Add or remove attributes.
- Modify the fields of the attributes, including the text, help text, activity status (required, optional, or conditional), sequence and input type (for example, changing from text box to a list box with pre-defined values).

The Component Request form input fields provide you with the following:

- Information a reviewer needs to make an informed decision as to whether you can allow a requester to use third-party components in the development process
- Those users with requester role permissions can create a new component request and save it as a draft request: This way, requesters have the option of editing and completing the request at a later date.

Required Request Form Input Fields (Attributes)

There are several mandatory input fields for each request form definition. These fields are required in order to have a valid request form definition. The fields can be located on any tab in the request form. Each of these fields require an input type = W (INPUT_TYPE_ column in the PAS_REQ_DEF_ATTR table) which stands for built-in widget. The field names (NAME_ column in the PAS_REQ_DEF_ATTR table) must be equal to those shown in the following table.

Table 9-1 ▪ Input Fields

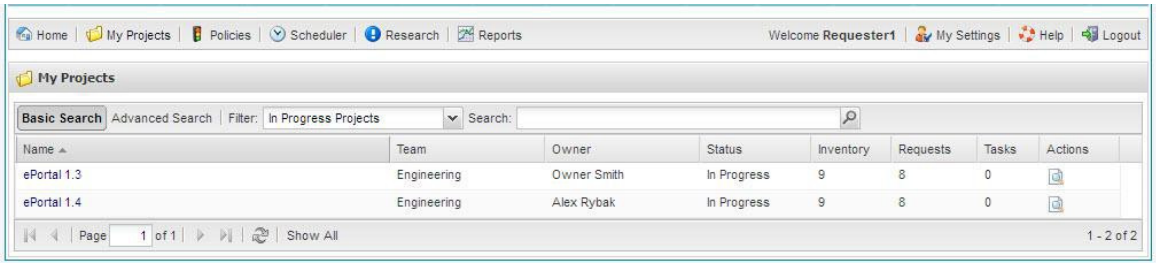
Field Name	Description
_palamida_projectName	Pull-down menu to select the project name, the value will be pre-selected if request is initiated from within a project
_palamida_componentName	Search widget to select the appropriate component for which the request is being made
_palamida_componentVersion	A constrained pull-down menu with version names associated with the selected component
_palamida_licenseName	A constrained pull-down menu with license names associated with the selected component
_palamida_reviewDeadline	A calendar widget to select the review deadline for the request: The current date is selected as the default value

Creating a Component Request

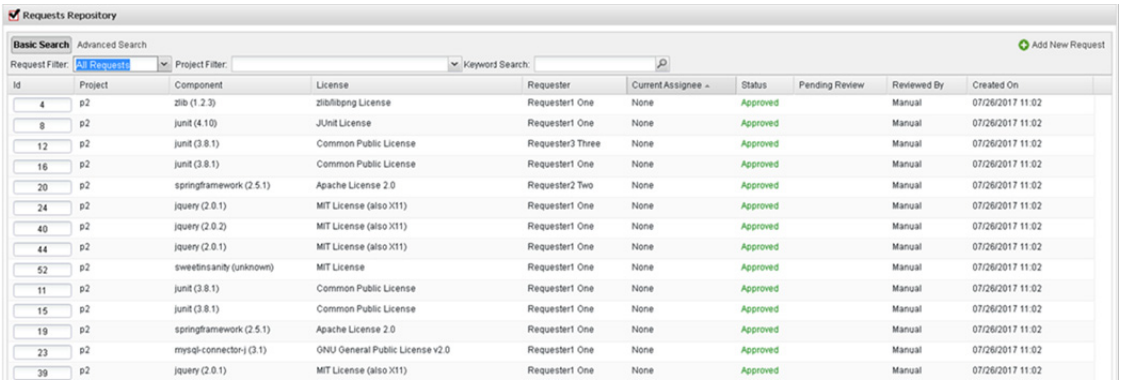
Task ***To create a component request, do as follows:***

1. Log in to Code Insight as a *Requester*:

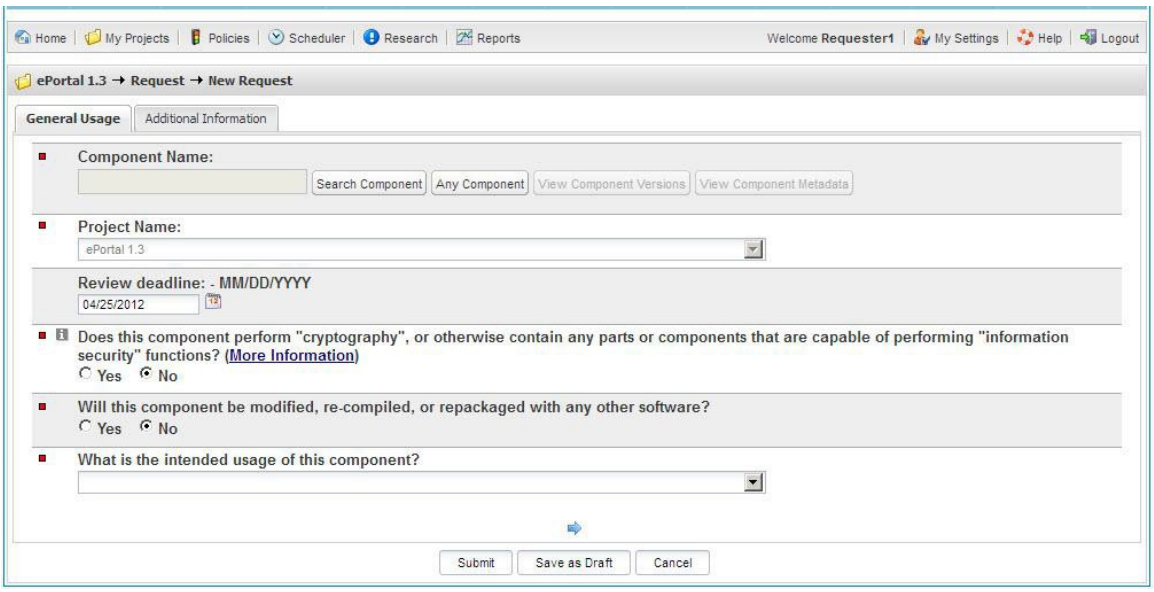
2. Click **My Projects**, and then click the **magnifying glass** icon in the **Actions** column:



3. Select the **Requests** tab. The **Request Dashboard** appears.



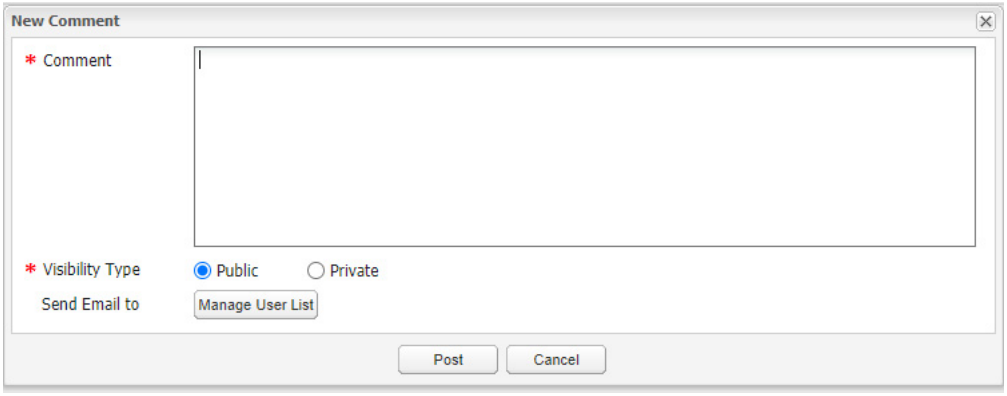
4. Click **Add New Request**. The **New Request** page appears.



5. Enter the appropriate information in the fields.
6. Select the **Additional Information** tab.
7. To enter any relevant comments in the **Comment** section, click **Post Comment**.
   a. Enter the comment text in the **Comment** field.
   b. Select whether it is **Public** (viewed by anyone on the **Additional Information** tab) or **Private** (viewed by selected users only).
   c. If the comment is **Public**, click the **Manage User List** to specify the users or the user lists to which to send an email notification about the comment. (The specified users are in addition to the requester, reviewers, and other request user roles that automatically receive the email.)

      Or

      If the comment is **Private**, use the **Manage User Visibility List** to specify the users who can see the comment and who will receive an email notification about its posting.
   d. Click **Post**.

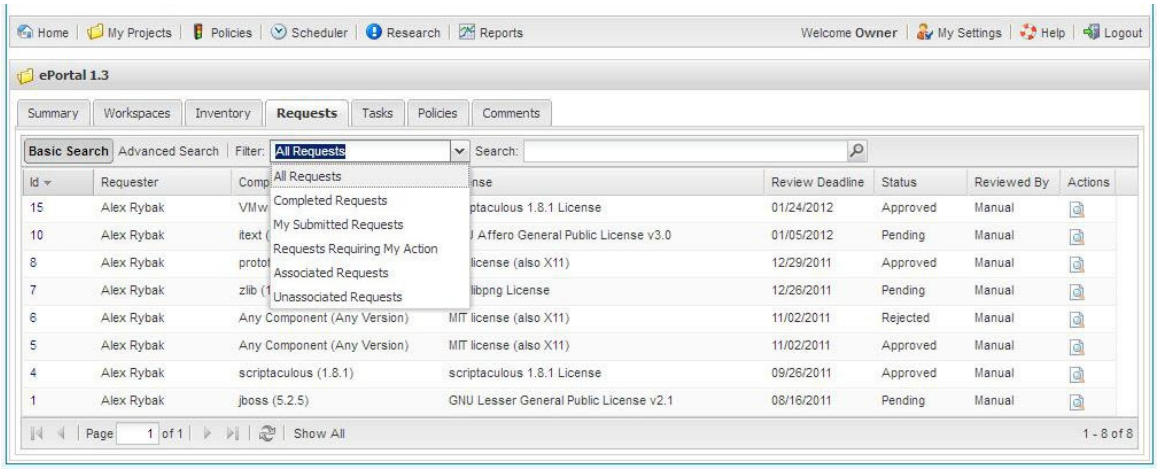


8. From the **Additional Information** tab, attach a file to the request if you want.
9. Submit the new request, or click **Save as Draft** to do additional work on your new request before submitting it for review.

Viewing Requests

Task ***To review requests related to your role, do as follows:***

1. Log into Code Insight.
2. Click **My Projects**, and then click the **magnifying glass** icon in the **Actions** column.
3. Select the **Requests** tab.
4. If you log in as an **Owner**, you can select **All Request**, **Completed Requests**, **My Submitted Requests**, **Requests Requiring My Action**, **Associated Requests**, **Unassociated Requests**.

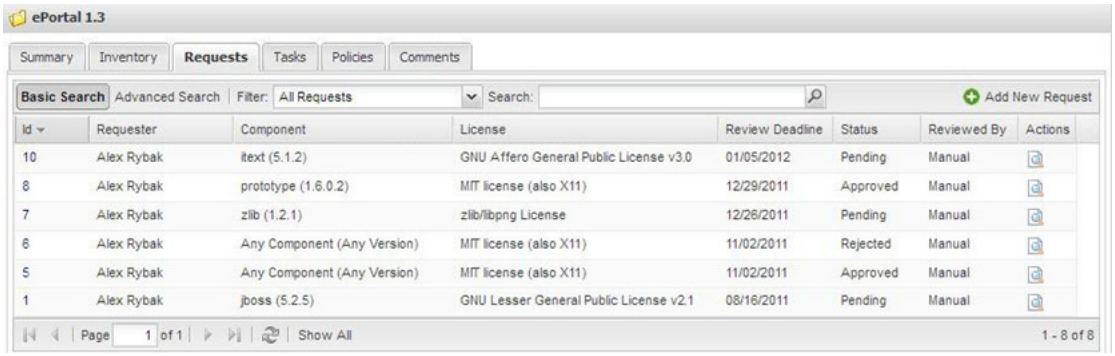


Note ▪ *If you are logged in as a Requester, you see another filter at the bottom of the* ***My Draft Requests*** *list.*

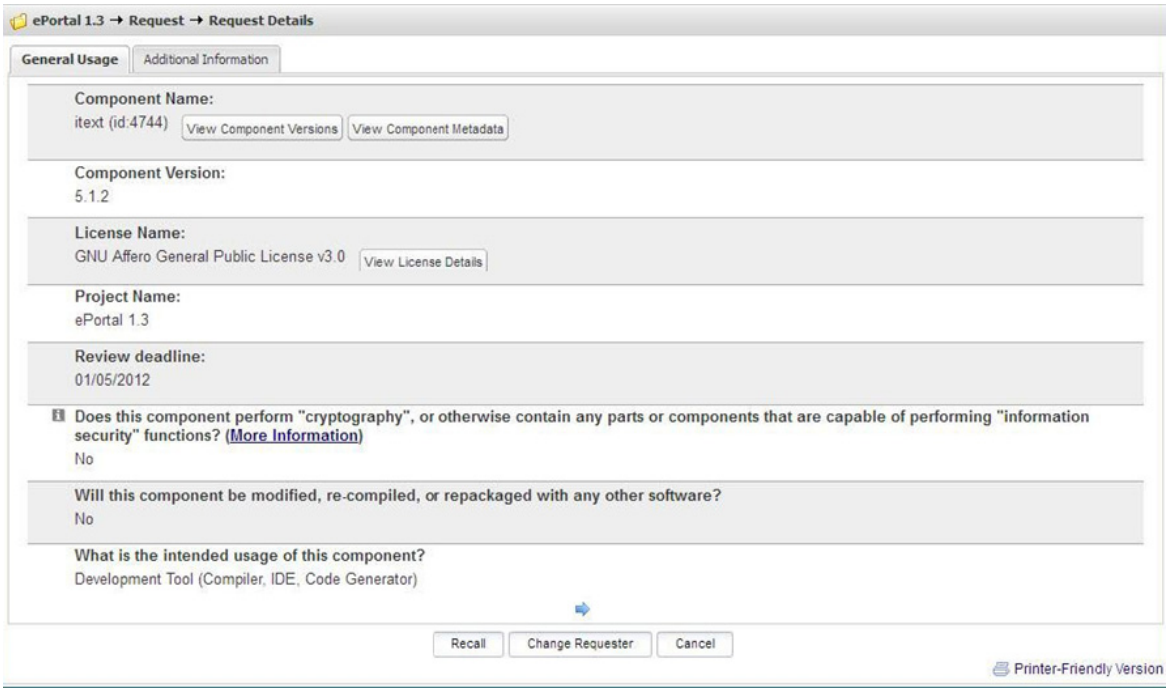
Viewing Request Details

Task ***To view request details, do the following:***

1. Log into Code Insight as a *Requester*. The **Requester Home** page appears.
2. Click **My Projects**. The **My Projects** page appears.
3. Choose a project, and click the associated **magnifying glass** icon in the **Actions** column.
4. Select the **Request** tab.



5. Select the **magnifying glass** icon next to the request you want to view. The **General Usage** tab appears.



Cloning a Request

Cloning a request saves the requester the time of filling in an entirely new request form for what might amount to making a small change for the request of a very similar component. If you would like to clone a request, do as follows.

Task ***To clone a request, do the following.***

1. Log into Code Insight as a *Requester*.
2. Click **My Projects**.
3. Choose a project, and click the associated **magnifying glass** icon in the **Actions** column.
4. Select the **Requests** tab.
5. Click the **magnifying glass** icon associated with an approved request you wish to clone. The **Requests Details** page appears.

   Note ▪ *The **Clone** button is only active on requests that have completed review.*

6. At the bottom of the **General Information** tab, click **Clone**. The system clones the request, and you will receive a notification at the top of the **Request** page that states the request, identified by a number, has been cloned.

   Note ▪ *If the request form definition has changed since the original request was created, the cloned request will map as many input fields and their values as possible. A system-generated comment will be added to the cloned request with a list of input fields that the system is unable to map.*

10

Customizing the Request Dashboard

This chapter provides the information to customize the **Request Dashboard** by adding an **Assigned Date** column.

This column shows the date when the request was last assigned to the current assignee. This information makes it easier to identify requests that are delayed in the review process and need to be prioritized. To add the **Assigned Date** column, follow these instructions.

Task ***To add the Assigned Date column, do the following:***

1. Add the following text to the requests.web.json file (located in `<FNCI_ROOT_DIRECTORY>/config/core/requests.web.json`). You may change the name of the column heading by modifying the “header” value and change the column order by modifying the “positionIndex” value.

```
{
    "dataIndex": "assignedDate",
    "derived": true,
    "positionIndex": -2,
    "header": "Assigned On",
    "width": 150
}
```

Note ▪ *The -2 refers to the 2nd position from the right.*

2. Launch a new browser session and navigate to the **Requests** dashboard, which has the Assigned Date column.

Note ▪ *Although this modification does not require a server restart, you may want to clear your browser cache if the column does not appear*

11

Reviewing Pending Requests

This section explains how to review pending requests and resubmit requests for review:

- Reviewing Pending Requests
- Resubmitting Requests for Review

Reviewing Pending Requests

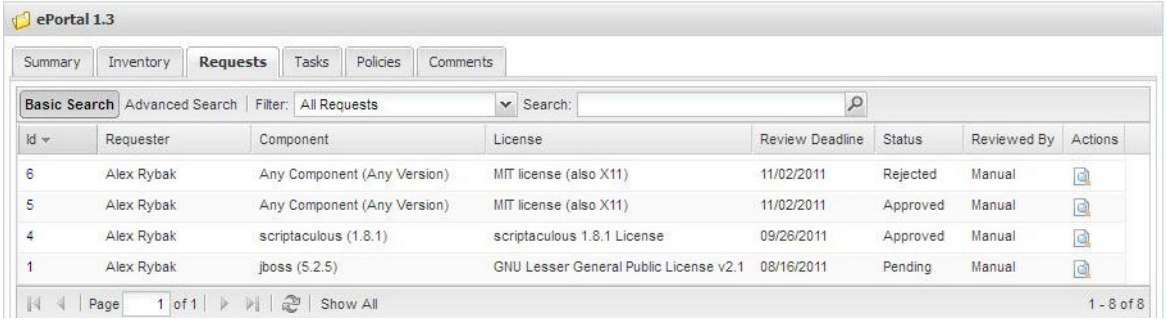
Reviewing requests is a part of a project request workflow. Any user in the system with the reviewer permissions is eligible for selection as a reviewer in a review level.

- Any reviewer in the review level may review a pending component request, thereby locking out the other peer reviewers.
- Once a request is reviewed, the request advances to the next review level as defined in the project workflow.

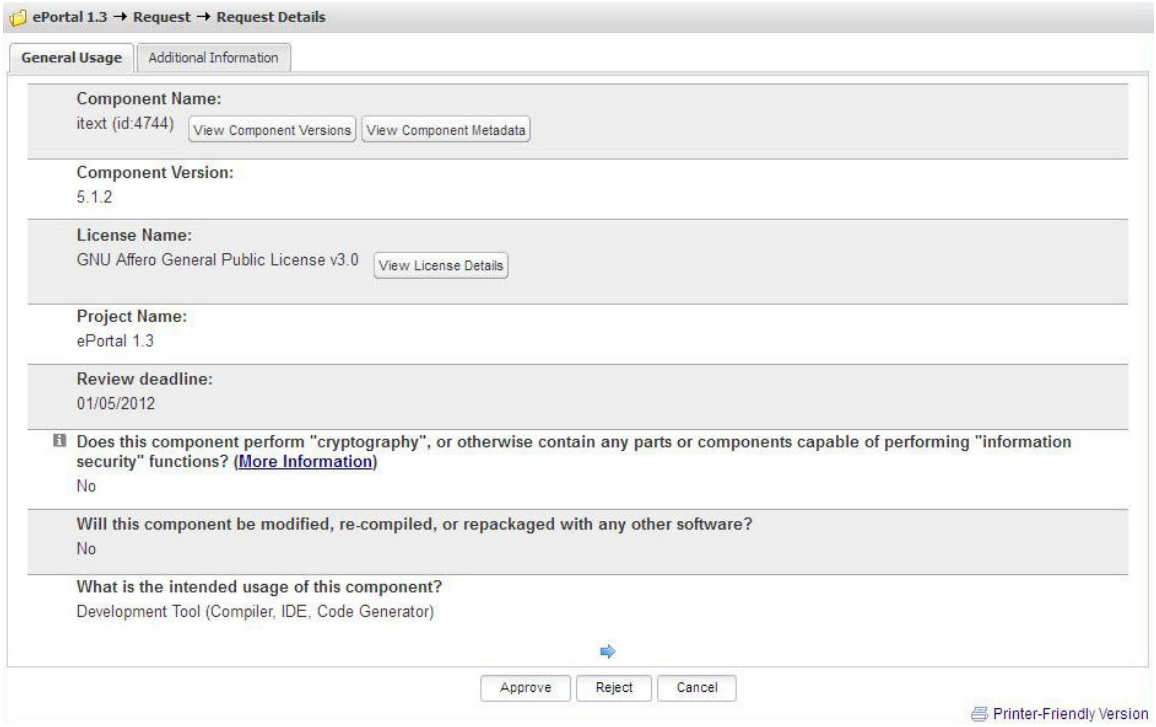
Reviewing Pending Requests from the My Tasks Tab

Task ***To review pending requests from the My Tasks Tab, do the following:***

1. Log into Code Insight as a **Reviewer**. Ensure the application administrator has set up your user account with Reviewer permissions.
2. Click the **Project** button on the **Main** menu to see a list of your projects.
3. Click the **magnifying glass** icon associated with the project for which you want to review requests. The **Project** page appears.
4. Click the **Request** tab. The **Review Task** page appears.



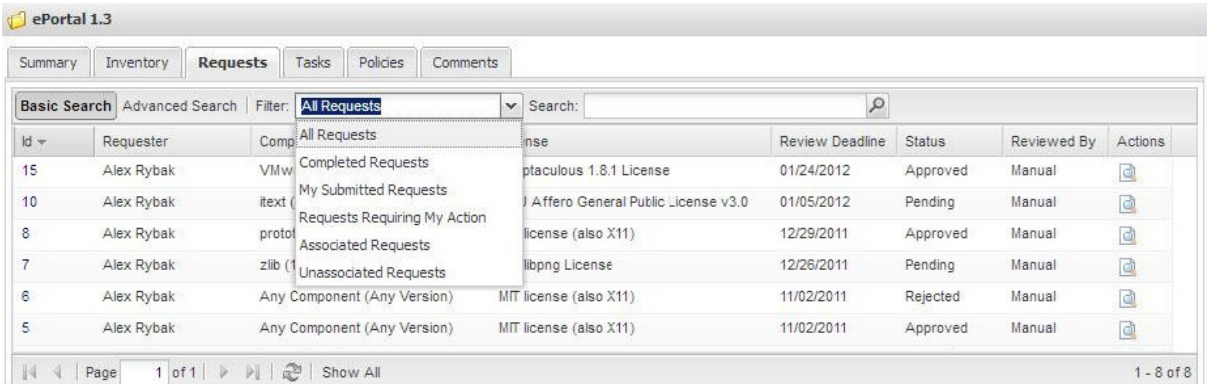
5. To review a request assigned to you, click the associated **magnifying glass** icon. A **Review** page appears on which you can view the associated request.



Reviewing Pending Requests from the Requests Tab

Task ***To review pending requests from the Requests tab, do the following:***

1. Log into Code Insight as a *Reviewer*.
2. Click the **My Project** button.
3. To select a project to review pending requests, click the **magnifying glass** icon. The **Project** page appears.
4. Click the **Requests** tab. A list of requests appears.



5. To sort this list, use the pull-down **Filter** menu. You can choose to view All Request, Completed Requests, My Submitted Requests, Requests Requiring My Action, Associated Requests, and Unassociated Requests. If you have Reviewer permissions, you will also see a Request filter for My Drafts.

6. To search for a component related to a request, enter a string in the **Search** field.

Resubmitting Requests for Review

Code Insight provides the ability to resubmit the request for review after editing it or adding policy without involving the requester, so that it can be reviewed as quickly as possible in the most automated fashion.

The ability to resubmit requests is useful in the following situations:

- **After the reviewer has edited and cleaned up the request**: reviewers often make edits to the request and resubmit it for review. Some businesses have a process in place that prevents or discourages the requester from selecting a license for a requested component due to trust issues and potential for user error. The requesters must make a guess at selecting a license or select “I don't know” as the license value. During triage review, a dedicated reviewer performs the necessary research and adds the proper license. At this point, the reviewer can recall the request and resubmit it for review.

- **When the request requires a policy review by the legal department**: if the legal department adds a new policy to the system and wants to check one or more pending requests against this policy, a member of the legal department can submit the requests for review without involving the original requester.

- **When the request requires a manual review**: if the legal reviewer wants to return the request to triage review for additional information, the reviewer can resubmit the request so it can go to triage without involving the original requester.

Task ***To resubmit a request, do the following:***

1. Log into Code Insight.

2. Click **My Project**.

3. Click the **magnifying glass** icon. The **Project** page appears.

4. Click the **Requests** tab. A list of requests appears.

5. Select the request you want to resubmit and click **Resubmit**. The selected request will be recalled and submitted for review.

12

Copying & Branching Projects

This section contains the following topics:

Introduction to Project Copy Feature

This feature allows project owners to copy a project as follows:

- Project Configuration Only
- Custom Project Copy: This option enables the Custom Project Copy tab, which offers many options for customizing your project copy operation.
- Complete Project Copy

You can use the complete project copy operation or use a project as a template to create a new project by using the custom project or project configuration copy operation.

The two main purposes of the project copy feature are to support the following scenarios:

- Reduce the administrative effort for multiple projects with similar options, users, and workflow configuration.
- Facilitate a branching scenario to mirror what is happening to the scanned codebase.

Copying a Project to Reduce Administration Effort

You can copy an existing project's settings to create a new project and in turn significantly reduce the administrative effort of re-defining all of the project settings including user provisioning and workflow definitions. For example:

- Project ePortal represents your Code Insight Portal product.

- Code Insight Manager, represented by project eManager, is developed by the same engineering team and has the same users and workflow definition as the Code Insight Portal product.
- Since the development team is large with hundreds of users, it's practical to copy the project settings from ePortal to eManager. Once the copy is completed, any necessary changes can be made to the new project. This process is much simpler than configuring the new project from scratch. It can be accomplished by copying project ePortal to a new target project eManager using the Type option "Project Configuration Only".

Copying a Project to Support Branching a Codebase

If a codebase is branched, the audit work needs to also be branched so that independent auditing results can be available for both codebases. Therefore, project data must be copied for the purposes of branching an audit.

The basic scenario is as follows:

- Project ePortal represents your Code Insight Portal product.
- The current release of ePortal is 1.2. This codebase has undergone some level of auditing and contains inventory, requests, and other data to represent the audit work and results.
- Development needs to begin on version 1.3 of ePortal.
- The codebase is branched, and now there is a codebase for the ePortal 1.2 and ePortal 1.3 product.
- Project "ePortal 1.2" needs to be copied to project "ePortal 1.3".
- The project owner for ePortal 1.2 can make a copy of the project, call it "ePortal 1.3," and map the codebase from the source project to the target project. As long as the corresponding file exists in the target project, all associated scan results and audit data are be copied to the target project.
- Along with the scan results and audit data, all existing inventory items, requests, and other source project data are also copied to the target project.
- Once the copy operation completes, the two projects are considered to be independent and any changes made to either project does not impact the other project.

Copy Type: Project Configuration Only

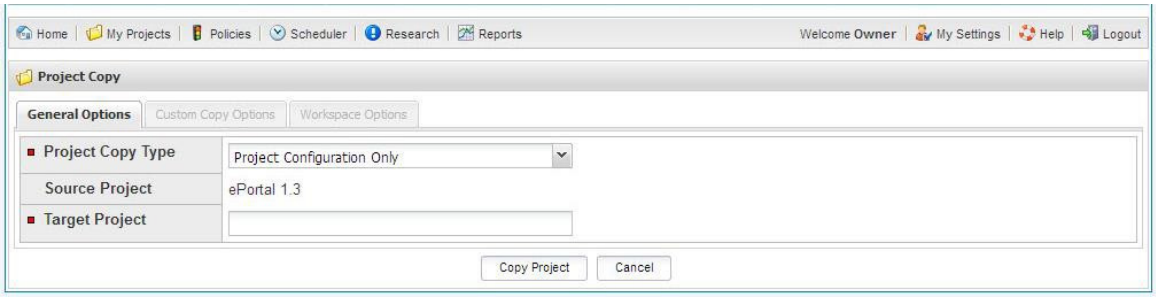
Project owners are the only ones with permission to copy projects. The following are the copy options available on the pull-down menu:

- Project Configuration Only
- Custom Project Copy
- Complete Project Copy

Task ***To copy a project configuration only, do the following:***

1. Log into Code Insight as an **Owner**.
2. Click the **Projects** button on the **Main** menu. The **My Projects** page appears.

3. Select a project from the list and click the **Project Copy** icon ( ) associated with that project. The **Project Copy** page appears.



The default copy command in the **Project Copy Type** pull-down menu is **Project Configuration only**.

4. Enter a name for the new project in the **Target Project** field.
5. Click the **Copy Project** button.

Data Copied During Project Configuration Copy Operation

This section explains what data are copied during a project configuration copy.

Project Settings: Workflow

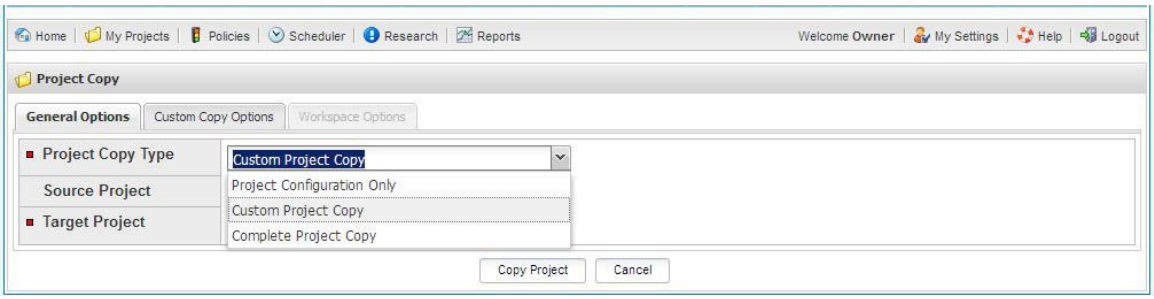
- Name is provided by the source project owner during the invocation of the project copy.
- Description is the same in the source project; the project owner can update this after the copy operation completes.
- Team is provided by the source project owner during the invocation of the project copy.
- Owner is the same as the source project; the project owner can update this after the copy operation completes.
- Project Summary Email Frequency is the same as the source project; the project owner can update this after the copy operation completes.
- Request Review Reminder Email Frequency is the same as the source project; the project owner can update this after the copy operation completes.
- Enable Inventory Quick Review is the same as the source project; the project owner can update this after the copy operation completes.
- Auto-Publish System-Detected Inventory is the same as the source project; the project owner can update this after the copy operation completes.
- Apply Policies to Inventory is the same as the source project; the project owner can update this after the copy operation completes.
- Request Form is the same as the source project; the project owner can update this after the copy operation completes.

- Review Levels are the same as the source project; the project owner can update these after the copy operation completes.
- Users—Observers are the same as the source project; the project owner can update these after the copy operation completes.
- Users—Auditors are the same as the source project; the project owner can update these after copy operation completes.
- Users—Security Analysts are the same as the source project; the project owner can update this after the copy operation completes.
- Users—Requesters are the same as the source project; the project owner can update this after the copy operation completes.
- Users—Reviewers are the same as the source project; the project owner can update these after the copy operation completes.

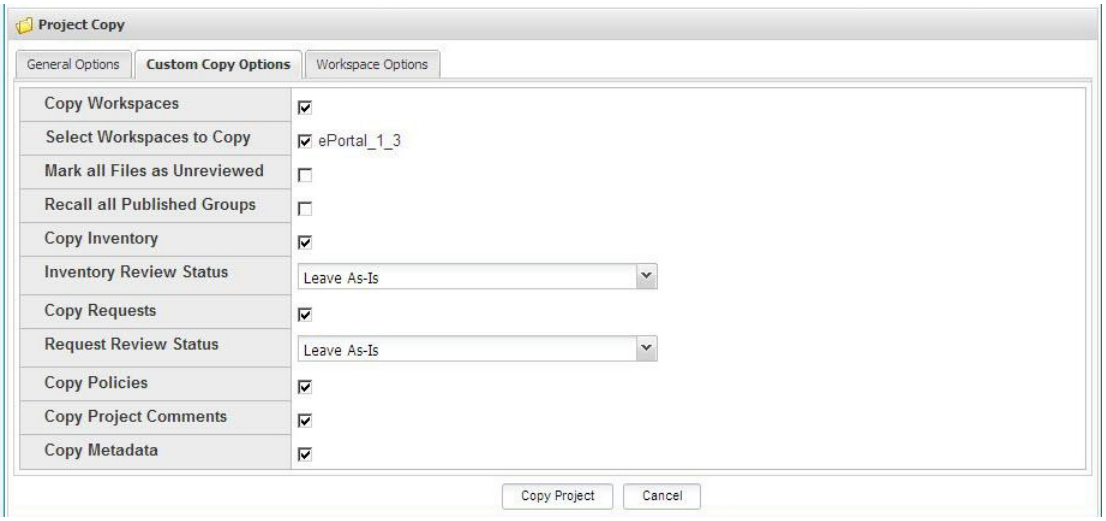
Copy Type: Custom Project Copy

Project owners are the only ones with permission to copy projects.

1. Log into Code Insight as an **Owner**.
2. Click the **My Project** button on the **Main** menu.
3. Select a project from the list and click the **Project Copy** icon ( ) associated with that project. The **Project Copy** page appears.



4. Select **Custom Project Copy** from the pull-down menu. This enables the Custom Copy Options tab.
5. Select the **Custom Copy Options** tab.



On the **Custom Copy Options** tab the following checkboxes are checked by default:

- Copy Workspaces
- Select Workspaces to Copy
- Copy Inventory
- Copy Requests
- Copy Policies
- Copy Project Comments
- Copy Metadata

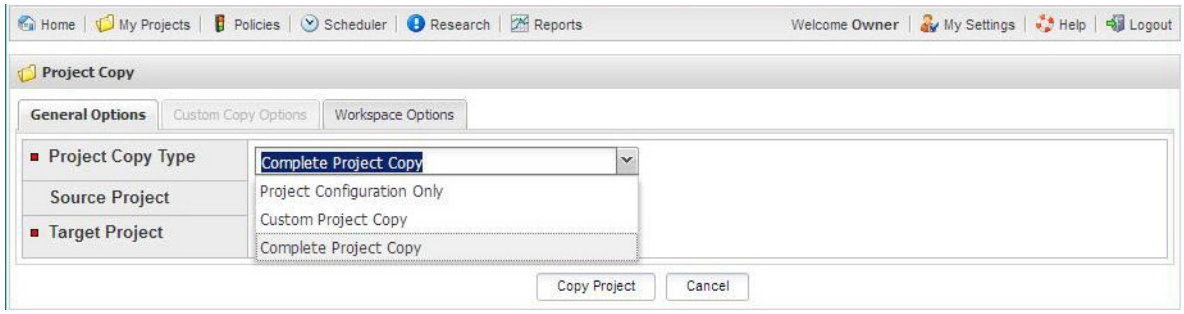
The **Inventory Review Status** and **Request Review Status** pull-down menus are set to Leave-As-Is by default. You can change the defaults to any combination of Reset to Ready for Review and Reset to Draft.

6. Click the **Copy Project** button.

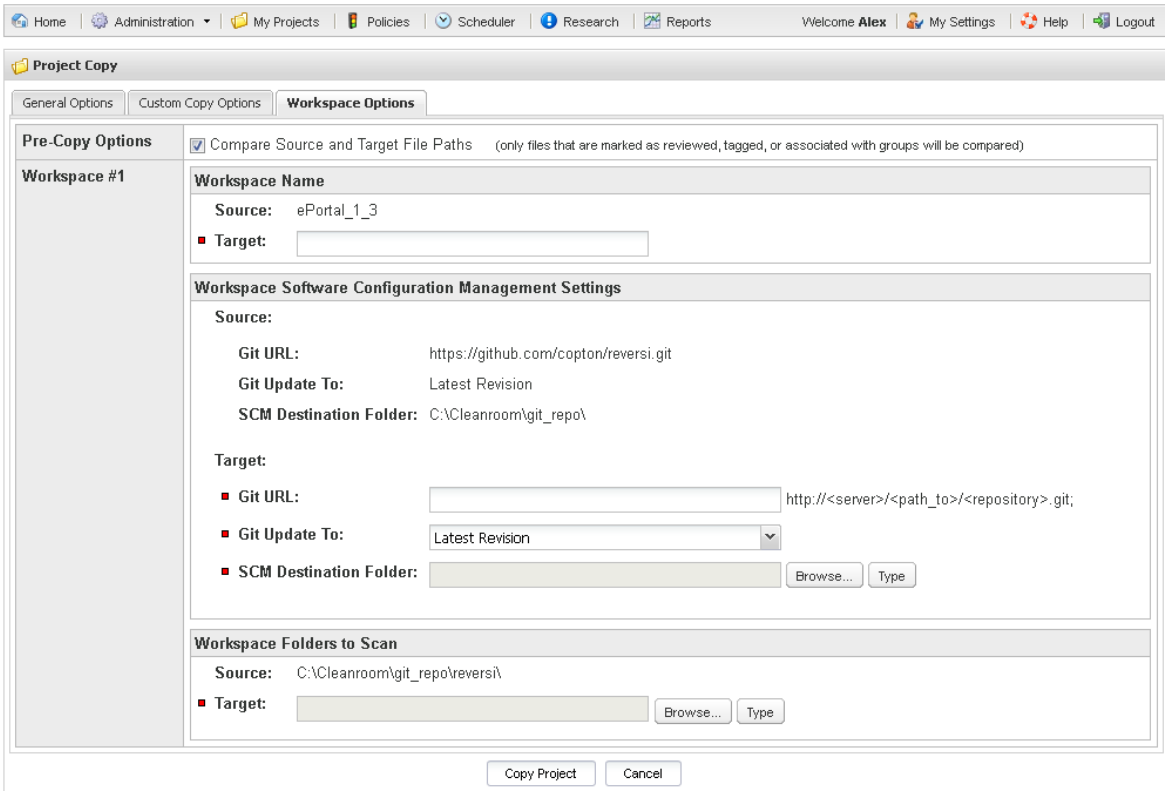
Copy Type: Complete Project

Project owners are the only ones with permission to copy projects.

1. Log into Code Insight as an **Owner**.
2. Click the **My Projects** button on the **Main** menu.
3. Select a project from the list and click on the **Project Copy** icon ( ) associated with that project. The **Project Copy** page appears.



4. Select **Complete Project Copy** from the **Project Copy Type** pull-down menu. The **Workspace Options** tab appears.



5. In the **Workspace Name** section, enter a name in the **Target** field.

6. In the **Workspace Folders to Scan** section, click on the **Browse** or **Type** button:

   - The **Browse** button opens a file system tree for the Scan Server on which you are working.
   - The **Type** button opens a dialog where you can enter the name of the input scan folder you wish to use.

7. On the **Workspace Options** tab, the Compare Source and Target File Paths checkbox is checked by default. Only files marked as reviewed, tagged, or associated with groups will be compared. Files that have not had any of these operations performed on them are not compared by the pre-copy analysis operation.This pre-copy codebase comparison lets you:

   - Know which files from the source codebase are not present in the target codebase: This means that files were removed after the codebase branching operation was performed.

- Know which target files are different from the source with the same relative path: This means that the target files were modified after the codebase branching operation was performed.
- Address file issues before the actual project copy, if appropriate.

Data Copied During the Copy Operation

This section explains which data is copied during the complete project copy.

Project Settings: Workflow

- Name is provided by the source project owner during the invocation of the project copy.
- Description is the same as the source project; the project owner can update this after the copy operation completes.
- Team is provided by the source project owner during the invocation of the project copy.
- Owner is the same as the source project; the project owner can update this after the copy operation completes.
- Project Summary Email Frequency is the same as the source project; the project owner can update this after the copy operation completes.
- Request Review Reminder Email Frequency is the same as the source project; the project owner can update this after the copy operation completes.
- Enable Inventory Quick Review is the same as the source project; the project owner can update this after the copy operation completes.
- Auto-Publish System-Detected Inventory is the same as the source project; the project owner can update this after the copy operation completes.
- Apply Policies to Inventory is the same as the source project; the project owner can update this after the copy operation completes.
- Request Form is the same as the source project; the project owner can update this after the copy operation completes.
- Review Levels are the same as the source project; the project owner can update these after the copy operation completes.
- Users—Observers are the same as the source project; the project owner can update these after the copy operation completes.
- Users—Auditors are the same as the source project; the project owner can update these after the copy operation completes.
- Users—Security Analysts are the same as the source project; the project owner can update these after the copy operation completes.
- Users—Requesters are the same as the source project; the project owner can update these after the copy operation completes.
- Users—Reviewers are the same as the source project; the project owner can update these after the copy operation completes.

Workspace Settings: General

- Name is provided by the source project owner during the invocation of the project copy.
- Description is the same as in source workspace; the project owner or auditor can update this after the copy operation completes.
- Folders to Scan are the same as the source workspace, and they are updated with the new Target Scan Paths that are provided by the source project owner during the invocation of the project copy.
- Excluded File Patterns are the same as the source workspace; the project owner or auditor can update these after the copy operation completes.
- Enable Incremental Scanning is the same as the source workspace; the project owner or auditor can update this after the copy operation completes.

Workspace Settings: Detection

All options are the same as the source workspace; the project owner or auditor can update these after the copy operation completes.

Workspace Settings: Source Code Options

All of the CodeRank™ attributes sliders are the same as the source workspace; the project owner or auditor can update these after the copy operation completes.

Workspace Settings: Automated Analysis

All options are the same as the source workspace; the project owner or auditor can update these after the copy operation completes.

Workspace Settings: Software Configuration Management (SCM)

The SCM settings from the source workspace are shown and can be updated and applied to the target workspace.

Workspace Reports

The workspace reports are not copied as part of the project copy operation. These reports can be generated after the target workspace scan completes, so the report data represents the target codebase and scan results.

File Tags

- Tag data is copied, and file paths are updated to reflect the target codebase.
- If a file with the same relative path existed in the source codebase, but is not present in the target codebase, its tag assignments are not copied over.

- A new tag “File Changed Since Workspace Copy” is applied to all files which have the same relative file path in the target codebase, but whose MD5 is different from the source codebase. This allows the project owner or auditor to review the file and determine whether additional auditing work is required due to the changes in that file. Furthermore, if the source file was marked as reviewed, then that tag assignment is removed (since this file has changed since the workspace copy).

Groups

- Group data is copied and file paths are updated to reflect the target codebase.
- Groups with zero files are recalled, but the group is not deleted from the project. Users can either add files to the empty group and re-publish (all existing inventory item data are retained from the source inventory item), or delete the group.

Inventory

- Inventory items from the source project are copied to the target project with all data preserved as-is, except for the associated project ID.
- The review status from the source inventory item is maintained in the target inventory item in the target project.
- All comments associated with the source inventory item are copied to the target inventory item in the target project with the original posted on and posted by values.
- All questions and associated answers associated with the source inventory item are copied to the target inventory item in the target project with the original posted on, posted by, answered on, and answered by values.
- All checklist items associated with the source inventory item are copied to the target inventory item in the target project with the original posted on and posted by values, as well as completed on and completed by, if the checklist items were marked as completed.
- All attachments associated with the source inventory item are copied to the target inventory item in the target project with the original uploaded on and uploaded by values.

Requests

- All of the requests from the source project are copied as-is to the target project with the exception of updating the project ID to the target project value. All of the original dates (created on, submitted on, reviewed on, etc.) from the source request are retained in the target request.
- Draft Requests are copied to the target project as an un-submitted draft request.
- Pending Requests are copied to the target project as a pending request. Any approvals that are already made as part of the completed portion of the request review workflow are preserved in the target project. All reviewers that have yet to take action on the pending request need to review it both in the source project as well as the target project.
- Completed requests are copied to the target project as completed request and retain the same IP review status and security review status as that of the source request.

Tasks

All of the tasks from the source project are copied as-is to the target project with the exception of updating the project ID to the target project value. All of the original dates (for example, created on, completed on) from the source task are retained in the target task.

Policies

Any policies with the source project explicitly selected as part of the scope via the **Project** selection widget are updated to include the target project as well. This does not apply to policies with a global or team scope, because the target project is already included as part of the team or global policy.

Comments

All project comments are copied with their original created by and created on values.

Pre-Copy Operations

To reduce the risk of failure of the project copy operation, check the following before starting the copy process:

- Whether target workspace names provided by the project owner already exist on the Scan Server.
- Whether each Scan Server is reachable.
- Whether there is enough disk space remaining on the Scan Server for the workspace copy operation.
- The Scan Server scheduler queue to ensure that there are no active or scheduled tasks for any source workspaces.
- Whether there are any workspaces which are blocked by a project owner or auditor modifying the workspace settings.
- Whether there are any workspaces that are opened by the project owner or auditor in Detector.
- Ensure that target paths across all workspaces in a project for which an auditor has used the file (applied a tag, added to a group, or marked as reviewed). To determine inconsistencies, perform the following actions:
  - Create a list of all file paths that are no longer present in new project codebase but were in the source codebase
  - Create a list of all file paths that are same relative path but different MD5 from the source codebase

Provide a file paths comparison summary before proceeding with the project copy, and allow users to cancel the copy if they choose.

Blocking Access to Source & Target Projects During Copy Operations

Access to the source project, target project, and associated workspaces needs to be blocked during the duration of the project copy operation. This includes web UI, Detector, and ScriptRunner API access.

Post-Copy Scan

All workspaces are scheduled for a full scan after the project copy completes to ensure that the scan results are accurate for the target codebase.

Email Notification

The project owner who invoked the project copy is notified by email when the copy operation has completed. At that point, the source and target project are available for use.

Target Project Cleanup Options after Copy

Once the project copy operation completes, there are several manual steps required to ensure that the target project accurately reflects the target codebase. The following list of tasks should be considered by the project owner, auditors, requesters, reviewers, security analysts, and policy administrators:

- Review and update (if necessary) the target project settings.
- Review and update (if necessary) the target workspace(s) settings.
- Software Configuration Management (SCM) options are not included in the project copy, so these properties needs to be set if the target workspaces are to be associated with a codebase managed via an SCM tool.
- If any workspace settings are changed, consider re-scanning the workspace to ensure that the scan results accurately reflect the target codebase.
- Run any necessary workspace reports so that the target codebase is properly reflected in the report contents.
- Review the scan results in Detector after the post project copy workspace scan(s) completes.
- Use the new tag “File Changed Since Workspace Copy” as a filter to review any files that have the same relative file path in the source and target codebase, but with file content that has changed. These files may have been altered by the SCM tool as part of the branching operation, or these files have actually undergone content changes via human editing. In any case, it is best for the auditor to review these files.
- Review any groups with zero (0) files. These are groups that have no files with the same relative path between the source and target codebase.
- Files may have moved around in the target codebase, and these files need to manually be added back into existing groups.
- Files may have been intentionally deleted, and the group may be deleted as well, since the inventory is no longer present in the target codebase.
- Review the list of inventory items, and ensure that they have the proper review status.
- Complete all pending reviews for items undergoing a full review. Reviewers need to review the same inventory items in both the source and target projects.
- Review any policies whose scope has been explicitly altered to include the target project. If the policies do not apply, manually adjust the policy scope.

About Concurrently Open Workspaces

Because project copying (custom and complete) requires opening workspaces, it is possible that if there are a large number of workspaces in a given project that project copy may fail. By default, the maximum allowable open workspaces on a given Scan Server is 50. Increase the `dispatcher.maxNumberOfOpenedWorkspaces` property in the `scanEngine.properties` file to allow more workspaces to be open at a given time. You will need to restart the Tomcat server for this change to take effect.

13

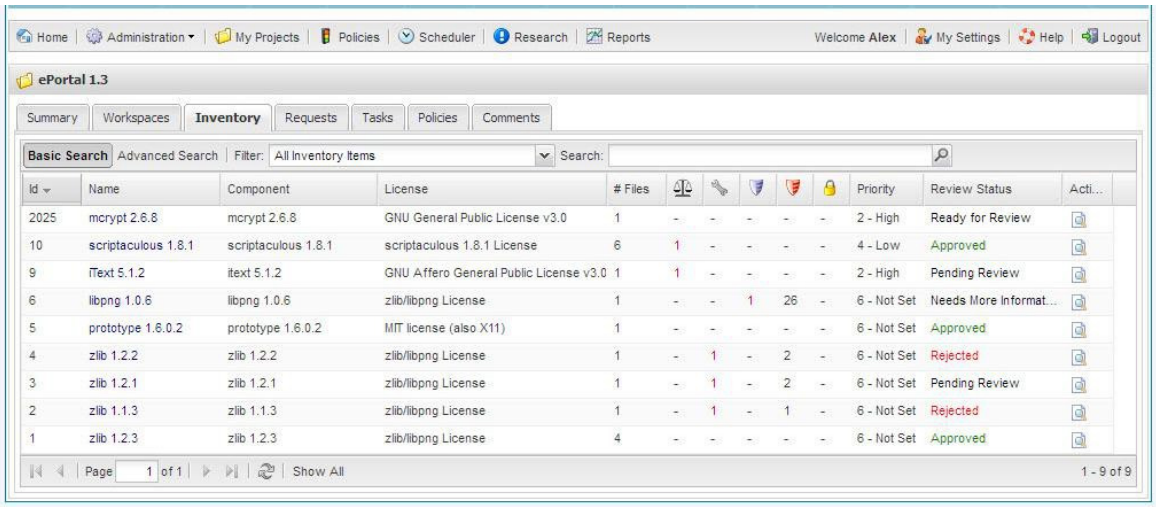
Reviewing Vulnerabilities

This chapter provides the procedure for doing a security scan by reviewing vulnerabilities.

When an inventory item is scheduled for a full review by a project owner and it has been through a review and reconciliation process to determine its IP compliance state, a secondary review by the security analyst assigned to the project must occur if the inventory item contains associated security vulnerabilities.

Task ***To review vulnerabilities, do the following:***

1. Log into Code Insight as a *Security Analyst*. A security analyst is a user with a project participant role who is assigned as a security analyst for a given project.
2. Click the **My Projects** button, and a list of projects opens.
3. Click the **magnifying glass** icon next to the project you want to review. The **Project** page opens.
4. Click the **Inventory** tab.



5. Select an inventory item, and click the associated **magnifying glass** icon. The **Inventory details** page appears.

6. In the **Components** line, click the **Vulnerability Shield** icon ( ) in the top left corner of the **Inventory Details page** to view vulnerabilities details. The **Contains the Following Security Vulnerabilities** dialog appears:

Score	Severity	Name	Description
10	HIGH	CVE-2004-0597	Multiple buffer overflows in libpng 1.2.5 and earlie...
7.8	HIGH	CVE-2010-0205	The png_decompress_chunk function in pngrutil.c in l...
7.5	HIGH	CVE-2010-1205	Buffer overflow in pngpread.c in libpng before 1.2.4...
7.5	HIGH	CVE-2002-1363	Portable Network Graphics (PNG) library libpng 1.2.5...
7.5	HIGH	CVE-2008-1382	libpng 1.0.6 through 1.0.32, 1.2.0 through 1.2.26, a...
7.5	HIGH	CVE-2006-3334	Buffer overflow in the png_decompress_chunk functior
6.8	MEDIUM	CVE-2009-0040	The PNG reference library (aka libpng) before 1.0.43...
6.8	MEDIUM	CVE-2011-2690	Buffer overflow in libpng 1.0.x before 1.0.55, 1.2.x...
5	MEDIUM	CVE-2007-2445	The png_handle_tRNS function in pngrutil.c in libpng...
5	MEDIUM	CVE-2009-5063	Memory leak in pngwutil.c in libpng before 1.2.39bet...
5	MEDIUM	CVE-2004-0421	The Portable Network Graphics library (libpng) 1.0.1...
5	MEDIUM	CVE-2006-7244	Memory leak in pngwutil.c in libpng 1.2.13beta1, and...
5	MEDIUM	CVE-2010-2249	Memory leak in pngrutil.c in libpng before 1.2.44, a...
5	MEDIUM	CVE-2011-2691	The png_err function in pngerror.c in libpng 1.0.x b...

7. To expand the descriptions, click the **Plus** icon to the left of the **Score** column.

8. Click the **CVE** links to read information about each vulnerability in the National Vulnerability Database.

Note ▪ *The score and severity displayed for each vulnerability are based on the CVSS v2 scoring system.*

9. From the **Change Inventory Status** pull-down menu, select **Approve**, **Reject** or **Schedule Full Review** of the inventory item when you have completed reviewing the associated vulnerabilities.

10. If action is required based on the vulnerability review, create security checklist items for the inventory item to track this work. For more information, see Revenera Support.

14

Metadata Framework

This section contains the following topics:

Metadata

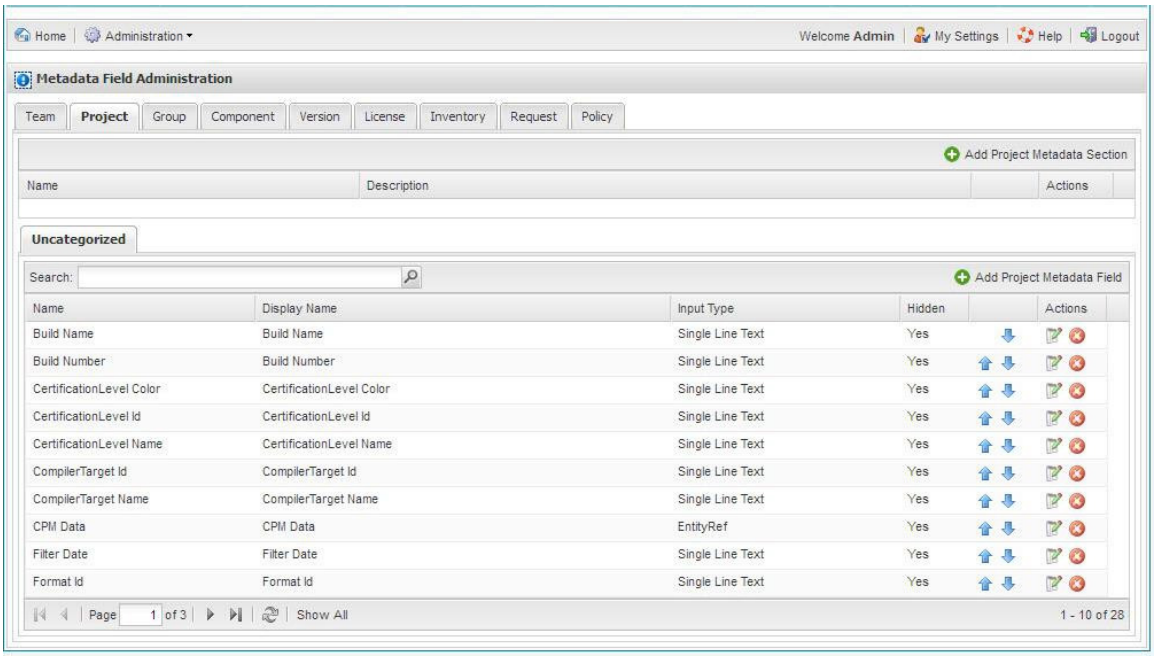
Code Insight contains a flexible metadata framework to allow support of additional information associated with all Code Insight-managed entities. This, in turn, makes the system more valuable as a repository for your code. Metadata support is provided via the Web UI as well as via public APIs using ScriptRunner.

Administering Metadata Sections and Fields

Task ***To administer metadata sections and fields for Code Insight-managed entities via the Web UI, do the following:***

1. Log into Code Insight as **Application Administrator**. Only a user with the Application Administrator permission can manage metadata sections and fields.
2. Click the **Administration** button on the Main menu, and select **Metadata** from the pull-down menu:

The **Metadata Field Administration** page opens to **Team**. In this example, click on the **Project** tab.



The following Code Insight-managed entities can support associated metadata fields organized into the metadata sections:

- Team
- Project
- Group
- Component
- Version
- License
- Inventory
- Request
- Policy

Metadata fields are organized into metadata sections for a given entity to allow for a logical grouping of related data. The Metadata Field Administration page is comprised of tabbed types of metadata and each tabbed page is split into two sections as shown in Figure 88.

The top portion of the page allows for management of metadata sections for Code Insight-managed entities. A metadata section definition consists of a name and description. Once metadata sections are defined, they can be ordered as desired by clicking on the **up** and **down** arrows in the metadata sections table.

Adding a Metadata Section

Task ***To add a metadata section, do the following:***

1. Log into Code Insight with **Application Administrator** permissions. Only a user with the Application Administrator permission is allowed to manage metadata sections and fields.
2. Click the **Administration** button on the Main menu, and select **Metadata** from the pull-down menu.

- Click on the tab for the type of Metadata to which you wish to add a section. In this example, select the Project tab.
- Click on the green Plus icon Add Metadata Section button.

3. The Add Metadata Section dialog box appears.

4. Complete the **Name** and **Description** fields.

Note ▪ *A section name is unique to an entity.*

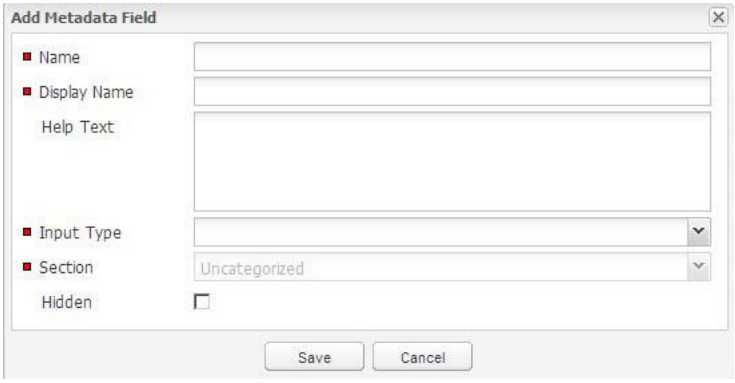
5. Click the **Save** button.

Adding a Metadata Field

The bottom portion of the **Metadata Field Administration tab** allows you to manage metadata fields for a given metadata section. A metadata field definition consists of a Name, Display Name, Help Text, Input Type, Section, and a Hidden Option. Once metadata fields are defined, they can be ordered as desired within their metadata section by clicking on the **up** and **down** arrows in the metadata fields table.

Task ***To create a metadata field, do as follows:***

1. Log in to the system with application administrator permissions. Only a user with the Application Administrator permission is allowed to manage metadata sections and fields.
2. Select **Administration** from the **Main** menu, and select Metadata from the pull-down menu.
3. Click the tab for the type of Metadata to which you wish to add a section. In this example, select the **Project** tab.
4. Click the **Plus** icon **Add Metadata Field** button. The **Add Metadata Field** dialog appears.



5. Complete the fields on the **Add Metadata Field** dialog:
   - Name: This is a unique identifier for the metadata field for a given entity. This is a required field.
   - Display Name: This is the label that shows up in the Web UI when displaying or assigning the metadata field value. This is a required field.
   - Help Text: If a value is entered here, a help icon will be shown next to the metadata field when displaying or assigning the metadata field value. This is an optional field.
   - Input Type: The input type will determine which validations need to occur when assigning a metadata value for the field as well as how to display the assigned value. The following input types are supported.
   - Date
   - Numbers
   - Integer (Whole Number)
   - Number with Decimal
   - Text Types
   - Single Line Text
   - Short Multiple Line Text
   - Long Multiple Line Text (over 4,000 characters)
   - URL
   - Yes/No
   - Section: The section to which this metadata field belongs. This value may be edited once the metadata field is defined. During the initial metadata field definition, the value is set to the metadata section for which the metadata field was created.
   - Hidden: If selected, this metadata field will not be shown in the Web UI, but is still accessible via public APIs. This is intended to be used for cases where a metadata field for a given entity needs to store data that should not be viewable or editable via the Web UI.
6. Click **Save**.

Metadata field administration is also supported via the public APIs.

Note ▪ *Any metadata field defined without a selected section will be located in the Uncategorized tab in the Web UI.*

The imported metadata fields can then be edited in the Web UI to organize them into the desired section. The process of defining metadata fields via the public APIs uses an XML file to define the metadata fields and a groovy script to import the data. Refer to the sample file (`sample-license- metadata-definitions.xml`) located in the `/scriptRunner/scripts` directory.

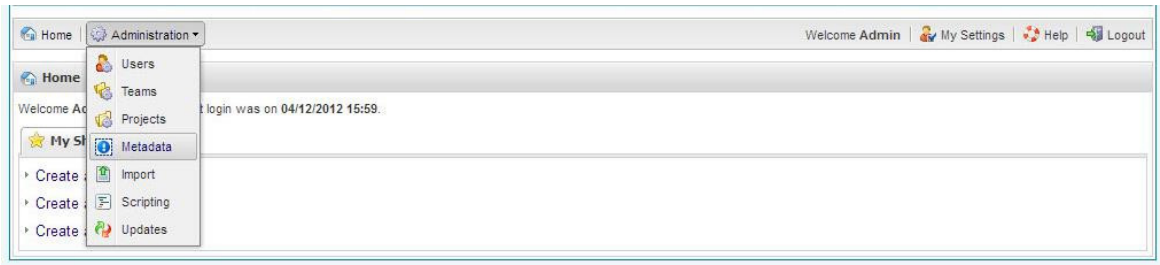
Viewing, Editing, and Assigning Metadata Values

Web UI access to metadata fields for viewing and assigning values is limited to the following entities:

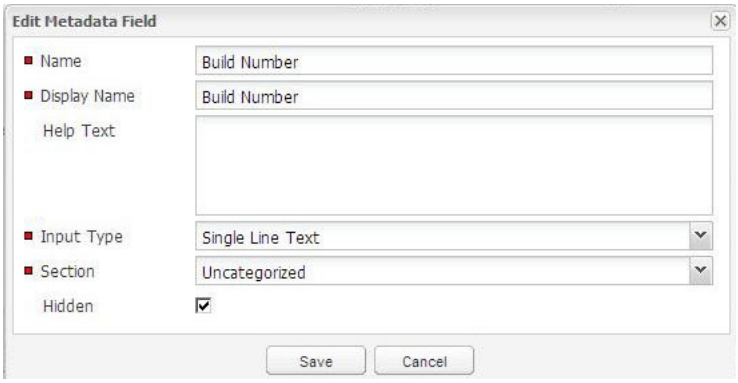
- Projects via the **Project Summary** page.
- Inventory items via the **Inventory Details** page.
- Requests via the **Request Details** page (**Additional Info** tab).
- Components via the research, inventory details, and request details pages.
- Component versions via the research, inventory details, and request details pages.
- Licenses via the research, inventory details, and request details pages.

Task ***To view, edit and assign metadata values associated with an inventory item via the Web UI, do as follows:***

1. Log into Code Insight as an **Application Administrator**. Only a user with **Application Administrator** permission is allowed to manage metadata sections and fields.
2. Click the **Administration** button on the Main menu, and select **Metadata** from the pull-down menu.

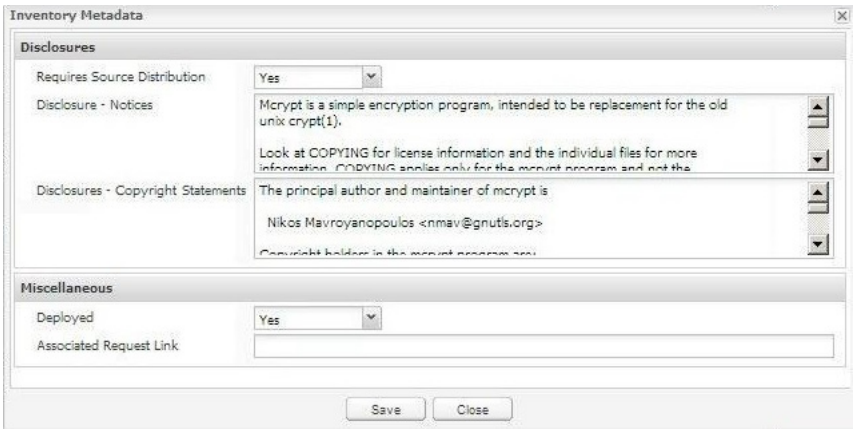


3. Click the tab for the type of Metadata to which you wish to add a section. In this example, select the **Project** tab.
4. Select an entity and click the **Pen and Paper** icon to view the metadata associated with that entity instance. The **Edit Metadata** dialog opens for that instance.



Another way to view metadata is to log into Code Insight as a user without administrator privileges. For example, log in as **Owner**.

5. Click the **My Projects** button on the **Main** menu.
6. Select the **magnifying glass** icon associated with the project inventory you wish to view.
7. Select the **Inventory** tab.
8. Select an inventory item and click one of the following icons:
   - Click the view component Metadata icon ( ) in the **Component** field at the top of the page.
   - Click the License Details icon ( ) next to the license name in the **Component** field at the top of the page or in the **Possible Licenses** section.
   - Click the **View Inventory Metadata** button (View Inventory Metadata) in the **Inventory Details page Inventory Metadata** section.
9. Click the **View Inventory Metadata** button. The Metadata information opens in View mode.
10. Click the **Edit** button at the bottom of the **Metadata** dialog, and the metadata information appears in Edit mode.



Note ▪ *Use of the **Edit** button is restricted to those users allowed to edit metadata values.*

Metadata fields are displayed in sections as defined by a user with Administrator permissions. If there is Help Text for a given **Metadata** field, a **Help Text** icon appears. You can hovered over or click on the Help Text to display it.

11. Modify the values in the **Metadata** fields, and click the **Save** button to save your changes.

Note ▪ *Access to Metadata Field values and assignment of values for Metadata fields is also supported via the public APIs.*

Using Public APIs to Access & Assign Metadata Field Values

Access to Metadata Field values and assignment of values for Metadata fields is also supported via the public APIs. The process of assigning Metadata Field values for Code Insight-managed entity instances via the public APIs uses a CSV (comma-separated) values file format to specify the Metadata Field values for an entity instance, and a groovy script to import the data.

For more information, see the sample file (`sample-license-metadata.csv`) located in the `/scriptRunner/scripts directory`.

15

Using the Product Catalog

Code Insight provides the Product Catalog so your company's offering (products and services) can be represented in a hierarchy within Code Insight. Once the Product Catalog items are in the system, they may be associated to requests and used to generate Third-Party Notices and other reports. This chapter describes discusses the Product Catalog and how to use it by describing the following use cases applicable to the Catalog:

- Associating Requests to Product Catalog Items
- Associating Inventory with Requests Across Projects
- Generating Third-Party Notices for Requests based on Product Catalog Items
- Reporting on Findings Based on Product Catalog Items
- Administering the Product Catalog
- Adding Product Catalog Items
- Viewing, Editing and Associating Product Catalog Items
- Associating Product Catalog Items with a Project
- Associating Product Catalog Items with a Request
- Associating Inventory with Requests across Projects
- Reporting on Requests (Filtered by Product Catalog)
- Reporting on Third-Party Notices for Requests (Filtered by Product Catalog)

Associating Requests to Product Catalog Items

A Code Insight user would like to file a request to use an OSS component and associate it to one or more versions of the product (i.e. ePortal 1.3, 1.4, 1.5) during or after request approval. With the product versions listed as items in the Product Catalog, the user can create the request from any project (including a non-audit Request project) and associate the request to one or more versions of the product via Product Catalog at any given any time. The user can also report on the request data and third party notices associated with the requests filtered by Product Catalog items. For more information, see Associating Product Catalog Items with a Request.

Associating Inventory with Requests Across Projects

A Code Insight reviewer would like to be able to associate an inventory item for zlib 1.2.3 to a request in another project (i.e. a Request Project) in order to determine its compliance state. This scenario can be achieved if both the project containing the inventory item and the request are associated to the same Product Catalog item.

Generating Third-Party Notices for Requests based on Product Catalog Items

A Code Insight legal user creates third-party notices for requests across multiple projects or a single global project. She would like to be able to generate a Third-Party Notices report for Requests based on her company's products and services. This can be done by associating the requests to the Product Catalog items and running the **Third Party Notices Report for Requests (Filtered by Product Catalog)**.

Reporting on Findings Based on Product Catalog Items

A Code Insight user may want to limit the scope of a report based on Product Catalog data. Code Insight currently offers the Request Report (Filtered by Product Catalog).

Administering the Product Catalog

The following attributes represent a Catalog item in Code Insight:

- Name (required)
- URL (optional)
- Description (optional)
- Parent Item (optional)

The following functionality is currently available for Product Catalog:

- Create/Delete a catalog item.
- Place catalog item in the correct position in Code Insight.
- Search for a catalog item across Projects/Requests.
- Associate a catalog item with an existing Project or existing Request.
- View associated Catalog items for a Project/Request.
- Report on all Requests associated with a Catalog item.
- Report on all Third-Party notices associated with a Catalog item.

Code Insight can be configured to display Product Catalog attributes. To modify how and where these attributes are displayed, modify the `product.catalog.web.json` file according to the steps below.

Projects

Task ***To enable the Product Catalog feature for projects, do the following:***

1. Open the `.json` file associated with your installation of Code Insight:

   ```
   <Code Insight_ROOT>/config/core/product.catalog.web.json
   ```

2. Set the following property to “true” to enable the Product Catalog attribute on the **Project Details** page:

   ```
   "project.details.product.catalog": { "enabled": true
   ```

   As an alternative, set the property to “false” to disable the Product Catalog attribute from the **Project Details** page.

3. By default the Product Catalog attribute will appear on the **Project Details** page with a label “Product Catalog”. To change the label text, modify the `entry.label.text` property:

   ```
   "applyTo": ["project.details.product.catalog"],
   "configuration": {
   "entry.label.text": "Product Catalog"
   ```

4. Be sure to restart the server after making modifications to the `.json` file.

Requests

Task ***To enable the Product Catalog feature for Requests, do the following:***

1. Open the `.json` file associated with your installation of Code Insight:

   ```
   <Code Insight_ROOT>/config/core/product.catalog.web.json
   ```

2. Set the following property to “true” to enable the Product Catalog attribute on the **Request** page:

   ```
   "request.details.product.catalog": { "enabled": true
   ```

3. Alternatively, set the above property to “false” to disable the Product Catalog attribute from the **Request Details** page.

4. By default the Product Catalog attribute will appear on the **Request Details** page with a label “Product Catalog”. To change the label text, modify the entry.label.text property:

   ```
   "applyTo": ["request.details.product.catalog"], "configuration": {
   "entry.required": false, "entry.label.text": "Product Catalog"
   ```

5. The following property determines where on the **Request Details** page the Product Catalog will appear:

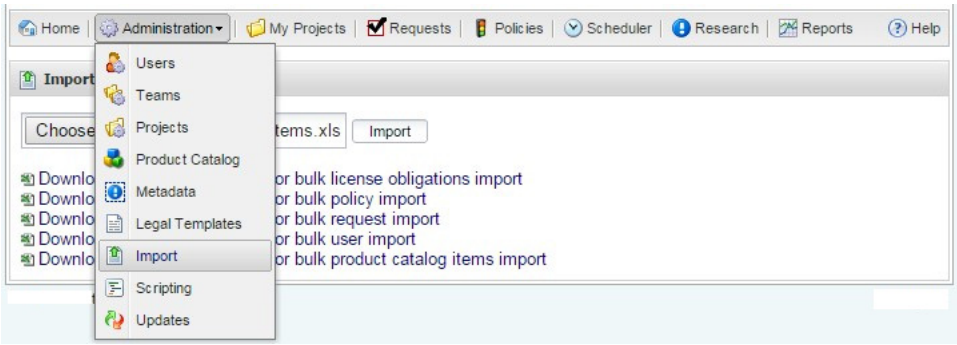
   ```
   "insert.after": "_palamida_projectName"
   ```

6. Be sure to restart the server after making modifications to the `.json` file.

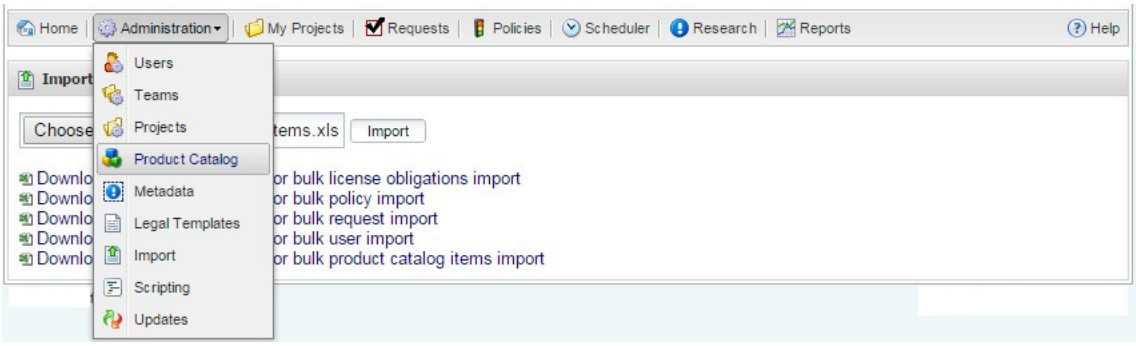
Adding Product Catalog Items

You can create Product Catalog items in Code Insight in the following ways:

- Import them using the **Import** option on the **Administration** pull-down menu.



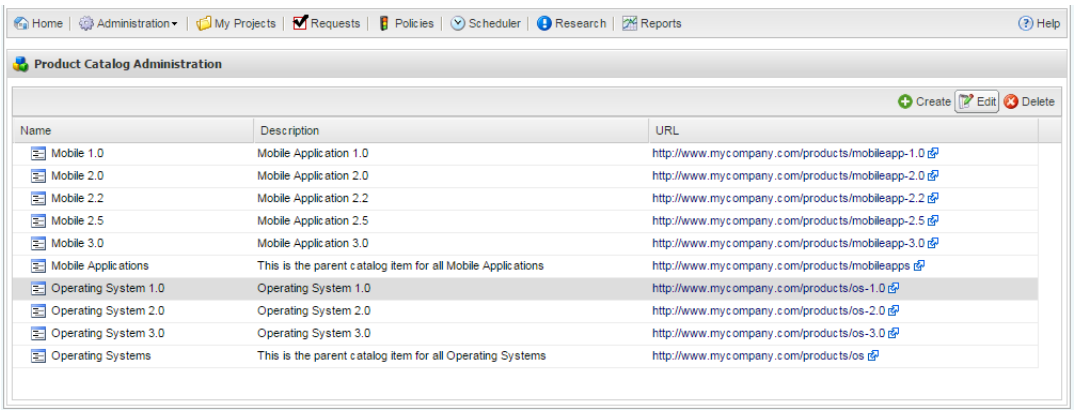
- Add them manually using the **Create** option available after selecting **Product Catalog** from the **Administration** pull-down menu:



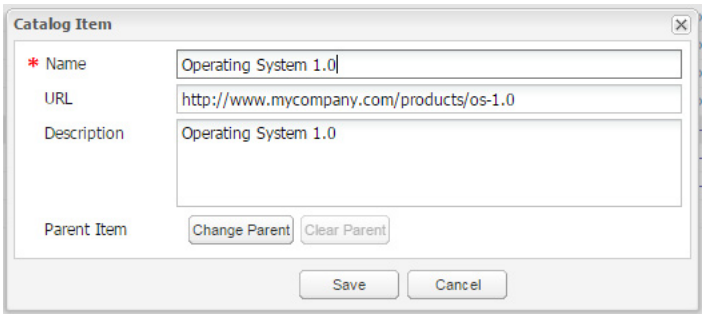
Viewing, Editing and Associating Product Catalog Items

Task ***To edit a Product Catalog Item, do the following:***

1. Select the item on the **Product Catalog Administration** page:



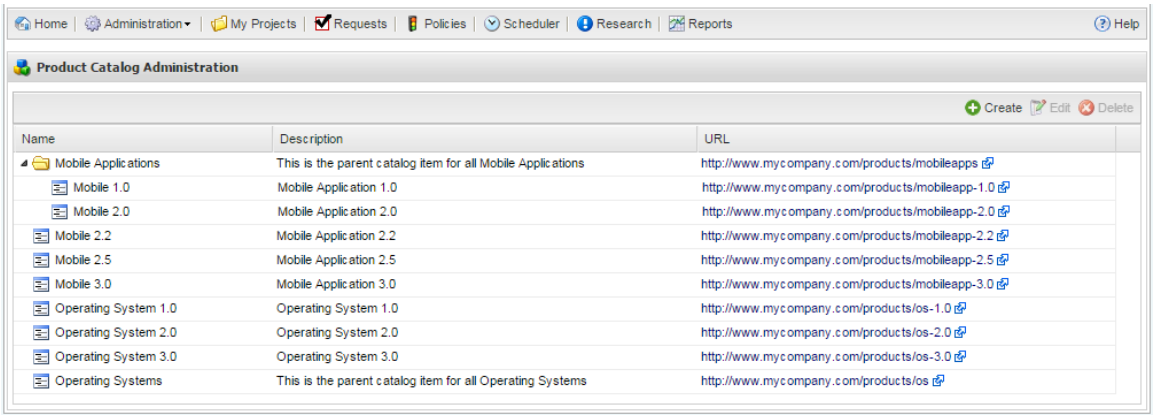
2. Click the **Edit** button. The **Catalog Item** dialog appears:



3. Modify the **Catalog Item** information:
   - Name
   - URL
   - Description
4. To edit the hierarchy of a Product Catalog Item by changing the parent on the item, click the **Change Parent** button in the **Catalog Item** dialog and select a parent to assign to the item.

Note ▪ *A catalog item can have only one Parent catalog item at any given time. A Parent catalog item can have more than one child at any given time.*

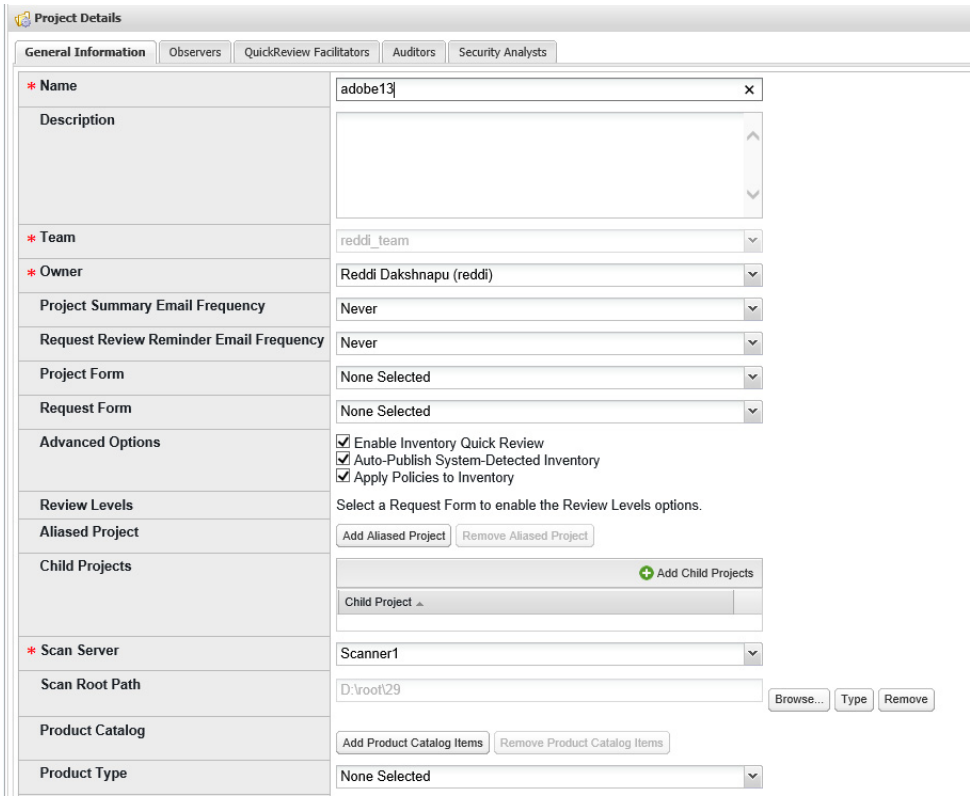
5. Click **Save**. Items that are parents are displayed with a folder icon in the **Product Catalog Administration** list.



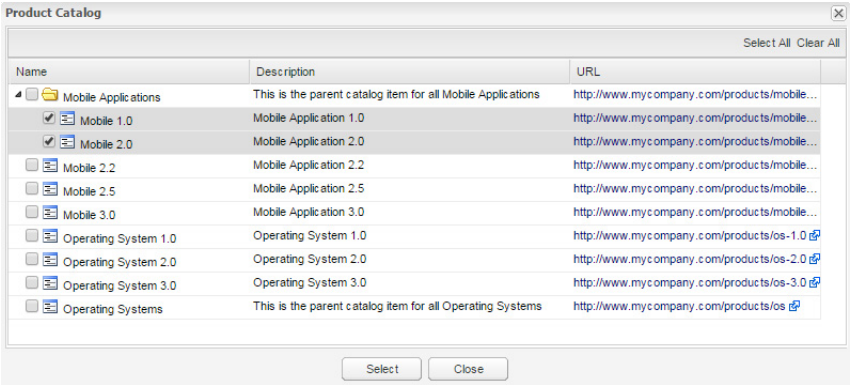
Associating Product Catalog Items with a Project

Task ***To associate Product Catalog items with a Code Insight Project, do the following:***

1. Go to **My Projects** and click the **Edit Project** button for the project you want to edit.
2. On the **Project Details** page, click the **Add Product Catalog** button to display Product Catalog items.



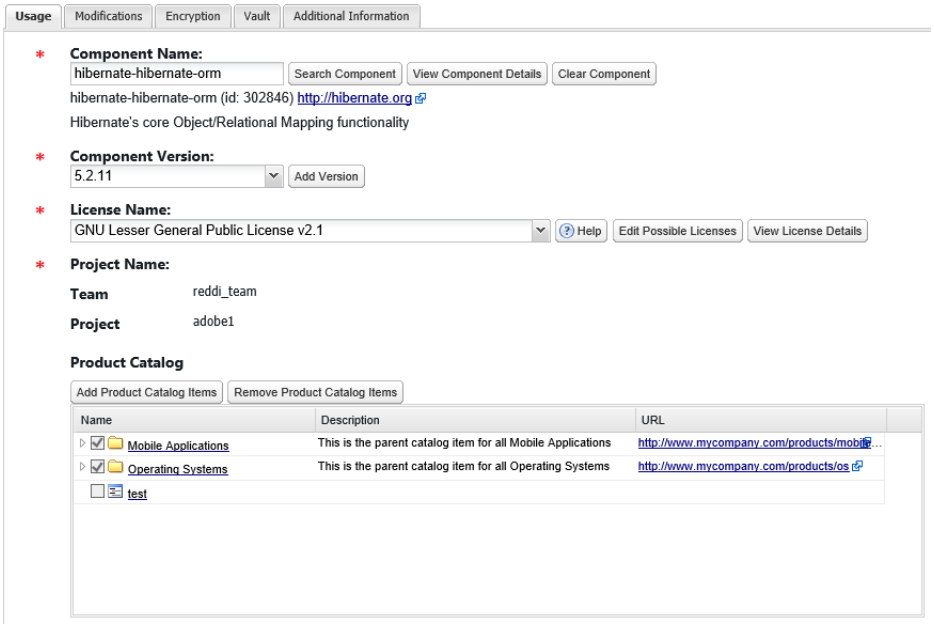
3. Check the items that you would like to associate with the project and click **Select**.



Associating Product Catalog Items with a Request

Task ***To associate Product Catalog items with a Code Insight Request, do the following:***

1. Navigate to the **Request Details** page.
2. Click Add Product Catalog Items button to add product catalog items.
3. Click Remove Product Catalog Items button to remove catalog items.



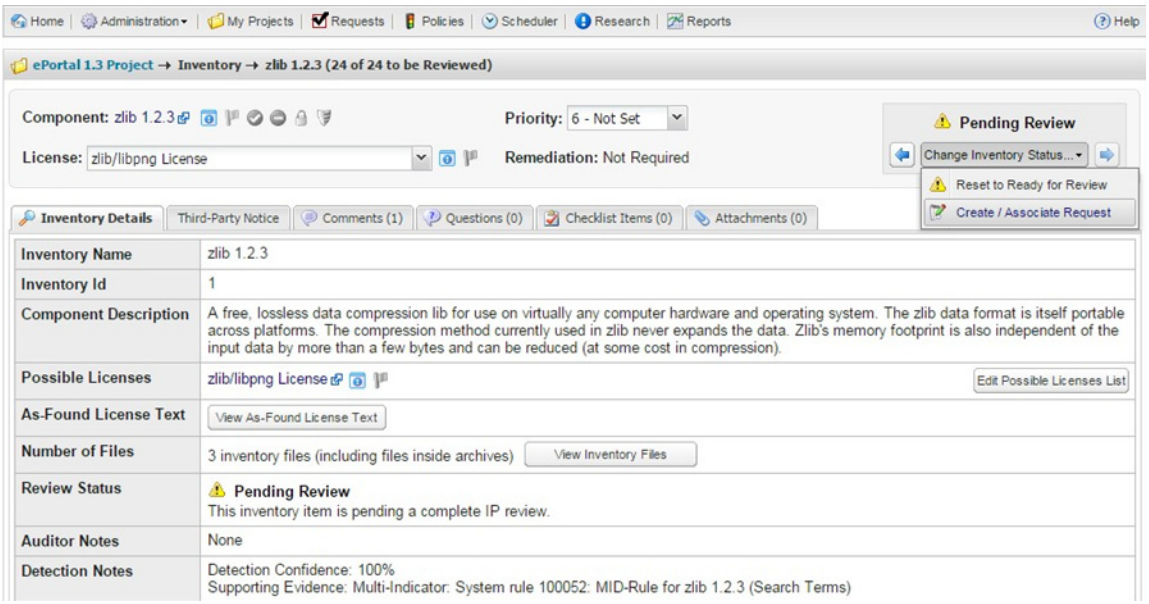
4. Check the items that you would like to associate with the **Request** and click **Select**.

Associating Inventory with Requests across Projects

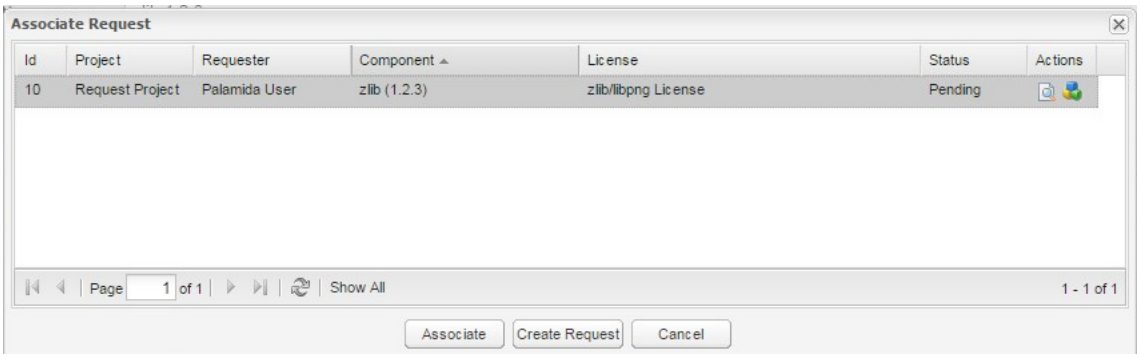
To search for a request across multiple projects when associating inventory, the inventory item and request must share a **Product Catalog** item.

Task ***To associate inventory with Requests across projects, do the following:***

1. Associate the Product Catalog item with the given project.
2. Associate the Product Catalog item with the request.



3. From the associated project, create an inventory item that matches the request component, version and license.
4. Schedule a Full Review for the inventory item and select the option to **Associate the Request**:



The request is available in the search results based on the Product Catalog association.

Reporting on Requests (Filtered by Product Catalog)

The Request Report (Filtered by Product Catalog) option presents all request data associated with the selected catalog items. (For more information about Code Insight report generation, see Generating Reports.)

Task ***To access the report, do the following:***

1. Select **Requests Report (Filtered by Product Catalog)** from the **Reports** menu.
2. Click the **Add Product Catalog** button and select the catalog items you would like to report on. The report output contains the catalog items and their associated requests:

Requests Report (Filtered by Product Catalog) Generated on Monday, April 20, 2015 at 4:14 PM

Download Report
Download Excel Version

There are **3 requests** from **10 catalog items** Show / Hide

Name	Description	URL
Mobile 1.0	Mobile Application 1.0	http://www.mycompany.com/products/mobileapp-1.0
Mobile 2.0	Mobile Application 2.0	http://www.mycompany.com/products/mobileapp-2.0
Mobile 2.2	Mobile Application 2.2	http://www.mycompany.com/products/mobileapp-2.2
Mobile 2.5	Mobile Application 2.5	http://www.mycompany.com/products/mobileapp-2.5
Mobile 3.0	Mobile Application 3.0	http://www.mycompany.com/products/mobileapp-3.0
Mobile Applications	This is the parent catalog item for all Mobile Applications	http://www.mycompany.com/products/mobileapps
Operating System 1.0	Operating System 1.0	http://www.mycompany.com/products/os-1.0
Operating System 2.0	Operating System 2.0	http://www.mycompany.com/products/os-2.0
Operating System 3.0	Operating System 3.0	http://www.mycompany.com/products/os-3.0
Operating Systems	This is the parent catalog item for all Operating Systems	http://www.mycompany.com/products/os

Catalog Items	Request ID	Requester	Component	Version	License	Status	Request Type	Creation Date
Mobile 1.0 Mobile 2.0	9 (URL)	Jane Doe	apache-jakarta-commons-codec	1.10	Apache License 2.0	Approved	Internal Use	04-07-2015
Operating System 1.0 Operating System 2.0 Operating System 3.0	8 (URL)	Jane Doe	jquery	1.9	MIT License (also X11)	Created	Internal Use	03-25-2015
Operating System 1.0 Operating System 2.0 Operating System 3.0	7 (URL)	Jane Doe	scriptaculous	1.9	MIT License (also X11)	Approved	Internal Use	03-25-2015

3. Click the **Show/Hide** button to display additional details for each catalog item attribute. HTML and Excel versions of the report are available for download.

Reporting on Third-Party Notices for Requests (Filtered by Product Catalog)

The **Third-Party Notices Report for Requests (Filtered by Product Catalog)** option presents all third-party notice data for requests associated with the selected catalog items. To run this report, third-party notices should be created for requests. (For more information about Code Insight report generation, see Generating Reports.)

Task ***To access the report, do the following:***

1. Open **Reports | Third-Party Notices Report for Requests (Filtered by Product Catalog)**.
2. Click **Add Product Catalog** and select the catalog items you would like to report on.
3. Click **Generate** to run the report.

4. The report output contains third-party notices only for those requests associated with the selected Product Catalog items. You can download the report in HTML format.

16

Performing Advanced Searches

This section provides information about advanced searches:

- Advanced Searches
- Advanced Project Search Example

Advanced Searches

Code Insight contains an advanced search interface that allows parametric searches to be used for filtering the list of projects, inventory items within a project, components, and licenses.

Advanced search supports multiple search criteria that must all be met for an item to be returned. The search criteria consist of an entity, field, operator, and value. The list of available entities varies based on which entity is being searched as shown below:

Project Advanced Search

- Project Description
- Project ID
- Project Name
- Project Owner
- Project Status
- Requester Name
- Project Metadata Fields
- Project Properties

Inventory Advanced Search

- Inventory
- Request
- Component
- License
- Inventory Properties

Request Advanced Search

- Request
- Inventory Fields
- Comment Text
- Has Comments
- ID
- Requester Name
- Review Deadline
- Status
- Component
- License
- Request Properties

Component Advanced Search

These are components associated with project inventory.

- Component Properties
- Component Fields
- Description
- Has Encryption
- Has Vulnerabilities
- ID
- Is Custom
- Name
- Component Metadata Fields

License Advanced Search

These are licenses associated with project inventory.

- Component
- License
- License properties
- License fields
- Comment Text
- Has Comments
- Has Family
- Has Obligations
- ID
- Is Custom
- Is Family
- Name
- Text
- License Metadata Fields

Policy (Web UI) Advanced Search

- Policy Properties

Team (Web UI) Advanced Search

- Team Properties

Group (Detector) Advanced Search

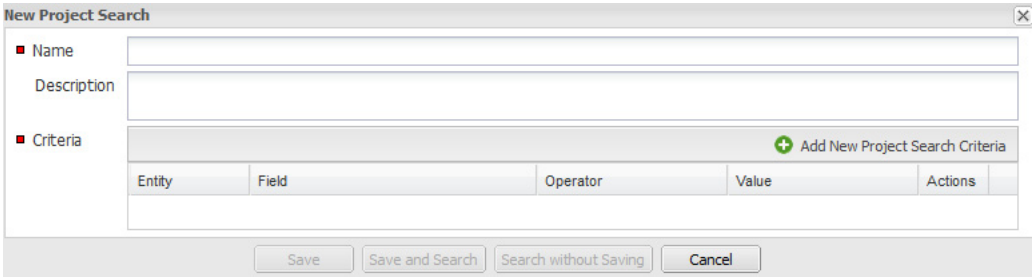
- Group Properties

Advanced Project Search Example

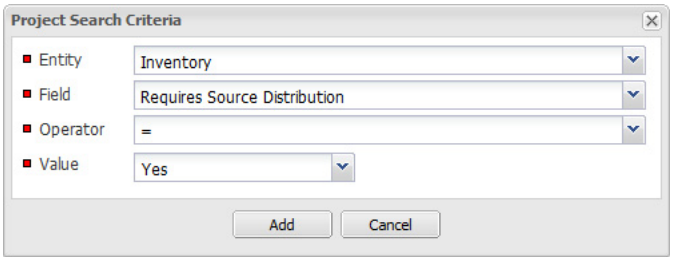
To search for projects using advanced search criteria, perform the following steps.

Task ***To search for projects using advanced search criteria, do the following:***

1. Navigate to **My Projects**.
2. Select **Advanced Search** in the project table header to switch to the advanced search mode.
3. Click the **View Saved Project Searched** button to edit/execute an existing advanced project search, or click the **Add New Project Search** button. The **Advanced Project Search** dialog appears:



4. Click the **Add New Project Search Criteria** button to define search criteria. The **Project Search Criteria** dialog appears.



5. Select and enter criteria for the search:
   - Select the entity to search by.
   - Select the entity field to search by.
   - Select the operator.
   - Select or enter the value for the entity field to search by.
6. Click **Add**.
7. Repeat as necessary to define other criteria for the search.
8. Click the **Save and Search** button or the **Search without Saving** button to invoke the search. To save the search without invoking it, click **Save**. The projects table is reloaded with all project records that match all the defined search criteria:

Note ▪ *When searching using the equals and not equals operator, only properties and/or metadata fields with assigned values are considered. Therefore if a license does not have a value for category and an advanced search criteria is defined as License Category != "Category 1", only those licenses with an assigned license category that is not equal to Category 1 will be returned.*

17

Code Insight Public APIs

This section is a starting point for learning about the Code Insight Public API. It will familiarize you with writing Groovy scripts for access to the Code Insight Public APIs from several contexts.

Public API Entry Points and Contexts

The following places in Code Insight are accessible via the public APIs.

ScriptRunner

ScriptRunner is a standalone module that allows for communication with the Code Insight Core Server and Scan Servers to perform various application functions. For more information, see “Using ScriptRunner” in the *Code Insight Installation and System Administration Guide*.

In addition, Code Insight provides Groovy scripts that can be used to import and export group definitions and associated file information, such as tags applied to files, along with all attendant custom data. For information, see Importing & Exporting Workspaces in the *Code Insight Installation and System Administration Guide*.

Web UI Administration Scripts Page

Custom Groovy scripts can be registered in the Code Insight central database via a SQL script and can be executed via the Script option on the Administration page in the Code Insight Web UI.

Note ▪ *These scripts are not interactive and therefore parameters cannot can be passed to the script.*

Web UI Custom Buttons

Some Code Insight customers have custom features and/or integrations that have been implemented using external Groovy scripts. These scripts are invoked by custom buttons in the Code Insight Web UI that are activated by a Code Insight license key (`Code Insight.key`) that enables such functionality. If your Code Insight license key does not activate these features, the custom buttons will not be visible.

Detector Groovy Console

Groovy scripts may be executed from the Code Insight Detector client either via the built-in Groovy console or via the Scripts menu. These scripts are limited to functionality available within a given Code Insight workspace, and must operate within that context.

Scan Server Autorun Scripts

There are several Code Insight scanner hooks that is used to execute custom logic at various phases of the workspace scan process. The following phases are supported: workspace open, on error, on success, before & after scan, after scan commit, and after post-scan analysis. These scripts are provided a `WorkspaceCover` object for convenience.

Workspace Reports

Code Insight workspace reports in some cases have been implemented in Groovy and make use of Code Insight public APIs. Workspace reports are provided a `WorkspaceCover` object and are limited to functionality available within a given Code Insight workspace.

Web Reports

Code Insight Web reports have all been implemented using Groovy scripts to allow customers to modify existing reports or create new ones.

Accessing Core & Scan Servers via ScriptRunner

ScriptRunner is a standalone module that allows communication with Code Insight Core Server and Scan Servers to perform various application functions. For more information about ScriptRunner, see “Using ScriptRunner” in the *Code Insight Installation and System Administration Guide*.

Note ▪ *If you are running ScriptRunner from a machine other than the Code Insight Core Server, you must use the -c flag to specify the URL of the Code Insight Core Server (i.e. `http://localhost:8888/Code Insight`). If using ScriptRunner for the first time, you will be prompted for a password. Below are examples:*

- **On Windows:** *`scriptRunner.bat -c <CORE_SERVER_URL> -u user file.groovy`*
- **On Linux:** *`./scriptRunner.sh -c <CORE_SERVER_URL > -u user file.groovy`*

The following operations are supported via ScriptRunner. They are organized by various covers (facades) that expose the various APIs. For more information, see **API Covers (Facades)**.

Table 17-1 ▪ Operation Types Supported via ScriptRunner

Operation Type	Operation
Admin Service Cover	• User, Team, Project, and Policy Management • Metadata Value Management (User, Team, Policy, Task) • Other Utility APIs
Metadata Service Cover	• Metadata Field Management
Request Service Cover	• Request Management
Reference Data Service Cover	• Component, Version, License, and Vulnerability Management • Metadata Value Management (Component, Version, License, and Vulnerability)
Project Data Cover	• Project Configuration Management • Workspace Management • Project Queue (Scans + Reports) Management • Project Inventory Management (Comments, Questions/Answers, Checklist Items, and Review Status) • Project Request Management • Project Task Management • Project Snapshot Management • Metadata Value Management (Project, Workspace, Inventory, and Request)
Auditor Service Cover	• Tag Management • Group Management • File Management • Metadata Value Management (File and Group)
Workspace Locator Cover	• Open Existing Workspaces
Workspace Cover	• Manage Workspace Settings • Obtain Workspace Scan Results • Codebase APIs

API Covers (Facades)

Code Insight public APIs are exposed via various covers (facades). These covers are the entry point for various service-level APIs that allow interfacing with the Code Insight Core Server and Scan Server.

- Admin Service Cover
- Metadata Service Cover
- Reference Data Service Cover
- Project Data Cover
- Auditor Service Cover
- Workspace Locator Cover
- Workspace Cover

Admin Service Cover

The `AdminServiceCover` is the entry point for Code Insight Core Server administrative public APIs. It supports various operations related to managing users, teams, projects, and policies. To use these APIs, the `AdminServiceCover` needs to be instantiated by passing in the Code Insight Core Server IP address. An example is shown below.

Constructor

```
AdminServiceCover(String Code InsightCoreServerIpAddress)
```

Sample Code

```
import com.Code Insight.script.AdminServiceCover;

// Enter core server IP address
String coreServerIpAddress = System.getProperty("core.server.url");
AdminServiceCover adminSrv = new AdminServiceCover(coreServerIpAddress);
if (adminSrv != null) {
   try {
      // Print all team names
      adminSrv.getTeamNames().each {
         team ->
         println("Team: " + team);
      }
   } catch (Exception e) {
      e.printStackTrace();
   }
} else {
   println("Unable to connect to Code Insight Core server: " + coreServerIpAddress);
}
```

Metadata Service Cover

The `MetadataServiceCover` is the entry point for Code Insight Core Server metadata management public APIs. To use these APIs, the `MetadataServiceCover` needs to be instantiated by passing in the Code Insight Core Server IP address. An example is shown below.

Constructor

```
MetadataServiceCover(String Code InsightCoreServerIpAddress)
```

Sample Code

```
import com.Code Insight.script.MetadataServiceCover;

// Enter core server IP address
String coreServerIpAddress = System.getProperty("core.server.url");
MetadataServiceCover mdSrv = new MetadataServiceCover(coreServerIpAddress);
if (mdSrv != null) {
   try {
      // Print all metadata field/values for a particular license
      Object mdValue = null;
      long licenseId = 20; // Apache License, Version 2.0
         mdSrv.getMetadataDefinitions(mdSrv.getEntityId("license")).each {
            mdDef ->
            mdValue = mdSrv.getValue(mdDef.getId(), licenseId);
            // MDValue always returns a list even if single value, get first
            // value if list is not empty
            if (mdValue != null && mdValue.size() > 0) {
            println("  " + mdDef.getDisplayName() + ": " + mdValue.get(0));
         }
      }
   } catch (Exception e) {
      e.printStackTrace();
   }
} else {
   println("Unable to connect to Code Insight Core server: " + coreServerIpAddress);
}
```

Reference Data Service Cover

The `ReferenceDataServiceCover` is the entry point for Code Insight Core Server component, version, license, and vulnerability entity management public APIs. To use these APIs, the `ReferenceDataServiceCover` must be instantiated by passing the Code Insight Core Server IP address. An example is shown below.

Constructor

```
ReferenceDataServiceCover(String Code InsightCoreServerIpAddress)
```

Sample Code

```
import com.Code Insight.script.ReferenceDataServiceCover;

// Enter core server IP address
String coreServerIpAddress = System.getProperty("core.server.url");
```

```
ReferenceDataServiceCover refDataSrv = new ReferenceDataServiceCover(coreServerIpAddress);
if (refDataSrv != null) {
   try {
      // Print all license family names
      refDataSrv.getLicenseFamilies().each {
         licId ->
         println("License Family: " + refDataSrv.getLicense(licId).getName());
      }
   } catch (Exception e) {
      e.printStackTrace();
   }
} else {
   println("Unable to connect to Code Insight Core server: " + coreServerIpAddress);
}
```

Project Data Cover

The `ProjectDataCover` is the entry point for Code Insight Core Server project data management public APIs. To use these APIs, the `ProjectDataCover` must be instantiated by passing the Code Insight Core Server IP address. An example is shown below.

Constructor

```
ProjectDataCover(String Code InsightCoreServerIpAddress)
```

Sample Code

```
import com.Code Insight.script.ProjectDataCover;

// Enter core server IP address
String coreServerIpAddress = System.getProperty("core.server.url");
ProjectDataCover pdCover = new ProjectDataCover(coreServerIpAddress);
if (pdCover != null) {
   try {
      // Print all inventory items for project
      String teamName = "Engineering";
      String projectName = "ePortal 1.3";
      pdCover.getInventoryForProject(teamName, projectName).each {
         invItem ->
         println("Inventory Item: " + invItem.getAuditGroupLabel());
      }
   } catch (Exception e) {
      e.printStackTrace();
   }
} else {
   println("Unable to connect to Code Insight Core server: " + coreServerIpAddress);
}
```

Auditor Service Cover

The `AuditorServiceCover` is the entry point for auditor APIs related to groups and tags. To use these APIs, the `AuditorServiceCover` must be instantiated by passing the Code Insight Core Server IP address. An example is shown below.

Constructor

```
AuditorServiceCover(String Code InsightCoreServerIpAddress)
```

Sample Code

```
import com.Code Insight.script.AuditorServiceCover;

// Enter core server IP address
String coreServerIpAddress = System.getProperty("core.server.url");
AuditorServiceCover auditCover = new AuditorServiceCover(coreServerIpAddress);
if (auditCover != null) {
   try {
      // Print all groups for project
      int projectId = 1;
      auditCover.getGroupIds(projectId).each {
         groupId ->
         println("Group: " + auditCover.getGroupDetails(groupId).getName());
      }
   } catch (Exception e) {
      e.printStackTrace();
   }
} else {
   println("Unable to connect to Code Insight Core server: " + coreServerIpAddress);
}
```

Workspace Locator Cover

The `WorkspaceLocatorCover` is instantiated by ScriptRunner and provided as a variable called `locator`. The `locator` is used to open a workspace. An example is shown below.

Constructor

```
import com.Code Insight.script.WorkspaceCover;
```

Sample Code

```
// Enter scan server IP address
String scanServerIPAddress = "127.0.0.1";
String workspaceName = "ePortal_13";
String serverUri = "http://" + scanServerIPAddress + "/";
String workspaceUri = serverUri + workspaceName;
locator.setServerUri(serverUri)
WorkspaceCover workspace;
try {
   workspace = locator.openWorkspace(workspaceUri);
   println("Workspace Name: " + workspace.getName());
   println("Workspace UUID: " + workspace.getUuid());
} catch (Exception e) {
   e.printStackTrace();
} finally {
   if (workspace) {
      workspace.close(); // Make sure you always close a workspace you open
   }
}
```

Workspace Cover

The `WorkspaceCover` is the entry point for workspace scan results and codebase access APIs. To use these APIs, the `WorkspaceLocatorCover` must be used to obtain a `WorkspaceCover` object that represents an open workspace. The `WorkspaceCover` can then be used to obtain access to the following covers that contain APIs for workspace scan results and codebase access:

CodebaseServiceCover

This is a deprecated cover and should not be used. Use the `WorkspaceCover.getFilesystem()` API instead to gain access to the codebase (files in the file system) that was scanned by the workspace.

ScanResultCover

This cover is used to obtain high-level information about the latest scan including the Compliance Library version, the scan date and duration, and the various file counts associated with the `workspaceultServiceCover`.

This cover is used to obtain scan results for a given workspace. There are APIs provided to obtain any scan results that can be viewed by the auditor in the Detector client.

SignatureServiceCover

This cover is used to obtain information about files, Java names, and releases known to a given workspace.

Sample Code

Below is an example for showing the number of scanned files and detected copyright holders:

```
import com.Code Insight.script.WorkspaceCover;

// Enter scan server IP address
String scanServerIPAddress = "127.0.0.1";
String workspaceName = "ePortal_13";
String serverUri = "http://" + scanServerIPAddress + "/";
String workspaceUri = serverUri + workspaceName;
locator.setServerUri(serverUri)
WorkspaceCover workspace;
try {
   workspace = locator.openWorkspace(workspaceUri);
   println("Workspace Name: " + workspace.getName());
   println("Workspace UUID: " + workspace.getUuid());
   println("Scanned Files: " + workspace.getScanResult().getScannedFileCount());
   println("Detected Copyright Holders...");
   workspace.getScanResultService().getCopyrightOwners().each {
      holder ->
      println("  " + holder.getName());
   }
} catch (Exception e) {
   e.printStackTrace();
} finally {
   if (workspace) {
      workspace.close(); // Make sure you always close a workspace you open
   }
}
```

18

Auditing and Analysis Overview

This section provides an overview of the what, how, and why auditors perform the tasks associated with their role.

- Tasks an Auditor Performs
- How Auditors Perform Tasks
- Why Auditors Perform Tasks

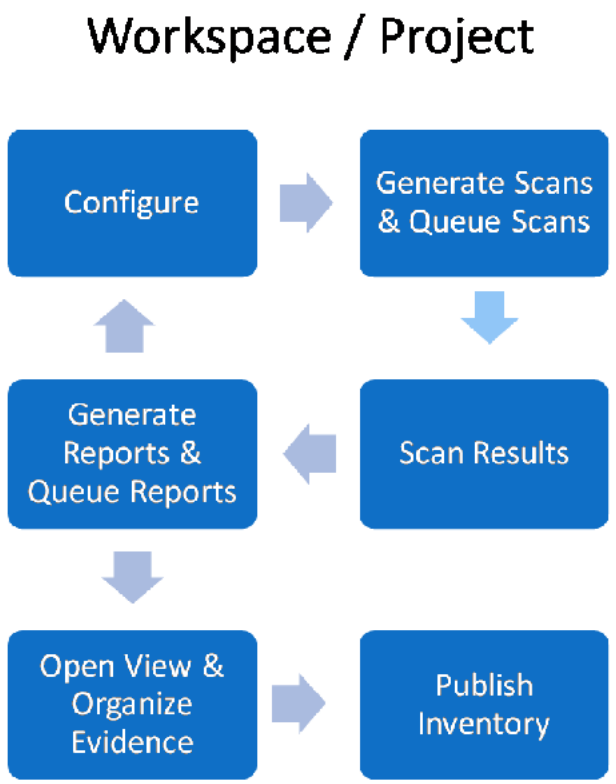
Tasks an Auditor Performs

An auditor has the job of analyzing evidence but not performing Intellectual Property (IP) compliance and vulnerability problem-solving actions. The general auditing and analysis steps are as follows:

1. Setting up the workspaces to scan codebases for third-party materials
2. Scheduling scans and reports
3. Launching Detector
4. Viewing scan and reports in the task queue
5. Viewing reports
6. Choosing the workspaces to open and analyze
7. Opening and accessing scan results
8. Viewing different types of evidence (for example, source code fingerprint matches, exact files [digest] matches, copyright matches, license matches, string search term matches, Java name matches, etc.)
9. Organizing and viewing the codebase scan evidence
10. Analyzing the codebase scan evidence and constructing groups and components to represent detected inventory
11. Publishing groups as inventory

12. Managing a catalog of custom Open Source Software (OSS) material

A general view of the auditor workflow is presented in the following diagram. If the scan is not providing you with the information you are seeking, you can re-configure the workspace and schedule a new scan to analyze.



How Auditors Perform Tasks

An overview of the way an auditor performs the various tasks is as follows:

1. Configure the workspace settings and details. This includes file settings, detection techniques to employ during the scan, sensitivities of the results, and specifics about the types of files to scan, such as extensions, etc.
2. After the workspace is configured, schedule scans and reports of the codebase in one or many workspaces.
   a. Scanning is best done in batches. To save time, queue your scans before leaving work. This way they can be sequentially performed through the night and not during work hours. Depending on the size of your code base, some scans and/or reports may take many hours.
   b. Forensic scans allow you to generate reports that categorize underlying evidence. The advantage of this is that you can look at reports, for example, while on an airplane without network connectivity, by printing out a static hard copy.
   c. Scan results are live on the server, and unless you change a code base or data library you don't need to scan a workspace again. Reports are persistent on the Scan Server.

3. Check the specific task queues to make sure you have scheduled and prioritized the scans and reports to your liking.
4. Once the scans are completed, launch Detector from the Web UI. The Detector forensic client allows you to choose one workspace from your completed list of scans and view, with varying levels of granularity, every piece of data (scan results) in that specific workspace's code base.
5. Analyze any evidence that might constitute a relevant compliance or security vulnerability issue in your code base. The evidence you analyze includes source code fingerprint matches, exact files [digest] matches, detected copyrights, detected license text or references, string search term matches, detected emails and URLs, Java name matches, etc. Sometimes there is an issue, but it has low relevance to your eventual software application's integrity. For example: You find an issue with a certain license match. But this license match issue is only relevant if you are distributing your codebase internationally. If your company has no history or intention of distributing your application internationally, you do not have a licensing issue that requires attention.
6. Organize and evaluate the information as it relates to your own internal compliance and security vulnerability policies.
7. Publish an inventory of components that are detected via the scan of the code base. These inventory items are then put through the compliance workflow to ensure that usage is permissible in your organization. This workflow may involve the security analysts, owner, and various reviewers.
8. Inventory with associated component versions containing known security vulnerabilities will require a review by the security analyst. Furthermore, all inventory items require an IP review by the reviewers, and these are configured by the project review workflow.

Why Auditors Perform Tasks

The main job function of the auditor is to analyze the results of the codebase scan, and generate an inventory (bill of materials) using the automated detection techniques as well as the other available third- party indicator evidence that resulted from the scan. By subsequently publishing the detected inventory, issues related to IP compliance and security vulnerabilities can be brought to the attention of the rest of the team.

The auditor does not make any compliance decisions. His or her sole responsibility is to analyze the scan results and generate a complete and accurate inventory of third-party materials. Once the inventory is published the rest of the workflow determines the validity of the existence of the inventory items.

19

Configuring Workspaces

This section contains the following topics:

Workspaces

The owner or assigned auditor configures project workspaces before launching a project. Workspaces are associated with specific projects and are usually scanned to ensure code compliance and to discover security vulnerabilities associated with the detected components.

Note ▪ *Only letters, numbers, underscores, and dashes may be used in the workspace name. To access workspace configuration settings, perform the following steps.*

Accessing Workspace Configuration Settings

To access workspace configuration settings, perform the following steps.

Task ***To access workspace configuration setting, do the followings:***

1. Log in to Code Insight as the *Auditor*.

2. You can get to the list of your active projects from the **Dashboard** or by clicking **My Projects** in the main menu.

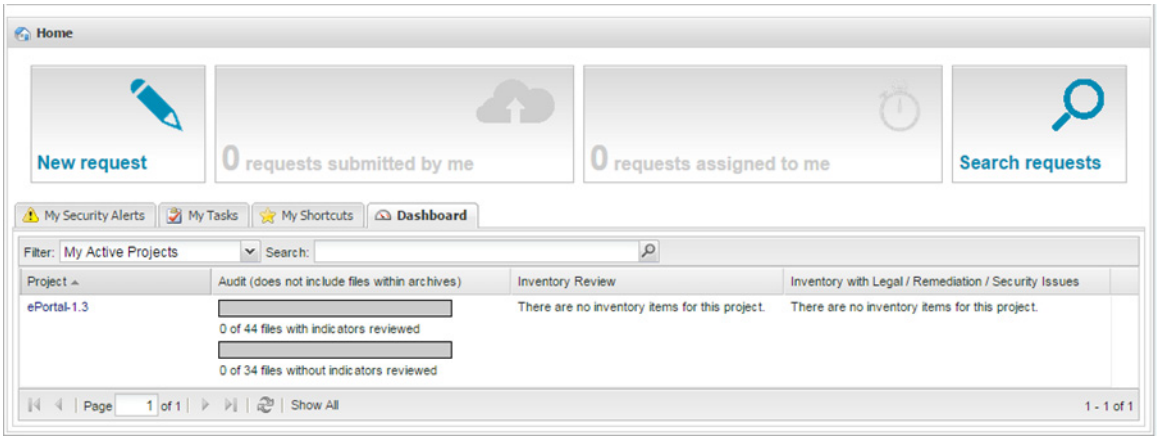
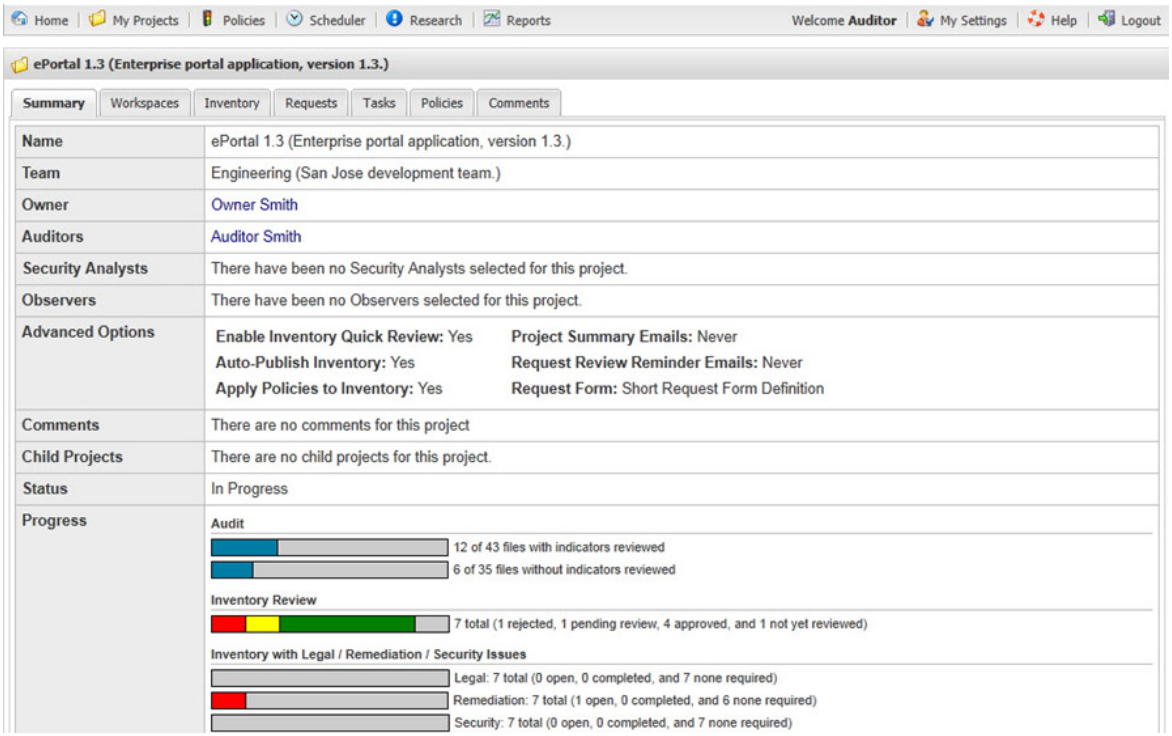


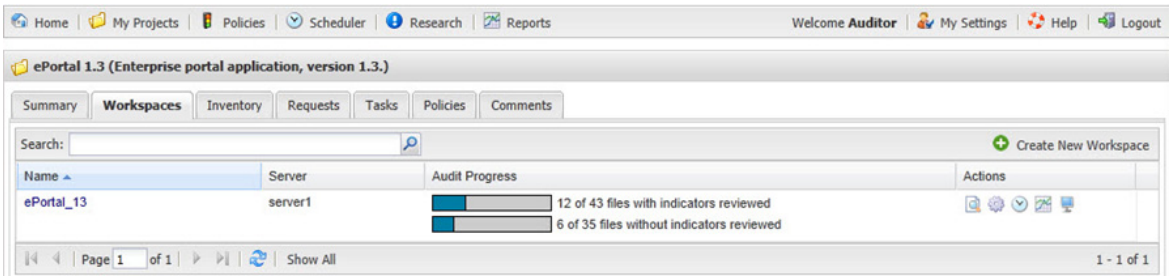
Figure 19-1: My Projects Dashboard

Figure 19-2: My Projects Screen

3. Click the project name (**Name** column) or the **magnifying glass** icon (**Actions** column). The **Summary** tab appears.



4. Select the **Workspaces** tab, which shows the project compliance and vulnerability workspaces associated with the project. The icons to the right of the project name allow you to perform the following activities:
   - View and edit workspace details (magnifying glass).
   - View and edit workspace settings (cog).
   - View the task queue for the selected workspace (clock).
   - View reports for the selected workspace (graph).
   - Launch the Detector forensic client for the selected workspace (computer monitor).

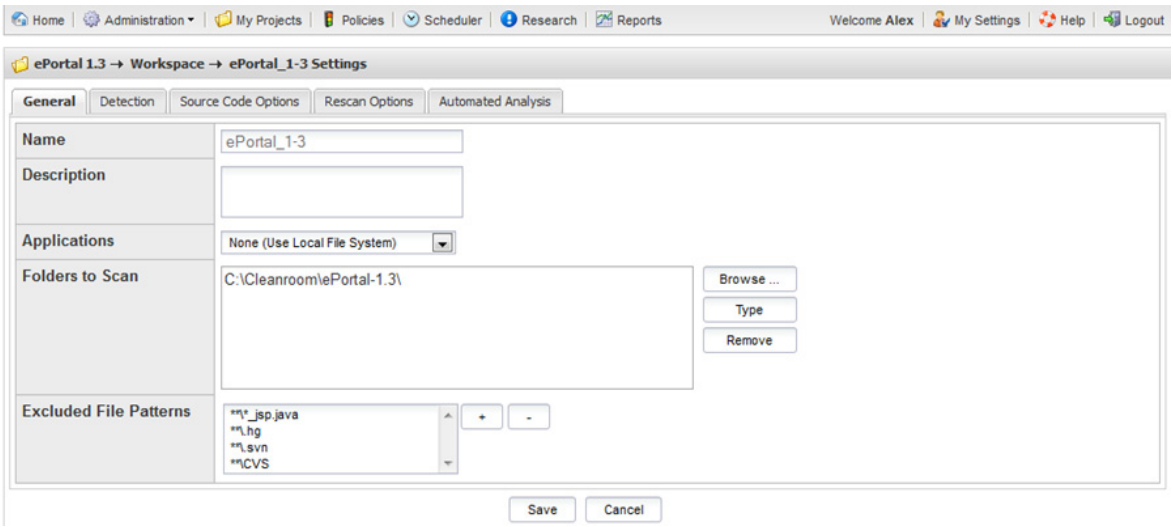


General Tab Tasks

To configure Workspace Settings on the **General** tab, perform the following steps.

Task ***To set workspace settings on the General tab, do the following:***

1. Click **My Projects > Project Name > Workspaces Tab >Cog** icon. The **General** tab appears:



2. The **General** tab allows you to enter a description for the workspace. If you are scanning a snapshot of your code base that is not tied to an SCM system, you can also choose folders to scan by clicking the Browse, Type, or Remove buttons. Note that the folders to scan may be restricted by the group to which this project belongs. The Application Administrator can restrict which folders are available for groups to scan by assigning the folder group permissions. Clicking on the Type button opens a dialog box requesting that you input a scan folder. To add a file pattern to exclude during a scan, click the Plus icon. The Minus icon removes excluded file patterns, so these files now appear in a scan.

Note ▪ *Windows hard links are not respected by the scanner and may cause file counts to be off in the workspace.*

3. Click **Save**.

Software Configuration Management Tab

If integration with ClearCase, Perforce, Subversion, Git, or TFS is enabled in the Scan Server SCM configuration file (`<CODE_INSIGHT_ROOT>/config/scanEngine/scm.properties`), the **Applications** pull-down menu in the Workspace Settings – General tab is populated with the available options.

Selecting any of the options enables the **Software Configuration Management** (SCM) tab which allows you to configure the SCM settings for the selected SCM application for the current workspace. Doing this ensures that files are up to date at the time of scanning.

See the *Installation and System Administration Guide* for detailed instructions related to SCM configuration for workspaces and project copy.

Detection Tab Tasks

To configure Workspace Settings on the Detection tab, perform the following steps.

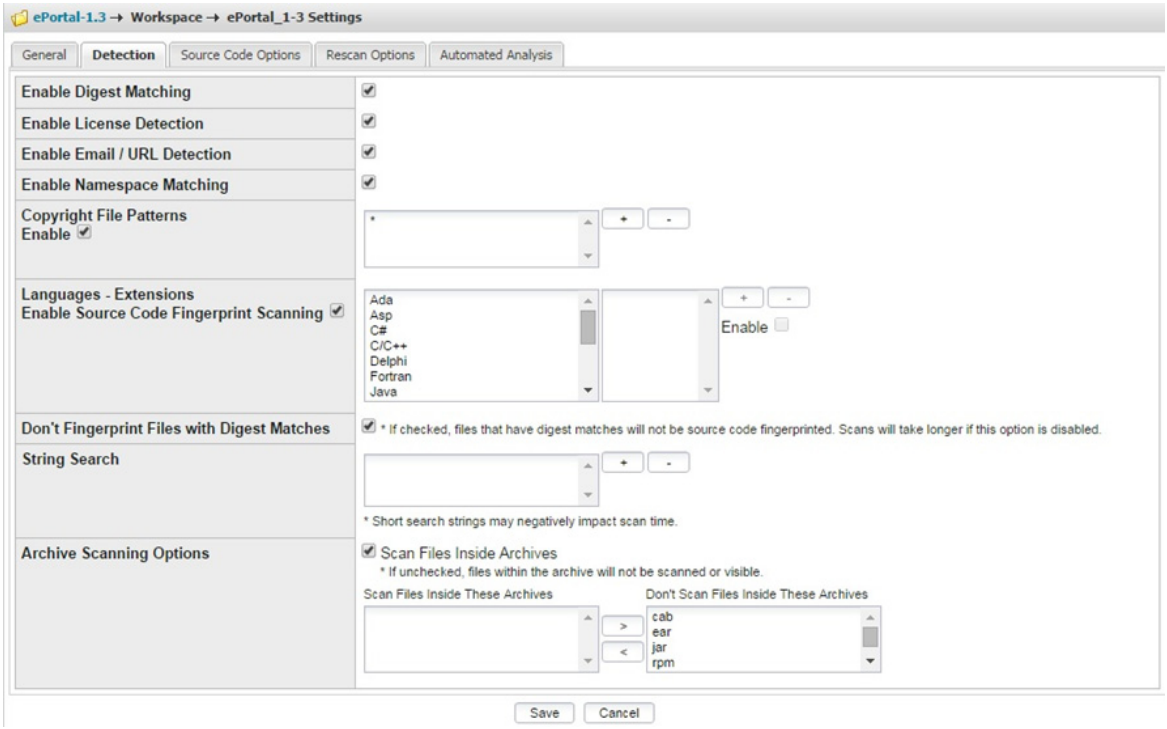
Task ***To configure workspace settings on the Detection tab, do the following:***

1. To access the Detection tab screen, follow the instructions in the General Tab Tasks.
2. Click the Detection tab. The Detection tab screen allows you to set the detection parameters. For example, you can enable digest matching, namespace matching, copyright file patterns, language extensions, source code fingerprint scanning, and string searches.
3. You can choose to exclude or include in the scan specific code languages and file extensions associated with those languages. To see the file extensions associated with the language that will be included in the scan, highlight a language and then select the **Enable** checkbox.
4. Check the **Scan Files Inside Archives** checkbox to perform a deep scan on the archives in your codebase. With this option enabled, the archive's outer file (such as foo.jar) and its inner files (all files inside foo.jar) are tagged for different evidence types. Both types of files will be visible in **Detector**, allowing you to perform file operations such as marking, tagging and adding to group.
5. The following table lists the supported archive types and evidence.

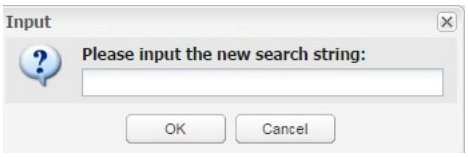
Table 19-1 ▪ Supported Archive Types and Evidence

Supported Types	Description
Supported Archive Types	ab, ear, jar, rpm, sar, tar, tar.bz, tar.bz2, tar.bzip2, tbz, tgz, war and zip are supported archive types. For the most up-to-date information on supported types, go to Workspace Settings -> Detection Tab -> Archive Scanning Options section at the bottom of the page to see a list of extensions supported in this release.
Supported Evidence Types	Exact Matches, Copyrights, Emails and URLs, Search Terms and Licenses are supported evidence types. Automated Analysis techniques such as Auto-WriteUp, POM File Analyzer and MID Rule Detection are also available for archives. See the Automated Analysis Tab section on how to enable these techniques.
Source Matches	Files inside archives will not be scanned and tagged for Source matches. For extended analysis on files inside archives, it is recommended that you manually unpack the archive before performing a scan.
Exact Matches	Archives that contain an Exact match (digest match) on the outer file will be tagged with the "Contains Exact File Matches" tag. Inner files of these archives will not be tagged for Exact matches. This behavior is in place as a performance enhancement and is reflected in the group/tag file counts. Inner files of archives with Exact matches will still be tagged for all other evidence types, including Copyrights, Emails and URLs, Search Terms and Licenses.

6. You can specify which archive types to include for deep scanning in the Archive Scanning Options section. To specify an archive type for deep scanning, select the extension associated with the archive from the list of available extensions. Use the **<** option to move the item to the **Scan Files Inside These Archives** area on the left. If you do not want to include an archive type for deep scanning, use the **>** option to move the extension to the **Don't Scan Files Inside these Archives** area on the right. For information on viewing and analyzing scan results for archive files, see Working with Archives.



7. To search the code base for a certain string, click the Plus icon in the **String Search** row. The Input dialog appears so you can enter the exact text string for which to search.



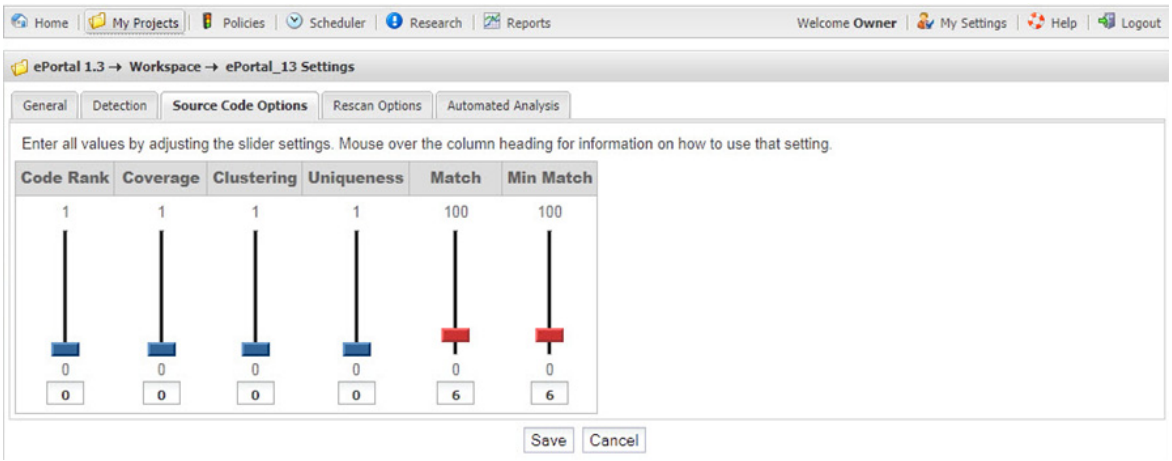
8. Click **OK** to save your detection settings.

Source Code Options Tab

To set the sensitivity of the workspace scan of the source code scan, perform the following steps.

Task ***To set the sensitivity of the workspace scan of the source code scan, do the following:***

1. To access the **Source Code Options** tab, follow the instructions in the General Tab Tasks section.
2. Select the **Source Code Options** tab. The **Source Code Options** tab opens. You can adjust the sliders to narrow the percentages of certain code elements and characteristics you want to see in a scan. This allows you to save time by analyzing priority compliance issues instead of every issue that might appear in a code base.



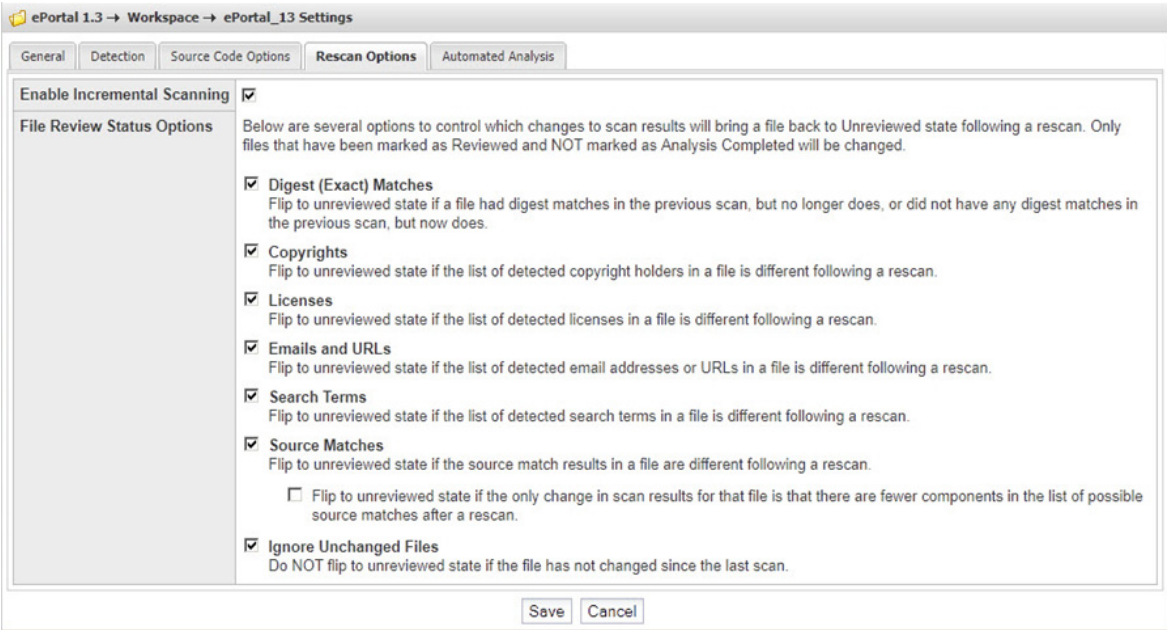
3. The sliders act as scan thresholds. All source matches with values below the sliders will be suppressed from the scan results. For example, if you set the Coverage slider to .75 (75 percent coverage), then the scan results will only source matches with a coverage value above 75%. All other matches will not be available. *Coverage* is a term used to identify the percentage of third-party files detected in the scanned source file.

Rescan Options Tab

The options on this tab allow you to control the conditions under which files that have been reviewed by the auditor are brought out of review during a rescan.

Task ***To configure workspace settings on the Rescan Options tab, do the following:***

1. To access the **Rescan Options** tab, follow the instructions in General Tab Tasks.
2. Select the **Rescan Options** tab to view available options.



You can disable incremental scanning and force a complete re-scan of the selected codebase. Incremental scans only scan files that have been added or modified since the last scan. Note, however, that the entire codebase is always scanned during the first scan. Additionally, during a rescan, the entire codebase is scanned if the Compliance Library has been updated or an NG-bridge update has occurred since the last scan.

3. Select any of the following options to control which changes in the scan results will return a file to an unreviewed state after a rescan:

   - **Digest (Exact) Matches**—Return to the unreviewed state if a file had digest matches in the previous scan, but no longer does, or did not have any digest matches in the previous scan, but now does.
   - **Copyrights**—Return to the unreviewed state if the list of detected copyright holders in a file is different following a rescan.
   - **Licenses**—Return to the unreviewed state if the list of detected licenses in a file is different following a rescan.
   - **Emails and URLs**—Return to the unreviewed state if the list of detected email addresses or URLs in a file is different following a rescan.
   - **Search Terms**—Return to the unreviewed state if the list of detected search terms in a file is different following a rescan.
   - **Source Matches**—Return to the unreviewed state if the source match results for a file are different following a rescan. The associated option clarifies the meaning of “different”:
     - If the associated option **Flip to unreviewed state if the only change in scan results for that file is that there are fewer components in the list...**. is *not* selected, the file status is reset to unreviewed only if the file *gains* component source matches after the rescan. In this way, you would need to re-analyze the file only if new evidence was discovered in the file.

- If this associated option *is* selected, the file status is reset to unreviewed whether the file *gains or loses* (or both gains and loses) component source matches after a rescan. In this way, you would need to re-analyze the file for any changes in source matches.

  Consider that the number of component sources matches can remain the same after a rescan if an equal number of source matches were gained and lost during the scan. However, based on the fact that source-match changes occurred, the file status will be set to unreviewed.

Note that the **Source Matches** option must be selected before the selection (or deselection) of the associated option goes into effect.

- **Ignore Unchanged Files**—Do *not* return to unreviewed state if the file has not changed since the last scan.

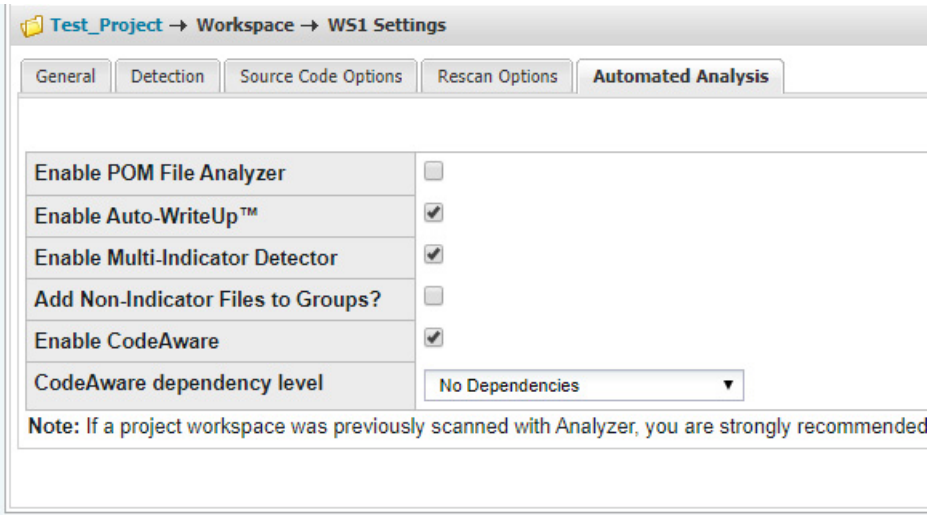
Note ▪ *Only files that have been marked as Reviewed and not marked as Analysis Completed will be changed.*

Automated Analysis Tab

To control which automated analysis techniques are executed after the scan completes, perform the following steps. (Refer to Automated Analysis for more details about these techniques.)

Task ***To configure workspace settings on the Automated Analysis tab, do the following:***

1. To navigate to the **Automated Analysis** tab, select **My Projects > *project_name* > Workspaces>*workspace_name*>Workspace Resources>Edit Workspace Settings**.
2. Open the **Automated Analysis** tab:



3. Select any of the following options for the automated analysis of the scan results for a workspace:

   - **Enable POM File Analyzer**—This automatically creates a group for the artifact for which the POM file exists, as well as for each dependency that is defined in the POM file. The dependency groups also contain the group id of the parent group in the new parentIds field. This data can be used to generate a hierarchical relationship between the top-level projects and their transitive dependencies. This analysis technique is disabled by default.

   - **Enable Auto-WriteUp™**—This automatically creates a group with associated files and populates it with group details such as Description, URL and As-Found License Text if all rule engine conditions are satisfied. For detailed information on this option, see Using Auto-WriteUp™. This analysis technique is enabled by default.

   - **Enable Multi-Indicator Detector (MID)**—This automatically creates a group with associated files if all MID rule conditions are satisfied. This analysis technique is enabled by default.

   - **Add Non-Indicator Files to Groups**—For each system group that is created, the indicator files (files that are indicators of the group and the reason the group exists) are always added to the group. However, you can also optionally have the system add all other files known to exist in the component (and version if available) associated with the group. In other words, any scanned file that has exact matches to the component (and version if available) can be added as an included file to the system created group. This option is disabled by default.

   - **Enable Analyzer**—(Available only if the Code Insight administrator has enabled the Analyzer in your Code Insight installation.) The Analyzer aids the auditing of third-party code by automatically identifying dependencies, packages, and source distributions. The Analyzer works in two ways. First, it can be used before a scan task to generate a Quick Assessment report that provides an overview of your codebase contents. Secondly, the Analyzer can execute automatically (or be scheduled to execute) after a scan to create component groups or inventory items. It also builds Third-Party Notices for groups and inventory items when the data is available and, for scheduled executions, produces a Group Builder report. Refer to Using the Analyzer to for details.

   - **Enable CodeAware**—CodeAware is basically the "next generation" of automated discovery, providing much of the same detection functionality as the Analyzer to build groups and auto-generate inventory.

     ***Note** ▪ If a workspace was previously scanned with the Analyzer, you are strongly recommended not to enable CodeAware in place of the Analyzer for subsequent scans, as this can result in duplicate groups. If you would like to replace Analyzer results with CodeAware results, you can do so in a new project to avoid duplicates.*

   - **CodeAware dependency level**—If Enable CodeAware is selected, you can specify the level of dependencies you want CodeAware to process:

     - **Only First Level Dependencies**—Direct (first-level) dependencies only.

     - **All Transitive Dependencies**—Both direct and transitive dependencies. Transitive dependencies are basically dependencies of dependencies.

     - **No Dependencies**—No dependency processing at all. (Default)

20

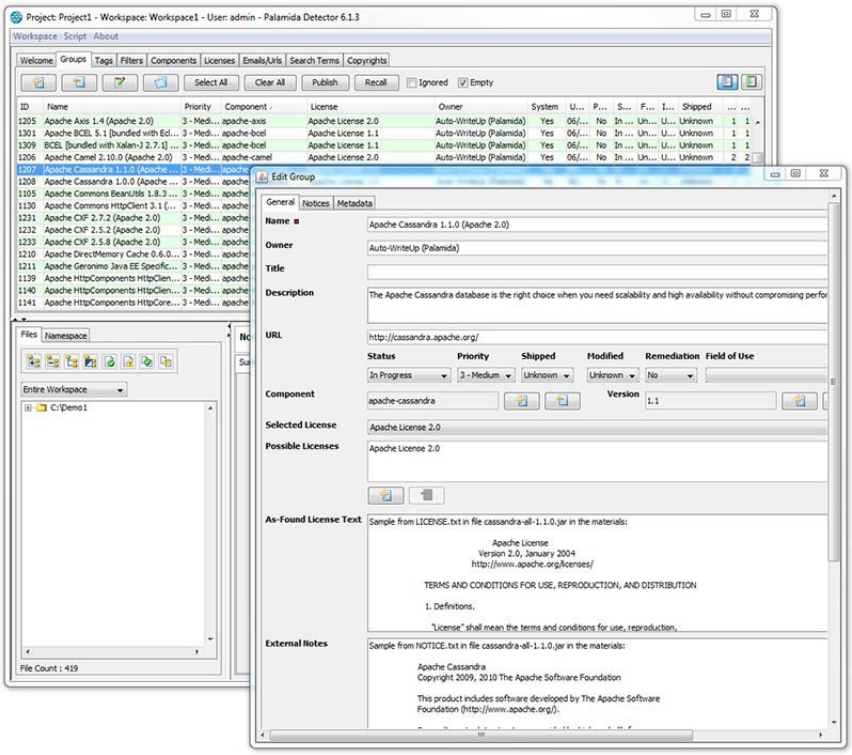
Using Auto-WriteUp™

This section discusses the Auto-WriteUp™ feature in the following topics:

- Auto-WriteUp™
- Configuring Auto-WriteUp™
- Frequently Asked Questions

Auto-WriteUp™

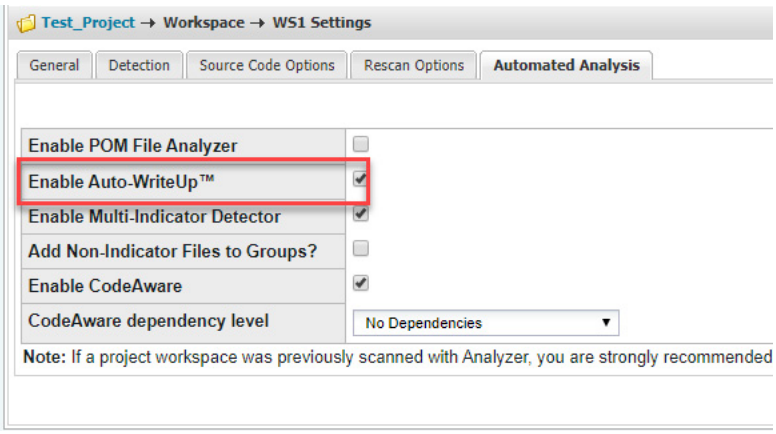
Auto-WriteUp™ is part of the automated analysis technologies distributed with Code Insight to assist the user with the auditing process. Based on a rules engine, Auto-WriteUp™ can automatically identify OSS content in a scanned code base and create populated groups in the Code Insight Detector Client.



After reviewing the results, the auditor can publish the groups to create project inventory.

Enabling Auto-WriteUp™

Auto-WriteUp™ can be enabled or disabled as part of the **Workspace Settings** before scheduling a scan.



Detection Rules

The Auto-WriteUp™ engine is driven by detection rules and updated weekly via the Electronic Update Service.

- Code Insight Rules
- High-Volume Rules

- User Defined Rules

Code Insight Rules

Rules created manually by Code Insight analysts are called Code Insight Rules. These rules can be created for virtually any component which contains unique releases. After a component is thoroughly researched, its rules will contain licensing information, any pertinent auditor notes, and any envelope issues (described in the next section).

High-Volume Rules

A forge is a repository of OSS software, typically a web site or an FTP site. For example, some popular forges include GitHub (http://www.github.com), SourceForge (http://www.sourceforge.com), and GNU (http://ftp.gnu.org/).

High-Volume Rules are created automatically by downloading available releases and associated metadata from a forge. Unlike manually created Code Insight Rules, High-Volume Rules only contain information that is available from the forge, which typically does not include information about envelope issues or auditors notes.

User Defined Rules

Starting with version 6.6.1 of Code Insight, users can define custom detection rules. To create custom Auto-WriteUp™ rules, contact your Code Insight account representative to set up a training session.

Configuring Auto-WriteUp™

Auto-WriteUp™ configuration is stored in the following location on the Code Insight Core Server:

`<code_insight_install_dir>/config/core/Auto-WriteUp.properties`

The following properties are supported:

Name	Value
groupOptions	A comma-separated list of any of the following options (including no options): • DO_VER • DO_ENV • SET_PRI • REF_PARENT_VER
maxHeaders	`<integer>`
useShortText	`true/false`
autoPublish	`ALL_GROUPS/CODE_INSIGHT_GROUPS/COMPLETE_GROUPS`
userRules	A comma separated list of file names or `<no value>`.

Important ▪ *Contact Revenera Support before editing the configuration file. Incorrect values may result in errors and a failed scan.*

Property Definitions

The following are property definitions for Auto-WriteUp's supported properties.

Property	Description	
groupOptions (comma separated list)	**SET_PRI**	Automatically assign priorities to groups created by Auto-WriteUp™. The following priorities are supported based on the license or set of licenses associated with the group: ● **P1**: Known copyleft OSS licenses (such as GPL, LGPL, MPL). ● **P2**: Unknown, proprietary, commercial, or modified licenses. ● **P3**: Known permissive OSS licenses (such as Apache, MIT, BSD).
	DO_VER	Create groups based on the component, version, and license of detected OSS content. When this option is omitted, Auto-WriteUp™ will group files based on the component and license only.
	DO_ENV	Create separate groups for envelope issues. Envelope issues (also known as embedded issues) are separate third- party content inside a parent component. Typically the third- party component is compiled, packaged, or copied inside the parent component (not to be confused with static or dynamic dependency linking). Envelope issues can be especially important if the third- party component is licensed differently than the parent component.
	REF_PARENT_VER	Explicitly reference the version of the parent group when creating envelope groups. This option has no effect if the DO_ENV or the DO_VER options are omitted.

Property	Description	
maxHeaders (integer, default: 3)	The maximum number of headers to use when providing evidence for a specific license text in the As-Found-License-Text field of a group in the Code Insight Detector Client. Typically, a single header refers to a URL or a file in the scanned code base.	
useShortText (true/false, default: true)	Specifies whether to use a short sample of the license text instead of the full license text when populating the As-Found-License-Text field of a group in the Code Insight Detector Client.	
autoPublish (ALL_GROUPS/Code Insight_GROUPS/ COMPLETE_GROUPS, default: COMPLETE_GROUPS)	Specifies which groups will be published if the Auto-Publish System-Detected Inventory is enabled in the Project Details.	
	ALL_GROUPS	Publish all groups created by Auto-WriteUp.
	Code Insight_GROUPS	Publish all groups created by Code Insight Rules.
	COMPLETE_GROUPS	Publish any groups which contain licensing information (all groups created by Code Insight Rules and a portion of groups created by auto-generated High-Volume Rules will contain licensing information).
userRules (comma separated list)	A list of file names indicating which user-specified detection rules to load before scanning. This property may be left blank if no user-specified rules are present. User-specified detection rules are saved in Microsoft® Excel® workbooks to the following directory on the Code Insight Core Server: `<Code_Insight_install_dir>/config/core/rules/`	

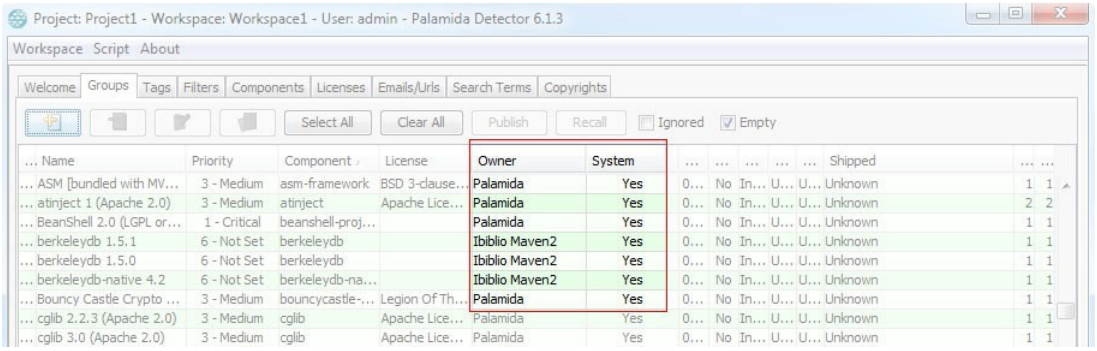
Frequently Asked Questions

The following questions are often asked about Code Insight Auto-WriteUp:

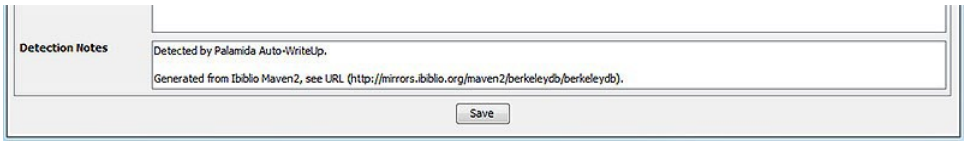
- How are Auto-WriteUp™, Code Insight Rules or High-Volume Rules Groups Marked?
- How can I tell which files were detected by Auto-WriteUp™?
- Why did Auto-WriteUp™ create duplicate groups?
- Why did Auto-WriteUp™ create groups with unpopulated fields?
- Why did Auto-WriteUp™ create groups without versions though I included the DO_VER option in the Auto-WriteUp™ configuration?
- How can I tell if a group contains envelope issues without creating envelope groups?
- After editing an Auto-WriteUp™ group, will I lose the changes if I rescan the workspace?

How are Auto-WriteUp™, Code Insight Rules or High-Volume Rules Groups Marked?

Each group created by Auto-WriteUp™ is marked as a System group. The **Owner** column in the Code Insight Detector Client lists the forge associated with the group. The Code Insight forge indicates a group was created with rules manually verified by Code Insight analysts. Other forges indicate the group was created by auto-generated High-Volume Rules.

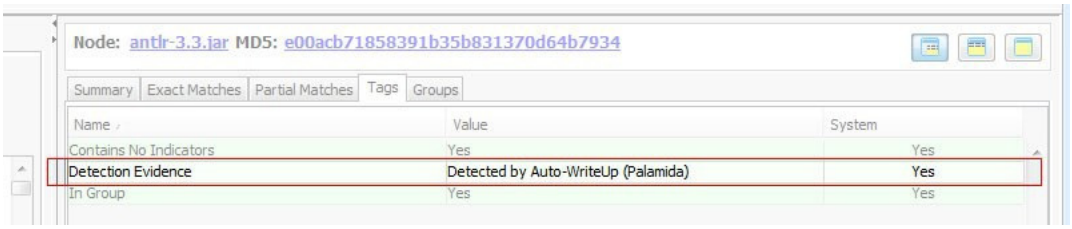


Each group has information in the **Detection Notes** field that describes the rules used to create the group.



How can I tell which files were detected by Auto-WriteUp™?

Files detected by Auto-WriteUp™ have a Detection Evidence tag indicating they were detected by Auto-WriteUp™.



Why did Auto-WriteUp™ create duplicate groups?

Auto-WriteUp™ does not create duplicate groups. However duplicate components may exist on forges. In that case it may appear that Auto-WriteUp™ created duplicate groups.

Why did Auto-WriteUp™ create groups with unpopulated fields?

Groups created from Code Insight Rules will always be filled out. Groups created from auto-generated High-Volume Rules include as much (or as little) information as can be gathered from the forge associated with the group.

Why did Auto-WriteUp™ create groups without versions though I included the DO_VER option in the Auto-WriteUp™ configuration?

Not all rules can be associated with appropriate versions.

How can I tell if a group contains envelope issues without creating envelope groups?

The External Notes field of the group will contain information about any embedded issues.

After editing an Auto-WriteUp™ group, will I lose the changes if I rescan the workspace?

No, Auto-WriteUp™ will not remove any changes to the group as part of a rescan.

Workspace Backup/Restore Scripts

If you plan to use the workspace backup and restore Groovy scripts with Auto-WriteUp™, contact Revenera Support to discuss how they interact.

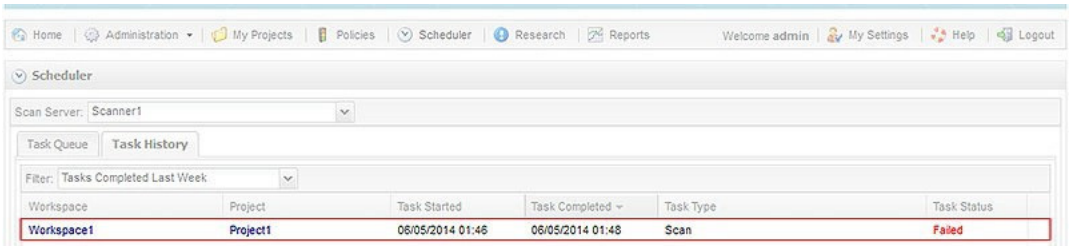
Troubleshooting

Code Insight Auto-WriteUp has the following troubleshooting topics:

- How do I know if an error occurred when Auto-WriteUp™ was running?
- If an error occurs when Auto-WriteUp™ is running, does the entire scan fail?
- Why are changes to the Auto-WriteUp.properties file not reflected after running a new scan?

How do I know if an error occurred when Auto-WriteUp™ was running?

The Scheduler Task History will report the scan as having failed. Auto-WriteUp™ will log all errors to `<Code_Insight_install_dir>/logs/Auto-WriteUp.log.X` files. If Auto-WriteUp™ is unable to start, errors will be logged to the Tomcat `stdout` facility.



If an error occurs when Auto-WriteUp™ is running, does the entire scan fail?

No, although the Scheduler Task History displays the scan status as failed, the scan results were committed before Auto-WriteUp™ was invoked.

Why are changes to the Auto-WriteUp.properties file not reflected after running a new scan?

When Auto-WriteUp™ runs for the first time on any workspace in a project, it saves its configuration to the project metadata. Subsequent rescans on any workspace in that project will read the configuration from the project metadata. It is currently not possible to run Auto-WriteUp™ with the new configuration without deleting all Auto-WriteUp™ groups from all workspaces in that project.

21

Automated Analysis

After a scan, the system runs a set of automated analysis techniques to sort through the scan results and create system groups with associated files, each group representing an automatically-detected inventory item. The system groups can then be promoted to inventory by way of auto-publishing or manually publishing. The following techniques are available:

- POM File Analyzer
- Auto-WriteUp™
- Multi-Indicator Detector (MID)
- CodeAware
- Analyzer

For a comparison of features supported by these techniques, see the later section, Comparison of the Analysis Techniques.

Refer to Automated Analysis Tab for instructions on how to enable one or more of these analysis techniques for scans on project workspaces.

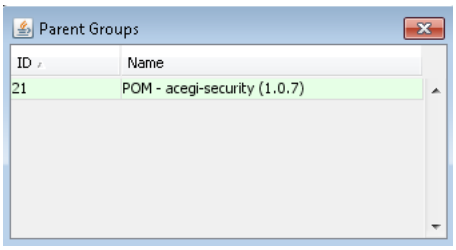
POM File Analyzer

This analysis technique performs dependency checking of Maven's Project Object Model (POM) files. POM files are XML representations of the structure of a Maven project and are contained in a `pom.xml` file. If you enable this option, the scanner automatically creates a group for the artifact for which the POM file exists, as well as for each dependency defined in the POM file. The dependency groups also contain the group id of the parent group in the new **parentIds** field. This data can be used to generate a hierarchical relationship between the top-level projects and their transitive dependencies.

Note ▪ *This analysis technique is disabled by default.*

Task ***To view parent groups that the current group is a dependency of in the POM file, do the following:***

1. Click the **Parents** hyperlink in the **Group Id** field. The Parent Groups dialog appears.



   This dialog displays all parent groups that declare the current group as a dependency in the POM file. If enabled, POM File Analyzer will be applied to files inside archives.

2. When you finish viewing the dependencies, click the **X** to close the dialog.

Auto-WriteUp™

This analysis technique is based on a rule engine that automatically identifies OSS content in the codebase. It creates groups that are populated with group details such as Description, URL and As-Found License Text. If enabled, Auto-WriteUp is applied to files inside archives. In the majority of cases, the Auto-WriteUp data is curated by the data library team and verified for accuracy.

For more information, see Using Auto-WriteUp™.

Note ▪ *Auto-WriteUp is enabled by default.*

Multi-Indicator Detector (MID)

This analysis technique relies on rules such as MD5-matching and filepath-matching to automatically identify OSS content in the codebase. It creates groups with associated files if all the MID rule conditions are satisfied. This analysis technique is enabled by default.

A set of standard MID rules are shipped with Code Insight and are added to the central database as part of the installation process. However, you can create custom MID rules that are unique to each Code Insight Scan Server.

The following sections provide more information:

- More About Custom MID Rules
- MID Rule Attributes

More About Custom MID Rules

You can create custom MID rules that are unique to each Code Insight Scan Server. The custom rules are defined in a Microsoft® Excel® spreadsheet that the system reads (along with the MID rules in the database) each time a workspace is scanned.

The MID rules spreadsheet can be found at the following location on the Code Insight Scan Server:

```
<CODE_INSIGHT_ROOT_DIR>\6.14.x\config\scanEngine\multi_indicator_detector.xls
```

Each row in the MID rule spreadsheet represents a single MID rule. For information about MID rule attributes, see MID Rule Attributes.

In case of a multi-Scan Server installation, you must keep these custom rules synchronized across all servers. The Code Insight Scan Server does not need to be restarted if a change is made to the MID rules spreadsheet.

MID Rule Attributes

Standard and custom MID rules are defined with these attributes.

Component and Version Information for Groups

The following columns are used to notify the Code Insight Scan Server which component and version (if known) is to be selected for the system group:

- **Rule Notes**—The value entered into this column shows up as the detection evidence value for a system-generated group. It is the explanation as to why a particular group was created. This value is optional.
- **Component ID**—The ID of the component to be associated with the system-generated group. You can look up a component id using the Research page of the Code Insight web UI. The component ID must exist in the system for the rule to be evaluated; otherwise the rule is skipped. This value is required.
- **Version ID**—The ID of the component version to be associated with the system-generated group. You can look up a component version ID using the Research page of the Code Insight web UI. The component version ID must exist in the system for the rule to be evaluated; otherwise the rule is skipped. This value is optional.

Rule Condition Criteria

The following data is also used to define MID rules. During a scan, any of these attributes can be used as criteria for MID rule conditions; at least one condition is required for each MID rule:

- **File Path Pattern**—File name or path to be considered for MID rule analysis; if path criteria are not met, the rule is skipped with no further processing (for example: *.dll).
- **MD5 Digest**—The MD5 digest of the file may be used as the most basic criteria, if the file has the following MD5 digest, then it is component Foo, version #.#. If the MD5 criteria are used, the search term and namespace conditions are not evaluated.
- **Search Terms**—All of the specified search terms must exist in the file for this condition to evaluate to true.

- **Java Names**—All of the specified Java names must exist in the file for this condition to evaluate to true. If a single package name is specified, the rule will match a Java source file or class file that belongs to the specified package or package with the specified prefix (for example, **com.foo** will match any package that starts with `com.foo`). It will also match an archive file (`.jar`,`.war`, `.ear`, `.zip`) that contains a file matching the package name. If multiple names are given, the rule will match only an archive file that contains all the specified packages.

CodeAware

CodeAware is basically the "next generation" of automated discovery, providing much of the same analysis functionality as the Analyzer to build groups and auto-generate inventory with relevant details, such as license and vulnerability information. CodeAware detection methods include package, search term, repository, digest, and component analyzers, as well as other intelligent parsing algorithms and live lookups. CodeAware detects various packages and source distributions. You can find details about the files and methods of detection used by CodeAware in the **Detection Notes** group field.

The following sections provide more information about CodeAware:

- CodeAware Requirements
- Recommendation When Running CodeAware
- Supported Ecosystems

Refer to Comparison of the Analysis Techniques for more information on the existing capabilities of CodeAware in comparison to the other automation techniques.

CodeAware Requirements

CodeAware requires outgoing TCP access on port 443 to access known repository sites for license and vulnerability information and other data. (It performs only lookups and never sends files or code snippets.) If using a proxy server, contact the Code Insight or network administrator to ensure that the external URLs required by Code Insight have been added to the proxy server's list of allowed sites.

Recommendation When Running CodeAware

If a workspace was previously scanned with the Analyzer, you are strongly recommended not to enable CodeAware in place of the Analyzer for subsequent scans, as this can result in duplicate groups. If you would like to replace Analyzer results with CodeAware results, you can do so in a new project to avoid duplicates.

Supported Ecosystems

CodeAware provides native support for operating in many development ecosystems The table below provides the following information about each ecosystem (each encompassing a language, package type, and public registry) that CodeAware 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.**—If top-level inventory is supported, the discovery of this component's direct (first-level) dependencies and transitive dependencies (that is, dependencies of dependencies).
- **Notes**—Link to notes (if available) pertaining to Code Insight's support of the specific ecosystem.

Table 21-1 ▪ 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++, 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	—
Go	glide	https://go-search.org	`glide.yaml`	✔	—	—	See Go Ecosystems.
	godep		`godeps.json`	✔	—	—	
	govendor		`vendor.json`	✔	—	—	
	module		`go.mod`	✔	—	—	
Java	Gradle	http://search.maven.org/	`build.gradle`	✔	✔	✔	—
	Maven		`pom.xml`	✔	✔	✔	—
JavaScript	Bower	https://registry.bower.io/packages/	`bower.json`	✔	✔	—	—
			`.bower.json`	✔	✔	—	—
			`package.json`	✔	✔	—	—

Table 21-1 ▪ Supported Ecosystems (cont.)

Language/ File Type	Package	Registry	Manifest File	Top-level Inv.	Direct Dep.	Trans. Dep.	Notes
.NET	NuGet	https://api.nuget.org/v3-flatcontainer/	.nupkg	✔	✔	✔	—
			.nuspec	✔	✔	✔	—
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	✔	✔	—	
RPM	RPM Header	N/A	.rpm	✔	N/A	N/A	—
Ruby	Gem	https://rubygems.org/api/v1/	.gem	✔	✔	—	See Ruby Ecosystems.
			Gemfile	✔	✔	—	
			.gemspec	✔	✔	—	
Swift, Obj-C	CocoaPods	N/A	Podfile.lock	✔	—	—	—
			.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 CodeAware automated analysis process:

- Git Ecosystems
- Go Ecosystems
- NPM Ecosystems
- PyPI Ecosystems
- Ruby Ecosystems
- Yarn Ecosystems
- Yocto Ecosystems

Conda Ecosystems

First level dependencies are supported for `index.json`, but the semver resolution of version is not yet supported.

Git Ecosystems

Code Insight scans configuration files inside `.git` folders encountered in a project codebase and uses identified information to create inventory items.

Go Ecosystems

Note the following for Go ecosystems:

- A golang project configured with a supported package manager must include a license file to enable Code Insight to discover it as top-level inventory.
- Currently, Code Insight supports the discovery of top-level inventory only in scans of pre-build Artifact source code.
- 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.

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`.
- 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` files.

Direct dependencies for the pre-build artifacts are retrieved from the `requirements.txt` file, as long as `PKG-INFO` or `setup.py` reside in the same directory as `requirements.txt`. (`PKG-INFO` or `setup.py` is needed to determine the top-level inventory to which the direct dependencies are associated.) In the absence of `requirements.txt`, the dependencies are reported from `install_requires` section in the `setup.py` file.

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.

Analyzer

Important ▪ *In Code Insight 6.13.3 and later, the Analyzer technique is available only if the administrator has enabled it in your Code Insight installation. Consult the Code Insight administrator for more information.*

The Analyzer technique uses a variety of methods to identify OSS content such as packages, source distributions, and dependencies. It creates groups and, optionally, a Group Builder or Quick Assessment workspace report. Analyzer detection methods include various package, search term, repository, digest and component analyzers with optional data lookups via Data Services (see Data Services). You can find details about the files and methods of detection used by the Analyzer in the **Detection Notes** group field. There are three ways to invoke analyzer:

- **Analyzer as Part of Scan**—As of version 6.13.1, the Analyzer automatically executes at the end of scans as long as it is enabled on the **Automated Analysis** tab for a project workspace (this is the default setting). If invoked in this manner, Analyzer creates groups for identified OSS content but does not generate a report.

- **Group Builder**—You can use the **Schedule Scan/Report** feature to schedule a scan that is followed by the generation of the Group Builder report. This method creates groups for identified OSS content (the same set of groups as created by invoking Analyzer as part of scan) and generates the Palamida Analyzer - Group Builder report. This workspace report also includes a Bill of Materials, Security Vulnerability information, Third-Party Notices information and Codebase Metrics for the given workspace. Note that the report requires at least one successful scan prior to execution of the report.
- **Quick Assessment**—Alternatively, you can use the **Schedule Scan/Report** feature to schedule the Quick Assessment report. This workspace report does not require a scan prior to execution of the report. It provides a quick overview of the codebase contents and includes a Bill of Materials, Security Vulnerability information, Third-Party Notices information and Codebase Metrics.

Data Services

The Analyzer uses Revenera's Data Services (DS) to supplement license information, component metadata, and known security vulnerabilities for the detected groups. Data Services (DS) is a RESTful API hosted on Amazon Web Services. DS provides information about Open Source projects from the Code Insight Data Library. Requests to Data Services take the form of an "HTTPS GET" request to the web service.

Files and code snippets are never sent to Data Services. The web service is only queried for information about Open Source projects as contained in the Code Insight Data Library (for example, component information, vulnerabilities, licenses, digest-to-component data and so on) to enable rich inventory building.

Analyzer Requirements

The Analyzer requires outgoing TCP access on port 443 to access Data Services. Proxy support can be enabled in `analyzer.properties`. (If using a proxy server, contact the Code Insight or network administrator to ensure that the external URLs required by Code Insight have been added to the proxy server's list of allowed sites.)

Comparison of the Analysis Techniques

All the analysis techniques described in this chapter generate *groups*, which are groupings of files identified by the same third-party component, version, and license (although a license might not be available for some dual or multi-licensed groups). Groups can then be automatically or manually *published*—that is, promoted to inventory and made available in Code Insight's Web user interface for further review and inclusion in your product's Bill of Materials.

The following table provides a comparison of other important features (aside from group creation) that these various analysis techniques support:

Table 21-2 ▪ Comparison of Analysis Techniques

	Detects dependencies (direct)	Detects dependencies (transitive)	Processes files inside archives	Populates additional group details	Populates As-Found license text	Supports custom rules	Creates custom versions
POM Analyzer	`pom.xml` only	No	Yes	No	No	No	No
MID Rules	No	No	Yes	No	No	Yes	No
Auto-WriteUp	No	No	Yes	Yes	Yes	Yes	No

Table 21-2 ▪ Comparison of Analysis Techniques (cont.)

	Detects dependencies (direct)	Detects dependencies (transitive)	Processes files inside archives	Populates additional group details	Populates As-Found license text	Supports custom rules	Creates custom versions
Analyzer*	Yes	`pom.xml` only	Yes	Yes	Yes	No	Yes
CodeAware	Yes	Yes	Yes	Yes	Yes	Yes (MID)	Yes

* Available only if enabled in your Code Insight installation.

22

Workspace Scanning & Report Scheduling

Information about performing a workspace scan and scheduling reports is presented in the following sections.

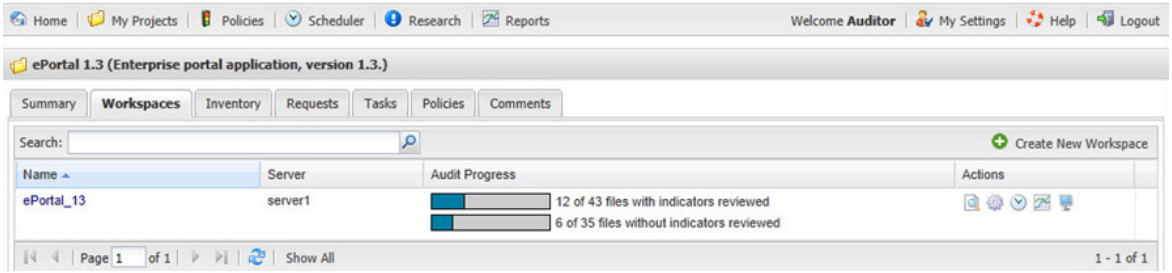
- Scheduling a Scan
- Generating a Report

Scheduling a Scan

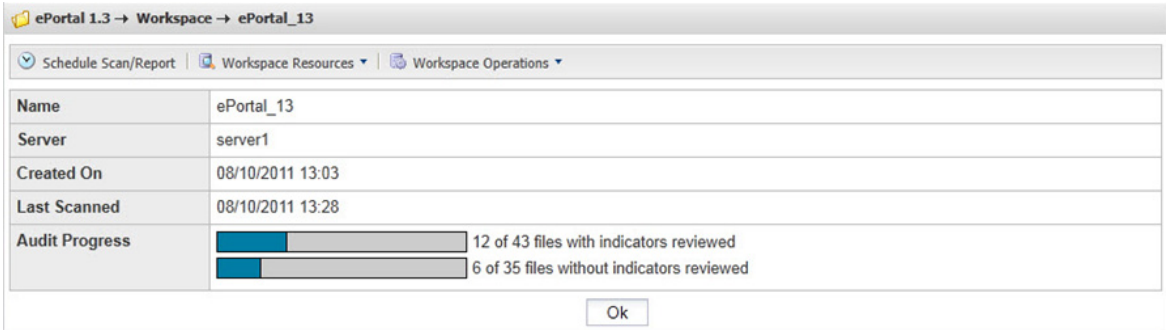
To schedule a scan or generate a report, perform the following steps.

Task ***To schedule a scan:***

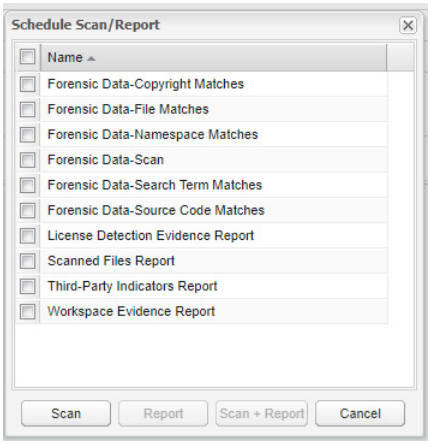
1. Log in to Code Insight as the **Auditor**.
2. Select the **My Projects** tab.
3. Choose a project, and click the project name or the associated **magnifying glass** icon in the Action column.
4. Click the **Workspaces** tab.



5. Click the name of the workspace link in the **Name** column or the associated **magnifying glass** icon in the Action column that you wish to scan. The **Workspace Details** page appears.



6. From the **Workspace Details** page, click **Schedule Scan/Report.** A list of available reports appears.



7. Select the reports that you want to generate after the scan completes, and then click the **Scan + Report** button.

8. To schedule a scan without reports, ensure all Report checkboxes are unchecked, and then click the **Scan** button.

9. If you configured an SCM application for the current workspace, an SCM sync/update/checkout will be performed based on the selected SCM application. If the operation fails, the scan will be terminated.

Note ▪ *If ClearCase is installed, the selected view is started and the selected VOBs are mounted. If the operation fails, the scan is terminated.*

See the *Installation and System Administration Guide* for detailed instruction related to SCM configuration for workspaces and project copy.

Generating a Report

To generate a report, see Generating Reports.

23

Workspace Resources

This section includes the following topics:

- Accessing Workspace Resources
- Editing Workspace Settings
- Launching Detector
- Viewing Tasks in Queue for a Workspace
- Viewing Reports for a Workspace

Accessing Workspace Resources

To access workspace resources, perform the following steps.

Task ***To access workspace resources, do the following:***

1. Log in to Code Insight as the *Auditor*.
2. Select the **My Projects** tab.
3. Choose a project, and click the project name or the associated **magnifying glass** icon in the **Action** column.
4. Select the **Workspaces** tab to open the **Project Workspace** view.
5. Click the **Workspace Resources** button and select one of the following:
   - Editing Workspace Settings
   - Launching Detector (after scheduling a scan)
   - Viewing Tasks in Queue for a Workspace
   - Viewing Reports for a Workspace.

Editing Workspace Settings

To edit a workspace, do the following:

Task ***To edit a workspace, do one of the following:***

- From the **Workspace Resources** pull-down, select **Edit Workspace Settings**.
- Click the **Cog** icon that corresponds to the workspace.

Note ▪ *The Workspace Details setting is the **magnifying glass** icon.*

Launching Detector

To launch Detector, perform the following steps.

Task ***To launch Detector, do one of the following:***

- From the **Workspaces Resources** pull-down, select **Launch Detector Client**.

  Note ▪ *You must schedule a scan before doing this.*

- Click the **Monitor** icon that corresponds to the workspace.

  Note ▪ *The first time that you launch Detector on a new client machine, you will be prompted to trust a JAR file from The Legion of the Bouncy Castle (http://www.bouncycastle.org/). Check the checkbox **Always trust content from this publisher**, then click **Run** to accept. This is an expected event and should only occur once.*

Viewing Tasks in Queue for a Workspace

To view the tasks in the queue, perform the following steps.

Task ***To view tasks in queue for a workspace, do one of the following:***

- From the **Workspaces Resources** pull-down, select **View Tasks in Queue**.
- Click the **Clock** icon that corresponds to the workspace.

Viewing Reports for a Workspace

To view reports, perform the following steps.

Task ***To view reports, do one of the following:***

- From the **Workspaces Resources** pull-down, select **View Reports**.
- Click the **Graph** icon that corresponds to the workspace.

24

Workspace Operations

Workspace operations include the following:

- Accessing Workspace Operations
- Renaming a Workspace
- Copying a Workspace
- Deleting a Workspace

Accessing Workspace Operations

To access workspace operations, perform the following steps.

Task ***To access workspace operations, do the following:***

1. Log in to Code Insight as the *Auditor*.
2. Select the **My Projects** tab.
3. Choose a project, and click the project name or the associated **magnifying glass** icon in the **Action** column.
4. Select the **Workspaces** tab to view the project compliance and vulnerability workspaces.
5. Click the **Workspace Operations** button and from the pull-down, select one of the following:
   - Renaming a Workspace
   - Copying a Workspace
   - Deleting a Workspace

Renaming a Workspace

To rename a workspace, perform the following steps.

Task ***To rename a workspace, do the following:***

1. From the **Workspace Operations** pull-down, select **Rename Workspace**. A dialog box appears, prompting you to enter a new name for the workspace.
2. Type a new name, and click **OK**.

Note ▪ *Only letters, numbers, underscores, and dashes can be used in the workspace name.*

Copying a Workspace

To copy a workspace, perform the following steps.

Task ***To copy a workspace, do the following:***

1. From the **Workspace Resources** pull-down, select **Copy Workspace**. A dialog box appears, prompting you to enter a name for the new workspace.
2. Type a new name for the copied workspace, and click **OK**.

Note ▪ *Only letters, numbers, underscores, and dashes may be used in the workspace name.*

Deleting a Workspace

To delete a workspace, perform the following steps.

Task ***To delete a workspace, do the following:***

1. From the **Workspace Resources** pull-down, select **Delete Workspace**. A dialog box appears, prompting you to confirm that this is the workspace you want to delete.
2. Click **OK** to delete the workspace.

25

Using the Analyzer

Important ▪ *In Code Insight 6.13.3 and later, the Analyzer technique is available only if the administrator has enabled it in your Code Insight installation. Consult your Code Insight administrator for more information.*

This section contains the following topics describing the Analyzer:

- The Analyzer
- Code Insight Data Services
- Frequently Asked Questions

The Analyzer

The Analyzer aids the auditing of third-party code by automatically identifying dependencies, packages, and source distributions. The Analyzer can be run at the following times:

- Before a scan task to get an overview of the contents of your codebase.
- After a scan to create component groups/inventory items.

The Analyzer provides the following functionality:

- Builds notices for groups/inventory items where possible to make creating third-party notices easier.
- Support for Composer (php) packages. If there is a composer.lock file, those defined dependencies and versions will be used. If there is no lockfile (for example, if the package has not been installed), composer.json files are parsed for uninstalled 'required' dependencies and the versions are set to the most recent version in the Packagist registry that satisfies the given composer.json's semantic versioning restrictions.
- Support for npm uninstalled node modules. Inventory items are created when there is a package.json with dependencies even when the corresponding node modules are not present. The npm Analyzer uses the npm registry to get package information and resolve semantic versioning. Versions are set to the most recent version in the npm registry that satisfies the semantic versioning restrictions in the given package.json file.

- Support for Tomcat webapps rebranding. There are webapps named codeInsightScanner and palamidaScanEngine. The update jar can be used in Code Insight versions 6.10.3+.

Requirements

The following are requirements to run the Analyzer:

- The Analyzer must be enabled on the **Automated Analysis** tab. See Automated Analysis Tab for details.
- A Code Insight Data Services-enabled key is required to enable the Analyzer to query Code Insight Data Services' open source data library using HTTPS GET requests. If you require a new key, contact Revenera Support.

Reports

Workspace reports include the following:

- **Analyzer--Quick Assessment** creates an html report of Analyzer findings.
- **Analyzer--Group Builder** creates an html report and creates groups in your project.

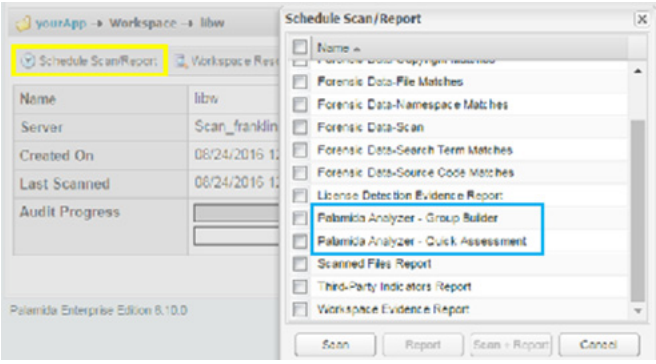
Permissions

The project owner and project auditor roles can run the Analyzer workspace tasks and view the resulting reports.

Running the Analyzer

Task ***To run the Analyzer, do the following:***

1. Click the **Schedule/Scan Report** button in the Workspace.



2. To run the Quick Assessment, run as a REPORT only (not in conjunction with a scan).

Note ▪ *The Analyzer will not do an SCM pull before analysis, though. SCM pulls are only done at scan time or can be triggered manually.*

The Group Builder should only be run after a scan. You can select this task at the same time you kick off a scan or wait until the scan is complete. Just as with a scan, it is recommended to close all Detector client windows before running.

Code Insight Data Services

Code Insight Data Services is a RESTful API hosted on Amazon Web Services. Data Services provides information about Open Source projects from the Code Insight Data Library. Queries can be made against the following entities:

- **Components**: open source projects and their metadata (URL, description, licenses, etc)
- **Versions**: the version names of components, vulnerabilities, per-version licenses, etc.
- **Licenses**: text, whether it is copyleft or compatible with versions of the GPL, etc.)
- **Vulnerabilities**: CVE, CPE info, scores, etc.

  Note ▪ *Code Insight supports only the CVSS v2 scoring system when reporting security vulnerability details.*

- **Digests**: exact md5 digest matches, auto-writeup rules, etc.

Requests to Data Services take the form of an HTTPS GET request to the web service. The following typical query is a request for the names of the licenses associated to Code Insight components with the name ‘zlib’:

```
/components?fields=component_license_names&query=name:zlib
```

The Analyzer is the first application built to use DATA SERVICES. The Analyzer does a combination of file parsing, built-in component analysis, intelligent reading of typical metadata sources (package.json, pom.xml, etc.), and queries to DATA SERVICES and other web services to create an open source inventory. DATA SERVICES supplements license information, component metadata, and known security vulnerabilities.

Important ▪ *Files and code snippets are never sent to DATA SERVICES, which is queried only for information about Open Source projects as contained in the Code Insight Data Library (e.g., component information, vulnerabilities, licenses, digest-to-component data) to enable rich inventory building.*

Ports

The Analyzer requires outgoing TCP access on port 443. Proxy support can be enabled in `analyzer.properties`.

Setup

Ensure you have a Data Services-enabled key file (that is, `palamida.key`), which is all that is required to use the Analyzer. Restart Tomcat if you are replacing an existing key. For an example of additional properties that may be configured, see `<CODE_INSIGHT_ROOT>/config/core/analyzer.properties.example`.

1. Copy the example file to `<CODE_INSIGHT_ROOT>/config/core/analyzer.properties`.
2. Restart Tomcat to try the configuration.

Performing a Quick Assessment with the Analyzer

You can use the Analyzer to perform a quick assessment of your project. The analyzer Quick assessment report can be run on any project in a workspace. The workspace does not need to be scanned to run the report.

Task ***To perform a quick assessment, do the following:***

1. Sign into Code Insight.
2. Select **My project**.
3. Select **My workspace**.
4. Click **Schedule Scan/Report**.
5. Click **Analyzer – Quick Assessment** from the menu.
6. Click **Report** to generate a report based on the unscanned workspace.
7. To access the report, select **View Reports** from the **Workspace Resources** menu. A list of generated reports appears.
8. Select the **Analyzer – Quick Assessment** report from the list.

Auto-creating Component Groups with the Analyzer

The analyzer Group builder is a task that runs on a project workspace during or after a scan task. This task can be run on any workspace. The task runs on the following:

- The scan results produced by the scanner.
- File contents for the scanned files of interest.

Task ***To auto-create component groups, do the following:***

1. Sign into Code Insight.
2. Select **My project**.
3. Select **My workspace**.
4. Click **Schedule Scan/Report**.

5. Click **Analyzer – Group Builder** from the menu.
6. Click **Report** to generate a report based on the workspace.
7. To access the report, select **View Reports** from the **Workspace Resources** menu. A list of generated reports appears.
8. Select the **Analyzer – Group Builder** report from the list.

Administration

The Analyzer requires a new key file (`palamida.key`) to enable access to Code Insight Data Services. If you use your old key, the workspace reports will fail. If your Code Insight server does not have internet access, you can turn off Data Services in analyzer.properties to apply limited analysis techniques.

The feature is enabled by default for all projects, but it can be globally disabled or enabled for specific projects. Contact Revenera Support to enable admin mode to turn off the Analyzer selectively.

The feature can be configured to run automatically with each scan without selecting the **Analyzer - Group Builder** report. Contact Revenera Support for instructions.

Updates

When the report task runs on the Scan Server, it automatically downloads and installs the update jar from the Core Server. The Electronic Update includes the latest Analyzer update jar.

Settings

System settings are controlled via a properties file on the Core Server in `config/core/analyzer.properties`. You can use the example file `config/core/analyzer.properties.example` to view the possible options and to create your own properties file.

Core Settings

Analyzer Use Data Services

Determines whether to use the FlexeNet Code Insight Data Services API for additional analysis techniques. The default is *true*.

Note ▪ *This requires a license key with Data Services enabled, and outgoing TCP access on port 443.*

Analyzer Use RubyGems API

Determines whether to use the RubyGems public API for additional analysis.

Note ▪ *This requires outgoing TCP access on port 443.*

Project Settings

Project settings are controlled via *Yes/No* Metadata fields. You can create these Metadata fields by running the configure`PalamidaAnalyzer.groovy` script, as above. The default values are controlled via `analyzer.properties`.

Analyzer Auto Schedule

Whether to auto-schedule to the report task for enabled projects. The default is *true*. To change the default, set `autoScheduleTask` in `analyzer.properties`.

Analyzer Auto Include

Whether to select "Include in Third-Party Notices" for the groups that the Analyzer creates. Groups will only be included if they have license text. The default is *true*. To change the default, set `autoIncludeNotices` in `analyzer.properties`.

Analyzer Override MID rules

Whether the Analyzer should take files out of MID rule groups with the same component, and delete the groups if they are empty afterwards. The default is *false*. To change the default, set `overrideMidrules` in `analyzer.properties`.

Analyzer Resolve Transitive Dependencies

Whether to include transitive dependencies. Currently this only affects the POM Analysis. The default is *true*. To change the default, set `transitiveDependencies` in `analyzer.properties`.

Frequently Asked Questions

The following are questions that are often asked by Code Insight users:

- How do I know if a group was created by the Analyzer?
- How do I know if a file has been analyzed by the Analyzer?
- How do I know what information the Analyzer used to create the group?
- Do I still have to audit and where should I start?
- Why do some groups have Unknown in the group name?
- What if the Analyzer finds a package that doesn't have a component?
- What happens to the Analyzer groups when you rescan a workspace?
- Does the Analyzer find vulnerabilities? How?
- Does the Analyzer do License Scanning/Copyright Detection/Source Code Fingerprint Analysis?
- Does the Analyzer mark files as reviewed?
- Why are my own Java modules or dependencies listed as inventory items?

How do I know if a group was created by the Analyzer?

Analyzer or *Component Analyzer* will appear in the group's **Owner** field. The group will also have Detection Notes that say “Detected by Analyzer” along with some information about the analysis techniques used.

How do I know if a file has been analyzed by the Analyzer?

Each file that the Analyzer places into a group receives a special tag of *Detection Evidence* with the value *Detected by Analyzer*. This tag can also be used in filters.

How do I know what information the Analyzer used to create the group?

Take a look at the group's detection evidence to see what analysis techniques and data sources the Analyzer used.

Do I still have to audit and where should I start?

The Analyzer is not intended to be a replacement for a good forensic open source audit. As always, you should decide what the appropriate depth of audit is for each of your code projects. In most cases, at least some additional audit work is highly recommend. Some good starting places would be to:

- Identify groups with unknown or unexpected licenses and use the FlexeNet Code Insight scan results to identify/verify the license.
- Check for “envelope” issues (cases where a component declares a permissive license but contains code from or a dependency on a component under an unacceptable license). It is a good idea to check high priority scan evidence such as search terms indicative of a copyleft license.
- Mark files associated to Analyzer groups as *Reviewed*; and then proceed with your normal audit process to explain remaining instances of open source licenses, third party copyrights, and exact digest matches. You can easily find all the files within Analyzer groups by creating a Tag Value Expression filter on Detection Evidence which contains *Analyzer*.

Why do some groups have *Unknown* in the group name?

The Analyzer uses a variety of techniques to try to determine the license of the inventory item. If none of the techniques pan out, the inventory item is marked *Unknown License*. For these items, the usual audit techniques should be applied in the Detector client to ascertain the license (if there is one).

What if the Analyzer finds a package that doesn't have a component?

The Analyzer will try a variety of techniques to find a component match for the group, but if it does not that field and the version field will be left blank. If you would like to add a component to the Data Library, contact Library-Request@Flexeerasoftware.com.

What happens to the Analyzer groups when you rescan a workspace?

The Analyzer will not delete any groups from the previous scan. New groups will be created as appropriate. You can sort the Groups list in Detector to see which groups were recently created/updated. Consider deleting the old groups unless you have made changes to them.

Does the Analyzer find vulnerabilities? How?

The Analyzer is not a vulnerability scanner in the sense of software like Fortify. Rather, the Analyzer is able to identify Open Source projects in your code and use the FlexeNet Code Insight Data Library to see whether any of the projects have been mapped to known/published security vulnerabilities by the Library team. These curated mappings help reduce false positives that often result from pure text-based CVE lookups.

Code Insight supports only the CVSS v2 scoring system when reporting security vulnerability details.

Does the Analyzer do License Scanning/Copyright Detection/Source Code Fingerprint Analysis?

No, the Analyzer does not do systematic license detection. However, the Analyzer does use the license scanning data in the Group Building mode on some occasions and uses a variety of techniques to get license information, but it is not a replacement for the FlexeNet Code Insight scanner and a good audit. In general, the Analyzer should be used in conjunction with the standard auditing methods for the discovery of open source licenses in the Detector client.

Does the Analyzer mark files as reviewed?

No, files should be marked as reviewed by an auditor when the depth of audit appropriate for the workspace has been completed. For some types of audit, it might be appropriate to mark the files as reviewed after a limited review for high-priority evidence types or envelope issues. For a more forensic audit, it might be appropriate to wait until more evidence has been evaluated (such as source code fingerprints).

Why are my own Java modules or dependencies listed as inventory items?

Some customers find it useful to visualize and understand their project hierarchy by placing the dependencies in the context of the module they are part of. If you do not find these items to be useful, delete the groups in Detector or do not publish them as inventory items.

26

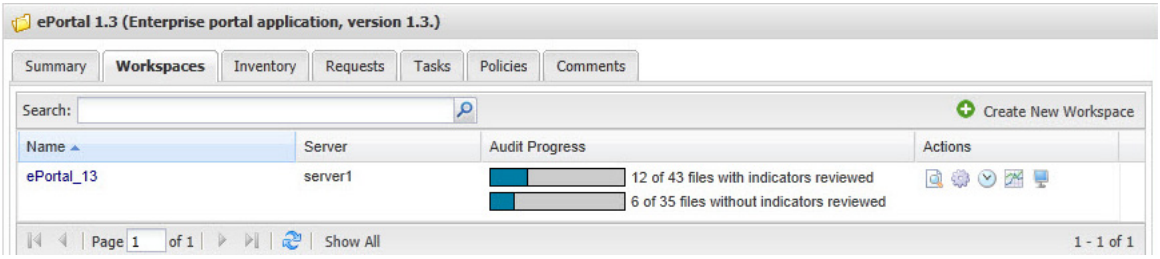
Accessing a Workspace to Analyze

This chapter describes how to access workspaces to analyze.

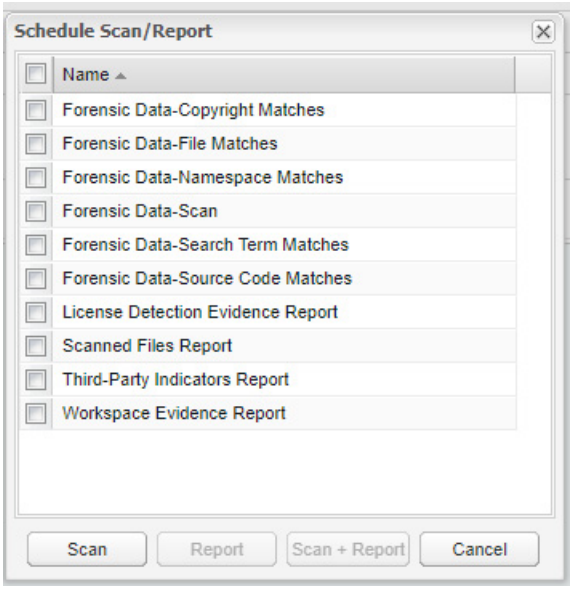
To schedule a scan and later analyze that scan, you need first to choose the workspace for which you want to analyze scan results.

Task ***To choose a workspace, do the following:***

1. Click the **My Projects** button.
2. Click the name of the project.
3. Click the **Workspace** tab.
4. Select a workspace to scan by clicking on the workspace link in the **Name** column.



5. Click the **Schedule Scan/Report** button. The **Schedule/Scan Report** dialog appears with a list of available reports to generate.



6. Click a specific report checkbox. You have the option to generate a report only or a scan *and* a report.

27

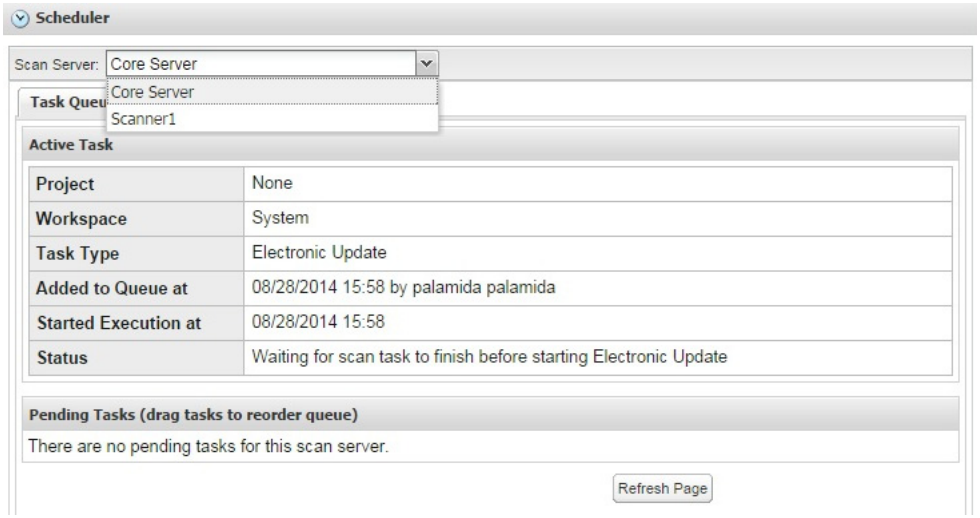
Viewing and Updating Scheduler Queuing

This chapter describes how to view the scheduler queue.

To view the scheduler queue, perform the following steps.

Task ***To view the scheduler queue, do the following:***

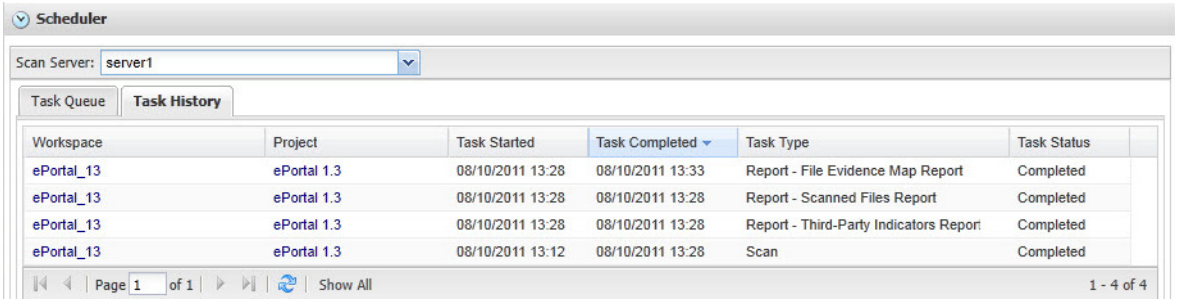
1. Click the **Scheduler** button in the **Main** menu.
2. Select a Scan Server from the Scan Server pull-down.



3. You can schedule as many scans as you wish. To prioritize a certain scan, hover over the boxed-in number in the right column and drag and drop the scan to a different position.
4. To move a scan up or down in completion priority, select the scan or report numbers and drag it to the spot you wish it to occupy in the list. You can also cancel a scan by clicking the red X icon. This feature allows you to maximize your time auditing because you don't have to wait for a scan to finish, just reorder the scan priorities.

5. The queue list automatically renumbers as you drag and drop the task into position. You can also click **Refresh Page** to refresh the page.

6. Select the **Task History** tab to view the time a task started and completed, as well as the task type and status.



28

Launching Detector & Viewing Scan Results

This section contains the following topics:

Opening Detector to View Scan Results

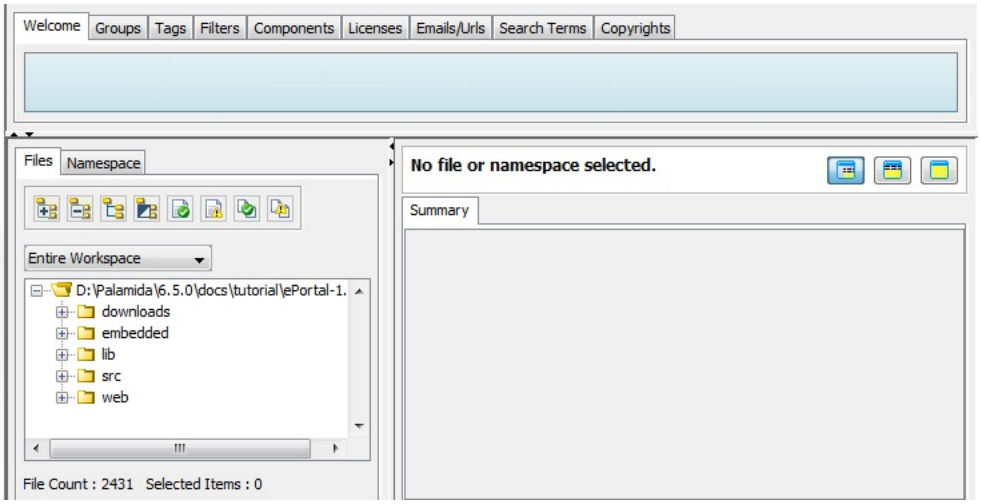
To open a workspace scan and analyze the code base before viewing the scan results, you must schedule a scan.

Note ▪ *For more information on scheduling a scan, see Scheduling a Scan.*

To view scan results in **Detector**, perform the following steps.

Task ***To open Detector to view scan results, do the following:***

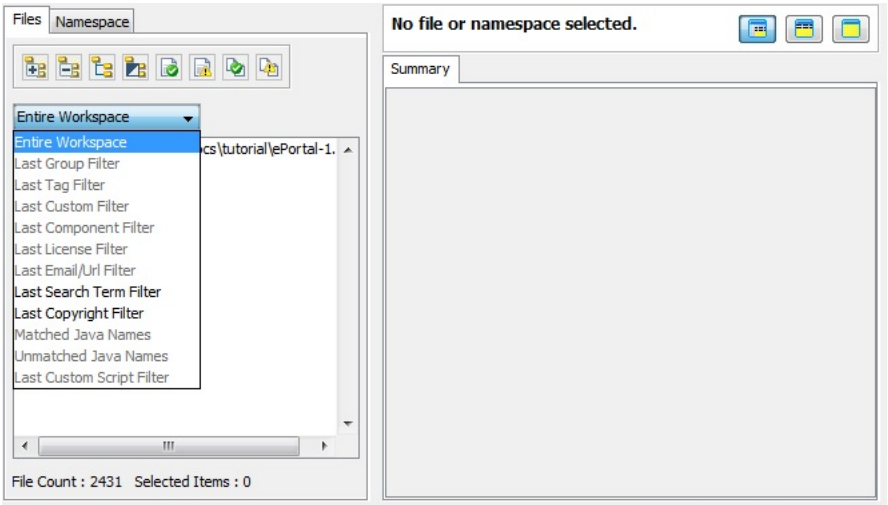
1. From Code Insight, click the **Monitor** icon associated with the workspace, after you have scheduled, and the system has completed, a scan.
2. The workspace opens automatically when you launch **Detector**.



Detector Tabs and File Tree Views

Clicking on the tabs in the upper pane of Detector allow you to access summary data related to last scan results. This includes project inventory and file counts for various third-party indicators.

Click the Plus (+) icon to the left of the folder icon in the lower pane **Files** tab file tree to expand the tree.



There are several tree manipulation buttons above the file tree.

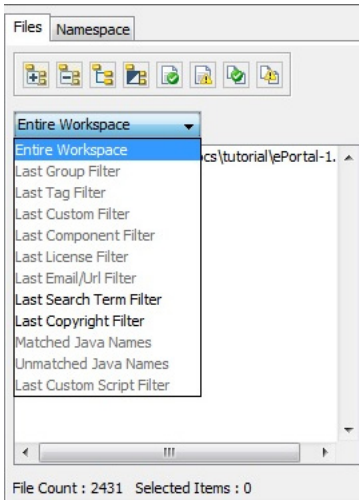
Table 28-1 ▪ Tree Manipulation Buttons

Button	Description
	Expands the tree hierarchy.
	Collapses the file tree.

Table 28-1 ▪ Tree Manipulation Buttons (cont.)

Button	Description
	Toggles between hierarchy and flat modes.
	Allows you to view the inverse tree in hierarchy and flat mode.
	Shows only reviewed files.
	Shows only unreviewed files.
	If the tree display is in flat mode, this button sorts the files in an ascending order.
	If the tree display is in flat mode, this button sorts the files in a descending order.

The pull-down menu above the file tree allows you to apply a filter from recent history.



29

Viewing Evidence

Information about viewing different types of evidence is presented in the following sections:

- Viewing Evidence
- Types of Evidence
- Filtering the File Tree
- Viewing SCM File Check-In History

Viewing Evidence

After you open a workspace, you can view the tabs in the upper panel of Detector. They are named according to the types of information you can access via the scan.

Task ***To view different types of evidence, do the following:***

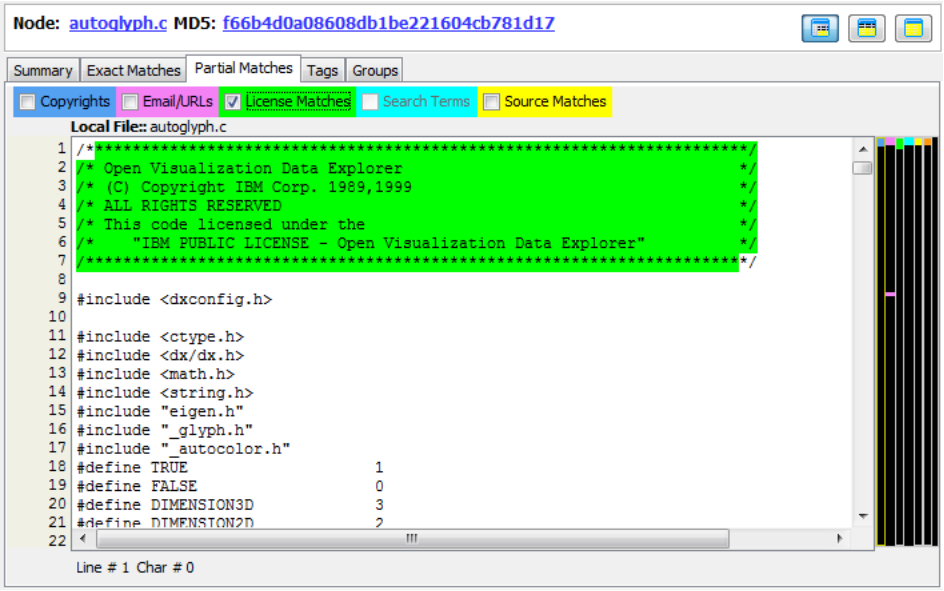
1. Double-click a row in any table within the top tabs; this action filters the file tree to only those files corresponding to the data in the selected table row.

2. Select a row in any table within the top tabs; and then click the **More Information** button ( ) to view more information about the current row in a popup window.

3. Click the **Add** button ( ) to add a new data element or click the **Delete** button ( ) to delete the currently selected element. The lower right panel in **Detector** contains detailed information about the selected node in the file tree.

   Note ▪ *These buttons are enabled only if additions or deletions are permitted.*

   The header above the lower-right panel contains summary information about the currently selected node in the file tree and hyperlinks to a search engine for the file name and its MD5 checksum.



Types of Evidence

There are several types of third-party indicators available as part of the scan results. These include exact and partial file matches, copyrights, emails and/or URLs, license matches, source code matches, and custom search term matches. You may also choose to ignore copyrights, emails, and URLs if in the Properties file you define the elements you wish the system to ignore. Selecting a node in the file tree shows the file details in the bottom-right panel in Detector. The display consists of a node header as shown in Types of Evidence, and the following tabs (if appropriate):

- Summary
- Exact Matches
- Partial Matches
- Ignoring Copyrights, Emails, and URLs
- Tags
- Groups

Summary

This tab contains file-level details about the currently selected node in the file tree.

Exact Matches

This tab contains a list of components (including policy data if available) which holds a file, which is an exact (bit-for-bit) match with selected node in the file tree.

Partial Matches

This tab contains various partial match details for the selected node in the file tree as shown in the four numbered sections in Tags.

- A set of checkboxes runs across the screen horizontally. These highlight various partial-match evidence examples in the selected node in the file tree, as do the following buttons.
- For Source Matches, a table with match details allows you to select which matches should be highlighted in the selected node in the file tree.
- This section contains the content of the selected node in the file tree on top of which highlights are shown.
- A set of color side bars corresponding to the evidence type checkboxes show the location of the highlights for each evidence type in the selected node in the file tree.

Ignoring Copyrights, Emails, and URLs

In order to ignore copyrights, emails, and URLs, you must set up the strings you wish to ignore in the following properties files: `ignoredCopyrights.txt` and `ignoredEmailURL.txt`. Both files are located in `<CODE_INSIGHT_ROOT_DIR>/6.14.x/config/scanEngine/`. In effect, the system will not flag those elements you set it to ignore. This file may be used to force the scanner to skip a detected email/URL.

- The purpose of getting Detector to ignore one of these elements is so you can treat a file as if it does not have third-party content if it only has an internal email or URL within a String literal or comment.
- Lines beginning with # and blank lines are ignored.
- Each line is treated as a string to match. (There is no regex supported).
- An email or URL is ignored if it contains any of the strings from this file.
- The comparison is case-insensitive.
- Avoid use of strings that are over 100 characters in length and those that have hyphens in the name.
- To ignore a set of URLs with a similar pattern, it is recommended that you generalize the notation when possible. For example, to ignore the following five URLs you should use the pattern `://connect.palamida.com` instead of listing each URL individually.

  `http://connect.palamida.com/EnterpriseEdition/release/details/304958/problem-with-treeview-doubleclick-event`

  `http://connect.palamida.com/EnterpriseEdition/release/details/367247/elementhost-set-child-doesnt-stop-listen`

  `https://connect.palamida.com/EnterpriseEdition/release/details/`

  `https://connect.palamida.com/EnterpriseEdition/release/details/694400`

  `https://connect.palamida.com/EnterpriseEdition/release/details/719443/c-chrono-headers-high-resolution-clock-d`

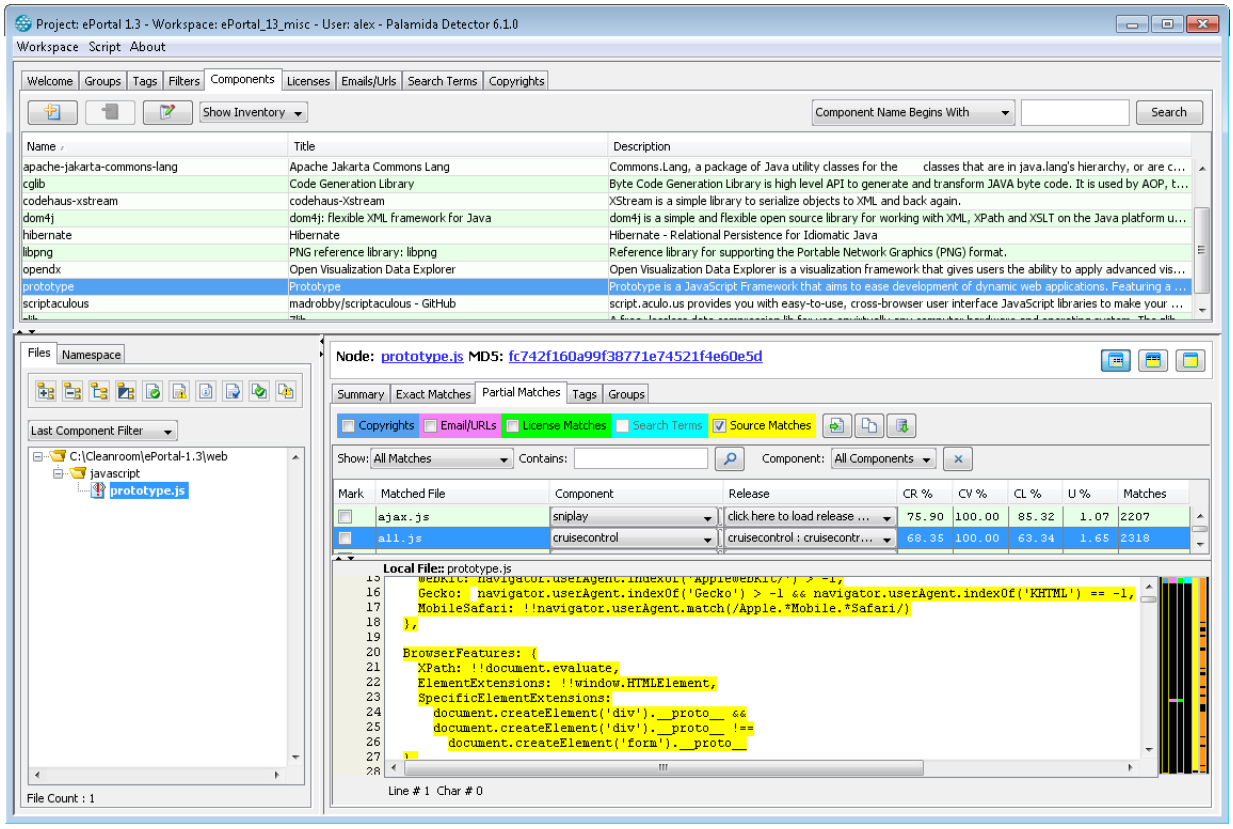
- Likewise, to ignore a set of emails such as those listed below you should use the pattern `@palamida.com` instead of listing each entry individually.

  `alex@palamida.com`

  `julia@palamida.com`

  `devnull@palamida.com`

Note ▪ *Although files that only contain ignored emails/URLs are not tagged as containing these, if you look at the file in Detector you see the supposedly ignored emails/URLs highlighted. This is because the highlighting function in Detector uses its own search rather than basing itself on scan results.*



Tags

This tab contains a list of tags applied to the currently selected node in the file tree.

Groups

This tab contains a list of groups to which the currently selected node in the file tree has been added.

Filtering the File Tree

Information about filtering the file tree is presented in the following sections:

- Displaying Only Desired Files (Nodes)
- Displaying Only Exact Matches
- Displaying Partial Matches

Displaying Only Desired Files (Nodes)

To filter the file tree to display only those files containing a specific type of evidence, perform the following steps.

Task ***To filter the file tree to display only desired files (nodes), do the following:***

1. Double-click a desired row in the **Tag** table on the **Tags** tab in **Detector**.
2. Other filter options include double-clicking on a desired Component, License, Copyright, or Search Terms from the corresponding tabs in the top panel of **Detector**.

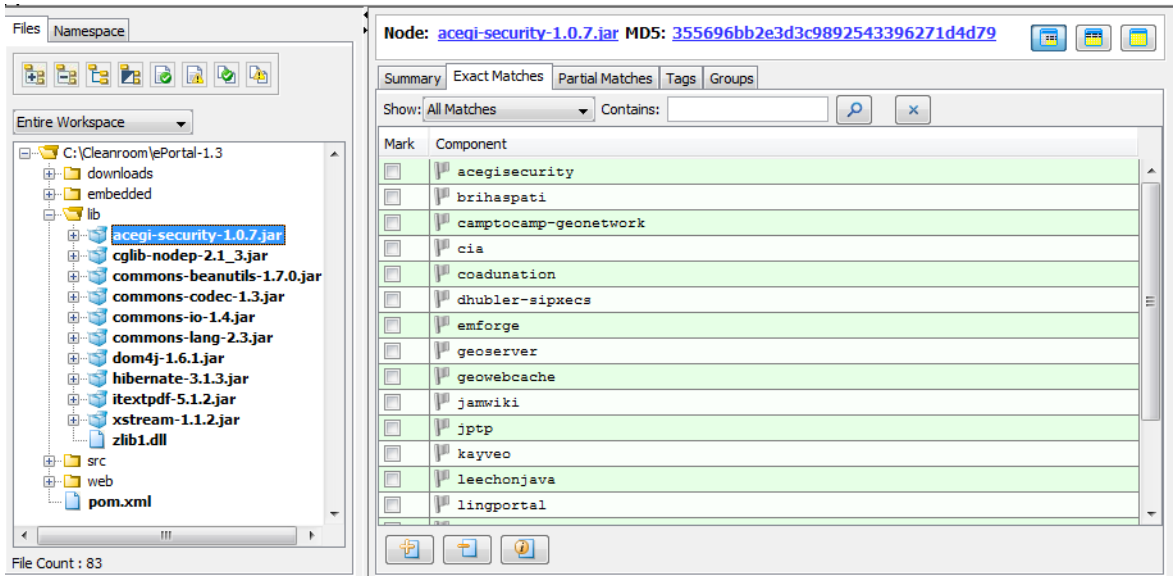
Note ▪ *See Custom Script-Based Filters for more information about Custom Groovy script-based filters you can develop for custom or complex filtering.*

Displaying Only Exact Matches

To filter the file tree to display only exact matches, perform the following steps.

Task ***To filter the file tree to display only exact matches, do the following:***

1. Click the node in the file tree for which you wish to review exact file matches.
2. Select the **Exact Matches** tab in the **File Details** panel. This shows a list of components that include files that are an exact (bit-for-bit) match to the currently selected node in the file tree. Policy icons indicate the component policies that exist for each row.
3. There are three action buttons below the table that allow you to remove a component from the list, add a component to the list, or ignore a particular match from the list.



Displaying Partial Matches

Information about filtering the file tree to display partial matches is presented in the following sections:

- Displaying Copyrights
- Displaying Emails & URLs
- Displaying License Matches
- Displaying Search Term Matches
- Displaying Source Code Fingerprint (SCF) Matches
- Displaying Source Code Fingerprint (SCF) Matches
- Displaying Java Name Matches

Note ▪ *For more information on how to control the sensitivity of copyright detection, see "Tuning Copyright Detection" in the Code Insight Installation and System Administration Guide.*

Displaying Copyrights

To display copyrights, perform the following steps.

Task ***To display copyrights, do the following:***

1. A list of detected copyright holders is available in the **Copyrights** tab in the top panel in Detector.
2. Click the node in the file tree for which you wish to review detected copyrights.
3. Select the **Partial Matches** tab in the file details panel and ensure that the **Copyrights** checkbox is checked to view the detected copyrights. Copyrights are highlighted in blue, and the corresponding vertical color bar to the right of the panel indicates matches which exist within the currently selected node in the file tree.

Displaying Emails & URLs

To display emails and URLs, perform the following steps:

Task ***To display emails and URLs, do the following:***

1. Click the node in the file tree for which you wish to review detected email addresses and URLs.
2. Click the **Partial Matches** tab in the file details panel and ensure that the **Emails/URLs** checkbox is checked in order to view the detected email addresses and URLs. Email addresses and URLs are highlighted in purple, and the corresponding vertical color bar to the right of the panel indicates the matches exist within the currently selected node in the file tree.

Ignoring Emails & URLs

You can choose to ignore emails and URLS. To do this, include the email or URL you wish to ignore in the Properties file the same way you include copyrights to ignore.

Displaying License Matches

To display license matches, perform the following steps.

Task ***To display license matches, do the following:***

1. Click the node in the file tree for which you wish to review detected license text and license references. Note that license matches are only highlighted in source files. Other file types are flagged (via tags) as containing license matches bug highlights within the partial matches tab, and are not available for non-source files.
2. Click the Partial Matches tab in the file details panel and ensure that the License Matches checkbox is checked to view the detected license text and license references. License matches (in source files only) are highlighted in green, and the corresponding vertical color bar to the right of the panel indicates matches which exist within the currently selected node in the file tree.

Displaying Search Term Matches

To display search term matches, perform the following steps.

Task ***To display search term matches, do the following:***

1. Click the node in the file tree for which you want to review search terms.
2. Select the **Partial Matches** tab in the **File Details** panel and ensure that the **Search Terms** checkbox is checked to view the detected search terms. Search terms are highlighted in light blue, and the corresponding vertical color bar to the right of the panel indicates matches that exist in the currently selected node in the file tree.

Displaying Source Code Fingerprint (SCF) Matches

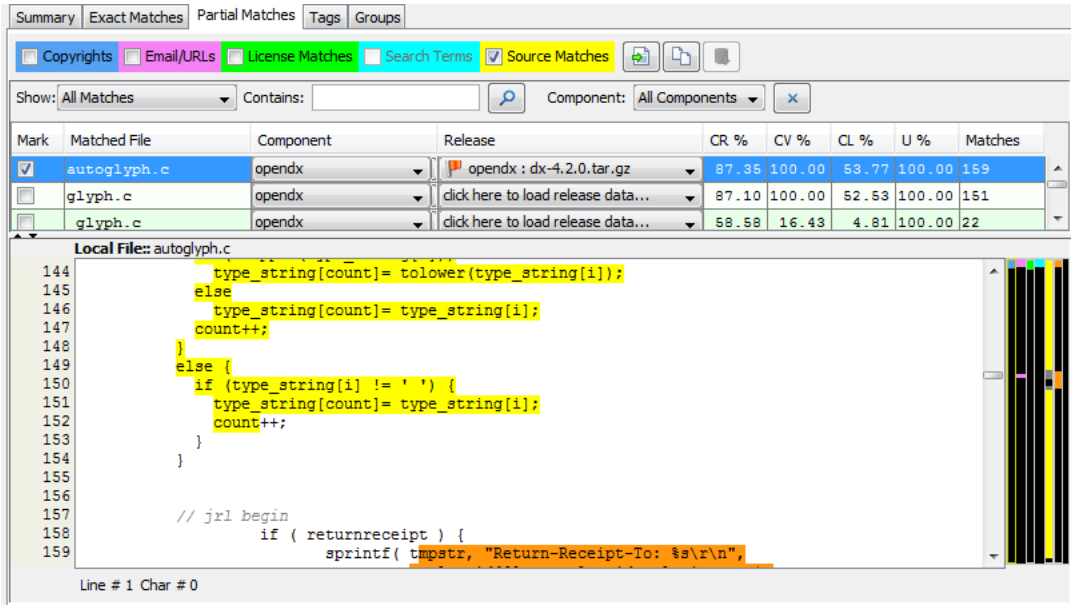
Source code fingerprints are snippets within files that match content in source files found in third-party components. Source code fingerprints act as identifiers of likely third-party content within the scanned file.

To display source code fingerprint (SCF) matches, perform the following steps.

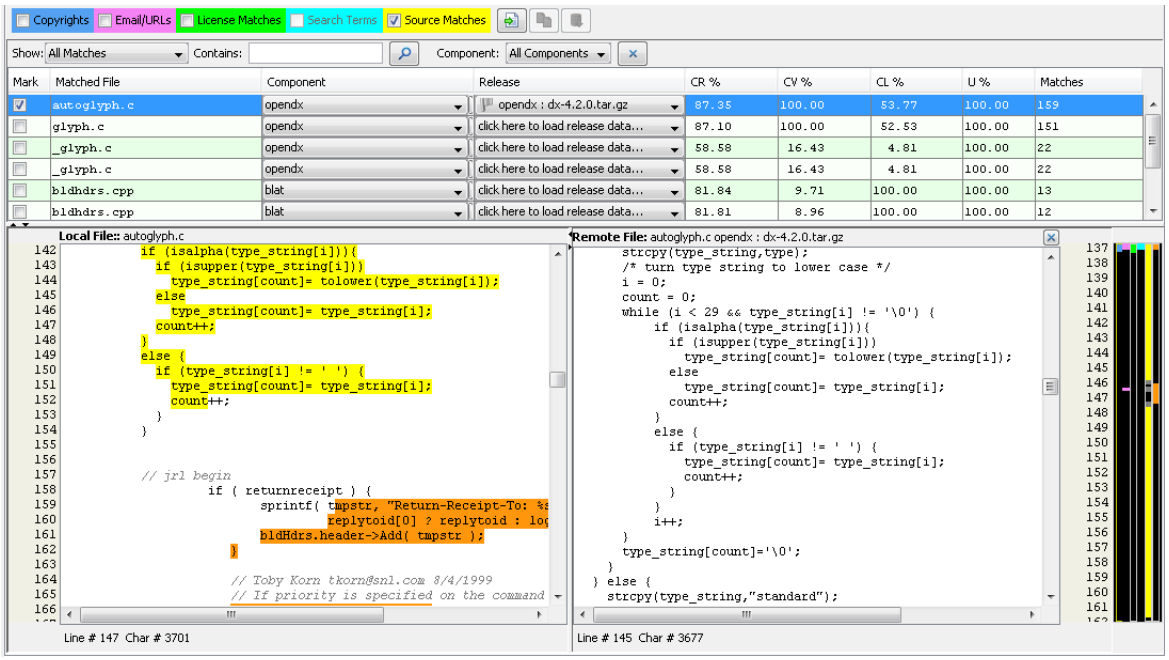
Task ***To display SCF matches, do the following:***

1. Click the node in the file tree for which you wish to review source code matches.
2. Select the **Partial Matches** tab in the file details panel and ensure that the **SCF Matches** checkbox is checked to view a list of third-party files for which there are source matches.
3. Select a row in the matches table to highlight the currently selected node in the file tree with the source code matches that were detected from the selected remote file. The data in the table can be sorted by any of the columns. The following acronyms are used in the column headings to save space:
   - **CR**: Code Rank
   - **CV**: Coverage
   - **CL**: Clustering
   - **U**: Uniqueness
4. Source code matches are highlighted in yellow, and the corresponding vertical color bar to the right of the panel indicates matches which exist within the currently selected node in the file tree.

   Note ▪ *If you set sliders in the **Workspace Settings** page via Detector, only your local (on your machine and with your machine login only) scan results view will be affected. The sliders filter the file tree to display only those files that contain source matches above the defined thresholds. The list of source matches in the **Partial Matches** tab will not be filtered, so no scan results are hidden for a given file. When a file is selected from a complete scanned files tree of a filtered view, all source matches above the scan-source-match thresholds are shown in the source matches table.*
5. There are three buttons to the right of the Evidence Type checkboxes at the top of the **Partial Matches** tab.
6. To open the remote file from the selected Release (Component) in the source matches table in a local test editor, click  .
7. To open the remote file from the selected Release (Component) in a panel next to the scanned source file (dual-pane), click  . You can then review these two files side-by-side. You can perform text searches in either pane using the [Ctrl]+ [F] shortcut.
8. To load all source code matches, click  . By default, only the first 500 matches are loaded. They are sorted in descending order by the number of matches they contain.



The **File Details Panel Partial Matches** tab with SCF Matches Dual-Pane View view appears:

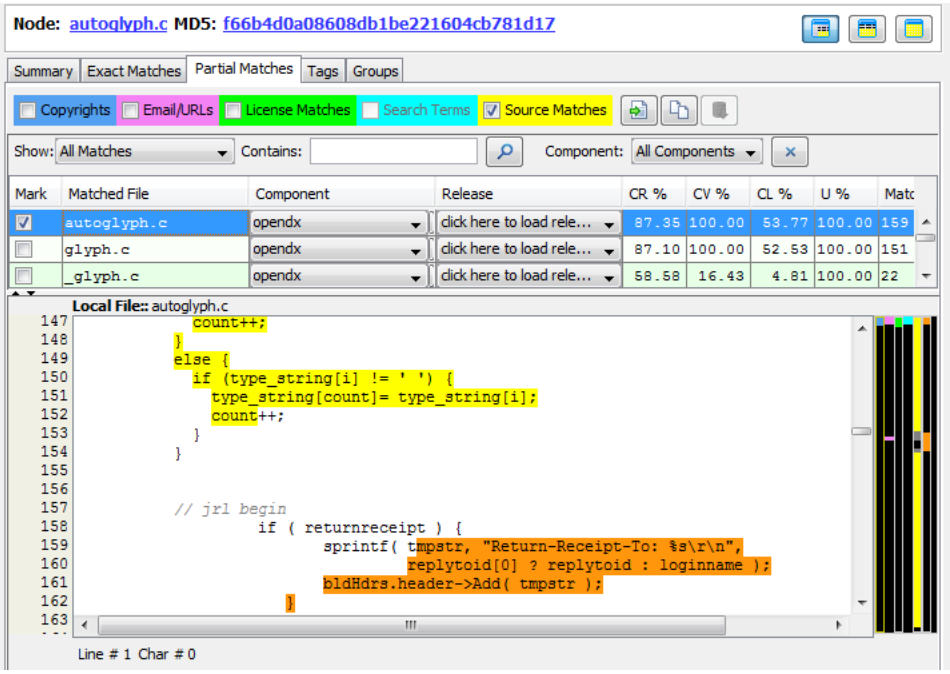


Displaying Source Code Fingerprint (SCF) Matches

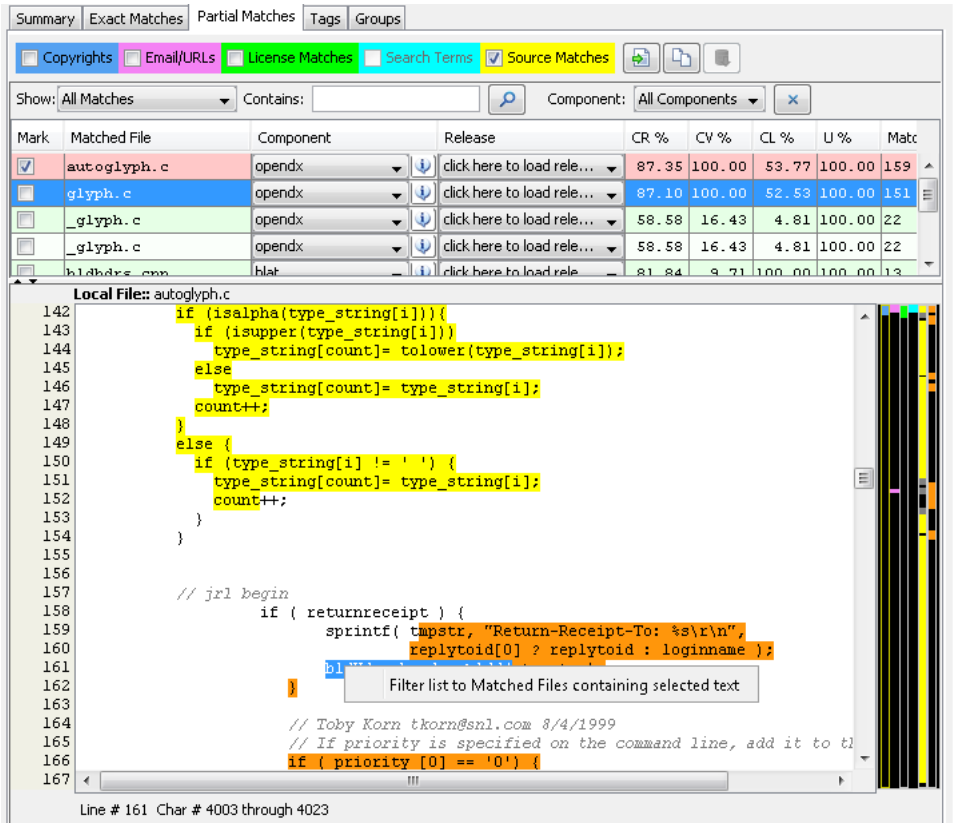
Because a typical scanned source file may contain source matches from multiple origins, Code Insight determines if all third-party content in the scanned file comes from a single source or from multiple sources.

This is done via the orange side bar and orange source match highlighting in the partial matches panel. It represents the union of all source matches across all matched files that are not represented via the currently selected matched file. This allows an auditor to quickly determine whether there are additional source code fingerprint matches in the scanned file that can be explained via an alternate set of matched files.

The yellow bar and highlights represent the Source Code Fingerprint matches found in selected files. The orange bar and highlights represent the Source Code Fingerprint matches found in non-selected files. The combination of the yellow and orange highlights represents the entire set of source code fingerprint matches in the scanned file from all files.



To filter the list of matched files to only those containing the selected text, select a section of text with orange highlights and right-click it and click the **Filter list to Matched Files containing selected text**.

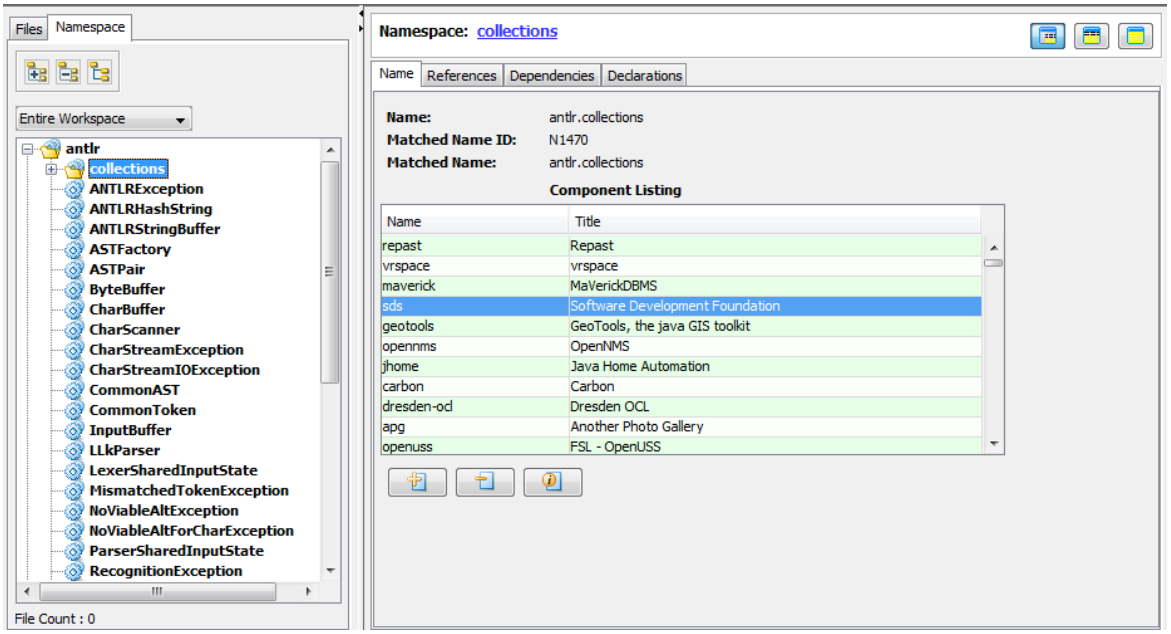


Displaying Java Name Matches

Java name space matches are Java classes in the code base that match Names in the signature Compliance Library.

Task ***To display Java name matches, do the following:***

1. Click the **Namespace** tab in the lower pane. The Java namespace file tree opens.
2. Select a bold filename, and double-click a line item to view the file information.



3. Click the **Information** icon for more information about the component. The **Component Info and Attribute** dialog box appears.

4. To assign a version number to the file from the existing list of versions or add a custom version to the list, click the **Plus** button (under the **Component Listing** pane). You can also remove a custom version from components, but you can't remove standard versions because they are predefined entries.

5. Click **Apply** on the Info icon (under the **Component Listing** pane) to apply your comments and changes.

Viewing SCM File Check-In History

If you configured an SCM application for the current workspace, right-clicking on a file in the file tree allows you to view the that file's recent SCM file check-in history.

Select the **View File Check-In History** option in the context menu.

30

Analyzing Files with Evidence (Third-Party Indicators)

This section explains how the auditor can use Detector for analyzing scan results in files with evidence. Information about analyzing files with evidence is presented in the following sections:

About Analyzing Files With Evidence

The auditor can use Detector for analyzing scan results in files with evidence. The basic auditing work process includes viewing files with a purpose in mind, and then taking action to mark those files that do or do not represent an indication of third-party content in the scanned code base.

The auditing tasks you can accomplish are as follows:

- **Research a file**—When a file contains third-party indicators, you can use the Exact or Partial Matches tab to review the evidence and decide which group best explains the existence of this file. If no such group exists, a new group can be created to explain the file.
- **Customize data**—You can associate a file that appears in the file tree with a product or add a license.
- **Annotate (tag) a file**—You can add information to a file by tagging the file with any type of data.
- **Analyze components and licenses**—You can view components and licenses that are part of your project inventory, or research any other components or licenses in the Code Insight Compliance Library.
- **Designate which files you have reviewed**—You can mark a file reviewed or unreviewed anytime during the auditing process.

- **Filter files**—You can filter evidence, much like filtering email. You can sort the files that fulfill certain filter criteria.
- **Add a file to a group**—Adding files that share characteristics with a group allows you to organize the scanned codebase. It also allows you to give a file a status, in order to slot it appropriately for review during the auditing workflow. Publishing groups results in project inventory.

Researching a File

When a file contains third-party indicators, you can use the Exact or Partial Matches tab to review the evidence and decide which group best explains the existence of this file.

To research (review all evidence) a file, perform the following steps.

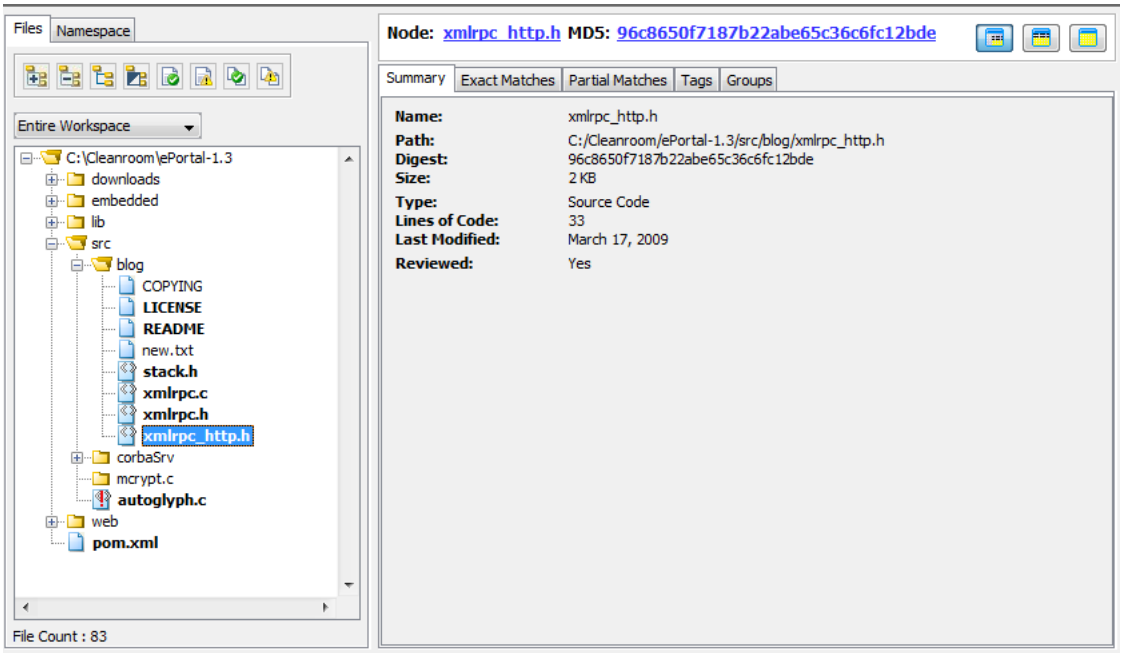
Task ***To research a file, do the following:***

1. Navigate to the file tree.

   Note ▪ *You can filter the tree using evidence tags or a custom filter.)*

2. Click on any file in the tree. Files are decorated depending on the types of evidence present in the file. Depending on whether the scanned file is source, archive, or other type of file, the icon preceding it is representative of the file type. The following file notations are available:

Notation	Description
bold	A filename that is in bold contains one or more exact matches.
(!)	A file with a red exclamation mark in front of it contains one or more source matches.
< >	A file with angle brackets in front of it is recognized as a source file based on the file extensions specified in Workspace Settings. ***Note*** ▪ *All files with exclamation marks also have < >.*
Regular weight, no exclamation mark	The absence of bold characters or exclamation marks indicates that the file is neither an exact match nor a source match.
Italics	A file in italics indicates that its status has not yet been received by the server.



3. Navigate through the **Evidence** tabs in the file details panel to review the various third- party indicators that were detected in the current file. To view different types of evidence, see Types of Evidence.

Designating Files That Have Been Reviewed

Files in the system can have the following states:

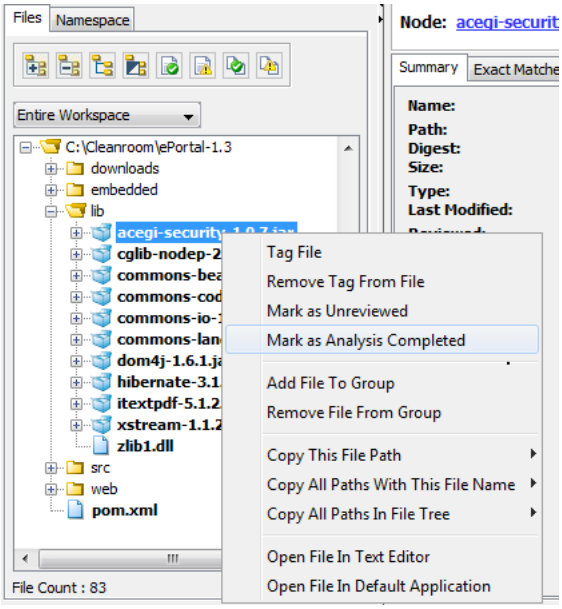
- Not Reviewed
- Reviewed
- Analysis Completed.

The states of Reviewed and Analysis Completed are related; therefore, only files that are marked as Reviewed can be marked as Analysis Completed and only files that are not marked as Analysis Completed can be unreviewed. A file that has been marked as Reviewed may be brought back to unreviewed state by the scanner during a rescan. See Rescan Options Tab for details. A file that has been marked as *Analysis Completed* cannot be brought out of that state by the scanner. This can only be done by the Auditor using the Detector client.

You can mark a file reviewed, unreviewed, or analysis completed by performing the following steps.

Task ***To designate a file that has been reviewed, do the following:***

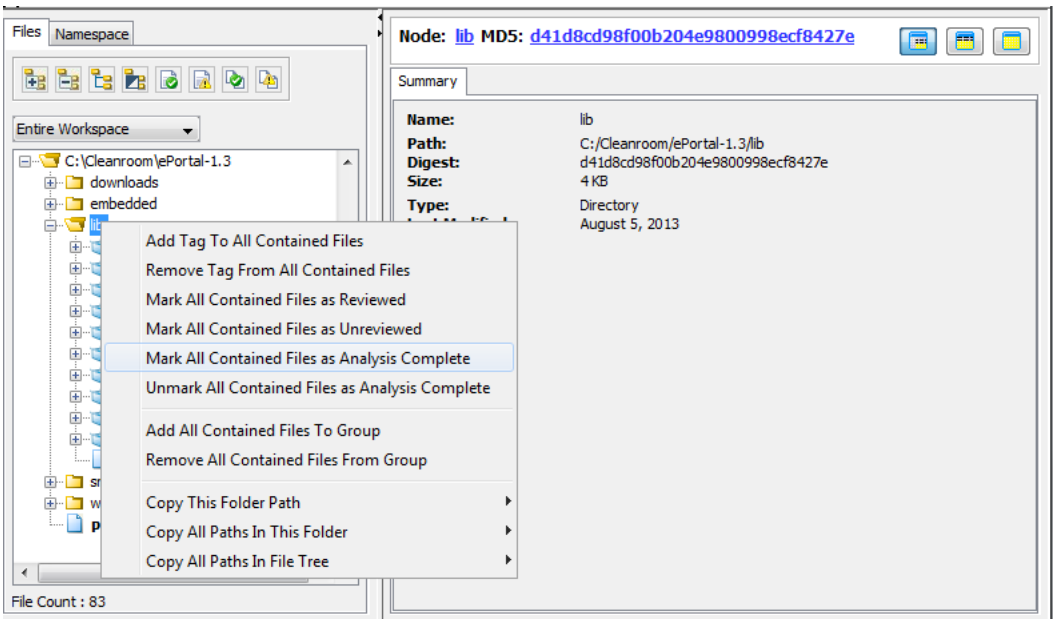
1. Click on a file in the file tree.
2. Right-click to access the commands **Reviewed** (or unreviewed) and **Analysis Complete**.



3. To designate that an entire directory has been *Reviewed* (or unreviewed) or *Analysis Completed*, select the directory.

4. Right-click on the selected directory.

5. Select the **Mark All Contained Files as Reviewed** (or unreviewed) and **Analysis Completed**.



Tagging Files and Archives

You can assign tags to files in order to associate arbitrary data to files or to allow custom filtering to be performed using tag values.

- Tagging a File

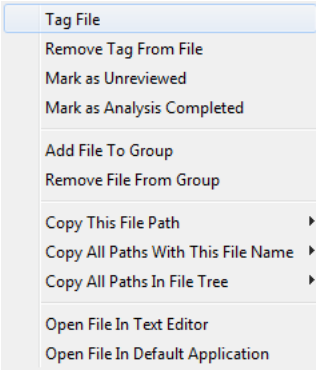
- Tagging an Entire Directory
- Assigning a Different Value to a Tag

Tagging a File

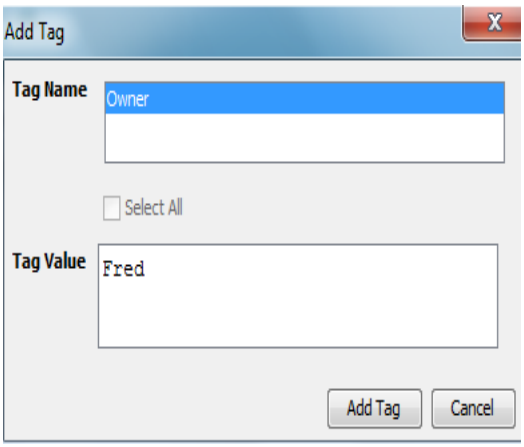
To tag a file, perform the following steps.

Task ***To tag a file, do the following, do the following:***

1. Select the file from the file tree that you wish to tag with one right mouse click.
2. Select the **Tag File** from the Right-Click menu.



The **Add Tag** dialog appears:



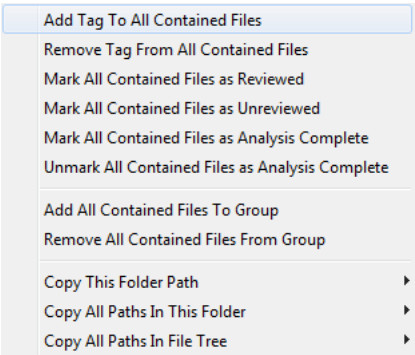
3. Click the **Tag Name** you want to apply. It is highlighted.
4. Click **Add Tag**.

Tagging an Entire Directory

To tag an entire directory, perform the following steps.

Task ***To tag an entire directory, do the following:***

1. Select the directory.
2. Right-click to access the command menu that pertains to directories:



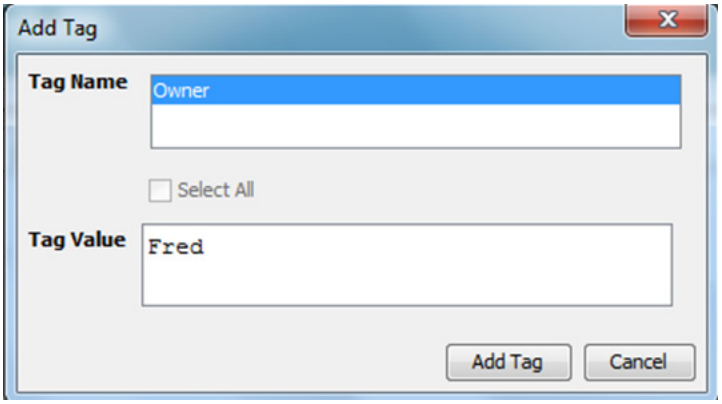
3. Select **Tag All Contained Files** from the menu.

Assigning a Different Value to a Tag

To assign a different value to a tag, perform the following steps.

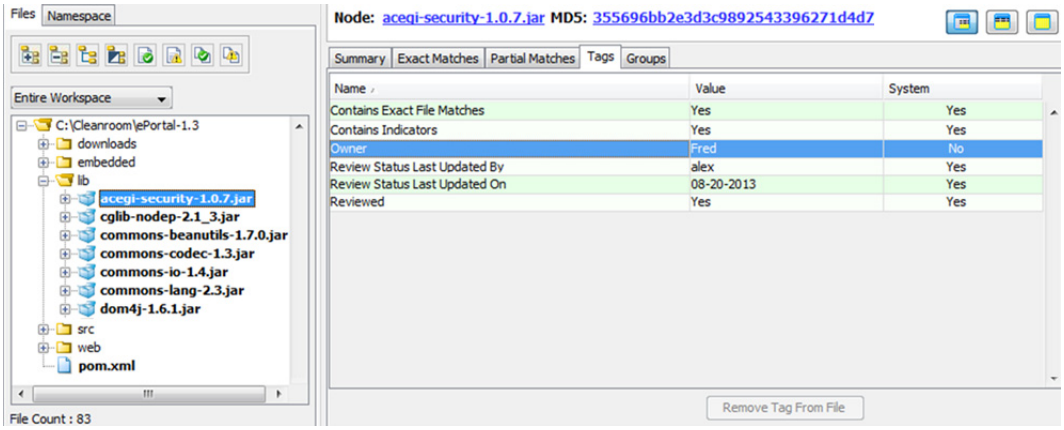
Task ***To assign a different value to a tag, do the following:***

1. Select **Tag File** from the right-click menu as shown in Assigning a Different Value to a Tag.
2. Double-click the **Tag Name** (in this case, **Owner**), and the **Add Tag** dialog appears.



3. Enter a different value.
4. Click **OK**.

5. Click **Add Tag** to apply the tag with the new value to the file or directory. The tag value appears on the **Tag** tab of the lower right-hand pane.



Filtering Files and Results

You can filter evidence, much like filtering email. You can sort the files that fulfill certain filter criteria, and by doing this, you can more easily find files and those components that require the same type of review process. You can also filter results by choosing the last filter you used during your research.

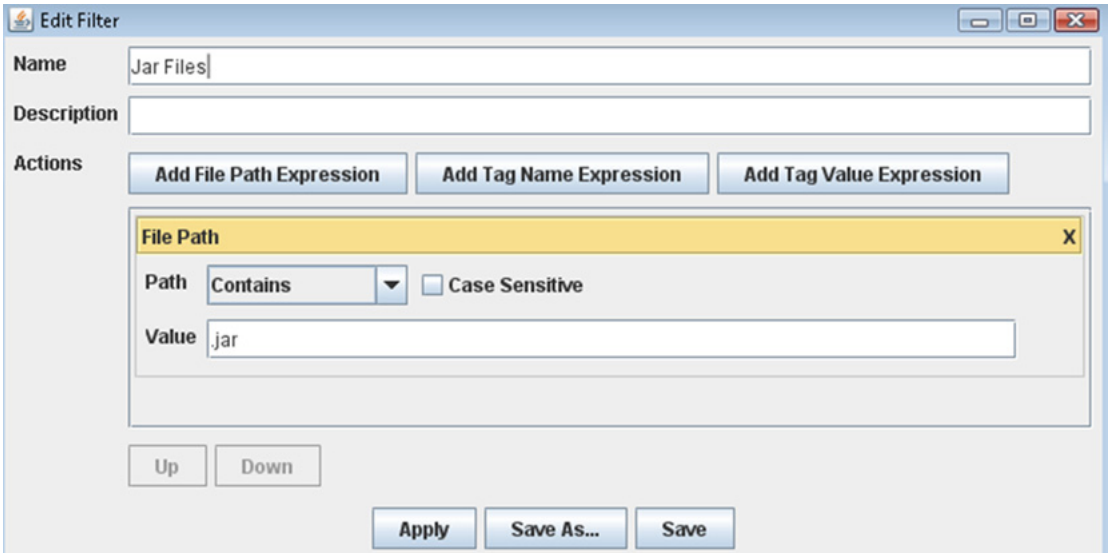
- Filtering Files
- Filter Expression Options

Filtering Files

To filter files based on file information, perform the following steps.

Task ***To filter files, do the following:***

1. Select the **Filter** tab in the upper pane, for example.
2. Select a line item from the **Filter** pull-down.
3. Click the **Edit** icon at the top of the tab. The **Edit Filter** page appears.



4. Click **Save As** to save the filter settings with a new name, or the **Save** button to save the filter settings as a future filter choice.

Filter Expression Options

This section contains details related to options you can set during file filtering.

- File Name/Path Expression
- Tag Name Expression
- Tag Value Expression
- Filtering Files by Last Filter Used

File Name/Path Expression

You can use File name expressions to filter files in the scanned codebase tree to include only those matching the criteria of the file name, path, or extension.

There are several operators available for this filter expression, and a case-sensitive comparison can be enabled if required.

Table 30-1 ▪ Operators

Operator	Description
Contains	Used to find all files that contain a certain String value in the file path
Does not Contain	Used to find all files that do not contain a certain String value in the file path
Equals	Used to find all files that exactly match the file path
Does not Equal	Used to find all files that are do not exactly match the file path

Table 30-1 ▪ Operators

Operator	Description
Starts With	Used to find all files that begin with a certain String value
Ends With	Used to find all files that end with a certain String value (file extension for example)

An example might be all files that contain “src” in the file path, and end in `.java`. So the following file would survive such a filter:

```
\project\modules\core\src\com\mycompany\class\foo.java
```

Tag Name Expression

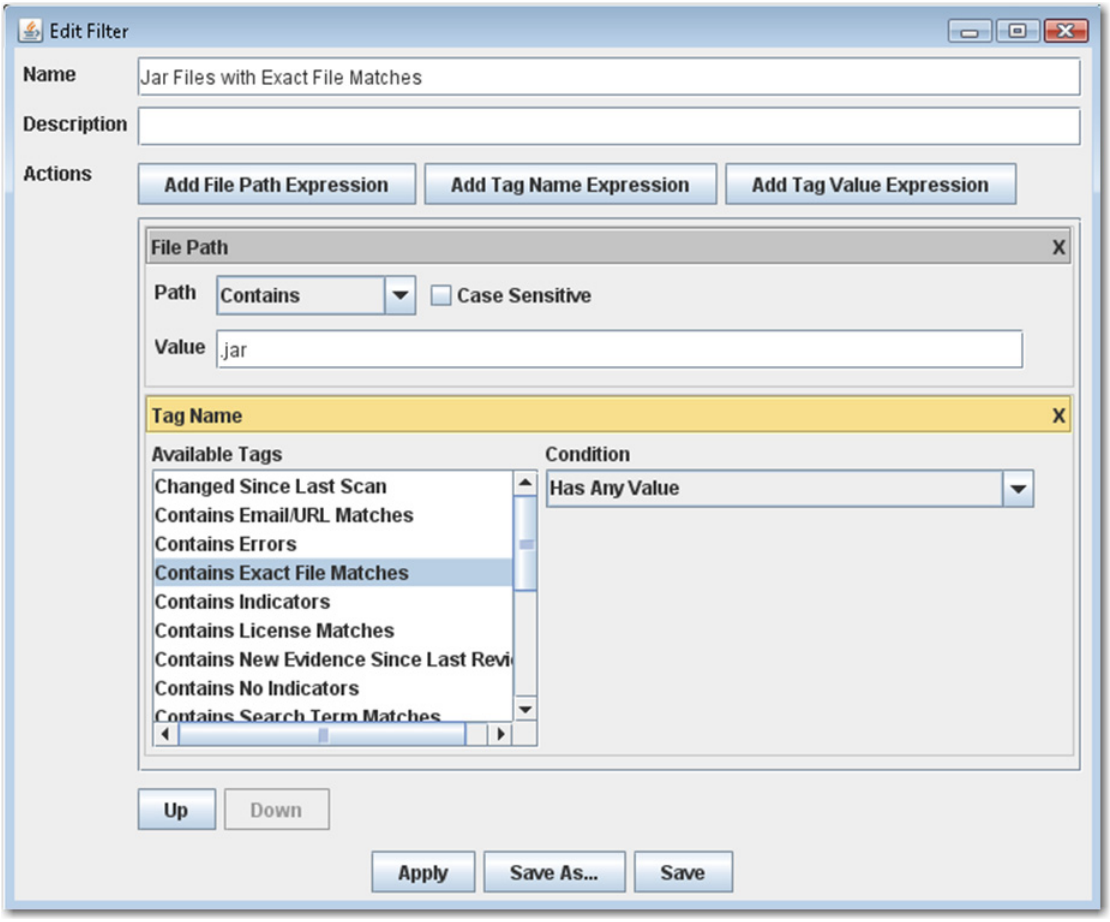
Tag name expressions can be used to filter files in the scanned codebase tree to include only those containing a particular tag assignment. This is the expression to use for checking whether a particular system evidence tag is assigned to a given file. An example would show all files with exact file matches that have (or have not) been marked as reviewed.

Note ▪ *Several tags can be selected. In such a case, the same operator is applied to each tag, and all selected tags must meet the defined criteria for a file to survive the filter.*

There are several operators available for this filter expression:

Table 30-2 ▪ Tag Name Expression Operators

Operator	Description
Has Any Value	Used to find all files that contain any value (including multiple) for the selected tags.
Has Single Value	Used to find all files that contain only a single value for the selected tags.
Has Multiple Values	Used to find all files that contain only multiple values for the selected tags.
Has No Value	Used to find all files that do not contain any values for the selected tags.



Tag Value Expression

Tag value expressions can be used to filter files in the scanned codebase tree to include only those containing a particular value assignment for a specific tag. For example, show all files that contain the string `John Smith` in the copyright block.

Note ▪ *Only one tag can be selected for this expression. Also, only files with a given tag assigned will be considered for the comparison. In other words, if Tag A does not have a value for File 1, then File 1 will not be a candidate for surviving a filter value expression for Tag A.*

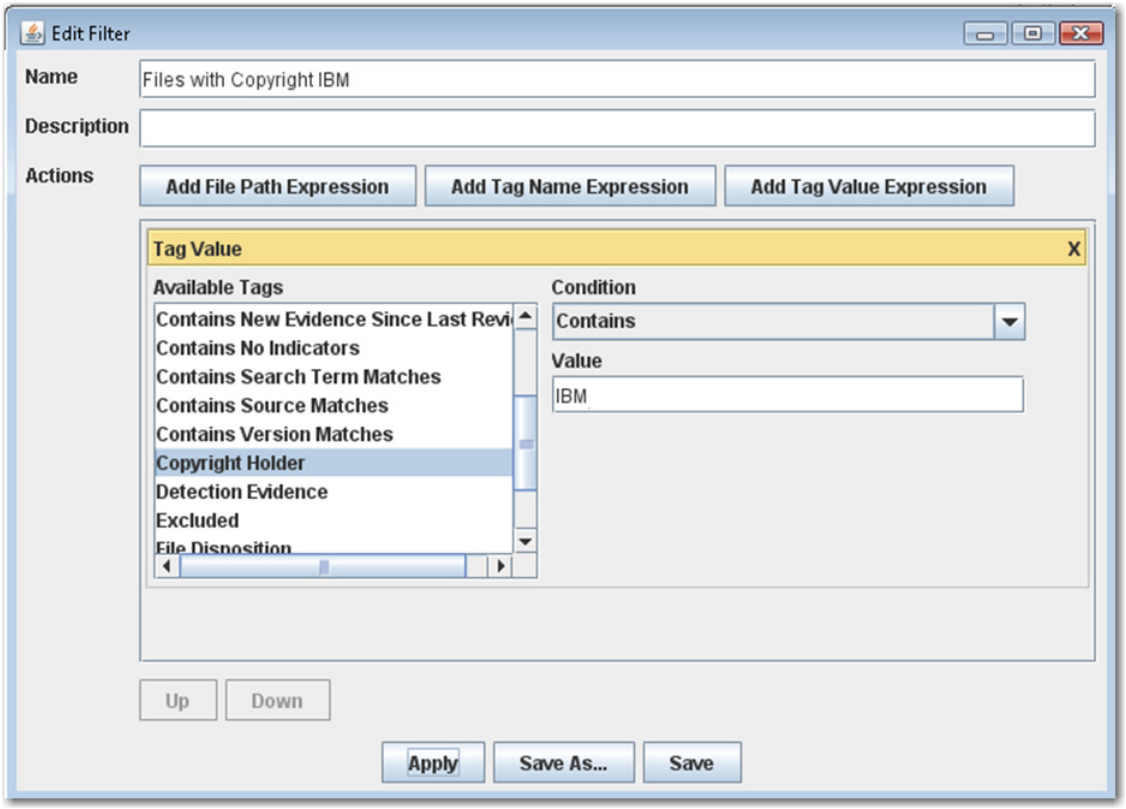
There are several operators available for this filter expression in addition to a **Has Only One Value** option that restricts the comparison to only those files that have a single value assigned for the selected tag:

Table 30-3 ▪ Tag Value Expression Operators

Operator	Finds...
Contains	Files with the selected tag assigned whose value contains the specified string.
Does not Contain	Files with the selected tag assigned whose value does not contain the specified string.

Table 30-3 ▪ Tag Value Expression Operators

Operator	Finds...
Equals	Files with the selected tag assigned whose value is an exact match with the specified string.
Does not Equal	Files with the selected tag assigned whose value is not an exact match with the specified string.
Starts With	Files with the selected rag assigned whose value begins with the specified string.
Ends With	Files with the selected tag assigned whose value ends with the specified string.

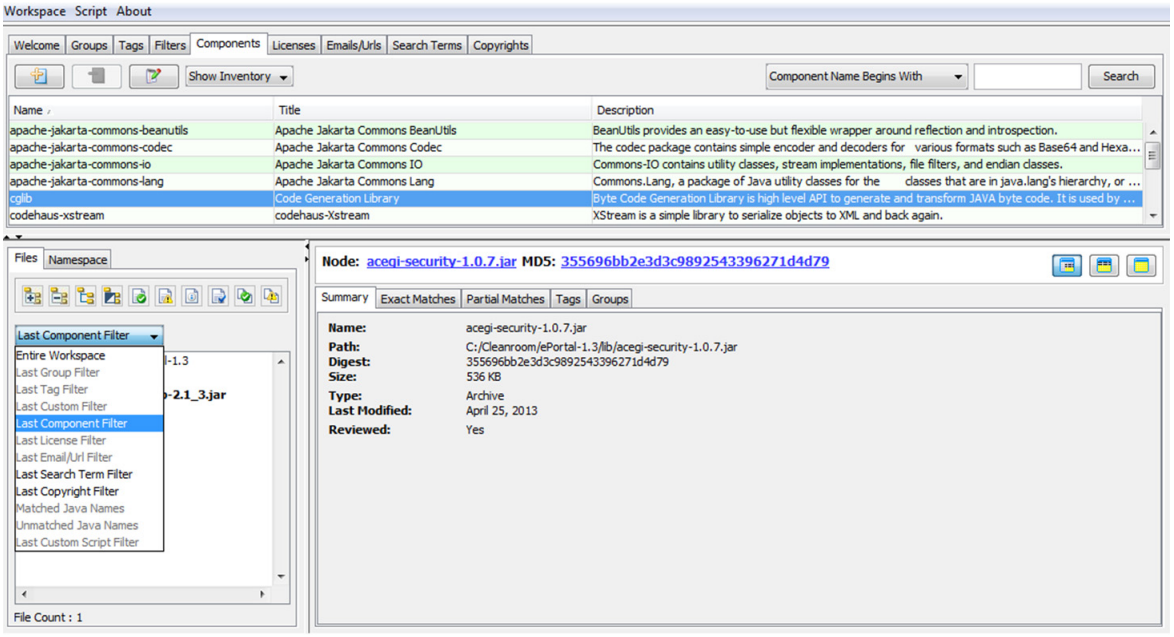


Filtering Files by Last Filter Used

To see the results of the last filter you used to view files in the file tree, perform the following steps.

Task ***To filter files by last filter used, do the following:***

1. Select a line item, for example, from the **Components** tab.
2. Select the last filter type you want to view from the pull-down on the **Files** tab.

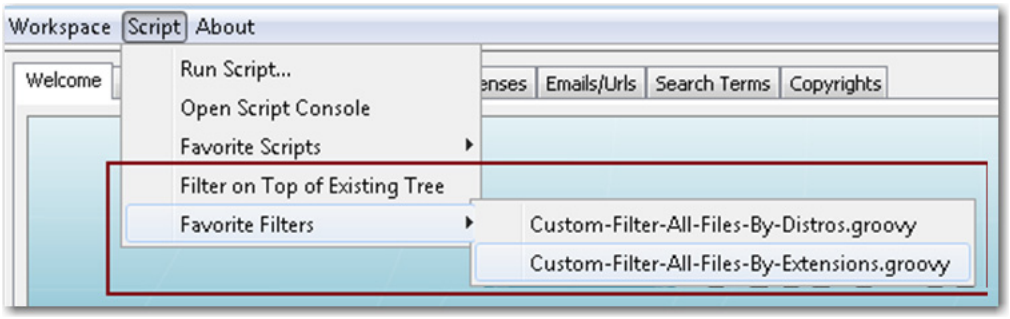


Custom Script-Based Filters

There may be cases where complex “custom” logic is required to filter the **Detector** file tree to only the desired files. A custom, Groovy script-based filtering option is available as an add-on to Code Insight.

The **Detector Script** menu contains a **Favorite Filters** option that lists all of the Groovy scripts that reside on the client system. The custom filter Groovy scripts should be placed in the following directory:

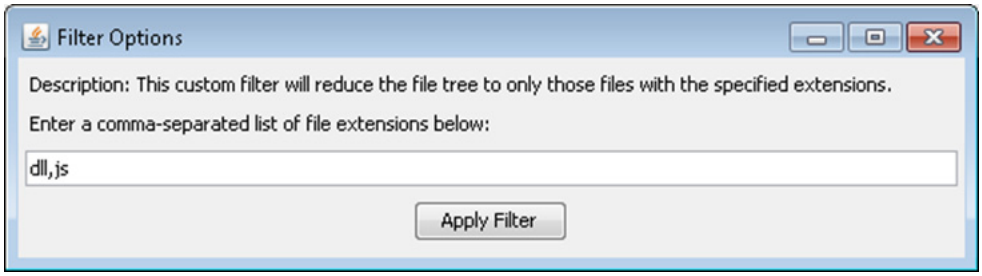
```
<USER_HOME>/.palamida/config/detector/custom_filters
```



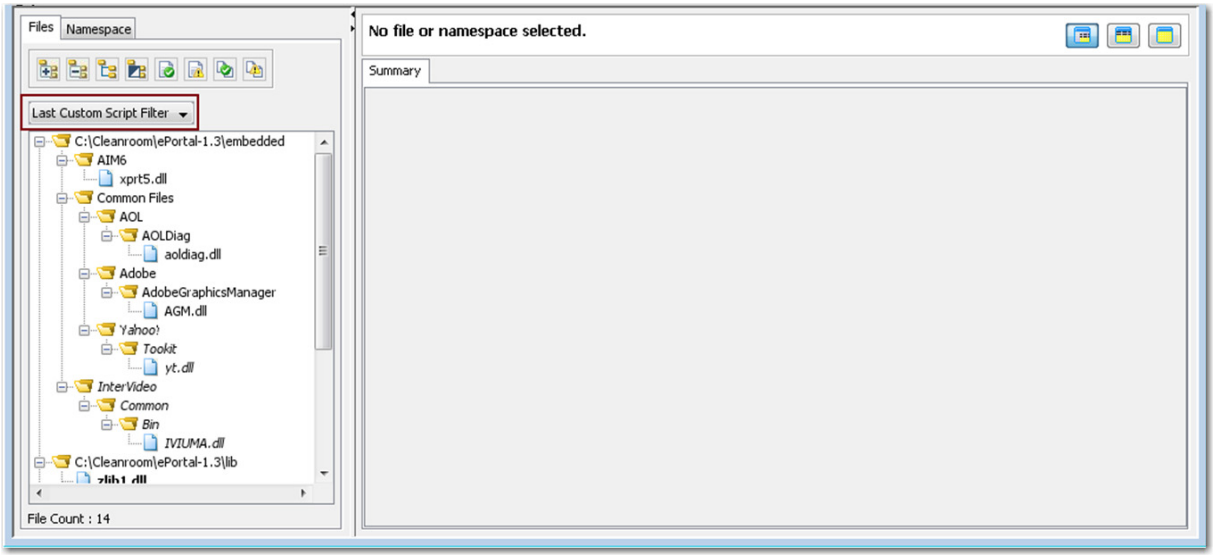
Theses Groovy script filters can implement any desired combination of criteria used to filter the list of files in the Detector file tree. The script is passed a list of existing file paths visible in the Detector file tree at the time of execution, and expects a list of file paths returned so that those files can be shown in the Detector file tree.

The **Filter on Top of Existing Tree** option allows the user to control whether the Detector file tree should show all files returned by the custom script filter, or only a subset from the files that are currently visible in the Detector file tree. This option allows sequential filters to be applied to continue filtering the tree until the desired files are remaining.

It is up to the custom script filter to invoke a Filter Options dialog if use-entered options are necessary.



After a custom filter is applied, use the **Last Used Filter** pull-down menu to reset the file tree to a previous filtered view. See Filtering Files by Last Filter Used for details.



31

Managing Inventory via Groups

Information about managing inventory via groups is presented in the following sections:

About Groups

Groups are used to organize scanned files. System-generated groups are populated with files that share version, component, or license information. For example, if a set of all files points to one license, it might be assigned a group by the system.

Publishing a group allows those working on the project to see the inventory associated with the project or workspace. As a project owner or auditor, you can create groups or assign files to existing groups. Files in a group can come from multiple workspaces in a project. You can view column headings associated with certain component names that refer to Workspaces as well as Projects.

Policies determine if a component in a group is allowed or disallowed. As an auditor, your job is to look all of the files in a workspace and determine which group best explains the existence (or origin) of that file.

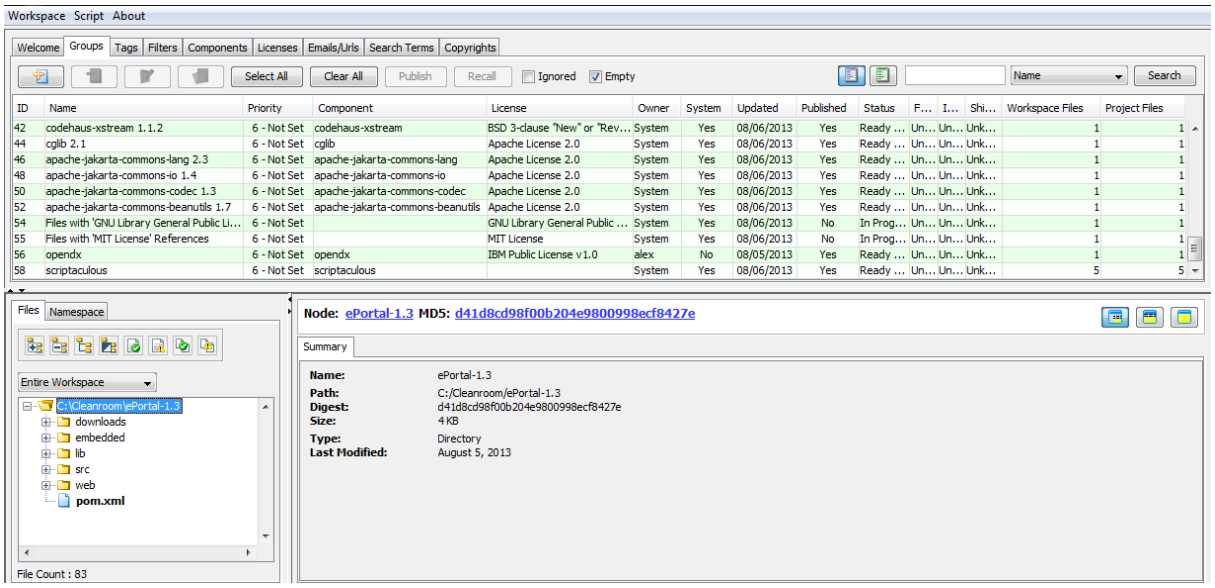
When you publish a group in Detector, in effect, you make it visible as inventory that is ready for review to the entire project team. These project team members can view inventory details including a complete file list associated with the inventory item (published group). You can use the Quick Review feature to immediately review each inventory item, or you can use the complete Code Insight workflow to review fully each inventory item using the request form and a formal review process.

The job of the auditor is to fully vet the scanned codebase, and explain the origin of each file by assigning each to appropriate groups. However, once a group is published, it is up to the project team to deal with the review and remediation of the files association with the inventory items (if necessary). Groups that are published from Detector can be viewed by clicking on the Inventory tab in the Project screen.

Recalling a group removes the inventory item from the project. This allows the auditor to do additional analysis on the associated files and re-publish the group or delete it if appropriate.

Managing Groups

Almost all group management can be accomplished from the **Groups** tab of Detector.



Creating a Group

To create a group, perform the following steps.

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

1. Click the **Plus** icon (on the top left of the top pane). The **Add New Group** dialog appears.
2. Enter the name for the new group and the owner.

3. Click **Add New Group** to add the group.

Deleting a Group

To delete a group, perform the following steps.

Task ***To delete a group, do the following:***

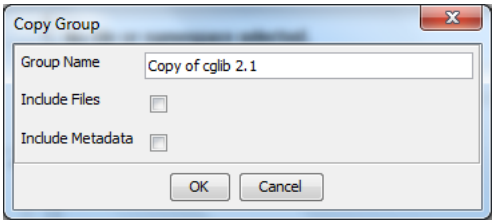
1. Select a Group in the table.
2. Click the **Minus** icon. The **Group Delete** dialog appears.
3. Click **Yes** to delete the group.

Copying an Existing Group

To copy an existing group, perform the following steps.

Task ***To copy an existing group, do the following:***

1. Select a Group in the table.
2. Click the **Copy** icon. The **Copy Group** dialog opens.



3. Configure the copy group options:
   - **Include Files**—Select to associate files from existing group with the new group.
   - **Include Metadata**—Select to set group metadata fields for the new group equal to those of the existing group.
4. Click **OK** to copy the group.

Managing Group Details

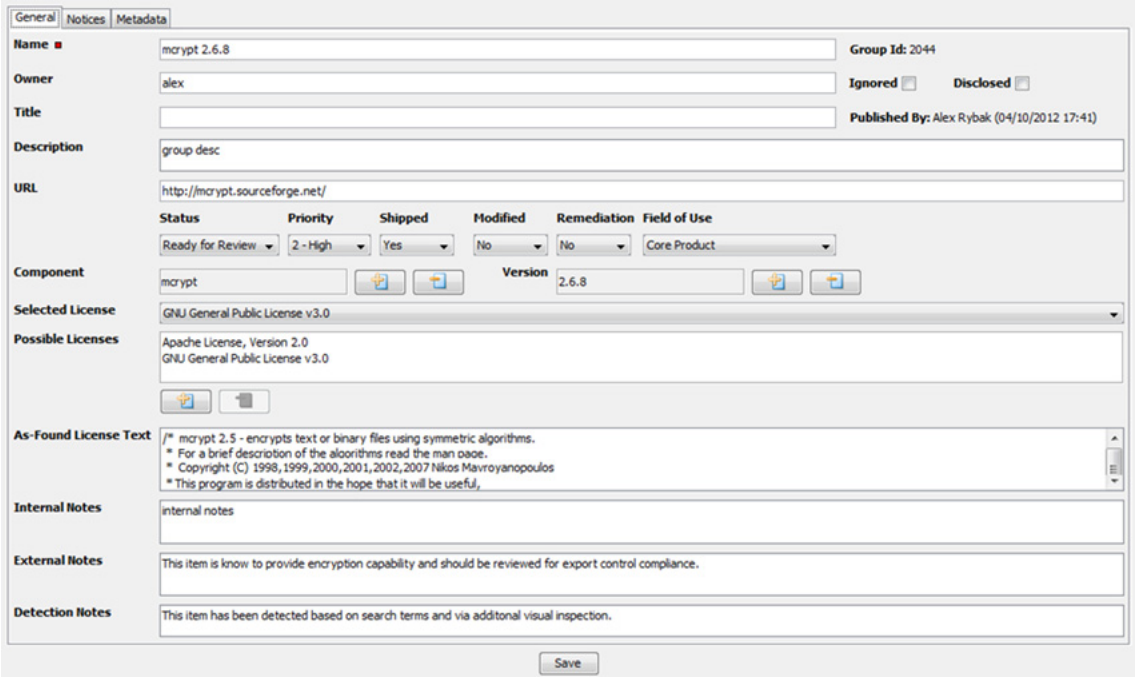
The Group Details tabbed screen provides a tabbed view of various group details. From the General, Notices, and Metadata tabs, you can manage information related to specific group information.

Table 31-1 ▪ Group Details Screen

Tabs	Description

General	• Group Name, Owner, Title, Description
	• URL
	• Specific criteria for the following values:
	  • Disclosed
	  • Status
	  • Priority
	  • Shipped
	  • Modified
	  • Remediation
	  • Field of Use
	• Source Distribution Required and URL
	• Components, Versions, Selected Licenses, Possible Licenses, As Found License text, Internal Notes, External Notes, and Detection Notes (read-only)

Notices	• Include in Third-Party Notices • Notice Title, Notice URL, Notice Attribution Statements, Notice Copyright Statements, Notice License URL, Notice License Text, Source Distribution URL, and an option to save current values as a default as well as overriding current values with an available set of default data.
Metadata	Includes any defined group metadata fields.



Managing Third-Party Notices Data for a Group

The use of third-party (TP) and open source software (OSS) is permitted under various open source licenses. In many cases, open source licenses carry with them an obligation that the user of the open source software must provide attribution to the original author. This obligation is generally met by generating a report that includes a list of open source and third-party materials and the licenses under which they are used for the shipping product or application.

Code Insight provides a standard third-party notices feature in the product to make this a simple byproduct of the auditing work performed in the product. The following fields comprise a third-party notice for the use of a particular piece of TP/OSS:

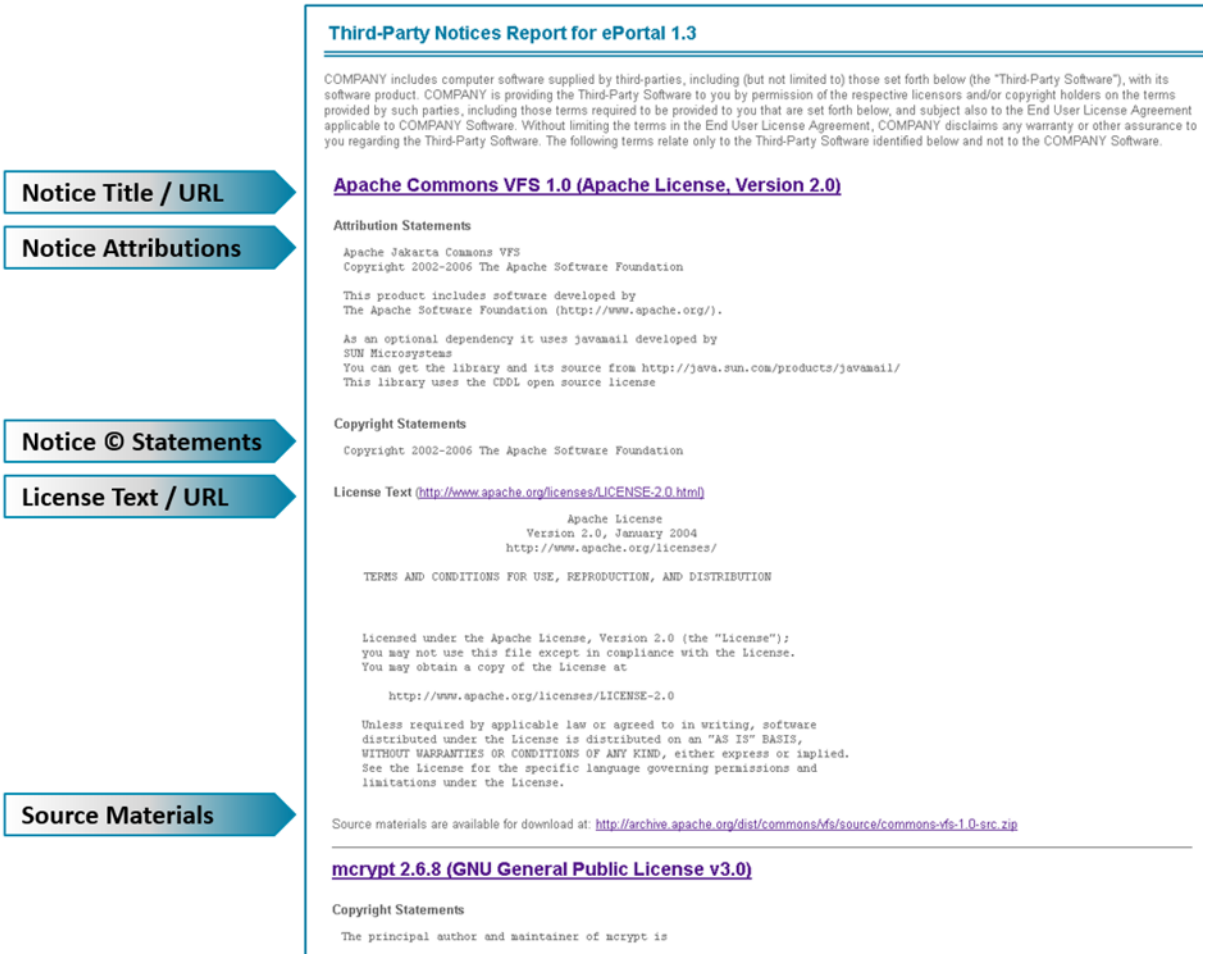
Table 31-2 ▪ Third Party Notices Fields

Field Name	Field Description
Include in Third-Party Notices	Used to control which audit findings are included in the Third-Party Notices report.
Notice Title	Recommended: Component Version (License) Example: Apache Ant 4.7.1 (Apache License, Version 2.0)
Notice URL	Hyperlink to desired location. Typically, this is to the project homepage.
Notice Attribution Statements	Aggregated list of attribution statements and other notices as-found in the codebase.

Table 31-2 ▪ Third Party Notices Fields (cont.)

Field Name	Field Description
Notice Copyright Statements	Aggregated list of “clear” copyright statements as-found in the codebase. May optionally paste in copyright statement from the license text.
Notice License URL	Hyperlink to desired location. Typically, this is to the location of the license text.
Notice License Text	The license text as it is to be shown in the Third-Party Notices report.
Source Distribution URL	Location where the source materials are available for a given third- party or open source software item if you are making this available.

The information entered in the third-party notice fields is used to produce an out-of-the-box third-party notices report (from the Web UI).

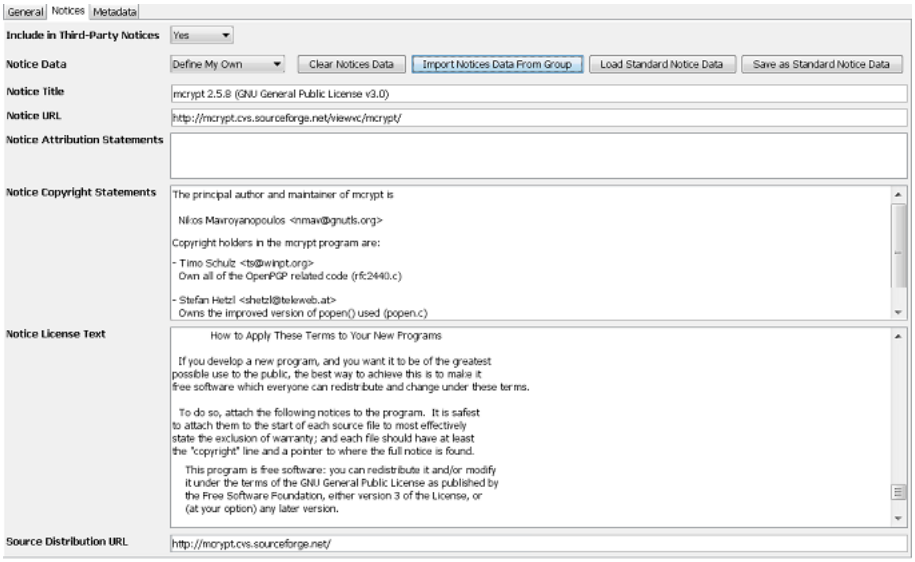


The report contains a block for each item that has been flagged to be included in the report consisting of the following values:

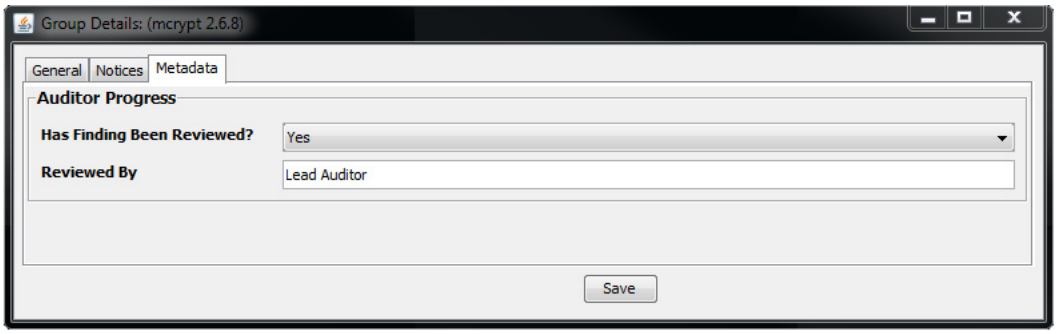
- Notice Title and optional Notice URL.
- An optional Notice Attribution section if the corresponding field has a value.
- An optional Notice Copyright section if the corresponding field has a value.
- An optional License Text and URL section if the corresponding fields have values.
- An optional Source Materials URL if the corresponding field has a value.

Third-party notice fields are located on the Notices tab of the Group Details window in Detector.

A third-party notice can either be defined from scratch for the current group or loaded (using the Load Standard Notice Data button) from the Standard Notices data (if available) and modified as needed. A standard notice applies if the component, version, and license of the notice matches that of the current group. If a standard notice is loaded, it can be modified and saved (using the Save as Standard Notice Data button) as an updated version of the standard notice, or just used for the current group.



The Import Notices Data From Group button may be used to copy data from the General Tab to the Notices Detail tab to avoid unnecessary rework if the information has already been entered.



Adding or Editing Group Details

To edit or add information to a group, perform the following steps.

Task ***To add or edit group details, do the following:***

1. Select the **Group** tab.
2. Click the **Edit** icon. The Groups Detail tabbed screen appears, allowing you to manage general group information, notes, and notices.

Setting Group Status Information

To set group status information, perform the following steps.

Task ***To set group status information, do the following:***

1. Select the **Groups** tab in the top pane.
2. Click the **Edit** icon. The **Group Details** page appears.
3. From the Status pull-down, select the correct status for a particular group.

Managing Licenses for a Group

There are situations where the selected component version for a group has multiple possible licenses— one of which must be selected. An auditor may encounter evidence during an audit that suggests this multiple possible licenses type of scenario, but may not have the ability to make the appropriate selection. In such a case, it is the auditor's responsibility to prepare an accurate list of possible licenses from which a legal user will eventually select the appropriate license from the Code Insight web UI during the inventory review process.

For example, a group contains two license fields: Possible Licenses and Selected License. The Possible Licenses field can be edited by the auditor to accurately reflect the list of licenses associated with a given group. The Possible Licenses list is defaulted to the licenses associated with the selected component version if version specific license data is available for the selected component version, or the component itself if version specific license data for the selected component version is not available, or if the version is not selected. The Possible License list can then be edited by the auditor by adding and/or removing any licenses. There is no requirement that the licenses in the Possible License list must be associated with the selected component and/or version. Any changes made to the list of licenses in the Possible Licenses list are for the current group only, and are not reflected in any subsequent groups with the same component and version.

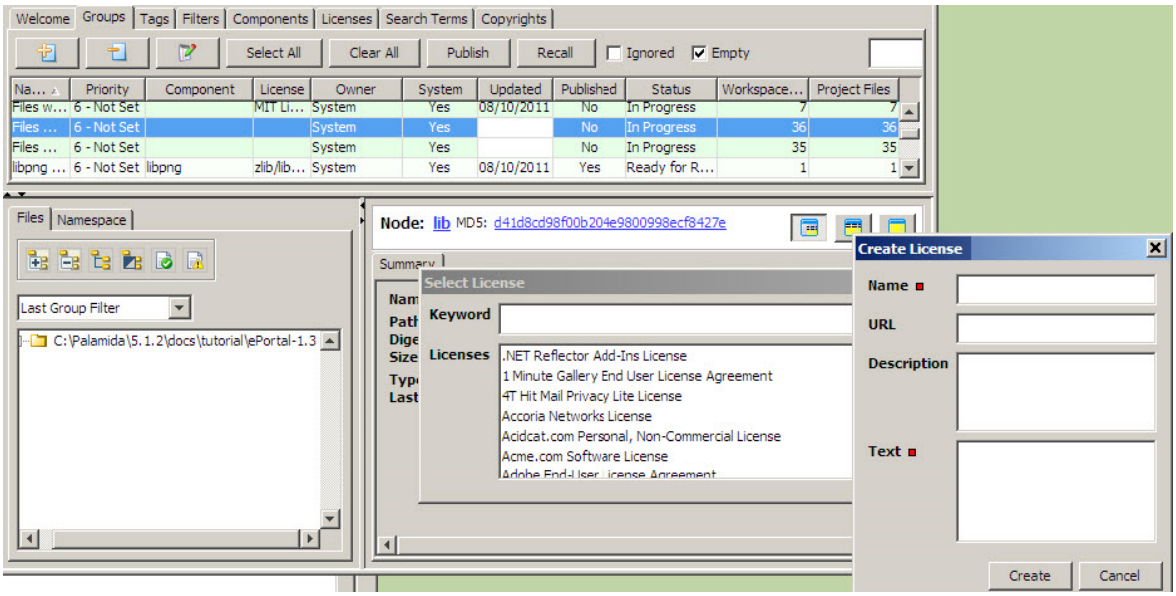
The list of licenses in the Possible Licenses box will define the possible values for the Selected License pull-down. If the auditor knows which license should be selected, then a selection can be made from the Selected License pull-down. This value can then be updated in the Code Insight web UI during inventory review.

To manage licenses for a group, perform the following steps.

Task ***To manage licenses for a group, do the following:***

1. Access the **Edit Group** dialog for the group to which you would like to add license information.
2. To add additional possible licenses to the list, click the Plus button under the **Possible Licenses** box. To remove existing licenses from the possible licenses list, select the license to remove, and click the Minus button under the **Possible Licenses** box.

   When you click the Plus button, the **Select License** box appears.
3. Select a license from this list or use the **Search** field to find a license.
4. If you do not see the license that you need, you can create a license by clicking on the Create License button on the **Select License** box.
5. The list of licenses in the Possible Licenses box will define the possible values for the Selected License pull-down. If the auditor knows which license should be selected, then a selection can be made from the Selected License pull-down. This value can then be updated in the Code Insight web UI during inventory review.



Viewing Groups Created by System

During a scan, the automated analysis detectors create system groups with component, version, and license information pre-selected. To view the system-created groups, perform the following steps.

Task ***To view system-created groups, do the following:***

1. Select the **Groups** tab in the top pane.
2. Click the **Owner** column, and sort the groups by those created by the system and those created by the users.

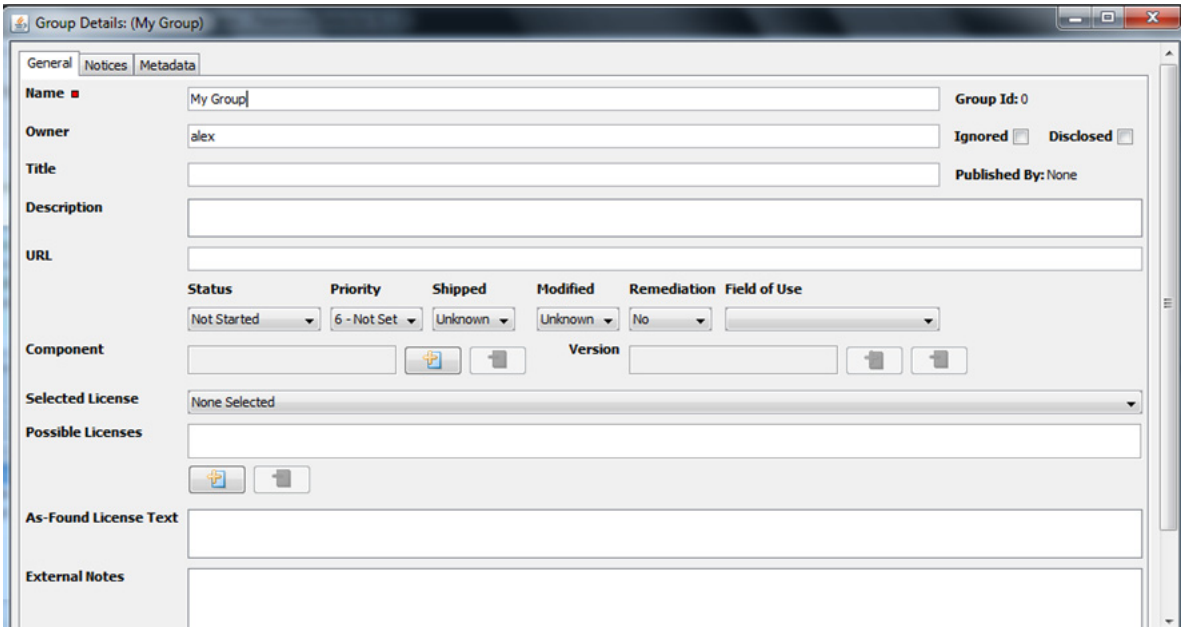
Creating User Groups

You may want manually to create groups to organize files representing inventory items or for other purposes such as to organize files to be reviewed by a specific auditor.

To create a user group, perform the following steps.

Task ***To create user groups, do the following:***

1. Select the **Groups** tab in the top pane.
2. Click the Plus icon to add a new group to the list of groups.



The **Create Group** box opens.

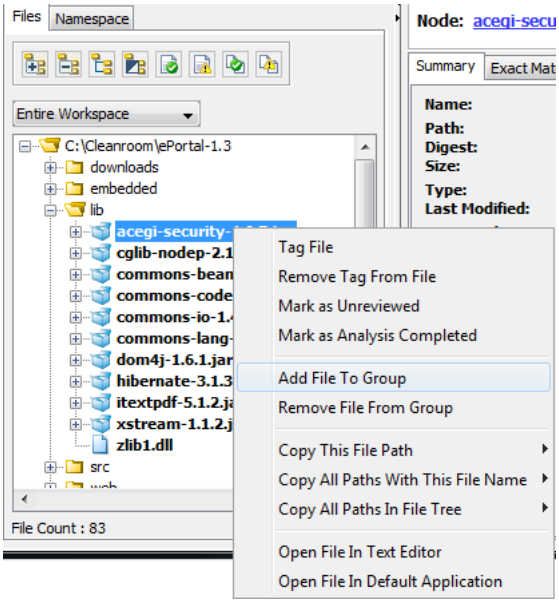
3. Enter a name and an owner for the new group. (The owner field defaults to the current user login name.)
4. Click the **Create** button.

Adding Files to a Group

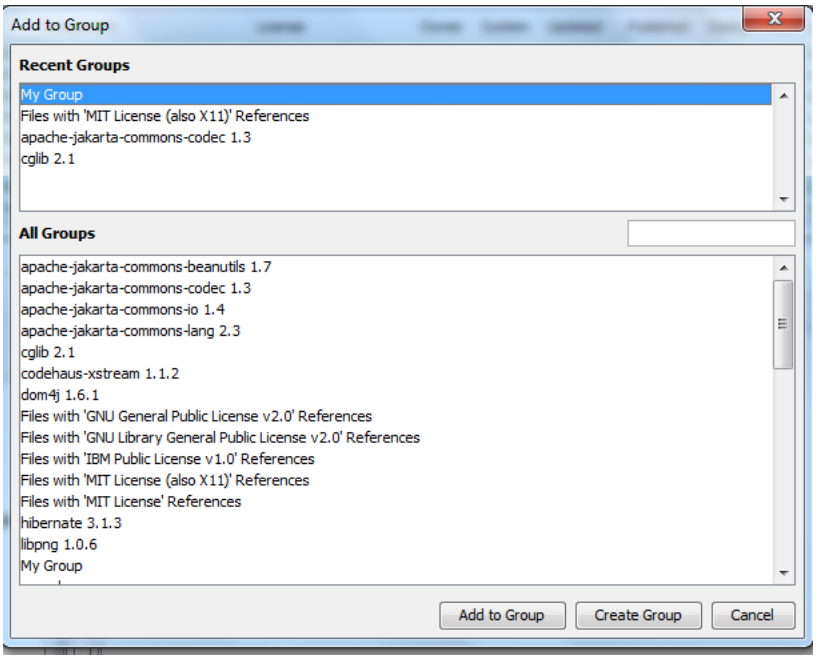
To add a files to a group, perform the following steps.

Task ***To add files to a group, do the following:***

1. Navigate to the file tree.
2. Right-click a single file or a directory (if you want to add all files within that directory).



3. Select **Add File to Group**. The **Add to Group** dialog appears.



4. Select a group you want to add the file.
5. Click **Add to Group**.

Publishing Groups

To publish groups, perform the following steps.

Task ***To publish groups, do the following:***

1. On the top pane on the **Groups** tab, select as many groups as you wish to publish by holding down the Control key on the keyboard and left clicking the mouse simultaneously.
2. Click **Publish** to publish only that group.
3. To publish all the groups, click **Select All**, and then click **Publish** in the top pane.

Recalling Groups

The following section contains information about different ways to recall groups.

- Recalling a Group from the Individual Group Pane
- Recalling Groups Bulk Fashion

Recalling a Group from the Individual Group Pane

To recall a group is to remove it from the published files, which constitute your inventory. You can recall a group only from the **Groups** tab.

Task ***To recall one group, do the following:***

1. Select the **Groups** tab.
2. Select the group to recall. To recall a group, the Published column on the **Groups** tab displays **Yes** to indicate that the group is published and part of inventory.
3. In the top pane, click **Recall**. The group you selected will be removed from the set of published files and inventory. The **Published** column associated with that group on the **Groups** tab now displays **No**.

Recalling Groups Bulk Fashion

If you need to recall several groups, save time by recalling groups in a bulk fashion.

Task ***To recall groups in a bulk fashion, do the following:***

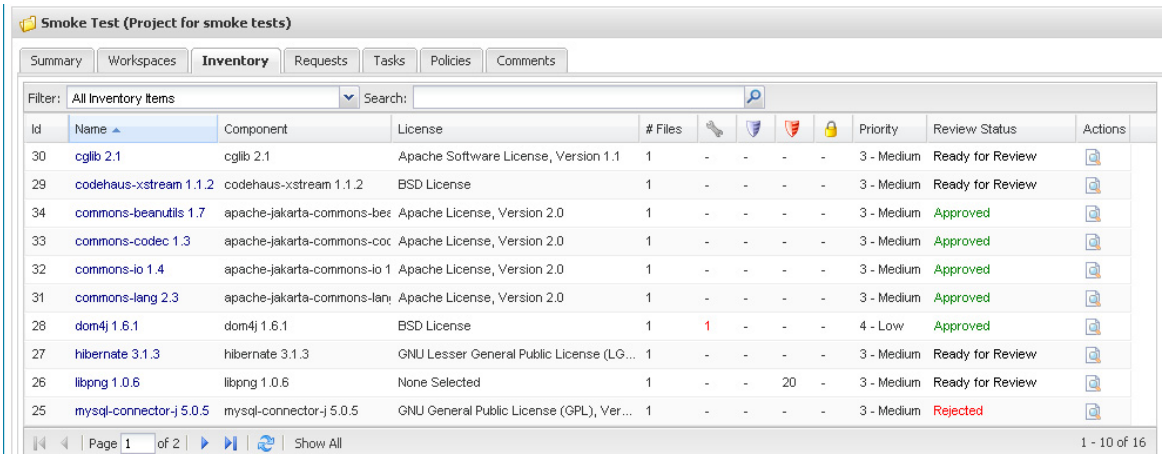
1. On the top pane on the **Groups** tab, select as many groups as you wish to recall by holding down the Control key on the keyboard and left clicking the mouse simultaneously.
2. Click **Recall** to recall all of the groups you selected.
3. To recall all of the groups, click **Select All**, and then click **Recall**.

Viewing and Confirming Published Inventory

After you have published groups, you can confirm that they were published successfully by reviewing the project inventory.

Task ***To view and confirm published inventory, do the following:***

1. Click **My Projects**.
2. Click a **Project Name** link.
3. Select the **Inventory** tab. To search and filter the list, enter terms in the **Filter** and **Search** fields.



32

Working with Archives

If your codebase contains supported archive types and was scanned with the “Scan Files Inside Archives” option enabled in Workspace Settings, you will be able to view different evidence types and operate on both the archive’s outer file (such as foo.jar) as well as for the archive’s inner files (all files located inside foo.jar) in Detector. For information on how to configure your workspace to scan files inside archives, see Detection Tab Tasks.

Supported Archive Types

Cab, ear, jar, rpm, sar, tar, tar.bz, tar.bz2, tar.bzip2, tbz, tgz, war and zip are supported archive types. For an up-to-date list of supported types, go to **Workspace Settings > Detection Tab > Archive Scanning Options** at the bottom of the page.

Types of Evidence

The following are the types of third-party indicators available as part of the scan results for archives and their contents:

- Exact Matches, Copyrights

- Emails and/or URLs
- Search Term matches
- License matches

Automated Analysis techniques such as Auto-WriteUp, POM File Analyzer and MID Rule Detection are also available for archives and are reflected in the scan results. See Automated Analysis Tab for information on how to enable these techniques.

Source Matches

Files inside archives are not scanned and tagged for Source matches. For extended analysis on files inside archives, it is recommended that you manually unpack the archive before performing a scan.

Exact Matches

There are two types of archives: those with exact matches and those without exact matches.

Archives with Exact Matches

During a Code Insight scan, archives that are found to contain an Exact match (digest match) on the outer file are tagged with the “Contains Exact File Matches” tag, while inner files of these archives are not tagged for Exact matches and therefore will not display the tag. This behavior is in place as a performance enhancement and is reflected in the group/tag file counts. These files are still analyzed and tagged for all other evidence types, including Copyrights, Emails and/or URLs, Search Term matches and License matches. The “Contains Indicators” tag is applied to inner files of archives with Exact matches only if one of these four evidence types is detected and is not applied otherwise.

For example, if a scan reveals that foo.jar has an Exact match to a component in the Code Insight CL, it will be tagged with the “Contains Exact File Matches” tag. Its inner files however, such as `foo.jar/License.txt`, will not display this tag but may display other tags such as the “Contains License Matches” and “Contains Indicators.”

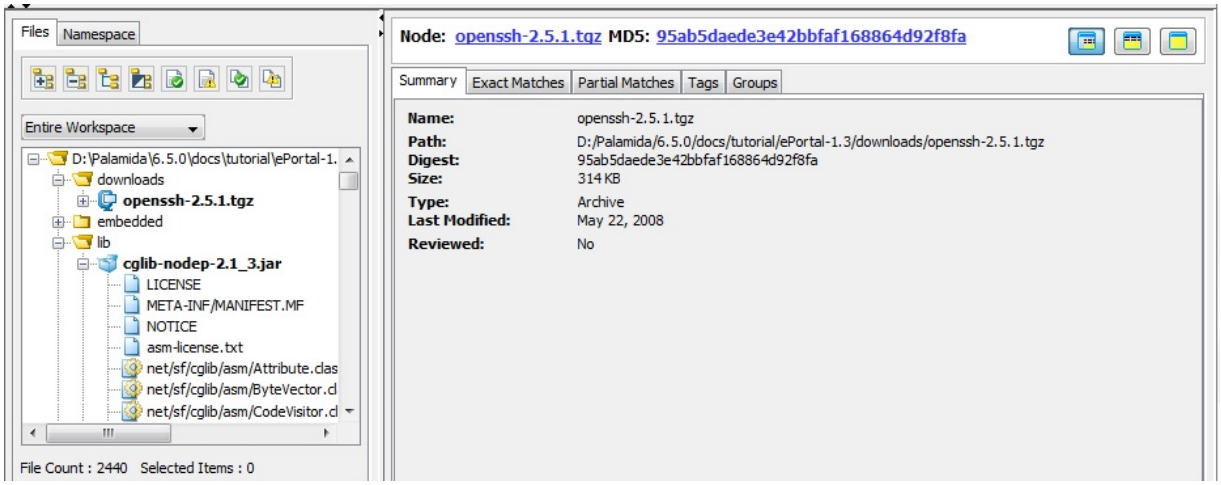
Archives without Exact Matches

Archives that do not contain an Exact match (digest match) on the outer file are further analyzed for Exact matches on the inner files. If an inner file is found to contain an Exact match, it will be tagged with the “Contains Exact File Matches” tag and the “Contains Indicators” tag.

For example, if a scan reveals that foo.jar does not have an Exact match to a component in the Code Insight CL, its inner files will be further analyzed for Exact matches. If foo.jar/License.txt is then identified as an Exact match to a component in the Code Insight CL, it will be tagged with the “Contains Exact File Matches” tag and the “Contains Indicators” tag, as well as any other tags reflecting the detected evidence.

Viewing Evidence

To access the contents of an archive press the (+) icon to the left of the archive name in the file tree located in the Files tab. This will expand the archive, allowing you to view evidence for and operate on the files inside the archive.



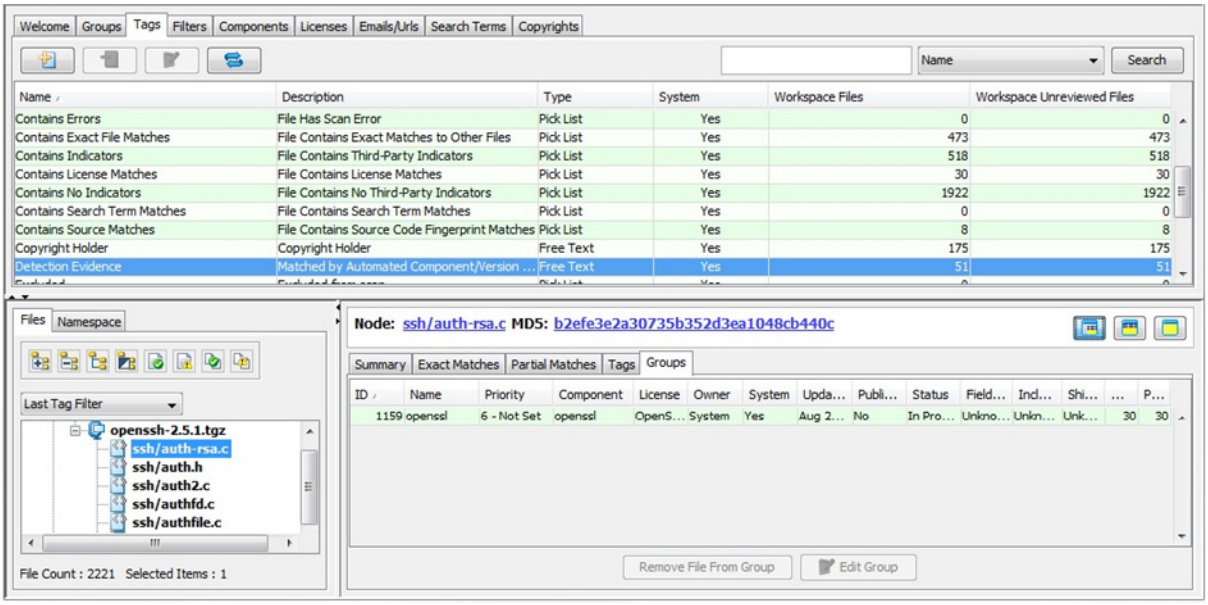
Groups

You can view the groups that an archive file belongs to by selecting the file in the file tree and switching to the Groups tab in the lower pane.

If a file inside an archive triggers an automated detection technique such as Multi-Indicator Detector (MID Rule) or Auto-WriteUp, the file will be added to the appropriate System Group. Alternatively, inner files of archives, the archive outer file, or both types of files can be manually added to a group.

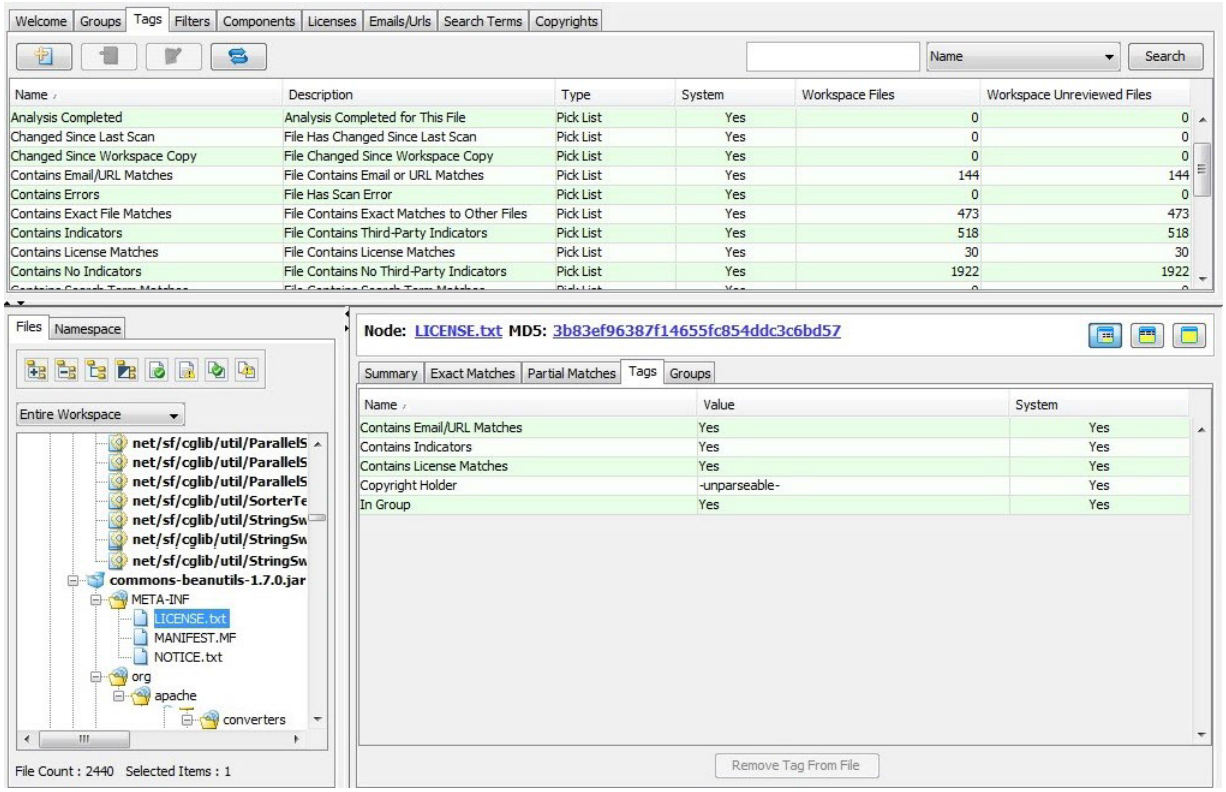
Note ▪ *Adding an inner file of an archive to a group will not automatically add the archive outer file to the group and adding an archive outer file to a group will not automatically add the archive's inner files to the group. Group file counts in the "Workspace Files" and "Workspace Unreviewed Files" column reflect this behavior and may be smaller than the File Count in the file tree.*

For more information on groups, see Managing Groups.



Tags

You can view tags for an archive file by selecting the file in the file tree and switching to the **Tags** tab in the lower pane.



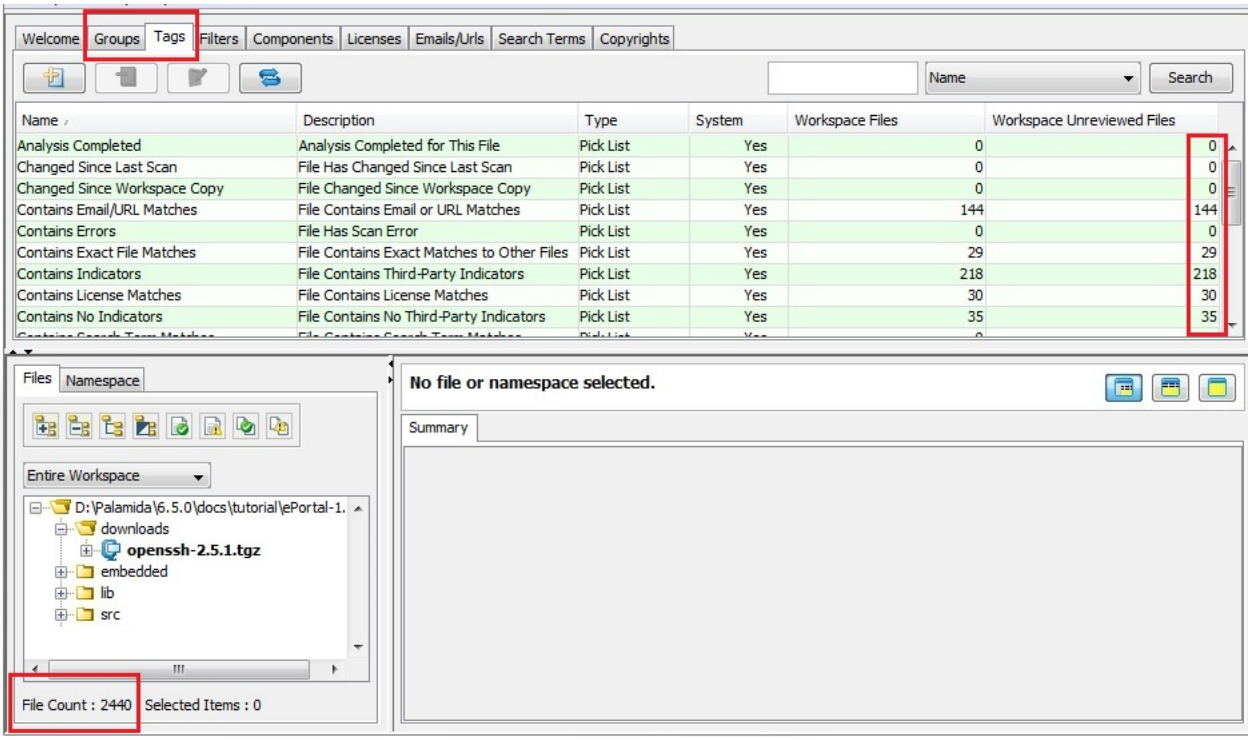
If a file inside an archive contains third-party indicators such as Exact Matches, the appropriate system tags will be applied to the file. Alternatively, you can manually add a tag to inner files of archives, the archive outer file, or both types of files.

Note ▪ *Adding a tag to inner files of an archive will not automatically add the tag to the archive outer file and adding a tag to the archive outer file will not automatically add the tag to the archive's inner files. Tag file counts in the "Workspace Files" and "Workspace Unreviewed Files" columns reflect this behavior and may be significantly smaller than the File Count in the file tree.*

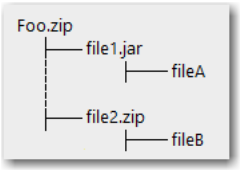
For more information on tags, refer to Tagging Files and Archives.

Archive File Counts/Nested Archives

The File Count located at the bottom left-hand corner of the Detector file tree represents the total number of files currently in the file tree. In the presence of nested archives (archives that contain multiple levels of sub-archives) the File Count for an archive in the tree will always consist of the outer-most archive file + each file inside the archive.



For example, the file count for foo.zip in the following tree structure, will equal 3. Only Foo.zip, fileA and fileB are included in the count while file1.jar and file2.zip are treated as directories and are not included in the count.



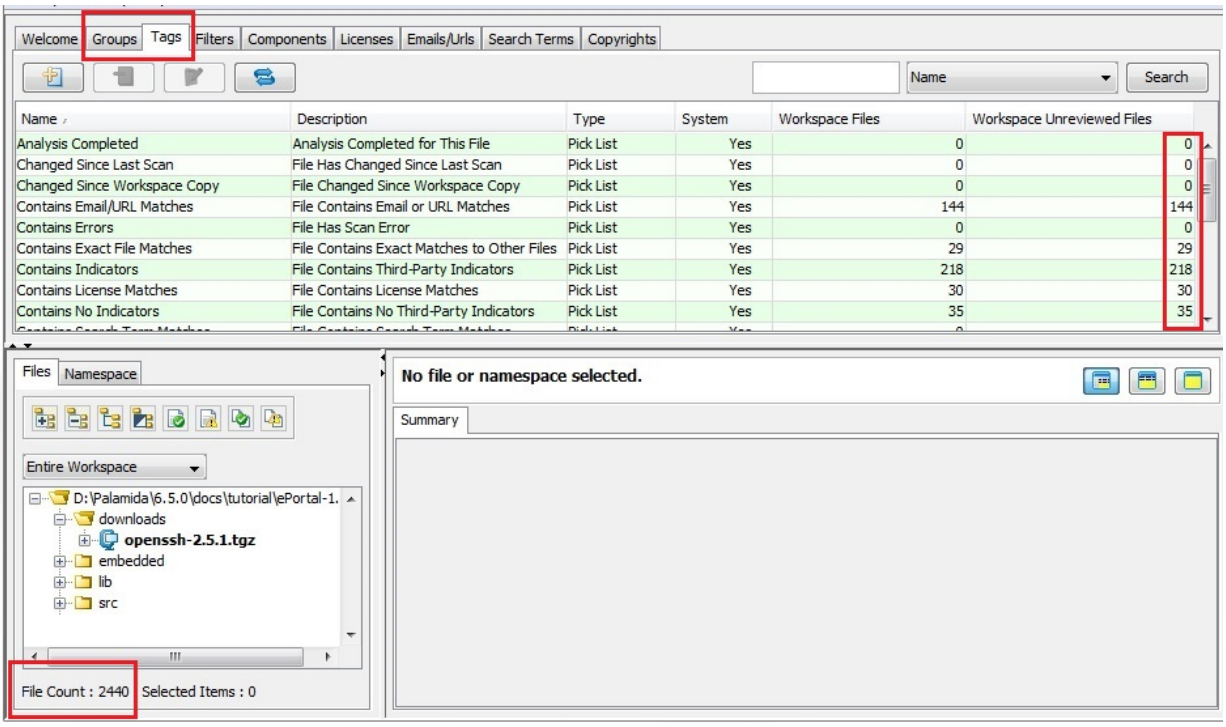
Note ▪ *The file counts in the "Workspace Files" and "Workspace Unreviewed Files" columns in the Groups and the Tags tabs may be significantly smaller than the File Count in the file tree. This is due to the fact that group/tag file counts do not always represent inner files of archives, while the File Count in the file tree always includes inner files of archives.*

Tagging an Archive for Scanning

The Tag Archive for Scanning operation can be used to tag a specific archive for future scanning. This is useful if for example you did not originally select the Scan Files Inside Archives option in Workspace Settings and want to scan one or more select archive files in your codebase. To tag an archive for future scanning, right-click the archive in the file tree and use the "Tag Archive for Scanning" option from the right-click menu. If you wish to tag all contained archives inside an archive or in a directory, use the "Tag All Contained Archives for Scanning" option instead.

Archive File Operations

Scanned archives are treated both as self-contained nodes (individual files) as well as directories that contain files and therefore both single-file operations as well as directory operations are available for archives in the right-click menu.



Applying any of the following single-file operations affects only the archive (outer file) without affecting the inner files:

- Tag File
- Remove Tag From File
- Mark as Reviewed
- Mark as Analysis Completed
- Tag Archive for Scanning
- Add File to Group
- Remove File from Group

Directory Operations for Archives

Applying the following directory operations affects only the files inside the archive without affecting the archive (outer file):

- Add Tag To All Contained Files
- Remove Tag From All Contained Files
- Mark All Contained Files as Reviewed
- Mark All Contained Files as Unreviewed
- Mark All Contained Files as Analysis Complete
- Unmark All Contained Files As Analysis Complete
- Tag All Contained Archives for Scanning
- Untag All Contained Archives for Scanning
- Add All Contained Files To Group
- Remove All Contained Files From Group

Bulk-File Operations for Archives

To select all the files inside the archive, select the archive (outer file), press and hold down the Shift key, and press the left mouse button.

33

Using Multi-Select for Bulk Operations

This chapter describes ways to select several files or folders by using the multi-select feature:

- Consecutive Selection
- Non-Consecutive Selection
- Selection of Files Inside Archives
- Clearing a Selection

Consecutive Selection

To select any consecutive group of files or folders, left-click the first item, press and hold down the Shift key, and left-click the last item.

Non-Consecutive Selection

To select non-consecutive files or folders, press and hold down the Ctrl key, and left-click each item that you want to select.

Selection of Files Inside Archives

To select all the files inside the archive, select the archive (outer file), press and hold down the Shift key, and then press the left mouse button.

Clearing a Selection

To clear a selection, click the selection again with the left-mouse button.

34

Detector Code Search

This section contains the following topics:

Detector

Detector Code Search provides the ability to quickly search across the scanned codebase for search terms, strings, file names, file paths or any combination. In addition to those capabilities, Detector Code Search provides the following:

- Multiple ways to invoke search.
- Search history per session
- Support for binary files
- Ability to add file types for indexing
- Ability to specify exclusions

Code Search Indexing

Code Search indexing is enabled on all workspaces by default. When the workspace is scanned for the first time, an index is created and placed into the `<workspaceDirectory>/<projectName>/index/` directory of your Code Insight installation. If the workspace is rescanned, the data is re-indexed, ensuring that the index is always up-to-date.

Disabling Code Search Indexing

Follow these instructions to disable Code Search indexing.

Task ***To disable Code Search indexing, do the following:***

1. Open scan.properties, located in `<FNCI_Root_Dir>/config/scanEngine/`.
2. Add the following property and set it to *false*:

   ```
   workspaceIndexing = false
   ```

3. Restart the server.

Launching Code Search

You can launch Code Search in any of the following ways:

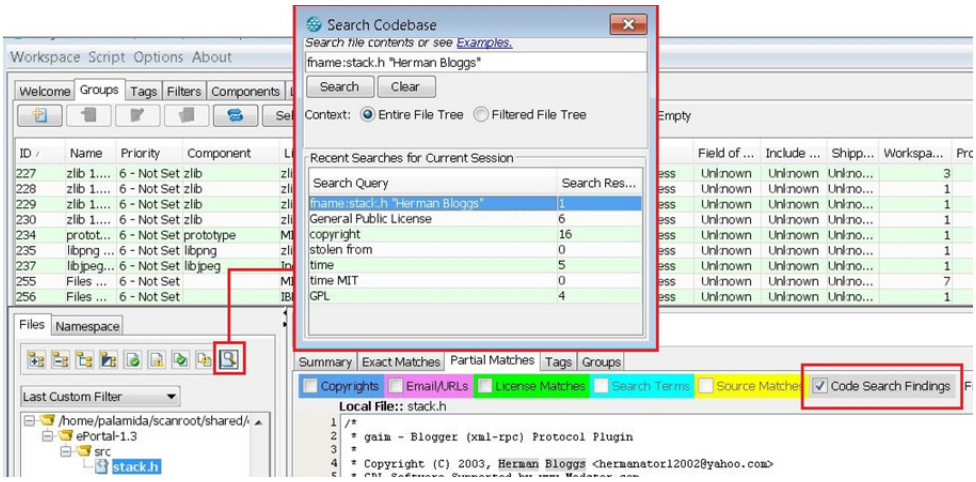
- By using the Code Search icon above the file tree.
- From the Partial Matches panel.
- From the file tree itself.

Using the Code Search Icon

Use the search icon above the file tree to open the **Search Codebase** dialog, which offers full search functionality, including the ability to select a context and view recent search history. The search supports simple or complex queries with wildcards and conjunction operators. See examples for constructing the search query in Code Search Examples.

Task ***To launch Code Search with the icon, do the following:***

1. Select the search icon ( ), which appears above the file tree. The **Search Codebase** dialog appears.



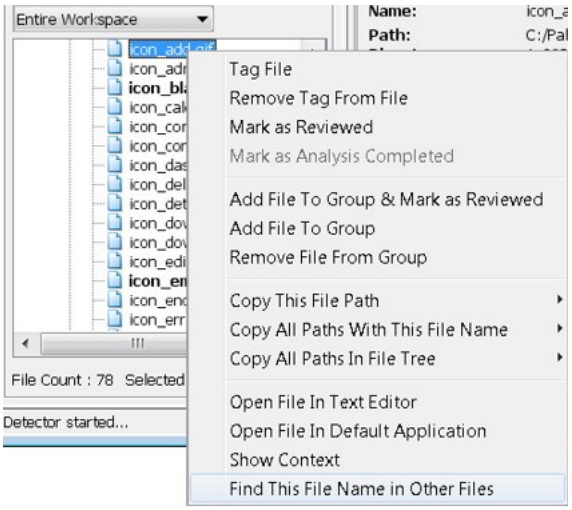
2. Type your query in the search field. For examples of search queries, see **Code Search Examples**.

Using the File Context Menu: File Name Search

A search for a file name contained in the codebase can be quickly executed by highlighting the file in the file tree, and selecting the option to Find This File Name in Other Files from the context menu. This is useful for finding references to files with an unknown origin in the code.

Task ***To use the File Name Search, do the following:***

1. In the file tree, highlight the name of the file you want to search for in your codebase. The context menu opens.



2. Select **Find This File Name in Other Files** from the context menu. Detector searches your codebase for the filename you selected.

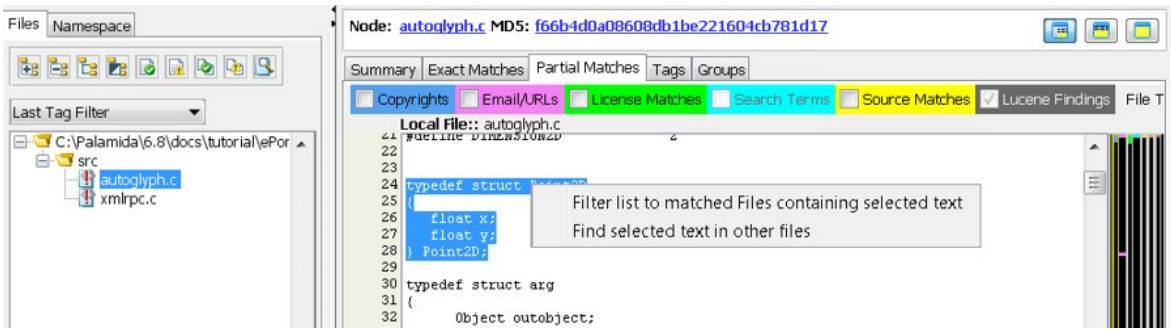
Note ▪ *This type of search always applies to the current context (filtered file tree).*

Using the Partial Matches Panel String Search

You can search directly from the Partial Matches panel. If you come across an interesting string, highlight it and select the Find selected text in other files option from the context menu to filter the file tree to only those files that contain the selected string.

Task ***To use the Partial Matches Panel String search, do the following:***

1. Select a file from the file tree and navigate to the **Partial Matches** tab.
2. Highlight a portion of the text that is of interest and select Find selected text in other files from the context menu.



The file tree is filtered to show only those files that contain the selected string.

Code Search Examples

The following table lists code search examples.

Table 34-1 ▪ Code Search Examples

Type	Search	Finds Files...
Search terms	`do it right`	Containing terms “do”, “it” and “right” anywhere in the file contents. The default conjunction operator is *AND* if no operator is specified.

Table 34-1 ▪ Code Search Examples (cont.)

Type	Search	Finds Files...
Search phrase	`"do it right"`	Containing phrase “do it right” in the file contents.
	`"jonny@revenera.com"`	Containing email phrase “jonny@revenera.com” in the file contents.
file name	`fname:stack.h`	That are named “stack.h”.
file path	`fpath:/home/palamida/ePortal- 1.3/* [Linux]` `fpath:c:/palamida/ePortal-1.3/* [Windows]`	That are located in the `ePortal-1.3` directory or below.
?	`te?t`	Containing “test” or “text” in the file contents.
*****	`test*`	Containing “test”, “testing” or “tested” in the file contents.
AND	`"jakarta apache" AND "Apache Lucene"`	Containing both “jakarta apache” and “Apache Lucene” in the file contents.
OR	`"jakarta apache" OR "Apache Lucene"`	Containing either “jakarta apache” or “Apache Lucene” in the file contents.
NOT	`"zlib compression library" NOT "zlibc Compressing File-I/O Library"?`	Containing “zlib compression library” but not “zlibc Compressing File-I/O Library” in the file contents.
grouping	`("Log4j" OR "Log4j 2") AND website`	Containing “Log4j”, “Log4j 2”, or both in addition to “website” in the file contents.
compounding	`fname:stack.h ("GPL" OR "General Public")`	That are named “stack.h” and have “GPL” or “General Public” in the file contents.

Additional Information

- The default conjunction operator is AND. If is no operator is specified between two terms, the AND operator applies.
- Searches are not case sensitive (including conjunction operators).
- You may use an asterisk (*) as the first character of a search.
- See Lucene Query Parser Syntax for advanced options.

Special Characters

A list of special characters and their interpretation is provided below. If you prefer these characters to be interpreted as search data, the characters must be escaped by placing a backslash (\) character immediately before. Within a phrase, characters are interpreted as search data and need not be escaped. A phrase is one or more terms surrounded by quotation marks.

Table 34-2 ▪ Special Characters

Special Characters	Description
*** 0-n**	n char wildcard
?	One-character wildcard
:	Follows field name
"	Begins or ends phrase
~	Similar or proximity search
!	Same as NOT
&&	Same as AND
||	Same as OR
( )	Groups terms
{ }	Exclusive range
[ ]	Inclusive range
****	Escapes next char
+	Preceding must have term
-	Preceding must not have term
^	Boosts term

Using Code Search for Binary Files

Code Search uses two different mechanisms for processing binary files. The mechanism used determines whether highlights are available in Detector for the search results.

- Highlighting is supported for file types specified by the indexBinList property. By default, these include dll, exe, jar, rpm file types.

- Highlighting is not supported for all other binary file types (not specified by the indexBinList property) such as doc, docx and ppt file types. If working with these types of files, you can use the CTRL-F command to locate specific search terms within the file.

Customizing Code Search Indexing

The following properties may be added to the scan.properties file to further customize indexing and processing of binary files during indexing.

Important ▪ *Restart the server if you add or modify any of these properties.*

Table 34-3 ▪ Code Search Binary Support Properties

Property	Details
workspaceIndexing = true	Set to *true* to enable indexing for the next scan. Set to *false* to turn it off. Default = true
indexExcludeList =	Comma separated list of file extensions to append to the internal predefined set of extensions (see Workspace Settings > General > Excluded File Patterns for existing list) that will be excluded from Code Search indexing.
indexSrcList =	Comma separated list of file extensions to append to the internal predefined set of extensions (see Workspace Settings > Languages – Extensions for existing list) that will be processed and indexed by Code Search. Default = empty
indexBinList = dll, exe, jar, rpm	Comma separated list of extensions to process as binary data (index without Tika using US-ASCII to minimize junk terms). Highlights are available. Default = dll, exe, jar, rpm
indexTikaParseLen = 10485760	Limits the number of characters per file processed by Apache Tika. Use -1 to process the entire file and 0 to not process any file using Tika. indexTikaParseLen = n implies Tika will process up to n characters per file. Default = 10485760

Note ▪ *All binary files not specified by the indexBinList property are processed by Apache Tika. Apache Tika typically generates a much larger index and can take a long time to run. In the case that you experience a system hang during indexing, it is recommended that you set "indexTikaParseLen = 0" to limit the number of characters processed by Tika.*

Detector Code Search APIs

Limited API support is available for Code Search. A ScriptRunner script may be used to invoke a REST API to search one or more indexes. There are methods to do the following:

- Search an index
- Search one or more indexes
- Test if an index exists

Contact Revenera Support for additional information on these APIs and a sample script.

35

Generating Custom Fingerprints

This section contains the following topics:

- Custom Fingerprints
- customFingerprints.groovy Options

Custom Fingerprints

Custom data, including custom source code fingerprints and custom digests, are managed centrally in the application. This means that custom fingerprints and digests will be picked up by all workspaces. If you need to isolate custom data to a particular workspace, contact Revenera Support.

Custom fingerprint generation is controlled by an XML configuration file. The template for this file is located in:

`scriptRunner/scripts/customFingerprints_template.xml`

All custom data is centralized, so you must always modify this file when generating custom fingerprints. The basic instructions for generating custom fingerprints are as follows:.

Task ***To generate custom fingerprints, do the following:***

1. Go to the `scriptRunner/scripts` directory and make a copy of the `customFingerprints_template.xml` file. You can edit this file or make a copy of it. For example purposes, we assume we are editing the `customFingerprints_template.xml` file in these instructions.
2. Edit the `customFingerprints_template.xml` file located in the `scriptRunner/scripts` directory as follows, and save the file when finished:

Line	Code
Line 14: Set the output directory to point to the corporate workspaces directory:.	`outputDir="Palamida/workspaces/CorporateWS">`

Line	Code
Line 22: Set the product name for which you are generating SCF.	`<product name="component_foo">`
Line 23: Set the release name for every distinct release for which you are generating SCF.	`<release name="component_foo_1.0.zip">` `<release name="component_foo_1.2.zip">`
Line 28: Provide a directory where the source code for the fingerprints is located.	`<fileset dir="/Palamida/custom/foo_directory">` ***Note*** *▪ You must provide the path to the directory, not the file path. Custom SCFs will be generated for all the files under this directory.*

3. Ensure the Code Insight Server is running.
4. Invoke `scriptRunner` to generate custom fingerprints:

```
<CODE_INSIGHT_ROOT_DIR>/6.14.x/scriptRunner/bin/./scriptRunner.sh customFingerprints.groovy -d
    customFingerprints_template.xml
```

or (if you are on Windows)

```
<CODE_INSIGHT_ROOT_DIR>/6.14.x/scriptRunner/bin/scriptRunner.bat customFingerprints.groovy
    -d customFingerprints_template.xml
```

5. Wait for the script to finish, which can take a long time if you are processing a large number of files.

Note *▪ It is no longer necessary to publish the CorporateWS with a script as it was in previous versions of the product.*

6. Rescan any compliance workspaces that you want to pick up the new custom fingerprints. New workspaces will automatically pick up the custom fingerprints.

customFingerprints.groovy Options

The following options can provide you more detailed match information:

Table 35-1 ▪ Additional Options for customFingerprints.groovy

Option	Description
-d	This option applied to the fingerprint compiler causes a digest record to be created for every file that is processed. If you do not specify the **-d** option, you will not get digest matches on exact copies of the files processed by the fingerprint compiler.
-a	This option generates fingerprints for files inside archives. Otherwise, the fingerprint compiler will not go inside archive files.

Table 35-1 ▪ Additional Options for customFingerprints.groovy (cont.)

Option	Description
-v	This option gives you more detailed information in the `scriptRunner/log/scriptRunner.log`.

36

Upload to Scan

The Upload to Scan feature allows the project Owner or auditor of a project to upload content to a designated Scan Server and (optionally) automatically invoke a scan/report on the uploaded content. Upload to Scan can be run as an upload only or as an upload immediately followed by a scan/report.

- Configuring Upload to Scan
- Prerequisites
- Using Upload to Scan

Configuring Upload to Scan

As a security precaution, Upload to Scan must be activated and configured by a system administrator in `core.properties` and `scanEngine.properties` before it is available for use.

Task ***To configure Upload to Scan, do the following:***

1. Stop Tomcat. You can use `shutdown.bat` from the `\bin` directory or a similar command.
2. Configure the following properties:
   - core.properties
   - scanEngine.properties
3. Restart Tomcat to make the Upload to Scan feature available in the Web UI.

Note ▪ *Upload to Scan will not work if any of these properties are missing. Sample values are provided below.*

core.properties

Configure the `core.properties` property as described below.

Table 36-1 ▪ core.properties

Property	Description
Set to "true" to enable upload `upload.to.scan.enable.core = true`	Enables upload functionality (without scanning) on the Core server from the Upload to Scan page. Required for Upload to Scan to work. ***Note ▪*** *Files are uploaded, never unpacked on the Core server.*
Max upload file size in MB `upload.to.scan.max.file.size = 10240` ***Note ▪*** *Must be an integer.*	Indicates the maximum supported file size for upload. Defaults to 10 GB. Files above max file size will not be uploaded.
Reports to be available on the Upload to Scan page (quoted, comma-delimited list) `upload.to.scan.reports.list = "Scanned Files Report","Third-Party Indicators Report","Palamida Analyzer - Quick Assessment,"Palamida Analyzer - Group Builder"`	Indicates the set of reports that should show up on the Upload to Scan page. Leave blank if you do not want any reports to be available to the user during Upload to Scan.
Set upload.to.scan.do.scan to "true" or "false": ● **True**—Allow upload followed by a scan/report task (Upload and Scan). ● **False**—Allow upload only (Upload). `upload.to.scan.do.scan = true`	Enables the scan functionality from the Upload to Scan page. If set to false, only upload is available without the option to scan.
Alias of the Scan Server to which files will be uploaded `upload.to.scan.scanner.alias = Scanner1`	Indicates the Scan Server to which files are uploaded from Upload to Scan. ***Note ▪*** *Specify only one Scan Server for the file upload. The upload process fails if more than one server alias is listed for this value.*

scanEngine.properties

Configure the `scanEngine.properties` property as described below

Table 36-2 ▪ scanEngine.properties

Property	Description
Set uploadToScanEnableScanner to "true" to enable Upload to Scan `uploadToScanEnableScanner = true`	Enables upload functionality on the Scan Server where this property file resides. Required for Upload to Scan to work.
Upload directory to write to `uploadToScanUploadDirectory = /home/upload_to_scan_folder`	Indicates the directory on the Scan Server where files will be uploaded during Upload to Scan. Required for Upload to Scan functionality. ***Important ▪*** *Must be an absolute path, equal to or a child of the `serverFileSystemRoot` directory.*

Web application uses this property to limit the root of server file system
Multiple roots are supported, but they have to be separated by "|". For example, in Windows:
`C:\\Windows|C:\\Folder1\\Folder2|D:\\Folder3`
`serverFileSystemRoot`
	Indicates the root directory on the Scan Server where files will be scanned. Users may not browse and scan directories outside of this root.
	Required for Upload to Scan functionality.
	Important ▪ *Must be equal to, or a parent of the `uploadToScanUploadDirectory`.*

Prerequisites

Upload to Scan has the following prerequisites:

- Supported File Types
- File Permissions
- Upload Size
- Using Upload to Scan
- Limitations
- Error Handling
- Security Features

Supported File Types

Upload to Scan supports the following file types:

- **Single file upload**: supported for all file types.
- **Multiple files**: may be uploaded in archived form (s, rpm, tgz, or tar.gz). If the system detects one of these file types, the file is expanded on the Scan Server and the compressed file and any temporary files are discarded. The contents of the archive are then scanned. If the extracted file contents include other compressed files, these files remain compressed. All other types of archives (e.g., exe or jar) are treated as single files.

File Permissions

All uploaded files have their permissions changed to “read only” and “non-executable”. All directories have their permissions changed to “read only” and “executable”.

On Linux systems, directories must be “executable” to be read.

Upload Size

The default maximum upload file size is 10240 MB or approximately 10 GB.

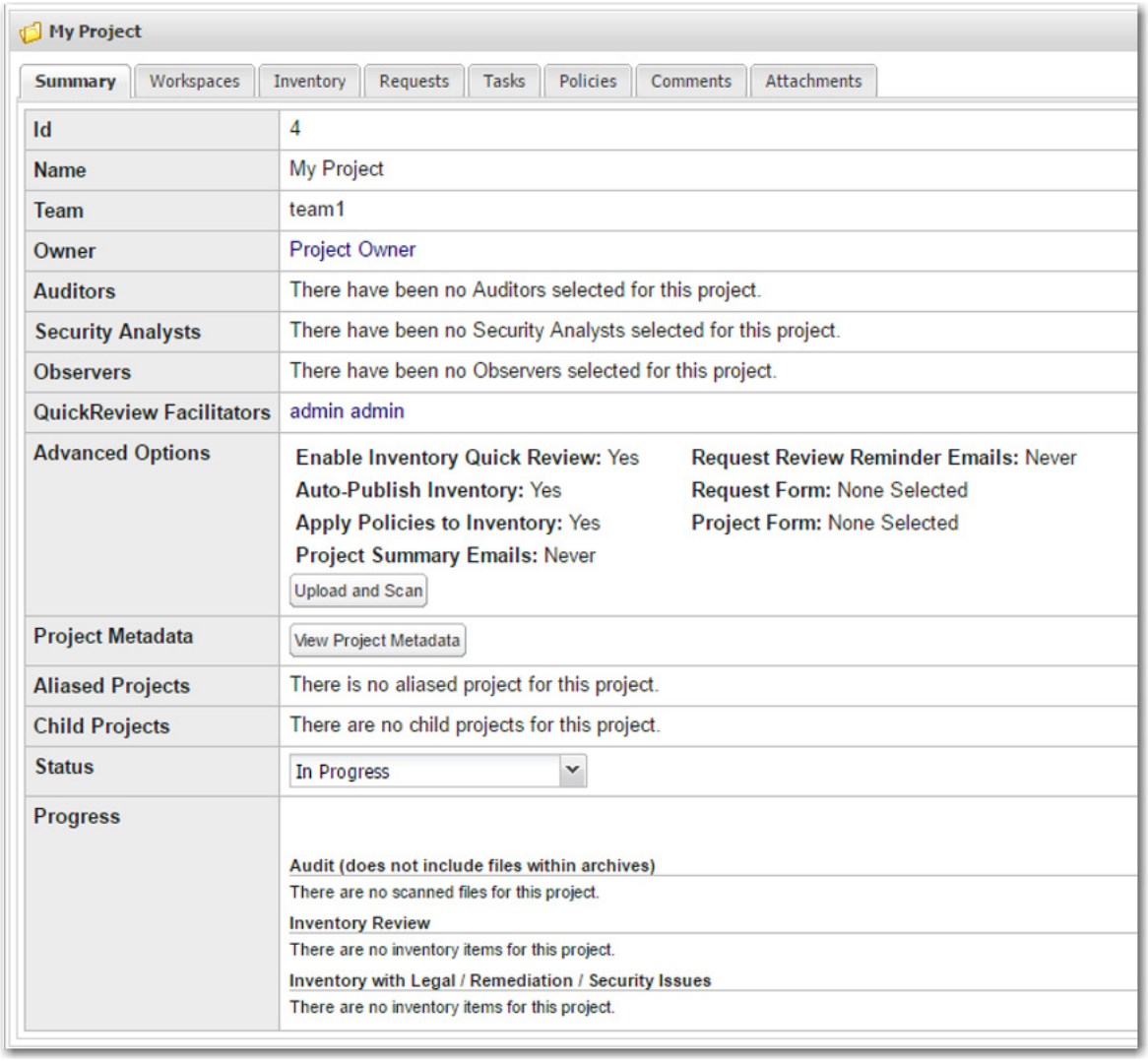
The `upload.to.scan.max.file.size` property in `core.properties` may be used to change the maximum upload size.

Using Upload to Scan

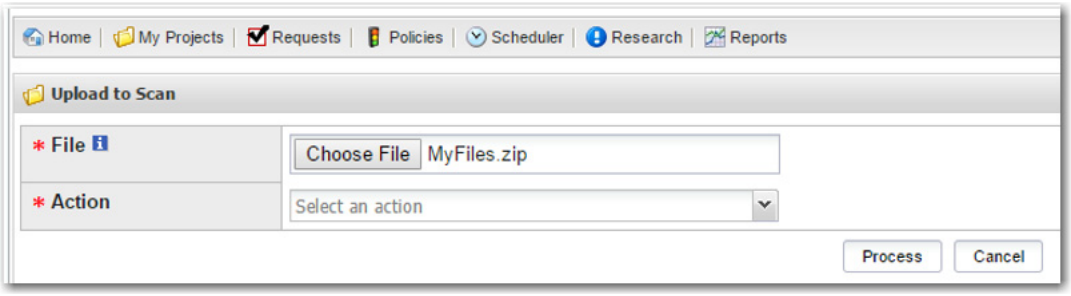
To use Upload to Scan, perform the following steps.

Task ***To use Upload to Scan, do the following:***

1. Log into Code Insight as an *auditor* or *project owner*.
2. Navigate to **My Project Details** for the project you want to upload and scan.
3. Select the **Summary** tab.



4. In the **Advanced Options** section, click **Upload and Scan**. Click **Choose File** and browse to the file or archive of files you want to upload. If uploading multiple files, ensure that they are placed into an archive of type zip, rpm, tgz or tar.gz prior to upload.



5. To upload a file or archive without scanning, select **Upload** in the **Action** menu and click **Process** to start the upload.

   The file or archive of files will be uploaded to the Scan Server into the directory specified by the `uploadToScanUploadDirectory` property.

Internal directories will be created based on the project ID and file name, as demonstrated in the following example:

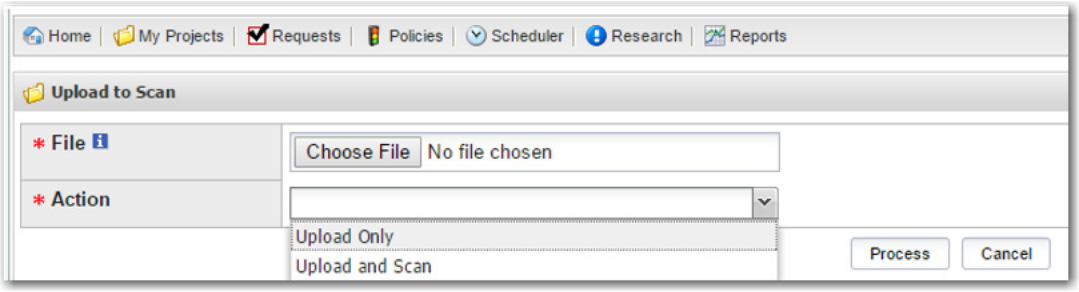
- **Single file**—Upload of `bar.dll` as part of Project1 with ID 1. The file is uploaded to the directory:

  ```
  /MyUploadDirectory/1/bar/bar.dll
  ```

- **Archive of files**—Upload of `foo.zip` containing 3 files as part of Project1 with ID 1:

  ```
  /MyUploadDirectory/1/foo/foo1
  /MyUploadDirectory/1/foo/foo2
  /MyUploadDirectory/1/foo/foo3
  ```

If the `uploadToScanUploadDirectory` already contains a file or archive with the same name, the user will be prompted to overwrite the files with the new content.



6. To upload a file or archive and automatically kick off a scan/report on the uploaded file do the following:

   a. Select **Upload and Scan** in the **Action** menu.

   b. When prompted, enter the name of the workspace to associate to the uploaded files to. Ensure that the name is unique in the system and follows standard naming conventions (e.g. does not include spaces or special characters).

   c. Check off one or more reports to run with the scan, or leave unchecked to do a scan only.

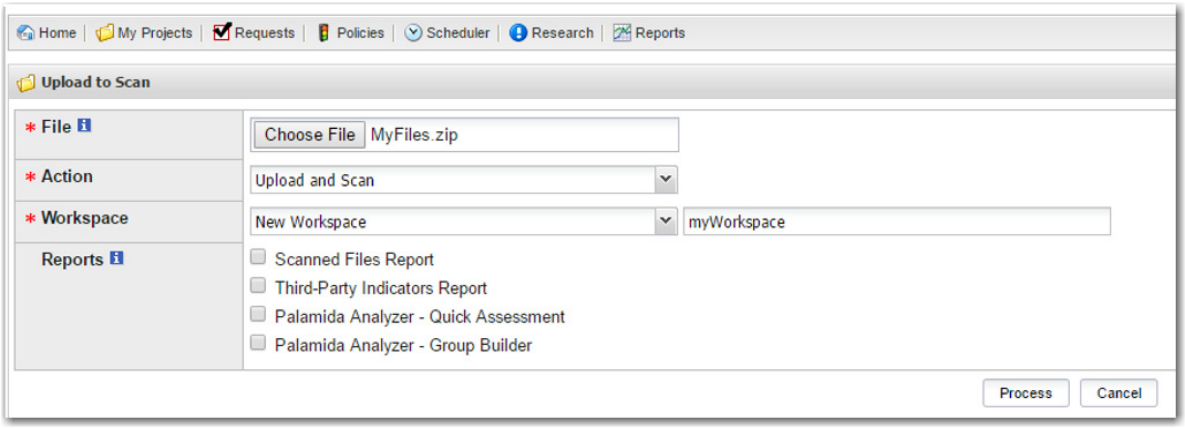
   d. Click **Process** when ready to start the upload and scan/report.

      - **Single file:** upload of `bar.dll` as part of Project1 with ID 1. The file is uploaded to the directory:

        ```
        /MyUploadDirectory/1/bar/bar.dll
        ```

      - **Archive of files:** upload of `foo.zip` containing 3 files as part of Project1 with ID 1:

        ```
        /MyUploadDirectory/1/foo/foo1
        /MyUploadDirectory/1/foo/foo2
        /MyUploadDirectory/1/foo/foo3
        ```



Limitations

Upload and Scan has the following limitations:

- Every **Upload and Scan** function will create a new workspace. To scan the uploaded content as part of an existing workspace, use the **Upload Only** function and edit the workspace to point to the directory containing the uploaded content.
- Files uploaded using the **Upload to Scan** function may not be delete from the Web UI. To delete uploaded content, a system administrator must access the Scan Server file system directly, navigate to the upload directory and delete the files.
- In a multi-Scan Server environment, only one Scan Server can be designated for **Upload to Scan** projects. Set the Scan Server alias of this scanner in `core.properties`.

Error Handling

If there are unacceptable exceptions in any of the above steps, an error message is generated on either the Core Server or Scan Server. The message contains a short description only if that description does not provide security-sensitive information

If the error happened on the Core Server, then the error is logged in `catalina.out` on the Core Server. If the error happened on the Scan Server, then the error is logged in `catalina.out` on the Scan Server.

Security Features

This section describes the security features in the Upload to Scan functionality:

- Core Server
- Scan Server

Core Server

User must be logged in to the Code Insight application, which provides all the usual security benefits, including use of the CSRF Token.

User must be the project's owner or one of the project's auditors.

Core.properties

If any of the following properties are set inappropriately, the Upload to Scan feature will not run:

- `upload.to.scan.enable.core` must be “true” (without quotes).
- `upload.to.scan.max.file.size` must be an integer greater than zero. Leave this property blank if you decide not to limit the size of the file.
- `upload.to.scan.scanner.alias` must be a string.

The Core Server handles all interactions with the user; the user will never directly access a Scan Server.

Uploaded files are managed by the Core Server as a simple array of bytes in memory (i.e., never written to disk), ensuring that potentially malicious files are never a threat to the Core Server.

A user performs a standard HTTP file transfer to upload files through the Code Insight application. In general, users require no additional physical or special permissions to access any Code Insight server; and they need no other software aside from their web browser to access the Code Insight application.

Scan Server

The Scan Server has the following security features.

scanEngine.properties Validation

If any of these properties are blank (or set incorrectly), then this servlet will not run:

- `uploadToScanEnableScanner` must be “true”.
- `uploadToScanUploadDirectory` must be a string.
- This directory must the same as (or a child directory of) the directory defined for the `serverFileSystemRoot` property, which is also set in the `scanEngine.properties` file.

Additional Security Features

Additional security features include the following:

- Scan Server only accepts HTTP Post submissions, never HTTP Get submissions.
- All uploaded/extracted files are set to read-only and non-executable.
- All extracted directories are set to read-only and executable.
- On Linux systems, directories must be executable to be read.
- The Upload to Scan process does *not* create symbolic links of any kind.

- The process checks every file as it is extracted and written to determine whether the file is a symbolic link. If the file is a symbolic link, it is deleted and a line is written to the Scan Server's log file:

  ```
  [INFO] Upload to Scan (scanner): Successfully deleted symbolic link: " + filePath
  ```

  where `filePath` is the full path to the symbolic link that was deleted. This process is identical for all uploaded files, whether the user uploads one file or an entire archive.

- Extracted files will only be written into the uploaded file's parent directory or a child directory of the uploaded file's parent directory. This keeps the upload from writing any content to other locations on the file system (e.g., a malicious attempt to upload an app to the Tomcat `webapps` directory or an accidental attempt to overwrite the '/' directory).